TENLAB

Xiangrong, Xu
xx2367@columbia.edu

Xincheng, Xie
xx2365@columbia.edu

Songqing, Ye
sy3006@columbia.edu

Senhong, Liu
sl4839@columbia.edu

1. Introduction 5
2. TENLab Tutorial 6
2.1. Getting started 6
2.1.1. Environment Setup 6
2.1.2. Compilation Guide 6
2.2. Legendary TENLab tensor 7
2.3. Godlike TENLab parallel functions 8
2.4. Elegant TENLab control flows 8
2.4.1. if-else statement 9
2.4.2. while statement 9
2.4.3. for statement 9
2.5. Rich TENLab built-in functions 9
3. TENLab Manual 11
3.1. Data Type 11
3.1.1. Primitive Data Types 11
3.1.2. Complex Data Types 11
3.2. Lexical Convention 12
3.2.1. Comments 12
3.2.2. Identifiers 12
3.2.3. Operators 12
3.2.3.1. Arithmetic Operators 12
3.2.3.2. Relational Operators 16
3.2.3.3. Logical Operators 16
3.2.4. Keywords 17
3.2.5. Delimiters 17
3.2.6. Literals 17
3.2.6.1. Integer literals 18
3.2.6.2. Float literals 18
3.2.6.3. Character literals 18
3.2.7. Tensor literals 18
3.3. Declarations 18
3.3.1. Tensor Declarations 18
3.3.1.1. From existing data 19
3.3.2. From shape and values with built-in function 19
3.3.3. Create numerical ranges 19
3.4. Function Declarations 19
3.5. Control Flow and Expressions 20
3.5.1. Control Flow 20
3.5.1.1. IF-ELSE Statements 20
3.5.1.2. For Statements 20
3.5.1.3. While Statements 21
3.5.2. Expression 21
3.5.2.1. Precedence and Associativity Rules 21
3.5.2.2. Assignment Expression 21

3.5.2.2.1. Tensor assignment 21
3.5.2.2.2. Function Assignment 22
3.6. User-Defined Parallel Environments 22
3.7. Built-in Functions 24
3.8. Sample Codes 25
4. Project Plan 27
4.1. Planning and specifications 27
4.2. Development 28
4.3. Testing 28
4.4. Coding Style 28
4.5. Project Timeline 29
4.6. Roles and Responsibilities 29
4.7. Environment 30
4.8. Project Log 30
30
5. Architectural Design 30
5.1. Tensor Structure (Xincheng Xie) 30
5.2. Project Architecture (Senhong Liu) 32
5.3. Scanner (Senhong Liu, Songqing Ye) 32
5.4. Parser and AST (Senhong Liu, Xiangrong Xu, Xincheng Xie, Songqing Ye) 32
5.5. Semant Check and SAST (Xincheng Xie, Senhong Liu, Xiangrong Xu, Songqing Ye) 33
5.6. Codegen (Senhong Liu, Xincheng Xie, Songqing Ye, Xiangrong Xu) 34
5.7. C++ Library (Xiangrong Xu, Songqing Ye, Xincheng Xie, Senhong Liu) 35
5.7.1. Built-in.cc 35
5.7.2. Pe.cc 35
5.7.3. Gc.cc 35
6. Test Plan 36
6.1. Test Suite and Automation 36
6.2. Sample Test One 36
6.3. Sample Test Two 65
6.4. Sample Test Three 94
6.5. Sample Test Four 140
7. Lessons Learned 325
7.1. Xiangrong Xu 325
7.2. Xincheng Xie 325
7.3. Songqing Ye 325
7.4. Senhong Liu 326
Appendix 327
Ast.ml 327
Parser.mly 329
Scanner.mll 337
Sast.ml 340
Semant.ml 344

codegen.ml 349
lib/builtin.cc 371
lib/gc.cc 390
lib/index.cc 392
lib/pe.cc 396
lib/tensor.cc 396
lib/tensor.h 398
Makefile 399
Generate.sh 400
Test.sh 403
Rundemo.sh 408
Tests: 410

1. Introduction

TENLab is an imperative language based on the idea that *Everything is Tensor*, just like Python's *Everything is Object*. Tensors are a type of data structure used in linear algebra, and like vectors and matrices, you can calculate arithmetic operations with tensors and tensor has been used extensively in machine learning and deep learning.That's also why we want to use tensor as the basic data structure of Tenlab.
Tenlab also enables users to easily define some parallel functions to implement a MapReduce model to efficiently do the matrix calculation. Like python, Tenlab also supports dynamic typing of variables, which means that variables could have different types during runtime. Also, there are various built-in functions for tensors. Finally, Tenlab has a simple automatic garbage collection to avoid memory leaks.
In a word, Tenlab is an easy, fast and flexible language.

The goals in the white book of Tenlab are (copied from the proposal):
• Implement a Python-style syntax language to support matrix computation.
• Potential optimization due to the static declaration of variables.
• If possible, implement a distributed model, e.g., MapReduce, to help accelerate the matrix computation.

Correspondingly, we realized:
• Warp everything in our language in tensors and develop our syntax based on tensors.
• Both syntactic check and dynamic check for tenors.
• A featured and user-friendly MapReduce parallel environment implementation.

Although by the limit of time, Tenlab is not perfect and there is still much work to do, we have basically achieved our goal in the white books.

2. TENLab Tutorial

2.1. Getting started

2.1.1. Environment Setup

First, pull our repository from Github.

```
git clone https://github.com/Senhongl/TENLab
cd TENLab
```

Download the Litorch library and move it to the current(./TENLab/) directory.

```
wget https://download.pytorch.org/libtorch/nightly/cpu/libtorch-shared-
with-deps-latest.zip
unzip libtorch-shared-with-deps-latest.zip
rm -rf libtorch-shared-with-deps-latest.zip
```

To run the code with providing Dockerfile, one can first build the image by (support only for x86-architecture)

```
docker build -t tenlab - < Dockerfile
```

Then start a container by

```
docker run --rm -it -v $("pwd"):/home/tenlab -w=/home/tenlab tenlab
```

2.1.2. Compilation Guide

Inside the docker container, compile the Coral compiler using the following commands. This will automatically run TENLab's test suite as well.

```
make
```

2.2. Legendary TENLab tensor

Declaring a normal tensor is trivial: use '[' and ']' to brace your tensor, use ',' to separate elements. Here is an example of declaring a 3×2 tensor:

```
a = [[2,3],[1,5],[6,7]];

print(a);
>> 2  3
   1  5
   6  7
   [ CPUIntType{3,2} ]
```

Now, how to access or index tensors? Similarly, use '[' and ']' to brace your indices and use ',' to separate different dimensions for indexing. For example,

```
print(a[[1,2],[0,1]]);
>> 1  7
   [ CPUIntType{2,} ]
```

Also, we support slicing, which can be used in tensor declaration and indexing. Slicing is 3 integers separated by ':' — start:end:step, which produces a one-dimensional tensor [start, end) with interval 'step'. For example,

```
print(1:10:2);
>> 1  3  5  7  9
   [ CPUIntType{5,} ]

print(a[1:3:1,0:2:1]);
>> 1  5
   6  7
   [ CPUIntType{,2} ]
```

In order to support various data structures, we introduce "vartensor", which does not have constraints like a normal one. Just use "var" before use define:

```
a = var[[2,3], [1.5], "abc"];
```

Here is an example of defining a binary tree using vartensor (nil is a tensor which has no data):

```
#[left, value, right]
tree = var[nil, 0, nil];
```

```
tree[0] = var[var[nil, 1, nil], 2, var[nil, 4, var[nil, 5, nil]]];
tree[2] = var[nil, 6, nil];
```

2.3. Godlike TENLab parallel functions

```
parallel_define toy_add {
    overload __+__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x+y;
                  return z;
            }
            reduce {
                  return f1+f2;
            }
      }
}
```

This is how one defines a parallel environment. See how easy the syntax is! The map/reduce functions take no arguments. The map functions share the parameter of the operator and the reduce function uses the name of map functions as parameters, and the value is the return value of the input parameters.

To apply the toy_add example, one simply use:

```
using toy_add;
a = 1+2
```

And the user will get a 3!

2.4. Elegant TENLab control flows

TENLab's control flows are very concise and elegant, the way of writing which is similar to that in C. There's some simple examples. If you want to learn more about details, refer to our manual or sample codes!

2.4.1. if-else statement

The forms of conditional statements are the following:

```
a = [1,0];
if (any(a)) {
      print("yes");
} else {
      print("no");
```

2.4.2. while statement

The forms of while loops are the following:

```
a = 1;
b = 2;

while (a) {
    if (b == 10) {
        a = 0;
    }
    print(b);
    b = b + 1;
}a = [1,0];
if (any(a)) {
      print("yes");
} else {
      print("no");
}
```

2.4.3. for statement

The forms of for loops are the following:

```
a = [1, 2, 3];
b = 1;
for (i in a) {
    print(i);
}
```

2.5. Rich TENLab built-in functions

Our language has very powerful matrix-related built-in functions as well as the arithmetic operations. You can see all the built-in functions and their functionalities in our language reference manual. As an instance, the below code is a part of our demo code for running linearn regression.

```
# Random Init
w = rand([8192]);
b = 0.;

# Mean Square Error loss function
def loss() {
    y_test = x_train * w + b;
    error = sum((y_test - y_train).^2 .* 0.5);

    return error / float_of(num_of_dimensions);
}

# Derivative of Loss w.r.t W
def dLdW() {
    y_test = x_train * w + b;
    dLdw = (y_test - y_train) * x_train / float_of(num_of_dimensions);

    return dLdw;
}
```

3. TENLab Manual

3.1. Data Type

All the data in TENLab are in "tensor" type. Even a simple number is a zero-dimensional tensor. However, all the tensors have an overall data type which is determined by the data type of the smallest granularity in a tensor. For example, an integer-type tensor means every element in this tensor is integer. Generally, a tensor has the following structure:

```
; structure {type, ndim, dims, data}
{i8, i8, [n x i64], [...]}
```

For any operations, they must operate on tensors with the same data type; otherwise, it will throw an error.

Data types in TENLab can be classified as three primitive data types and one complex data type.

```
Type: int | float | char | var
```

3.1.1. Primitive Data Types

Primitive data types are fundamental data types in TENLab. Every type has its fixed size in memory and some related properties. For a tensor whose every element's primitive type is the same, then the tensor is also this primitive type; otherwise, if elements in a tensor have different primitive types, then this tensor belongs to the complex data type.

1. `int`

`int` means integer. It stores a 32-bit numeric value in memory.

2. `float`

`float` means floating-point number. It stores a 64-bit numeric value in memory. It follows IEEE Standard for Floating-Point Arithmetic (IEEE 754).

3. `char`

`char` means characters. It stores a 8-bit numeric value in memory. It is encoded by ASCII.

3.1.2. Complex Data Types

When data types in a tensor are various, then it is a `var` type tensor, which means elements in the tensor can contain all of three primitive data types. It is just like a nested version of different types of tensors. For example,

```
ts = var([['c', 1], 1.3]);
```

3.2. Lexical Convention

3.2.1. Comments

Comments serve as an annotation or a programmer-readable explanation in the source code. Any tokens in the comment will not be parsed.
TENLab supports two forms of comment:

1. Inline comment: adds `#` at the beginning of the comment, and the comment will stop at the end of the current line.

```
x = 0; # this is a comment
```

2. Comment block: adds `'''` at both the beginning of the comment and the end of the comment block to make multiple lines of comments.

```
'''
This is
a multi-line
comment
'''
```

3.2.2. Identifiers

Identifier in TENLab is a case-sensitive ASCII sequence of one or more letters, digits and underscore `_`. The first character in an identifier must be a letter. Identifier cannot be a keyword.

```
# valid identifiers
helloTENLab
hello_TENlab
TENLab_ranks_1st_in_the_most_popular_programming_languages

# invalid identifiers
int
1st_TENLab
```

3.2.3. Operators

3.2.3.1. Arithmetic Operators

The following list shows basic arithmetic operations TENLab supports. Denote **A** and **B** as the first operand and the second operand in a binary operation. Constants are in the form of a 0-dimension tensor, and they should only appear on the right side of operators. Operators with '.' means element-wise operations.

Attention: A binary arithmetic operator should have operands of the same numeric types, that is to say any binary arithmetic operation between an int-type tensor and an float-type tensor is not allowed.

Addition(+): **A** and **B** should have the same dimension, or **B** is a 0-dim tensor.

Addition examples:

```
A = [0, 1, 1, 0];
B = [2, 3, 1, 5];
A - B;  # A and B should have the same dimension

return
    [-2, -2, 0, -5]

A + 1;  # the second operand is a 0-dim tensor

return
    [1, 2, 2, 11]

C = [0.2, 0.3, 0.4, 0.6];
A + C;  # not allowed!

A + 0.5;    # not allowed!
```

Subtraction(-): **A** and **B** should have the same dimension, or **B** is a 0-dim tensor.

Subtraction examples:

```
A = [0, 1, 1, 0];
B = [2, 3, 1, 5];
A - B  # A and B should have the same dimension

return
    [-2, -2, 0, -5]

A - 1;  # the second operand is a 0-dim tensor

return
    [-1, 0, 0, -1]
```

Matrix Multiplication(*): **A** should be an m by n tensor and **B** should be n by p tensor, or **B** is a 0-dim tensor.

Matrix multiplication examples:

```
A = [[1, 3, 5], [2, 4, 7]];
```

```
B = [[-5, 8, 11], [3, 9, 21], [4, 0, 8]];

A * B;  # A should be an m by n tensor and B should be n by p tensor

return
    [[24, 35, 114],[30, 52, 162]]

A * 2;

return
    [[2, 9, 25], [4, 16, 49]]
```

Element-wise multiplication(.*): **A** and **B** should have the same dimension

Element-wise multiplication examples:

```
A = [1, 0, 3];
B = [2, 3, 7];

A .* B;  # A should be an m by n tensor and B should be n by p tensor

return
    int([2, 0, 21])
```

Division(/): **B** should be a 0-dim tensor. . The return value is an float-type tensor.

Division examples:

```
A = [4, 5, 10];

A / 2;  # B should be a 0-dim tensor

return
    [2.0, 2.5, 5.0]
```

Power(^): **B** should be a 0-dim tensor.

Power examples:

```
A = [2, 3, 5];

A ^ 2  # B should be a 0-dim tensor

return
    [4, 9, 25]
```

Element-wise power(.^): **A** and **B** should have the same dimension.

Element-wise power examples:

```
A = [2, 3, 5];
B = [1, 0, 3];

A .^ B  # A and B should have the same dimension

return
    [2, 1, 125];
```

Transpose('): Transpose is a unary operator.

Element-wise power examples:

```
A = [[16, 2], [5, 11], [3, 8]];

A';  # A and B should have the same dimension

return
    [[15, 5, 3],[2, 11, 8]]
```

Mod(%): **B** should be a 0-dim tensor.

Mod examples:

```
A = [4, 6, 10];

A % 3;  # B should be a 0-dim tensor

return
    [1, 0, 1]
```

Floor Division(//): **B** should be a 0-dim tensor. The return value is an int-type tensor.

Remainder examples:

```
A = [2, 3, 5];

A // 2  # B should be 0-dim tensor

return
    int([1, 1, 2])
```

3.2.3.2. Relational Operators

Relational operators are used to determining how two operands relate to each other. The relational operators always take two tensors as inputs. The return value of all the following relational operations is a logical tensor with elements set to logical 1 (**true**) where tensors **A** and **B** satisfy the operation; otherwise, the element is logical 0 (**false**).

Operator Name	Description
==	Determine equality
>=	Determine greater than or equal to
>	Determine greater than
<=	Determine less than or equal to
<	Determine less than
!=	Determine inequality

Examples are as follows:

```
X = [1, 0, 0, 0, 1];
Y = [0, 0, 1, 1, 0];
X >= Y;

return
    [1, 1, 0, 0, 1]
```

3.2.3.3. Logical Operators

TENLab also provides logical AND, OR and NOT operations. Logical operators should only appear between two expressions.

Operator Name	Description
&&	Logical OR operator
||	Logical AND operator
!	Logical NOT operator

Examples are as follows:

```
X = [1 0 0 0 1];
Y = [0 0 1 1 0];
!any(X);
```

```
return
    [0];

any(X) || all(Y)

return
    [1];
```

3.2.4. Keywords

Keywords are reserved words that are case-sensitive and cannot be used as identifiers. The following keywords are included in TENLab:

1. **Control**: if, elif, else, for, while, in, continue, break, return, read, print, exit, define
2. **Type**: int, float, string, void
3. **Parallel related:** parallel_define, overload, map, reduce, __*__, __+__, __-__, using, end
4. **Tensor related**:

- cat: concatenate two tensors
- shape: return the shape of the tensor

Common tensor operations will be supported in the standard library.

3.2.5. Delimiters

A delimiter is a series of one or more characters used to define the boundary between separate, independent regions in plain text or other data streams.

1. **Whitespaces**: Any whitespace character such as a new line, a horizontal tab or a carriage return, used to separate tokens or statements from each other
2. Opening and closing **braces** '{' and '}', usually surrounding a block of statements to be treated as a unit, for example, the body of the method
3. Opening and closing **parentheses** '(' and ')', usually surrounding parts/pieces of an expression for the purposes of overriding the default precedence or simply for clarity, or surrounding the test part of an if/while statement
4. **Commas** ',', used to separate items in tensors.
5. **Colons** ':', used in a tensor declaration to index a specific range

3.2.6. Literals

Literals in TENLab can be divided into four categories: integer, float, character and tensor. Note that since everything is tensor, the first three are also a special zero-dimensional tensor.

3.2.6.1. Integer literals

An integer literal is a sequence of digits. This literal is only for decimal. For negative integers, add '-' in front of the digit sequence. '+' is not allowed to be prefixed. For example,

```
1341  -2123
```

3.2.6.2. Float literals

The first type of floating point literal is two sequences of digits with a dot between these two sequences. Either one of these two sequences could be empty, but not both. For negative integers, add '-' in front of the whole sequence (two sequences of digits plus a dot). '+' is not allowed to be prefixed.
The second type of floating point literal is two sequences with 'e' between these two sequences. Neither one of these two sequences could be empty. The sequence in the front is the first type of floating point literal or the integer literal. The sequence in the back must be the integer literal. For negative integers, add '-' in front of the whole sequence. The digit sequence behind the 'e' can also be prefixed with '-'.

The following are the examples:

```
3.2   .3124 123.  -1.2  -1.2e-10          1e-10
```

3.2.6.3. Character literals

A character literal is a single ASCII character enclosed by single quotes. For example,

```
'c'   '\0'  '\n'
```

3.2.7. Tensor literals

A tensor literal is grouped by brackets, with commas to separate different elements. Each element can be one of the four literals. For primitive data type tensors, dimensions and data type of each element must be matched. For example,

```
a = [[1, 2], [3, 4], [5, 6]];
b = var([['a', 2, 4.], [4, '2']]);
```

3.3. Declarations

3.3.1. Tensor Declarations

A tensor can be initialized in a couple of ways shown below.

3.3.1.1. From existing data

The users can wrap the existing data with brackets and a data type to declare a tensor:

```
A = [0, 1, 2]; # A is now declared as a 1-dimensional tensor with 3
elements in it
```

To declare a 0-dim tensor, one can either declare with a given data type or not. For example,

```
# giving a data type
A = 0;

# without giving any data type
A = 'a';
```

3.3.2. From shape and values with built-in function

The user can also create a tensor with the built-in functions, e.g., zeros(), ones(). These will be introduced in the following sections.

3.3.3. Create numerical ranges

One can also declare a tensor by giving the following expression:

```
Numerical-0: Numerical-1: Numerical-2
```

The Numerical-0 represents the beginning value of an interval, while Numerical-1 represents the end value of the interval. The interval includes the Numerical-0 but does not include the Numerical-1. And Numerical-2 represents the spacing between the values.

Here is an example of how to declare a tensor with such a statement.

```
A = 0:3:1;
A;

return
    [0, 1, 2]
```

3.4. Function Declarations

Function declaration in TENLab is first given a keyword, def, then followed by an identifier as the function name. Arguments are contained in parentheses after the function name. The body of a function is delimited by curly brackets. The function could either contain a return statement or not, but it could only return one tensor. Overall, the function declaration should be in the following form:

```
def foo(x, y) {
    statements;
    return z; # this is not required
}
```

3.5. Control Flow and Expressions

3.5.1. Control Flow

3.5.1.1. IF-ELSE Statements

In TENLab, we support the following three forms of conditional statements:

```
# if-only statement
if (expression) {
    statements;
}


# if-else statement
if (expression) {
    statements;
} else {
    statements;
}

# if-elif statement
if (expression) {
    statements;
} elif (expression) {
    statements;
}
```

The expression will be evaluated from top to down. When one of them is evaluated as non-zero, then the corresponding series of statements will be executed. The expression must be delimited in parentheses. And the statements must be delimited in curly braces.

3.5.1.2. For Statements

For statements in TENLab is iterated over a list, which could be created and initialized in a tricky way as in the previous section. The for statement should be formed as:

```
for (identifier in list) {
```

```
      statements;
}
```

Similar to the conditional statements, the statements inside the loop need to be delimited in curly braces.

3.5.1.3. While Statements

While statements in TENLab is pretty much the same as the conditional statements except for in a loop form:

```
while (expression) {
    statements;
}
```

3.5.2. Expression

3.5.2.1. Precedence and Associativity Rules

The associativity rules of all the operators are shown in the following table, in order of precedence from top to bottom.

Operators	Associativity
=	right
||	left
&&	left
== !=	left
>= > <= <	left
+ -	left
* .* / % //	left
^ .^	right
‘	left
!	right

3.5.2.2. Assignment Expression

3.5.2.2.1. Tensor assignment

After declaration, the value in the tensor can still be modified, the user can access the tensor by index and modify it by giving a value. For example,

```
A = [0, 1, 2]
A[0] = 1;
A;

return
   [1, 1, 2]
```

3.5.2.2.2. Function Assignment

Although TENLab doesn't support returning multiple tensors, one can pack multiple tensors in a tensor and return. When calling the function, you can unpack and assign multiple return values to different variables. For example,

```
x, y = foo(x, y);
```

3.6. User-Defined Parallel Environments

In TENLab, matrix operations can be paralleled in a map-reduce way. Users can define their own set of parallel functions to be used, called a parallel environment. Currently, we only allow parallel computation for +(Addition), -(Subtraction) and *(Matrix Multiplication).
To define a parallel environment, you do:

```
parallel_define <environment_name> {
    overload <operator_name> (X, Y) {
        map <name_of_function_one> {
            statements;
            return <some_variable>;
        }
        map <name_of_function_two> {
            statements;
            return <some_variable>;
        }
        reduce {
            return <some_variable>;
        }
    }
}parallel_define toy_add {
    overload __+__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x+y;
```

```
                return z;
            }
            reduce {
                return f1+f2;
            }
        }
}
```

Here map stands for the map functions, and reduce is for the reduce function. You do not need to specify all the operators. The not mentioned ones are treated as default. Note that there are three differences between map/reduce function definition and usual function definition:1. map/reduce functions must have return value. Otherwise, a runtime error will occur. 2. Reduce function takes no arguments, and no names. In the reduce function, the name of map functions represents the return value of map functions. 3. Map functions also take no arguments, and the operator's arguments are their default argument. A concrete example is you may overload addition like this:

```
parallel_define EnvironmentTest {
    overload __+__ (x, y) {
        map f1 {
            z = x[0:2:1] + y[0:2:1];
            return z;
        }
        map f2 {
            z = x[2:4:1] + y[2:4:1];
            return z;
        }
        reduce {
            return cat(f1, f2, 0);
        }
    }
}
```

To implement the function you defined, you can simply add using <environment name> to the beginning of the block and end <environment name> to end of the block:

```
using <environment_name>;

statements;

end <environment_name>;
```

By doing so, the operations in the block, if specified by the environment, will use its implementation in environment definition. It is also possible to omit the end command. In that case, this parallel environment will be in effect until the end of this file.

In the future, a standard library with parallel implementation provided called std_parallel may be supported. One can easily enjoy the power of multithreading by adding one single line in the front of your code and see everything becomes magically faster.

3.7. Built-in Functions

Matrix Control	
any(x)	If all elements of x are zero, return 0; otherwise, return 1
all(x)	If any element of x is zero, return 0; otherwise, return 1
sum(x)	Returns the sum of all elements in x
ones(x)	Return a tensor filled ones of the given shape x
zeros(x)	Return a tensor filled zeros of the given shape x
rand(x)	Return a tensor filled float number ranging from 0 to 1
len(x)	Return the first dimension of shape(x)
Arithmetic	
int_of(x)	Convert x to an int tensor
float_of(x)	Convert x to a float tensor
floor(x)	Round every element in x towards negative infinity
ceil(x)	Round every element in x towards positive infinity
round(x)	Round every element in x towards closest decimal
abs(x)	Take abstract value of every element in x
log(x)	Compute log value of every element in x
Matrix computation	
inverse(x)	Compute the inverse of matrix x
solve(A,b)	Solve Ax=b, return x

3.8. Sample Codes

```
# Design a parallel environment with Strassen algorithm, which computes
matrix multiplication in O(N^2.8)
parallel_define Strassen {
    overload __*__ (x, y) {
        map f1 {
            l1 = shape(x);
            l2 = shape(y);
                  x11 = x[0:l1[0]//2:1,0:l1[1]//2:1];
                  x22 = x[l1[0]//2:l1[0]:1,l1[1]//2:l1[1]:1];
                  y11 = y[0:l2[0]//2:1,0:l2[1]//2:1];
                  y22 = y[l2[0]//2:l2[0]:1,l2[1]//2:l2[1]:1];
            return (x11+x22)*(y11+y22);
        }
        map f2 {
            l1 = shape(x);
            l2 = shape(y);
                  x21 = x[l1[0]//2:l1[0]:1,0:l1[1]//2:1];
                  x22 = x[l1[0]//2:l1[0]:1,l1[1]//2:l1[1]:1];
                  y11 = y[0:l2[0]//2:1,0:l2[1]//2:1];
            return (x21+x22)*y11;
        }
        map f3 {
            l1 = shape(x);
            l2 = shape(y);
                  x11 = x[0:l1[0]//2:1,0:l1[1]//2:1];
                  y12 = y[0:l2[0]//2:1,l2[1]//2:l2[1]:1];
                  y22 = y[l2[0]//2:l2[0]:1,l2[1]//2:l2[1]:1];
            return x11*(y12-y22);
        }
        map f4 {
            l1 = shape(x);
            l2 = shape(y);
                  x22 = x[l1[0]//2:l1[0]:1,l1[1]//2:l1[1]:1];
                  y11 = y[0:l2[0]//2:1,0:l2[1]//2:1];
                  y21 = y[l2[0]//2:l2[0]:1,0:l2[1]//2:1];
            return x22*(y21-y11);
        }
        map f5 {
            l1 = shape(x);
            l2 = shape(y);
                  x11 = x[0:l1[0]//2:1,0:l1[1]//2:1];
```

```
                y22 = y[l2[0]//2:l2[0]:1,l2[1]//2:l2[1]:1];
                x12 = x[0:l1[0]//2:1,l1[1]//2:l1[1]:1];
            return (x11+x12)*y22;
        }
        map f6 {
            l1 = shape(x);
            l2 = shape(y);
                x21 = x[l1[0]//2:l1[0]:1,0:l1[1]//2:1];
                y11 = y[0:l2[0]//2:1,0:l2[1]//2:1];
                x11 = x[0:l1[0]//2:1,0:l1[1]//2:1];
                y12 = y[0:l2[0]//2:1,l2[1]//2:l2[1]:1];
            return (x21-x11)*(y11+y12);
        }
        map f7 {
            l1 = shape(x);
            l2 = shape(y);
                x22 = x[l1[0]//2:l1[0]:1,l1[1]//2:l1[1]:1];
                y22 = y[l2[0]//2:l2[0]:1,l2[1]//2:l2[1]:1];
                y21 = y[l2[0]//2:l2[0]:1,0:l2[1]//2:1];
                x12 = x[0:l1[0]//2:1,l1[1]//2:l1[1]:1];
            return (x12-x22)*(y21+y22);
        }
        reduce {
            c11 = f1+f4-f5+f7;
            c12 = f3+f5;
            c21 = f2+f4;
            c22 = f1-f2+f3+f6;
            return cat(cat(c11,c12,0),cat(c21,c22,0),1);
        }
    }
}

using Strassen;

# Markov Process
P = float([[3/4, 1/4], [1/4, 3/4]]); # transition matrix
s = float([[0.2, 0.8]]); # initial state
# judge difference of two matrices' elements is less than 1e-5
def diff(prev, curr)
{
    if (shape(prev) != shape(crr))
    {
        exit(-1);
```

```
    }
    delta = prev - cur;
    flag = 1;
    shape_t = shape(delta);
    for (i in 0:shape_t[0]:1)
    {
        for (d in delta[i,0:shape_t[1]:1])
        {
            if (abs(d) > 1e-5)
            {
                flag = 0;
            }
        }
    }
    return flag;
}
# multiply state and transition matrix
def mulPs(s, P)
{
    return s, s * P;
}
# check state after four transitions
print(s * (P^4));
# Iterates until stable
s_prev = zeros(shape(s));
while (diff(s_prev, s) == 0)
{
    s_prev, s = mulPs(s, P);
}
```

4. Project Plan

4.1. Planning and specifications

Group members meet once a week after class or over zoom to check in each member's progress and make sure that everyone is on the right track and on the same page. During our team meeting, each of us provided an update on what they were working on currently, discussed design/implementation details and made decisions together. After ensuring that we were

constantly making forward progress on our language, we made plans on tasks to do for the next week.

As mentioned above, our project proposal included not only matrix manipulation but also some other parallel functions. As we only have limited time, we have to set priority to the features we are going to implement, and our priority queue is like tensor creation, tensor operations (basic arithmetic operations, indexing, slicing) control flows, parallel functions, built-in functions, automatic garbage collection. Fortunately we have realized most of these functionalities, and hope we can hone for the rest in the future.

4.2. Development

At the beginning of the project, when we were still learning about Ocaml grammar, we only had a high-level idea of what language we wanted to have, and to compile the language for what we might need. We started our project after our first assignment, in which we just learned how to use Ocaml to write a calculator.

We gradually built the scanner, parser and AST, and wrote our language reference manual at the same time. After the LRM, For the Hello World assignment, we started to write Sast and codegen. We spent the final weeks expanding our sast and codegen to support more features and testing our compiler.

Even before the presentation we were still polishing our language and compiler, and trying to give better sample codes which could show the characteristics of our language as much as possible.

4.3. Testing

Every time we made a new feature we would create a corresponding test case to ensure its correctness before we merged it to the main branch of our github repository. Tests featured both success and fail cases. More details please refer to the Test Plans.

4.4. Coding Style

Tenlab is mostly developed with OCaml (for compiler) and C++ code (for library). We tried to write clean, readable, modular, OCaml code. We often followed these general guidelines when writing our compiler:

1. Indent clearly. This is very important to make our ocaml code readable.
2. Commit frequently.
3. Use descriptive function names to make it easier to understand the code.
4. All AST types are camel cases and all SAST types are the related AST type with a capital ‘S’ in front.
5. Every time before committing, make sure there’s no error and all test cases are runnable.

6. Write robust test cases for any code modifications and try to find both positive (passing test cases ie. if something should work, that it works) and negative (failure test cases ie. if something should fail, that it fails) test cases.
7. Write comments whenever things are unclear or if comments would be helpful to future readers.
8. Simplify programs when and if possible.

4.5. Project Timeline

Dated	**Milestone**
Oct. 8th	Proposal
Oct. 29th	LRM
Nov. 1st	First version of scanner.mll
Nov. 3rd	First version of parser.mly
Nov. 6th	Parser support PE, tensor, build-in functions
Nov. 10th	First simple Hello World
Nov. 14th	Parser for new tensor
Nov. 17th	Sement for expression
Nov. 19th	Simple codegen for tensor
Nov. 19th ~ Dec. 20th	Keep adding functionality and tests
Dec. 22th	Presentation

4.6. Roles and Responsibilities

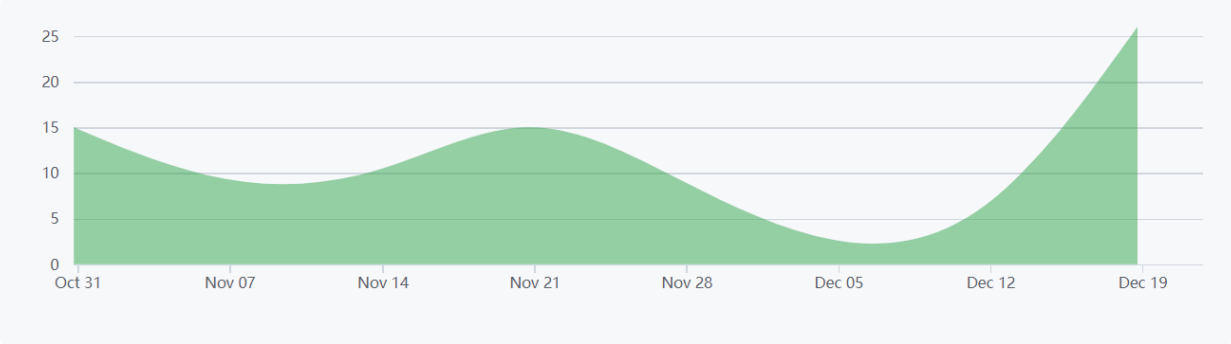
Name	**Roles**	**Responsibilities**
Xiangrong Xu	Tester	Tensor operations, Built-in functions(from parser to codegen), Relevant Testing
Xincheng Xie	Project Manager	Tensor/Vartensor-related syntax, semant & codegen Tensor/Vartensor runtime checking

		Tensor/Vartensor Indexing
Songqing Ye	PE Programmer	Parallel Environment (from parser to codegen); Test for PE and STMT
Senhong Liu	System Designer	Architecture Design. Basic expression and statement implementation. LLvm translation.

4.7. Environment

Operating Systems: MacOS, Ubuntu 20.04
Languages: OCaml, C++
Compiler: ocamlnative, gcc, clang
Text Editor: VS Code
Version Control: GitHub
Documentation: Google Docs, Latex

4.8. Project Log



5. Architectural Design

5.1. Tensor Structure (Xincheng Xie)

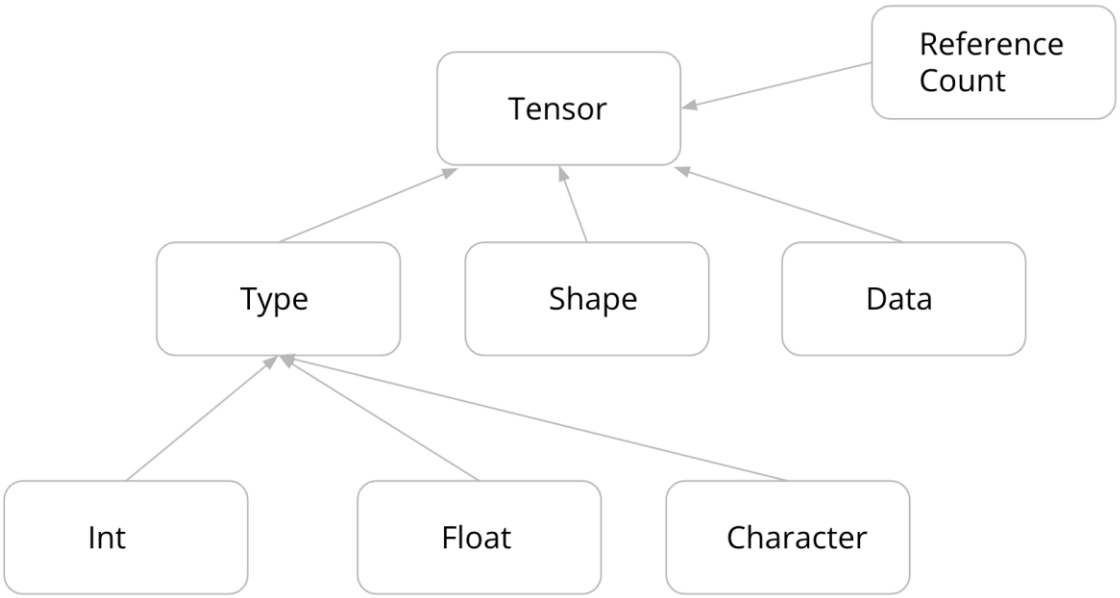
Since everything is a tensor in our language, we first briefly talk about the structure of the tensor. Tensor is a customer structure we defined, who basically have five fields:

1. Type: imply the data type of tensor, we support int, float, var and characters.
2. Ndim: the number of dimensions of the data.
3. Dims: size of each dimension

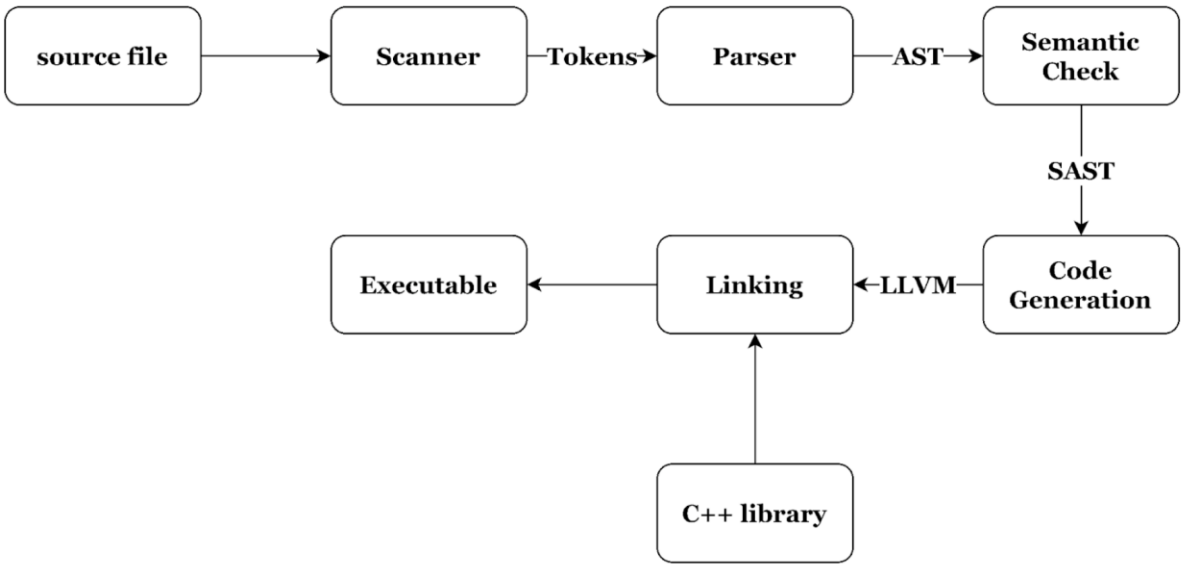
4. Data: pointer points to the data field (allocated in heap)
5. Rc: reference count for garbage collection.

```
typedef struct tensor
{
    int8_t type;
    int8_t ndim;
    int64_t *dims;
    void *data;
    int8_t rc;
} tensor;
```

Above is a high-level overview of the data structure. Integer tensors have type 0; Float tensors have type 1; Char tensors have type 2; Vartensor have type 3. We also define a nil tensor which does not have any data and has magical type 21.



5.2. Project Architecture (Senhong Liu)



5.3. Scanner (Senhong Liu, Songqing Ye)

The scanner (scanner.mll) takes as input a TENLab source program and tokenizes the program as a stream of tokens, e.g., identifiers, keywords and literals, etc. It also ignores the comments words as we described before. The scanner will also report an error if any invalid tokens exist.

5.4. Parser and AST (Senhong Liu, Xiangrong Xu, Xincheng Xie, Songqing Ye)

The parser (parser.mly) takes in the tokens produced by the scanner and uses the rules we defined in AST (ast.ml) to generate an abstract syntax tree. If the tokens can be successfully streaming into the next part, the program is at least syntactically correct. Otherwise, an exception about the parser error will be throwed.
As we mentioned before, everything is a tensor. We define two kinds of tensor:

1. Regular tensor

We parse regular tensors recursively. As a tensor must be braced by '[' and ']', and different sub-tensors in the same dimension must be separated by ','. Here is an abstract rule of how to parse a regular tensor:

```
tensor → [tensor,tensor_list] | [tensor] | 0-dim-tensor
tensor_list → tensor,tensor_list | tensor
```

2. Vartensor

Vartensor's parse is similar to regular one. However, vartensor is fixed to one dimension with various pointers to different tensors. Here is the abstract parse rule:

```
vartensor → [expr,vartensor_list] | [expr]
tensor_list → expr | expr, tensor_list
```

5.5. Semant Check and SAST (Xincheng Xie, Senhong Liu, Xiangrong Xu, Songqing Ye)

The semantic check (semant.ml) takes the abstract syntax tree as input and then decomposes it according to the rules defined in the ast.ml. Since TENLab is a dynamic typed language, we won't take too much type check here, because most of the type check will be conducted in runtime.

But static checking is still necessary for non-var tensors. We will automatically check the dimension recursively. Starting from the 0-dimensional tensor, which is the smallest element of a tensor. In a tensor, multiple 0-dimensional tensors are connected by ',’. If two tensors are connected by ',', they are in the same dimension, and they need to have the same type and the same shape.

Take [[1,2],[3,4]] as the first example, at the bottom end of recursion we must have 1, 2, 3, and 4. They each have shape [-1] and type "int". (Take -1 for simplicity.) 1 and 2 are connected by ',', and they have the same type and shape. We add up the shapes and get 1,2 which has shape [-2] and type "int". Same for 3,4. Then, we encounter '[' and ']'. We increase the number of dimensions: convert the current negative to positive, and add -1 to the front. Therefore, we get [1,2] has shape [-1,2] and type "int". Same for [3,4]. Similarly, [1,2] and [3,4] are connected by ',' and they have the same shape and type, thus [1,2],[3,4] has [-2,2]. We encounter '[' and ']' again: [[1,2],[3,4]] has [-1,2,2]. This semantic check completes and we verify the first one must be -1 and discard it.

Since it's too complicated to support the binary or unary operators with tensors in LLvm, we introduce a C++ library for matrix calculation, i.e., Libtorch. We basically define every possible tensor calculation by calling the built-in functions. More than 20 built-in functions implemented with C++ are supported so far. The key point is since the tensor calculation is actually computing during runtime, after compilation, we can check the type of variables while calling the built-in functions to do the dynamic semantic check. For example, we only support the multiplication between two tensors with the same types. At runtime, if the types of two tensors are inconsistent, an exception will be thrown to warn the users. So all the dynamic semantic checking is done in the built-in functions.

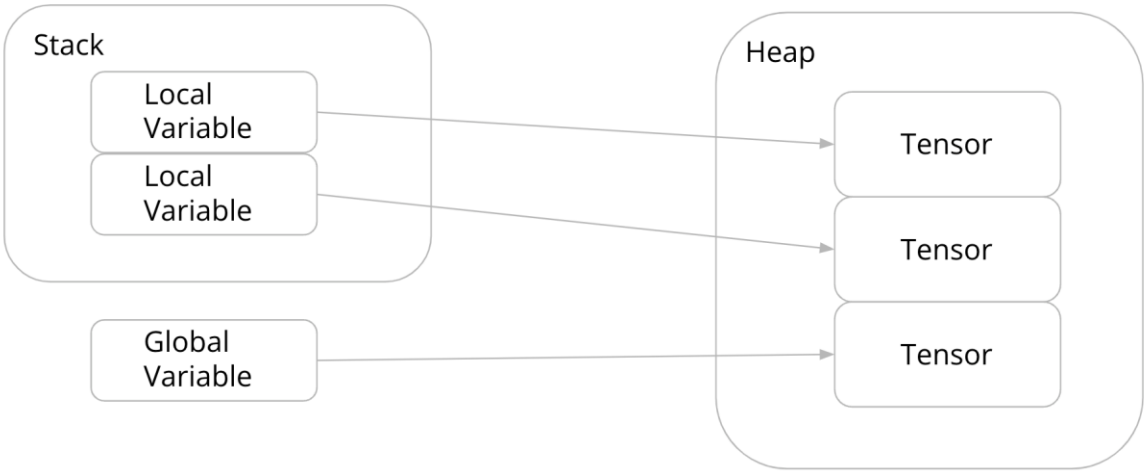
SAST (sast.ml) defines some new rules that bind tensors to inferred types and dimensions. If expressions that are not tensor will be bound with a void dimension and types, which will be ignored in the next part.

If the abstract syntax tree is checked semantically correct through the static semantic check, a semantically-checked abstract tree (sast) will be generated and streamed into the codegen (codegen.ml). Otherwise, a semantic exception will be thrown to the users.

For the parallel environment, semant checking first takes in the correct corresponding parameters as local variables. Besides ordinary checking on statements, it also counts the number of parallel maps, so that we can know how many parameters to pass to the reduce function. Also, it checks if an operator exists. If not, it gives a code generator a hint that it should use default.

5.6. Codegen (Senhong Liu, Xincheng Xie, Songqing Ye, Xiangrong Xu)

The code generator generally traverses the tree to generate code in post-order fashion and translate them into the llvm ir. By traversing the semantically-checked abstract syntax tree (sast), the variables and the function pointers will be added to the symbol table for later reference. We also introduce a customer data structure, namespace, that represents the current environment while traversing sast. Overall, the high level of memory management is shown below.



More tricky things are about the parallel environment, to generate a parallel environment, codegen first uses the same function generator to generate the map and reduce functions, with special names in the form of <PEname>+<Operatorname>+<Mapfunctioname/”reduce”>. Then for every environment, we generate one global function pointer array and one global function pointer. In the main function, before we generate any statements, we first store the value of corresponding function to the global function pointer variables, and when we call an

operator in the environment, instead of calling the operator function, we call the parallel function with the function pointer stored, with that the operator is paralleled.

5.7. C++ Library (Xiangrong Xu, Songqing Ye, Xincheng Xie, Senhong Liu)

We implemented our standard library In C++, and all these external C++ calls are processed and declared in LLVM.

5.7.1. Built-in.cc

As it's very difficult to do tensor operations using only LLVM, we turn to Libtorch, which is PyTorch in C++, to help us complete the tensor calculations.
This file includes all the operations we need to support tensor data type declaration, tensor operations and built-in functions. For each operation, we have a pair of functions. For example *add_t* function means it accept torch::Tensor and returns the output of addition calculated by torch function in tensor* type, while *add* function will be denoted as 'extern' so that llvm could find it. We also check the type and the shape of the tensors to see whether this operation is allowed.

5.7.2. Pe.cc

This is the corresponding function for the parallel environment. This function is fairly neat, it takes in a function pointer array for map functions, the number of map functions, a function pointer for the reduce function, and two operands. The function just parallelly runs the map function and stores the result, then calls the reduce function to return the result.

5.7.3. Gc.cc

We implement a simple version of garbage collection using the method of the reference count. Whenever a tensor is assigned to a variable, we will call the increase_rc function to increase the reference count of the tensor. Similarly, whenever a tensor is de-reference, the reference count will be decreased by 1. Finally, whenever the reference count is 0, the memory space allocated to the tensor will become invalid by calling free().

6. Test Plan

6.1. Test Suite and Automation

We divide our tests into 5 categories, including tensor-test, built-in-test, stmt-test, pe-test and fails, responsible for testing certain features. Tensor-test accounts for arithmetic operations, relational operations and logical operations between tensors. Built-in-test tests the functionality of tensor-related operations, e.g. shape(), cat(), print(). Stmt-test tests check whether control flows work or not, and pe-test can test whether parallel functions run correctly under a user-defined environment. Fails record some illegal operations in naming, using tensors or writing codes.

All test cases are stored in ./tests/ folder and end with the suffix '.tl', which is an abbreviation of TENLab. Success test cases are started with 'test-*', whose output is expected to be conform with the results written in its corresponding '*.out' file. Fail cases are started with 'fail*', and its output should be consistent with the error message pre-recorded in '*.err' file.

Our automation testing is inspired by Micro-C testing. The generate.sh script helps to generate .s file and the test.sh script compiles and runs all of the *.tl files in the /test folder and compares the output to the corresponding *.out file. This script is run when TENLab is compiled using make.

6.2. Sample Test One

In the first sample we implemented quicksort. Here you can see some applications of user-defined function, global variable, and control flow, built-in functions.

```
a = [5,3,9,6,3,2,1,7,8,4];

def partition(A, p, r) {
    n = rand([1]);
    n = int_of((n[0]+1.) .* float_of(r) .* 10.);
    n = (n % (r - p + 1)) + p;
    t = A[n];
    A[n] = A[r];
    A[r] = t;
    x = A[r];
    i = p - 1;
    for (j in p:r:1) {
        if (A[j] <= x) {
            i = i + 1;
            t = A[i];
            A[i] = A[j];
            A[j] = t;
```

```
        }
    }
    t = A[i+1];
    A[i+1] = A[r];
    A[r] = t;
    return i+1;
}

def quicksort(A, p, r) {
    q = 0;
    if (p < r) {
        q = partition(A, p, r);
        quicksort(A, p, q-1);
        quicksort(A, q+1, r);
    }
    return 1;
}

l = shape(a);
quicksort(a, 0, l[0]-1);
print(a);
```

```
; ModuleID = 'TENLab'
source_filename = "TENLab"

%tensor_t = type { i8, i8, i8*, i8*, i8 }

@a = global i8* null
@l = global i8* null

define i8 @main() {
entry:
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 1, i8* %ndims
  %malloccall1 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims = bitcast i8* %malloccall1 to [1 x i64]*
  %dims_as_i8ptr = bitcast [1 x i64]* %dims to i8*
```

```
  %elmptr = getelementptr [1 x i64], [1 x i64]* %dims, i64 0, i64 0
  store i64 10, i64* %elmptr
  %malloccall2 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i32* getelementptr
(i32, i32* null, i32 1) to i64), i64 10) to i32))
  %data = bitcast i8* %malloccall2 to [10 x i32]*
  %data_as_i8ptr = bitcast [10 x i32]* %data to i8*
  %elmptr3 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 0
  store i32 5, i32* %elmptr3
  %elmptr4 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 1
  store i32 3, i32* %elmptr4
  %elmptr5 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 2
  store i32 9, i32* %elmptr5
  %elmptr6 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 3
  store i32 6, i32* %elmptr6
  %elmptr7 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 4
  store i32 3, i32* %elmptr7
  %elmptr8 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 5
  store i32 2, i32* %elmptr8
  %elmptr9 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 6
  store i32 1, i32* %elmptr9
  %elmptr10 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 7
  store i32 7, i32* %elmptr10
  %elmptr11 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 8
  store i32 8, i32* %elmptr11
  %elmptr12 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 9
  store i32 4, i32* %elmptr12
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %lhsptr = load i8*, i8** @a
  call void @increase_rc(i8* %tensor)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %tensor, i8** @a
  %a = load i8*, i8** @a
  %shape = call i8* @shape(i8* %a)
  %lhsptr13 = load i8*, i8** @l
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr13)
  store i8* %shape, i8** @l
  %a14 = load i8*, i8** @a
```

```
  %malloccall15 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor16 = bitcast i8* %malloccall15 to %tensor_t*
  %dtype17 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 0
  store i8 0, i8* %dtype17
  %ndims18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 1
  store i8 0, i8* %ndims18
  %malloccall19 = tail call i8* @malloc(i32 0)
  %dims20 = bitcast i8* %malloccall19 to [0 x i64]*
  %dims_as_i8ptr21 = bitcast [0 x i64]* %dims20 to i8*
  %malloccall22 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data23 = bitcast i8* %malloccall22 to [1 x i32]*
  %data_as_i8ptr24 = bitcast [1 x i32]* %data23 to i8*
  %elmptr25 = getelementptr [1 x i32], [1 x i32]* %data23, i32 0, i32 0
  store i32 0, i32* %elmptr25
  %dimsptr26 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 2
  store i8* %dims_as_i8ptr21, i8** %dimsptr26
  %dataptr27 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 3
  store i8* %data_as_i8ptr24, i8** %dataptr27
  %rc28 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 4
  store i8 0, i8* %rc28
  %tensor29 = bitcast %tensor_t* %raw_tensor16 to i8*
  %malloccall30 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor31 = bitcast i8* %malloccall30 to %tensor_t*
  %dtype32 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 0
  store i8 0, i8* %dtype32
  %ndims33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 1
  store i8 0, i8* %ndims33
  %malloccall34 = tail call i8* @malloc(i32 0)
  %dims35 = bitcast i8* %malloccall34 to [0 x i64]*
  %dims_as_i8ptr36 = bitcast [0 x i64]* %dims35 to i8*
  %malloccall37 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data38 = bitcast i8* %malloccall37 to [1 x i32]*
  %data_as_i8ptr39 = bitcast [1 x i32]* %data38 to i8*
  %elmptr40 = getelementptr [1 x i32], [1 x i32]* %data38, i32 0, i32 0
  store i32 0, i32* %elmptr40
  %dimsptr41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 2
  store i8* %dims_as_i8ptr36, i8** %dimsptr41
  %dataptr42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 3
  store i8* %data_as_i8ptr39, i8** %dataptr42
  %rc43 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 4
```

```
  store i8 0, i8* %rc43
  %tensor44 = bitcast %tensor_t* %raw_tensor31 to i8*
  %malloccall45 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor46 = bitcast i8* %malloccall45 to %tensor_t*
  %dtype47 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 0
  store i8 3, i8* %dtype47
  %ndims48 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 1
  store i8 1, i8* %ndims48
  %malloccall49 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims50 = bitcast i8* %malloccall49 to [1 x i64]*
  %dims_as_i8ptr51 = bitcast [1 x i64]* %dims50 to i8*
  %elmptr52 = getelementptr [1 x i64], [1 x i64]* %dims50, i64 0, i64 0
  store i64 1, i64* %elmptr52
  %malloccall53 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data54 = bitcast i8* %malloccall53 to [1 x i8*]*
  %data_as_i8ptr55 = bitcast [1 x i8*]* %data54 to i8*
  %elmptr56 = getelementptr [1 x i8*], [1 x i8*]* %data54, i64 0, i64 0
  store i8* %tensor44, i8** %elmptr56
  %dimsptr57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 2
  store i8* %dims_as_i8ptr51, i8** %dimsptr57
  %dataptr58 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 3
  store i8* %data_as_i8ptr55, i8** %dataptr58
  %rc59 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 4
  store i8 0, i8* %rc59
  %tensor60 = bitcast %tensor_t* %raw_tensor46 to i8*
  %l = load i8*, i8** @l
  %access_tensor = call i8* @index_get(i8* %l, i8* %tensor60)
  %malloccall61 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor62 = bitcast i8* %malloccall61 to %tensor_t*
  %dtype63 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 0
  store i8 0, i8* %dtype63
  %ndims64 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 1
  store i8 0, i8* %ndims64
  %malloccall65 = tail call i8* @malloc(i32 0)
  %dims66 = bitcast i8* %malloccall65 to [0 x i64]*
  %dims_as_i8ptr67 = bitcast [0 x i64]* %dims66 to i8*
  %malloccall68 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data69 = bitcast i8* %malloccall68 to [1 x i32]*
  %data_as_i8ptr70 = bitcast [1 x i32]* %data69 to i8*
  %elmptr71 = getelementptr [1 x i32], [1 x i32]* %data69, i32 0, i32 0
```

```
  store i32 1, i32* %elmptr71
  %dimsptr72 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 2
  store i8* %dims_as_i8ptr67, i8** %dimsptr72
  %dataptr73 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 3
  store i8* %data_as_i8ptr70, i8** %dataptr73
  %rc74 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 4
  store i8 0, i8* %rc74
  %tensor75 = bitcast %tensor_t* %raw_tensor62 to i8*
  %tmpOp = call i8* @subtract(i8* %access_tensor, i8* %tensor75)
  %ret = call i8* @quicksort(i8* %a14, i8* %tensor29, i8* %tmpOp)
  %a76 = load i8*, i8** @a
  call void @print(i8* %a76)
  ret i8 0
}

declare i8* @add(i8*, i8*)

declare i8* @subtract(i8*, i8*)

declare i8* @negative(i8*)

declare i8* @mult(i8*, i8*)

declare i8* @dotmul(i8*, i8*)

declare i8* @divide(i8*, i8*)

declare i8* @floordivide(i8*, i8*)

declare i8* @matpow(i8*, i8*)

declare i8* @dotpow(i8*, i8*)

declare i8* @mod(i8*, i8*)

declare i8* @transpose(i8*)

declare i8* @equal(i8*, i8*)

declare i8* @notequal(i8*, i8*)

declare i8* @greater(i8*, i8*)

declare i8* @greaterequal(i8*, i8*)
```

```
declare i8* @less(i8*, i8*)

declare i8* @lessequal(i8*, i8*)

declare i8* @range(i8*, i8*, i8*)

declare void @print(i8*)

declare i8* @logicaland(i8*, i8*)

declare i8* @logicalor(i8*, i8*)

declare i8* @logicalnot(i8*)

declare i32 @len(i8*)

declare i8* @zeros(i8*)

declare i8* @cat(i8*, i8*, i8*)

declare i8* @shape(i8*)

declare i8* @ones(i8*)

declare i8* @tensor_rand(i8*)

declare i8* @sum(i8*, i8*)

declare i8* @any(i8*)

declare i8* @all(i8*)

declare i8* @tensor_abs(i8*)

declare i8* @tensor_log(i8*)

declare i8* @tensor_floor(i8*)

declare i8* @tensor_ceil(i8*)

declare i8* @tensor_round(i8*)

declare i8* @int_of(i8*)
```

```
declare i8* @float_of(i8*)

declare i8* @inverse(i8*)

declare i1 @bool_of_zero(i8*)

declare i8* @pe_calc(i8* (i8*, i8*)**, i32, i8* (i8**)*, i8*, i8*)

declare i8* @index_get(i8*, i8*)

declare i8* @index_get_int(i8*, i32)

declare void @index_put(i8*, i8*, i8*)

declare void @increase_rc(i8*)

declare void @decrease_rc(i8*)

declare noalias i8* @malloc(i32)

define i8* @partition(i8* %0, i8* %1, i8* %2) {
entry:
  %A = alloca i8*
  store i8* %0, i8** %A
  %p = alloca i8*
  store i8* %1, i8** %p
  %r = alloca i8*
  store i8* %2, i8** %r
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 1, i8* %ndims
  %malloccall1 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims = bitcast i8* %malloccall1 to [1 x i64]*
  %dims_as_i8ptr = bitcast [1 x i64]* %dims to i8*
  %elmptr = getelementptr [1 x i64], [1 x i64]* %dims, i64 0, i64 0
  store i64 1, i64* %elmptr
  %malloccall2 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall2 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
```

```
  %elmptr3 = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 1, i32* %elmptr3
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %rand = call i8* @tensor_rand(i8* %tensor)
  %n = alloca i8*
  store i8* null, i8** %n
  %lhsptr = load i8*, i8** %n
  call void @increase_rc(i8* %rand)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %rand, i8** %n
  %malloccall4 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor5 = bitcast i8* %malloccall4 to %tensor_t*
  %dtype6 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 0
  store i8 0, i8* %dtype6
  %ndims7 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 1
  store i8 0, i8* %ndims7
  %malloccall8 = tail call i8* @malloc(i32 0)
  %dims9 = bitcast i8* %malloccall8 to [0 x i64]*
  %dims_as_i8ptr10 = bitcast [0 x i64]* %dims9 to i8*
  %malloccall11 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data12 = bitcast i8* %malloccall11 to [1 x i32]*
  %data_as_i8ptr13 = bitcast [1 x i32]* %data12 to i8*
  %elmptr14 = getelementptr [1 x i32], [1 x i32]* %data12, i32 0, i32 0
  store i32 0, i32* %elmptr14
  %dimsptr15 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 2
  store i8* %dims_as_i8ptr10, i8** %dimsptr15
  %dataptr16 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 3
  store i8* %data_as_i8ptr13, i8** %dataptr16
  %rc17 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 4
  store i8 0, i8* %rc17
  %tensor18 = bitcast %tensor_t* %raw_tensor5 to i8*
  %malloccall19 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor20 = bitcast i8* %malloccall19 to %tensor_t*
  %dtype21 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 0
  store i8 3, i8* %dtype21
```

```
  %ndims22 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 1
  store i8 1, i8* %ndims22
  %malloccall23 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims24 = bitcast i8* %malloccall23 to [1 x i64]*
  %dims_as_i8ptr25 = bitcast [1 x i64]* %dims24 to i8*
  %elmptr26 = getelementptr [1 x i64], [1 x i64]* %dims24, i64 0, i64 0
  store i64 1, i64* %elmptr26
  %malloccall27 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data28 = bitcast i8* %malloccall27 to [1 x i8*]*
  %data_as_i8ptr29 = bitcast [1 x i8*]* %data28 to i8*
  %elmptr30 = getelementptr [1 x i8*], [1 x i8*]* %data28, i64 0, i64 0
  store i8* %tensor18, i8** %elmptr30
  %dimsptr31 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 2
  store i8* %dims_as_i8ptr25, i8** %dimsptr31
  %dataptr32 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 3
  store i8* %data_as_i8ptr29, i8** %dataptr32
  %rc33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 4
  store i8 0, i8* %rc33
  %tensor34 = bitcast %tensor_t* %raw_tensor20 to i8*
  %n35 = load i8*, i8** %n
  %access_tensor = call i8* @index_get(i8* %n35, i8* %tensor34)
  %malloccall36 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor37 = bitcast i8* %malloccall36 to %tensor_t*
  %dtype38 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 0
  store i8 1, i8* %dtype38
  %ndims39 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 1
  store i8 0, i8* %ndims39
  %malloccall40 = tail call i8* @malloc(i32 0)
  %dims41 = bitcast i8* %malloccall40 to [0 x i64]*
  %dims_as_i8ptr42 = bitcast [0 x i64]* %dims41 to i8*
  %malloccall43 = tail call i8* @malloc(i32 ptrtoint (double* getelementptr (double, double* null,
i32 1) to i32))
  %data44 = bitcast i8* %malloccall43 to [1 x double]*
  %data_as_i8ptr45 = bitcast [1 x double]* %data44 to i8*
  %elmptr46 = getelementptr [1 x double], [1 x double]* %data44, i64 0, i64 0
  store double 1.000000e+00, double* %elmptr46
  %dimsptr47 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 2
  store i8* %dims_as_i8ptr42, i8** %dimsptr47
  %dataptr48 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 3
  store i8* %data_as_i8ptr45, i8** %dataptr48
  %rc49 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 4
  store i8 0, i8* %rc49
```

```
  %tensor50 = bitcast %tensor_t* %raw_tensor37 to i8*
  %tmpOp = call i8* @add(i8* %access_tensor, i8* %tensor50)
  %r51 = load i8*, i8** %r
  %float_of = call i8* @float_of(i8* %r51)
  %tmpOp52 = call i8* @dotmul(i8* %tmpOp, i8* %float_of)
  %malloccall53 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor54 = bitcast i8* %malloccall53 to %tensor_t*
  %dtype55 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 0
  store i8 1, i8* %dtype55
  %ndims56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 1
  store i8 0, i8* %ndims56
  %malloccall57 = tail call i8* @malloc(i32 0)
  %dims58 = bitcast i8* %malloccall57 to [0 x i64]*
  %dims_as_i8ptr59 = bitcast [0 x i64]* %dims58 to i8*
  %malloccall60 = tail call i8* @malloc(i32 ptrtoint (double* getelementptr (double, double* null,
i32 1) to i32))
  %data61 = bitcast i8* %malloccall60 to [1 x double]*
  %data_as_i8ptr62 = bitcast [1 x double]* %data61 to i8*
  %elmptr63 = getelementptr [1 x double], [1 x double]* %data61, i64 0, i64 0
  store double 1.000000e+01, double* %elmptr63
  %dimsptr64 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 2
  store i8* %dims_as_i8ptr59, i8** %dimsptr64
  %dataptr65 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 3
  store i8* %data_as_i8ptr62, i8** %dataptr65
  %rc66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 4
  store i8 0, i8* %rc66
  %tensor67 = bitcast %tensor_t* %raw_tensor54 to i8*
  %tmpOp68 = call i8* @dotmul(i8* %tmpOp52, i8* %tensor67)
  %int_of = call i8* @int_of(i8* %tmpOp68)
  %lhsptr69 = load i8*, i8** %n
  call void @increase_rc(i8* %int_of)
  call void @decrease_rc(i8* %lhsptr69)
  store i8* %int_of, i8** %n
  %n70 = load i8*, i8** %n
  %r71 = load i8*, i8** %r
  %p72 = load i8*, i8** %p
  %tmpOp73 = call i8* @subtract(i8* %r71, i8* %p72)
  %malloccall74 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor75 = bitcast i8* %malloccall74 to %tensor_t*
  %dtype76 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 0
  store i8 0, i8* %dtype76
  %ndims77 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 1
```

```
  store i8 0, i8* %ndims77
  %malloccall78 = tail call i8* @malloc(i32 0)
  %dims79 = bitcast i8* %malloccall78 to [0 x i64]*
  %dims_as_i8ptr80 = bitcast [0 x i64]* %dims79 to i8*
  %malloccall81 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data82 = bitcast i8* %malloccall81 to [1 x i32]*
  %data_as_i8ptr83 = bitcast [1 x i32]* %data82 to i8*
  %elmptr84 = getelementptr [1 x i32], [1 x i32]* %data82, i32 0, i32 0
  store i32 1, i32* %elmptr84
  %dimsptr85 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 2
  store i8* %dims_as_i8ptr80, i8** %dimsptr85
  %dataptr86 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 3
  store i8* %data_as_i8ptr83, i8** %dataptr86
  %rc87 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 4
  store i8 0, i8* %rc87
  %tensor88 = bitcast %tensor_t* %raw_tensor75 to i8*
  %tmpOp89 = call i8* @add(i8* %tmpOp73, i8* %tensor88)
  %tmpOp90 = call i8* @mod(i8* %n70, i8* %tmpOp89)
  %p91 = load i8*, i8** %p
  %tmpOp92 = call i8* @add(i8* %tmpOp90, i8* %p91)
  %lhsptr93 = load i8*, i8** %n
  call void @increase_rc(i8* %tmpOp92)
  call void @decrease_rc(i8* %lhsptr93)
  store i8* %tmpOp92, i8** %n
  %n94 = load i8*, i8** %n
  %malloccall95 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor96 = bitcast i8* %malloccall95 to %tensor_t*
  %dtype97 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 0
  store i8 3, i8* %dtype97
  %ndims98 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 1
  store i8 1, i8* %ndims98
  %malloccall99 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims100 = bitcast i8* %malloccall99 to [1 x i64]*
  %dims_as_i8ptr101 = bitcast [1 x i64]* %dims100 to i8*
  %elmptr102 = getelementptr [1 x i64], [1 x i64]* %dims100, i64 0, i64 0
  store i64 1, i64* %elmptr102
  %malloccall103 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data104 = bitcast i8* %malloccall103 to [1 x i8*]*
  %data_as_i8ptr105 = bitcast [1 x i8*]* %data104 to i8*
  %elmptr106 = getelementptr [1 x i8*], [1 x i8*]* %data104, i64 0, i64 0
```

```
  store i8* %n94, i8** %elmptr106
  %dimsptr107 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 2
  store i8* %dims_as_i8ptr101, i8** %dimsptr107
  %dataptr108 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 3
  store i8* %data_as_i8ptr105, i8** %dataptr108
  %rc109 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 4
  store i8 0, i8* %rc109
  %tensor110 = bitcast %tensor_t* %raw_tensor96 to i8*
  %A111 = load i8*, i8** %A
  %access_tensor112 = call i8* @index_get(i8* %A111, i8* %tensor110)
  %t = alloca i8*
  store i8* null, i8** %t
  %lhsptr113 = load i8*, i8** %t
  call void @increase_rc(i8* %access_tensor112)
  call void @decrease_rc(i8* %lhsptr113)
  store i8* %access_tensor112, i8** %t
  %r114 = load i8*, i8** %r
  %malloccall115 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor116 = bitcast i8* %malloccall115 to %tensor_t*
  %dtype117 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 0
  store i8 3, i8* %dtype117
  %ndims118 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 1
  store i8 1, i8* %ndims118
  %malloccall119 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims120 = bitcast i8* %malloccall119 to [1 x i64]*
  %dims_as_i8ptr121 = bitcast [1 x i64]* %dims120 to i8*
  %elmptr122 = getelementptr [1 x i64], [1 x i64]* %dims120, i64 0, i64 0
  store i64 1, i64* %elmptr122
  %malloccall123 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data124 = bitcast i8* %malloccall123 to [1 x i8*]*
  %data_as_i8ptr125 = bitcast [1 x i8*]* %data124 to i8*
  %elmptr126 = getelementptr [1 x i8*], [1 x i8*]* %data124, i64 0, i64 0
  store i8* %r114, i8** %elmptr126
  %dimsptr127 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 2
  store i8* %dims_as_i8ptr121, i8** %dimsptr127
  %dataptr128 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 3
  store i8* %data_as_i8ptr125, i8** %dataptr128
  %rc129 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 4
  store i8 0, i8* %rc129
  %tensor130 = bitcast %tensor_t* %raw_tensor116 to i8*
  %A131 = load i8*, i8** %A
```

```
  %access_tensor132 = call i8* @index_get(i8* %A131, i8* %tensor130)
  %n133 = load i8*, i8** %n
  %malloccall134 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor135 = bitcast i8* %malloccall134 to %tensor_t*
  %dtype136 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 0
  store i8 3, i8* %dtype136
  %ndims137 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 1
  store i8 1, i8* %ndims137
  %malloccall138 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims139 = bitcast i8* %malloccall138 to [1 x i64]*
  %dims_as_i8ptr140 = bitcast [1 x i64]* %dims139 to i8*
  %elmptr141 = getelementptr [1 x i64], [1 x i64]* %dims139, i64 0, i64 0
  store i64 1, i64* %elmptr141
  %malloccall142 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data143 = bitcast i8* %malloccall142 to [1 x i8*]*
  %data_as_i8ptr144 = bitcast [1 x i8*]* %data143 to i8*
  %elmptr145 = getelementptr [1 x i8*], [1 x i8*]* %data143, i64 0, i64 0
  store i8* %n133, i8** %elmptr145
  %dimsptr146 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 2
  store i8* %dims_as_i8ptr140, i8** %dimsptr146
  %dataptr147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 3
  store i8* %data_as_i8ptr144, i8** %dataptr147
  %rc148 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 4
  store i8 0, i8* %rc148
  %tensor149 = bitcast %tensor_t* %raw_tensor135 to i8*
  %A150 = load i8*, i8** %A
  call void @index_put(i8* %A150, i8* %tensor149, i8* %access_tensor132)
  %t151 = load i8*, i8** %t
  %r152 = load i8*, i8** %r
  %malloccall153 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor154 = bitcast i8* %malloccall153 to %tensor_t*
  %dtype155 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 0
  store i8 3, i8* %dtype155
  %ndims156 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 1
  store i8 1, i8* %ndims156
  %malloccall157 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims158 = bitcast i8* %malloccall157 to [1 x i64]*
  %dims_as_i8ptr159 = bitcast [1 x i64]* %dims158 to i8*
  %elmptr160 = getelementptr [1 x i64], [1 x i64]* %dims158, i64 0, i64 0
```

```
  store i64 1, i64* %elmptr160
  %malloccall161 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data162 = bitcast i8* %malloccall161 to [1 x i8*]*
  %data_as_i8ptr163 = bitcast [1 x i8*]* %data162 to i8*
  %elmptr164 = getelementptr [1 x i8*], [1 x i8*]* %data162, i64 0, i64 0
  store i8* %r152, i8** %elmptr164
  %dimsptr165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 2
  store i8* %dims_as_i8ptr159, i8** %dimsptr165
  %dataptr166 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 3
  store i8* %data_as_i8ptr163, i8** %dataptr166
  %rc167 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 4
  store i8 0, i8* %rc167
  %tensor168 = bitcast %tensor_t* %raw_tensor154 to i8*
  %A169 = load i8*, i8** %A
  call void @index_put(i8* %A169, i8* %tensor168, i8* %t151)
  %r170 = load i8*, i8** %r
  %malloccall171 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor172 = bitcast i8* %malloccall171 to %tensor_t*
  %dtype173 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 0
  store i8 3, i8* %dtype173
  %ndims174 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 1
  store i8 1, i8* %ndims174
  %malloccall175 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims176 = bitcast i8* %malloccall175 to [1 x i64]*
  %dims_as_i8ptr177 = bitcast [1 x i64]* %dims176 to i8*
  %elmptr178 = getelementptr [1 x i64], [1 x i64]* %dims176, i64 0, i64 0
  store i64 1, i64* %elmptr178
  %malloccall179 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data180 = bitcast i8* %malloccall179 to [1 x i8*]*
  %data_as_i8ptr181 = bitcast [1 x i8*]* %data180 to i8*
  %elmptr182 = getelementptr [1 x i8*], [1 x i8*]* %data180, i64 0, i64 0
  store i8* %r170, i8** %elmptr182
  %dimsptr183 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 2
  store i8* %dims_as_i8ptr177, i8** %dimsptr183
  %dataptr184 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 3
  store i8* %data_as_i8ptr181, i8** %dataptr184
  %rc185 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 4
  store i8 0, i8* %rc185
  %tensor186 = bitcast %tensor_t* %raw_tensor172 to i8*
  %A187 = load i8*, i8** %A
```

```
  %access_tensor188 = call i8* @index_get(i8* %A187, i8* %tensor186)
  %x = alloca i8*
  store i8* null, i8** %x
  %lhsptr189 = load i8*, i8** %x
  call void @increase_rc(i8* %access_tensor188)
  call void @decrease_rc(i8* %lhsptr189)
  store i8* %access_tensor188, i8** %x
  %p190 = load i8*, i8** %p
  %malloccall191 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor192 = bitcast i8* %malloccall191 to %tensor_t*
  %dtype193 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 0
  store i8 0, i8* %dtype193
  %ndims194 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 1
  store i8 0, i8* %ndims194
  %malloccall195 = tail call i8* @malloc(i32 0)
  %dims196 = bitcast i8* %malloccall195 to [0 x i64]*
  %dims_as_i8ptr197 = bitcast [0 x i64]* %dims196 to i8*
  %malloccall198 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data199 = bitcast i8* %malloccall198 to [1 x i32]*
  %data_as_i8ptr200 = bitcast [1 x i32]* %data199 to i8*
  %elmptr201 = getelementptr [1 x i32], [1 x i32]* %data199, i32 0, i32 0
  store i32 1, i32* %elmptr201
  %dimsptr202 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 2
  store i8* %dims_as_i8ptr197, i8** %dimsptr202
  %dataptr203 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 3
  store i8* %data_as_i8ptr200, i8** %dataptr203
  %rc204 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 4
  store i8 0, i8* %rc204
  %tensor205 = bitcast %tensor_t* %raw_tensor192 to i8*
  %tmpOp206 = call i8* @subtract(i8* %p190, i8* %tensor205)
  %i = alloca i8*
  store i8* null, i8** %i
  %lhsptr207 = load i8*, i8** %i
  call void @increase_rc(i8* %tmpOp206)
  call void @decrease_rc(i8* %lhsptr207)
  store i8* %tmpOp206, i8** %i
  %p208 = load i8*, i8** %p
  call void @print(i8* %p208)
  %r209 = load i8*, i8** %r
  call void @print(i8* %r209)
  %p210 = load i8*, i8** %p
  %r211 = load i8*, i8** %r
```

```
  %malloccall212 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor213 = bitcast i8* %malloccall212 to %tensor_t*
  %dtype214 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor213, i32 0, i32 0
  store i8 0, i8* %dtype214
  %ndims215 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor213, i32 0, i32 1
  store i8 0, i8* %ndims215
  %malloccall216 = tail call i8* @malloc(i32 0)
  %dims217 = bitcast i8* %malloccall216 to [0 x i64]*
  %dims_as_i8ptr218 = bitcast [0 x i64]* %dims217 to i8*
  %malloccall219 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data220 = bitcast i8* %malloccall219 to [1 x i32]*
  %data_as_i8ptr221 = bitcast [1 x i32]* %data220 to i8*
  %elmptr222 = getelementptr [1 x i32], [1 x i32]* %data220, i32 0, i32 0
  store i32 1, i32* %elmptr222
  %dimsptr223 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor213, i32 0, i32 2
  store i8* %dims_as_i8ptr218, i8** %dimsptr223
  %dataptr224 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor213, i32 0, i32 3
  store i8* %data_as_i8ptr221, i8** %dataptr224
  %rc225 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor213, i32 0, i32 4
  store i8 0, i8* %rc225
  %tensor226 = bitcast %tensor_t* %raw_tensor213 to i8*
  %tmpOp227 = call i8* @range(i8* %p210, i8* %r211, i8* %tensor226)
  %tmp = alloca i8*
  store i8* null, i8** %tmp
  %lhsptr228 = load i8*, i8** %tmp
  call void @increase_rc(i8* %tmpOp227)
  call void @decrease_rc(i8* %lhsptr228)
  store i8* %tmpOp227, i8** %tmp
  %tmp229 = load i8*, i8** %tmp
  call void @print(i8* %tmp229)
  %idxptr = alloca i32
  store i32 0, i32* %idxptr
  %p230 = load i8*, i8** %p
  %r231 = load i8*, i8** %r
  %malloccall232 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor233 = bitcast i8* %malloccall232 to %tensor_t*
  %dtype234 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor233, i32 0, i32 0
  store i8 0, i8* %dtype234
  %ndims235 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor233, i32 0, i32 1
  store i8 0, i8* %ndims235
  %malloccall236 = tail call i8* @malloc(i32 0)
```

```
  %dims237 = bitcast i8* %malloccall236 to [0 x i64]*
  %dims_as_i8ptr238 = bitcast [0 x i64]* %dims237 to i8*
  %malloccall239 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data240 = bitcast i8* %malloccall239 to [1 x i32]*
  %data_as_i8ptr241 = bitcast [1 x i32]* %data240 to i8*
  %elmptr242 = getelementptr [1 x i32], [1 x i32]* %data240, i32 0, i32 0
  store i32 1, i32* %elmptr242
  %dimsptr243 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor233, i32 0, i32 2
  store i8* %dims_as_i8ptr238, i8** %dimsptr243
  %dataptr244 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor233, i32 0, i32 3
  store i8* %data_as_i8ptr241, i8** %dataptr244
  %rc245 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor233, i32 0, i32 4
  store i8 0, i8* %rc245
  %tensor246 = bitcast %tensor_t* %raw_tensor233 to i8*
  %tmpOp247 = call i8* @range(i8* %p230, i8* %r231, i8* %tensor246)
  %length = call i32 @len(i8* %tmpOp247)
  br label %for

for:                                              ; preds = %merge, %entry
  %idx = load i32, i32* %idxptr
  %new_idx = add i32 %idx, 1
  store i32 %new_idx, i32* %idxptr
  %condition = icmp sgt i32 %new_idx, %length
  br i1 %condition, label %merge363, label %for_body

for_body:                                         ; preds = %for
  %j = alloca i8*
  store i8* null, i8** %j
  %indicator_as_tensor = call i8* @index_get_int(i8* %tmpOp247, i32 %idx)
  store i8* %indicator_as_tensor, i8** %j
  %j248 = load i8*, i8** %j
  %malloccall249 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor250 = bitcast i8* %malloccall249 to %tensor_t*
  %dtype251 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor250, i32 0, i32 0
  store i8 3, i8* %dtype251
  %ndims252 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor250, i32 0, i32 1
  store i8 1, i8* %ndims252
  %malloccall253 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims254 = bitcast i8* %malloccall253 to [1 x i64]*
  %dims_as_i8ptr255 = bitcast [1 x i64]* %dims254 to i8*
  %elmptr256 = getelementptr [1 x i64], [1 x i64]* %dims254, i64 0, i64 0
```

```
  store i64 1, i64* %elmptr256
  %malloccall257 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data258 = bitcast i8* %malloccall257 to [1 x i8*]*
  %data_as_i8ptr259 = bitcast [1 x i8*]* %data258 to i8*
  %elmptr260 = getelementptr [1 x i8*], [1 x i8*]* %data258, i64 0, i64 0
  store i8* %j248, i8** %elmptr260
  %dimsptr261 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor250, i32 0, i32 2
  store i8* %dims_as_i8ptr255, i8** %dimsptr261
  %dataptr262 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor250, i32 0, i32 3
  store i8* %data_as_i8ptr259, i8** %dataptr262
  %rc263 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor250, i32 0, i32 4
  store i8 0, i8* %rc263
  %tensor264 = bitcast %tensor_t* %raw_tensor250 to i8*
  %A265 = load i8*, i8** %A
  %access_tensor266 = call i8* @index_get(i8* %A265, i8* %tensor264)
  %x267 = load i8*, i8** %x
  %tmpOp268 = call i8* @lessequal(i8* %access_tensor266, i8* %x267)
  %bool = call i1 @bool_of_zero(i8* %tmpOp268)
  br i1 %bool, label %then, label %else

merge:                                            ; preds = %else, %then
  br label %for

then:                                             ; preds = %for_body
  %i269 = load i8*, i8** %i
  %malloccall270 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor271 = bitcast i8* %malloccall270 to %tensor_t*
  %dtype272 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 0
  store i8 0, i8* %dtype272
  %ndims273 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 1
  store i8 0, i8* %ndims273
  %malloccall274 = tail call i8* @malloc(i32 0)
  %dims275 = bitcast i8* %malloccall274 to [0 x i64]*
  %dims_as_i8ptr276 = bitcast [0 x i64]* %dims275 to i8*
  %malloccall277 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data278 = bitcast i8* %malloccall277 to [1 x i32]*
  %data_as_i8ptr279 = bitcast [1 x i32]* %data278 to i8*
  %elmptr280 = getelementptr [1 x i32], [1 x i32]* %data278, i32 0, i32 0
  store i32 1, i32* %elmptr280
  %dimsptr281 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 2
  store i8* %dims_as_i8ptr276, i8** %dimsptr281
```

```
  %dataptr282 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 3
  store i8* %data_as_i8ptr279, i8** %dataptr282
  %rc283 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 4
  store i8 0, i8* %rc283
  %tensor284 = bitcast %tensor_t* %raw_tensor271 to i8*
  %tmpOp285 = call i8* @add(i8* %i269, i8* %tensor284)
  %lhsptr286 = load i8*, i8** %i
  call void @increase_rc(i8* %tmpOp285)
  call void @decrease_rc(i8* %lhsptr286)
  store i8* %tmpOp285, i8** %i
  %i287 = load i8*, i8** %i
  %malloccall288 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor289 = bitcast i8* %malloccall288 to %tensor_t*
  %dtype290 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 0
  store i8 3, i8* %dtype290
  %ndims291 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 1
  store i8 1, i8* %ndims291
  %malloccall292 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims293 = bitcast i8* %malloccall292 to [1 x i64]*
  %dims_as_i8ptr294 = bitcast [1 x i64]* %dims293 to i8*
  %elmptr295 = getelementptr [1 x i64], [1 x i64]* %dims293, i64 0, i64 0
  store i64 1, i64* %elmptr295
  %malloccall296 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data297 = bitcast i8* %malloccall296 to [1 x i8*]*
  %data_as_i8ptr298 = bitcast [1 x i8*]* %data297 to i8*
  %elmptr299 = getelementptr [1 x i8*], [1 x i8*]* %data297, i64 0, i64 0
  store i8* %i287, i8** %elmptr299
  %dimsptr300 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 2
  store i8* %dims_as_i8ptr294, i8** %dimsptr300
  %dataptr301 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 3
  store i8* %data_as_i8ptr298, i8** %dataptr301
  %rc302 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 4
  store i8 0, i8* %rc302
  %tensor303 = bitcast %tensor_t* %raw_tensor289 to i8*
  %A304 = load i8*, i8** %A
  %access_tensor305 = call i8* @index_get(i8* %A304, i8* %tensor303)
  %lhsptr306 = load i8*, i8** %t
  call void @increase_rc(i8* %access_tensor305)
  call void @decrease_rc(i8* %lhsptr306)
  store i8* %access_tensor305, i8** %t
  %j307 = load i8*, i8** %j
```

```
  %malloccall308 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor309 = bitcast i8* %malloccall308 to %tensor_t*
  %dtype310 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor309, i32 0, i32 0
  store i8 3, i8* %dtype310
  %ndims311 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor309, i32 0, i32 1
  store i8 1, i8* %ndims311
  %malloccall312 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims313 = bitcast i8* %malloccall312 to [1 x i64]*
  %dims_as_i8ptr314 = bitcast [1 x i64]* %dims313 to i8*
  %elmptr315 = getelementptr [1 x i64], [1 x i64]* %dims313, i64 0, i64 0
  store i64 1, i64* %elmptr315
  %malloccall316 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data317 = bitcast i8* %malloccall316 to [1 x i8*]*
  %data_as_i8ptr318 = bitcast [1 x i8*]* %data317 to i8*
  %elmptr319 = getelementptr [1 x i8*], [1 x i8*]* %data317, i64 0, i64 0
  store i8* %j307, i8** %elmptr319
  %dimsptr320 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor309, i32 0, i32 2
  store i8* %dims_as_i8ptr314, i8** %dimsptr320
  %dataptr321 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor309, i32 0, i32 3
  store i8* %data_as_i8ptr318, i8** %dataptr321
  %rc322 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor309, i32 0, i32 4
  store i8 0, i8* %rc322
  %tensor323 = bitcast %tensor_t* %raw_tensor309 to i8*
  %A324 = load i8*, i8** %A
  %access_tensor325 = call i8* @index_get(i8* %A324, i8* %tensor323)
  %i326 = load i8*, i8** %i
  %malloccall327 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor328 = bitcast i8* %malloccall327 to %tensor_t*
  %dtype329 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor328, i32 0, i32 0
  store i8 3, i8* %dtype329
  %ndims330 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor328, i32 0, i32 1
  store i8 1, i8* %ndims330
  %malloccall331 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims332 = bitcast i8* %malloccall331 to [1 x i64]*
  %dims_as_i8ptr333 = bitcast [1 x i64]* %dims332 to i8*
  %elmptr334 = getelementptr [1 x i64], [1 x i64]* %dims332, i64 0, i64 0
  store i64 1, i64* %elmptr334
  %malloccall335 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
```

```
  %data336 = bitcast i8* %malloccall335 to [1 x i8*]*
  %data_as_i8ptr337 = bitcast [1 x i8*]* %data336 to i8*
  %elmptr338 = getelementptr [1 x i8*], [1 x i8*]* %data336, i64 0, i64 0
  store i8* %i326, i8** %elmptr338
  %dimsptr339 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor328, i32 0, i32 2
  store i8* %dims_as_i8ptr333, i8** %dimsptr339
  %dataptr340 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor328, i32 0, i32 3
  store i8* %data_as_i8ptr337, i8** %dataptr340
  %rc341 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor328, i32 0, i32 4
  store i8 0, i8* %rc341
  %tensor342 = bitcast %tensor_t* %raw_tensor328 to i8*
  %A343 = load i8*, i8** %A
  call void @index_put(i8* %A343, i8* %tensor342, i8* %access_tensor325)
  %t344 = load i8*, i8** %t
  %j345 = load i8*, i8** %j
  %malloccall346 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor347 = bitcast i8* %malloccall346 to %tensor_t*
  %dtype348 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor347, i32 0, i32 0
  store i8 3, i8* %dtype348
  %ndims349 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor347, i32 0, i32 1
  store i8 1, i8* %ndims349
  %malloccall350 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims351 = bitcast i8* %malloccall350 to [1 x i64]*
  %dims_as_i8ptr352 = bitcast [1 x i64]* %dims351 to i8*
  %elmptr353 = getelementptr [1 x i64], [1 x i64]* %dims351, i64 0, i64 0
  store i64 1, i64* %elmptr353
  %malloccall354 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data355 = bitcast i8* %malloccall354 to [1 x i8*]*
  %data_as_i8ptr356 = bitcast [1 x i8*]* %data355 to i8*
  %elmptr357 = getelementptr [1 x i8*], [1 x i8*]* %data355, i64 0, i64 0
  store i8* %j345, i8** %elmptr357
  %dimsptr358 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor347, i32 0, i32 2
  store i8* %dims_as_i8ptr352, i8** %dimsptr358
  %dataptr359 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor347, i32 0, i32 3
  store i8* %data_as_i8ptr356, i8** %dataptr359
  %rc360 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor347, i32 0, i32 4
  store i8 0, i8* %rc360
  %tensor361 = bitcast %tensor_t* %raw_tensor347 to i8*
  %A362 = load i8*, i8** %A
  call void @index_put(i8* %A362, i8* %tensor361, i8* %t344)
  br label %merge
```

```
else:                                             ; preds = %for_body
  br label %merge

merge363:                                         ; preds = %for
  %i364 = load i8*, i8** %i
  %malloccall365 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor366 = bitcast i8* %malloccall365 to %tensor_t*
  %dtype367 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor366, i32 0, i32 0
  store i8 0, i8* %dtype367
  %ndims368 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor366, i32 0, i32 1
  store i8 0, i8* %ndims368
  %malloccall369 = tail call i8* @malloc(i32 0)
  %dims370 = bitcast i8* %malloccall369 to [0 x i64]*
  %dims_as_i8ptr371 = bitcast [0 x i64]* %dims370 to i8*
  %malloccall372 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data373 = bitcast i8* %malloccall372 to [1 x i32]*
  %data_as_i8ptr374 = bitcast [1 x i32]* %data373 to i8*
  %elmptr375 = getelementptr [1 x i32], [1 x i32]* %data373, i32 0, i32 0
  store i32 1, i32* %elmptr375
  %dimsptr376 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor366, i32 0, i32 2
  store i8* %dims_as_i8ptr371, i8** %dimsptr376
  %dataptr377 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor366, i32 0, i32 3
  store i8* %data_as_i8ptr374, i8** %dataptr377
  %rc378 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor366, i32 0, i32 4
  store i8 0, i8* %rc378
  %tensor379 = bitcast %tensor_t* %raw_tensor366 to i8*
  %tmpOp380 = call i8* @add(i8* %i364, i8* %tensor379)
  %malloccall381 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor382 = bitcast i8* %malloccall381 to %tensor_t*
  %dtype383 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor382, i32 0, i32 0
  store i8 3, i8* %dtype383
  %ndims384 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor382, i32 0, i32 1
  store i8 1, i8* %ndims384
  %malloccall385 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims386 = bitcast i8* %malloccall385 to [1 x i64]*
  %dims_as_i8ptr387 = bitcast [1 x i64]* %dims386 to i8*
  %elmptr388 = getelementptr [1 x i64], [1 x i64]* %dims386, i64 0, i64 0
  store i64 1, i64* %elmptr388
```

```
  %malloccall389 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data390 = bitcast i8* %malloccall389 to [1 x i8*]*
  %data_as_i8ptr391 = bitcast [1 x i8*]* %data390 to i8*
  %elmptr392 = getelementptr [1 x i8*], [1 x i8*]* %data390, i64 0, i64 0
  store i8* %tmpOp380, i8** %elmptr392
  %dimsptr393 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor382, i32 0, i32 2
  store i8* %dims_as_i8ptr387, i8** %dimsptr393
  %dataptr394 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor382, i32 0, i32 3
  store i8* %data_as_i8ptr391, i8** %dataptr394
  %rc395 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor382, i32 0, i32 4
  store i8 0, i8* %rc395
  %tensor396 = bitcast %tensor_t* %raw_tensor382 to i8*
  %A397 = load i8*, i8** %A
  %access_tensor398 = call i8* @index_get(i8* %A397, i8* %tensor396)
  %lhsptr399 = load i8*, i8** %t
  call void @increase_rc(i8* %access_tensor398)
  call void @decrease_rc(i8* %lhsptr399)
  store i8* %access_tensor398, i8** %t
  %r400 = load i8*, i8** %r
  %malloccall401 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor402 = bitcast i8* %malloccall401 to %tensor_t*
  %dtype403 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 0
  store i8 3, i8* %dtype403
  %ndims404 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 1
  store i8 1, i8* %ndims404
  %malloccall405 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims406 = bitcast i8* %malloccall405 to [1 x i64]*
  %dims_as_i8ptr407 = bitcast [1 x i64]* %dims406 to i8*
  %elmptr408 = getelementptr [1 x i64], [1 x i64]* %dims406, i64 0, i64 0
  store i64 1, i64* %elmptr408
  %malloccall409 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data410 = bitcast i8* %malloccall409 to [1 x i8*]*
  %data_as_i8ptr411 = bitcast [1 x i8*]* %data410 to i8*
  %elmptr412 = getelementptr [1 x i8*], [1 x i8*]* %data410, i64 0, i64 0
  store i8* %r400, i8** %elmptr412
  %dimsptr413 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 2
  store i8* %dims_as_i8ptr407, i8** %dimsptr413
  %dataptr414 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 3
  store i8* %data_as_i8ptr411, i8** %dataptr414
  %rc415 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 4
```

```
  store i8 0, i8* %rc415
  %tensor416 = bitcast %tensor_t* %raw_tensor402 to i8*
  %A417 = load i8*, i8** %A
  %access_tensor418 = call i8* @index_get(i8* %A417, i8* %tensor416)
  %i419 = load i8*, i8** %i
  %malloccall420 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor421 = bitcast i8* %malloccall420 to %tensor_t*
  %dtype422 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor421, i32 0, i32 0
  store i8 0, i8* %dtype422
  %ndims423 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor421, i32 0, i32 1
  store i8 0, i8* %ndims423
  %malloccall424 = tail call i8* @malloc(i32 0)
  %dims425 = bitcast i8* %malloccall424 to [0 x i64]*
  %dims_as_i8ptr426 = bitcast [0 x i64]* %dims425 to i8*
  %malloccall427 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data428 = bitcast i8* %malloccall427 to [1 x i32]*
  %data_as_i8ptr429 = bitcast [1 x i32]* %data428 to i8*
  %elmptr430 = getelementptr [1 x i32], [1 x i32]* %data428, i32 0, i32 0
  store i32 1, i32* %elmptr430
  %dimsptr431 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor421, i32 0, i32 2
  store i8* %dims_as_i8ptr426, i8** %dimsptr431
  %dataptr432 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor421, i32 0, i32 3
  store i8* %data_as_i8ptr429, i8** %dataptr432
  %rc433 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor421, i32 0, i32 4
  store i8 0, i8* %rc433
  %tensor434 = bitcast %tensor_t* %raw_tensor421 to i8*
  %tmpOp435 = call i8* @add(i8* %i419, i8* %tensor434)
  %malloccall436 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor437 = bitcast i8* %malloccall436 to %tensor_t*
  %dtype438 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor437, i32 0, i32 0
  store i8 3, i8* %dtype438
  %ndims439 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor437, i32 0, i32 1
  store i8 1, i8* %ndims439
  %malloccall440 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims441 = bitcast i8* %malloccall440 to [1 x i64]*
  %dims_as_i8ptr442 = bitcast [1 x i64]* %dims441 to i8*
  %elmptr443 = getelementptr [1 x i64], [1 x i64]* %dims441, i64 0, i64 0
  store i64 1, i64* %elmptr443
  %malloccall444 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
```

```
  %data445 = bitcast i8* %malloccall444 to [1 x i8*]*
  %data_as_i8ptr446 = bitcast [1 x i8*]* %data445 to i8*
  %elmptr447 = getelementptr [1 x i8*], [1 x i8*]* %data445, i64 0, i64 0
  store i8* %tmpOp435, i8** %elmptr447
  %dimsptr448 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor437, i32 0, i32 2
  store i8* %dims_as_i8ptr442, i8** %dimsptr448
  %dataptr449 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor437, i32 0, i32 3
  store i8* %data_as_i8ptr446, i8** %dataptr449
  %rc450 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor437, i32 0, i32 4
  store i8 0, i8* %rc450
  %tensor451 = bitcast %tensor_t* %raw_tensor437 to i8*
  %A452 = load i8*, i8** %A
  call void @index_put(i8* %A452, i8* %tensor451, i8* %access_tensor418)
  %t453 = load i8*, i8** %t
  %r454 = load i8*, i8** %r
  %malloccall455 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor456 = bitcast i8* %malloccall455 to %tensor_t*
  %dtype457 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 0
  store i8 3, i8* %dtype457
  %ndims458 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 1
  store i8 1, i8* %ndims458
  %malloccall459 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims460 = bitcast i8* %malloccall459 to [1 x i64]*
  %dims_as_i8ptr461 = bitcast [1 x i64]* %dims460 to i8*
  %elmptr462 = getelementptr [1 x i64], [1 x i64]* %dims460, i64 0, i64 0
  store i64 1, i64* %elmptr462
  %malloccall463 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data464 = bitcast i8* %malloccall463 to [1 x i8*]*
  %data_as_i8ptr465 = bitcast [1 x i8*]* %data464 to i8*
  %elmptr466 = getelementptr [1 x i8*], [1 x i8*]* %data464, i64 0, i64 0
  store i8* %r454, i8** %elmptr466
  %dimsptr467 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 2
  store i8* %dims_as_i8ptr461, i8** %dimsptr467
  %dataptr468 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 3
  store i8* %data_as_i8ptr465, i8** %dataptr468
  %rc469 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 4
  store i8 0, i8* %rc469
  %tensor470 = bitcast %tensor_t* %raw_tensor456 to i8*
  %A471 = load i8*, i8** %A
  call void @index_put(i8* %A471, i8* %tensor470, i8* %t453)
  %i472 = load i8*, i8** %i
```

```
  %malloccall473 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor474 = bitcast i8* %malloccall473 to %tensor_t*
  %dtype475 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor474, i32 0, i32 0
  store i8 0, i8* %dtype475
  %ndims476 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor474, i32 0, i32 1
  store i8 0, i8* %ndims476
  %malloccall477 = tail call i8* @malloc(i32 0)
  %dims478 = bitcast i8* %malloccall477 to [0 x i64]*
  %dims_as_i8ptr479 = bitcast [0 x i64]* %dims478 to i8*
  %malloccall480 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data481 = bitcast i8* %malloccall480 to [1 x i32]*
  %data_as_i8ptr482 = bitcast [1 x i32]* %data481 to i8*
  %elmptr483 = getelementptr [1 x i32], [1 x i32]* %data481, i32 0, i32 0
  store i32 1, i32* %elmptr483
  %dimsptr484 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor474, i32 0, i32 2
  store i8* %dims_as_i8ptr479, i8** %dimsptr484
  %dataptr485 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor474, i32 0, i32 3
  store i8* %data_as_i8ptr482, i8** %dataptr485
  %rc486 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor474, i32 0, i32 4
  store i8 0, i8* %rc486
  %tensor487 = bitcast %tensor_t* %raw_tensor474 to i8*
  %tmpOp488 = call i8* @add(i8* %i472, i8* %tensor487)
  ret i8* %tmpOp488
}

define i8* @quicksort(i8* %0, i8* %1, i8* %2) {
entry:
  %A = alloca i8*
  store i8* %0, i8** %A
  %p = alloca i8*
  store i8* %1, i8** %p
  %r = alloca i8*
  store i8* %2, i8** %r
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall1 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall1 to [0 x i64]*
```

```
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall2 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall2 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %q = alloca i8*
  store i8* null, i8** %q
  %lhsptr = load i8*, i8** %q
  call void @increase_rc(i8* %tensor)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %tensor, i8** %q
  %p3 = load i8*, i8** %p
  %r4 = load i8*, i8** %r
  %tmpOp = call i8* @less(i8* %p3, i8* %r4)
  %bool = call i1 @bool_of_zero(i8* %tmpOp)
  br i1 %bool, label %then, label %else

merge:                                            ; preds = %else, %then
  %malloccall49 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor50 = bitcast i8* %malloccall49 to %tensor_t*
  %dtype51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 0
  store i8 0, i8* %dtype51
  %ndims52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 1
  store i8 0, i8* %ndims52
  %malloccall53 = tail call i8* @malloc(i32 0)
  %dims54 = bitcast i8* %malloccall53 to [0 x i64]*
  %dims_as_i8ptr55 = bitcast [0 x i64]* %dims54 to i8*
  %malloccall56 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data57 = bitcast i8* %malloccall56 to [1 x i32]*
  %data_as_i8ptr58 = bitcast [1 x i32]* %data57 to i8*
  %elmptr59 = getelementptr [1 x i32], [1 x i32]* %data57, i32 0, i32 0
  store i32 1, i32* %elmptr59
  %dimsptr60 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 2
  store i8* %dims_as_i8ptr55, i8** %dimsptr60
```

```
  %dataptr61 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 3
  store i8* %data_as_i8ptr58, i8** %dataptr61
  %rc62 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 4
  store i8 0, i8* %rc62
  %tensor63 = bitcast %tensor_t* %raw_tensor50 to i8*
  ret i8* %tensor63

then:                                             ; preds = %entry
  %A5 = load i8*, i8** %A
  %p6 = load i8*, i8** %p
  %r7 = load i8*, i8** %r
  %ret = call i8* @partition(i8* %A5, i8* %p6, i8* %r7)
  %lhsptr8 = load i8*, i8** %q
  call void @increase_rc(i8* %ret)
  call void @decrease_rc(i8* %lhsptr8)
  store i8* %ret, i8** %q
  %A9 = load i8*, i8** %A
  %p10 = load i8*, i8** %p
  %q11 = load i8*, i8** %q
  %malloccall12 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor13 = bitcast i8* %malloccall12 to %tensor_t*
  %dtype14 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 0
  store i8 0, i8* %dtype14
  %ndims15 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 1
  store i8 0, i8* %ndims15
  %malloccall16 = tail call i8* @malloc(i32 0)
  %dims17 = bitcast i8* %malloccall16 to [0 x i64]*
  %dims_as_i8ptr18 = bitcast [0 x i64]* %dims17 to i8*
  %malloccall19 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data20 = bitcast i8* %malloccall19 to [1 x i32]*
  %data_as_i8ptr21 = bitcast [1 x i32]* %data20 to i8*
  %elmptr22 = getelementptr [1 x i32], [1 x i32]* %data20, i32 0, i32 0
  store i32 1, i32* %elmptr22
  %dimsptr23 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 2
  store i8* %dims_as_i8ptr18, i8** %dimsptr23
  %dataptr24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 3
  store i8* %data_as_i8ptr21, i8** %dataptr24
  %rc25 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 4
  store i8 0, i8* %rc25
  %tensor26 = bitcast %tensor_t* %raw_tensor13 to i8*
  %tmpOp27 = call i8* @subtract(i8* %q11, i8* %tensor26)
  %ret28 = call i8* @quicksort(i8* %A9, i8* %p10, i8* %tmpOp27)
```

```
  %A29 = load i8*, i8** %A
  %q30 = load i8*, i8** %q
  %malloccall31 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor32 = bitcast i8* %malloccall31 to %tensor_t*
  %dtype33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 0
  store i8 0, i8* %dtype33
  %ndims34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 1
  store i8 0, i8* %ndims34
  %malloccall35 = tail call i8* @malloc(i32 0)
  %dims36 = bitcast i8* %malloccall35 to [0 x i64]*
  %dims_as_i8ptr37 = bitcast [0 x i64]* %dims36 to i8*
  %malloccall38 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data39 = bitcast i8* %malloccall38 to [1 x i32]*
  %data_as_i8ptr40 = bitcast [1 x i32]* %data39 to i8*
  %elmptr41 = getelementptr [1 x i32], [1 x i32]* %data39, i32 0, i32 0
  store i32 1, i32* %elmptr41
  %dimsptr42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 2
  store i8* %dims_as_i8ptr37, i8** %dimsptr42
  %dataptr43 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 3
  store i8* %data_as_i8ptr40, i8** %dataptr43
  %rc44 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 4
  store i8 0, i8* %rc44
  %tensor45 = bitcast %tensor_t* %raw_tensor32 to i8*
  %tmpOp46 = call i8* @add(i8* %q30, i8* %tensor45)
  %r47 = load i8*, i8** %r
  %ret48 = call i8* @quicksort(i8* %A29, i8* %tmpOp46, i8* %r47)
  br label %merge

else:                                             ; preds = %entry
  br label %merge
}
```

6.3. Sample Test Two

In the second demo you can see the use of var tensor, with which you can use char/string in tensor! A very powerful feature.

```
# [vertex_name, visited, adjacency list]
uw = var["undershorts", 0, nil];
pt = var["pants", 0, nil];
```

```
se = var["shoes", 0, nil];
sk = var["socks", 0, nil];
wt = var["watch", 0, nil];
bl = var["belt", 0, nil];
ts = var["shirt", 0, nil];
ti = var["tie", 0, nil];
jc = var["jacket", 0, nil];

vertices = var[wt, sk, ts, uw, pt, se, bl, ti, jc];
sorted_v = var[nil, nil, nil, nil, nil, nil, nil, nil, nil];

uw[2] = var[pt, se];
pt[2] = var[bl, se];
sk[2] = var[se];
bl[2] = var[jc];
ts[2] = var[bl, ti];
ti[2] = var[jc];

def dfs_visit(U, time, L) {
    U[1] = 1;
    if (U[2] != nil) {
        for (v in U[2]) {
            if (v[1] == 0) {
                time = dfs_visit(v, time, L);
            }
        }
    }
    time = time - 1;
    L[time] = U;
    return time;
}

def dfs(V, L) {
    time = shape(L);
    for (u in V) {
        if (u[1] == 0) {
            time = dfs_visit(u, time, L);
        }
    }
    return time;
}

dfs(vertices, sorted_v);
```

```
for (v in sorted_v) {
    print(v[0]);
}
```

```
; ModuleID = 'TENLab'
source_filename = "TENLab"

%tensor_t = type { i8, i8, i8*, i8*, i8 }

@a = global i8* null
@l = global i8* null

define i8 @main() {
entry:
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 1, i8* %ndims
  %malloccall1 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims = bitcast i8* %malloccall1 to [1 x i64]*
  %dims_as_i8ptr = bitcast [1 x i64]* %dims to i8*
  %elmptr = getelementptr [1 x i64], [1 x i64]* %dims, i64 0, i64 0
  store i64 10, i64* %elmptr
  %malloccall2 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i32* getelementptr
(i32, i32* null, i32 1) to i64), i64 10) to i32))
  %data = bitcast i8* %malloccall2 to [10 x i32]*
  %data_as_i8ptr = bitcast [10 x i32]* %data to i8*
  %elmptr3 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 0
  store i32 5, i32* %elmptr3
  %elmptr4 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 1
  store i32 3, i32* %elmptr4
  %elmptr5 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 2
  store i32 9, i32* %elmptr5
  %elmptr6 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 3
  store i32 6, i32* %elmptr6
  %elmptr7 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 4
  store i32 3, i32* %elmptr7
  %elmptr8 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 5
  store i32 2, i32* %elmptr8
  %elmptr9 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 6
```

```
  store i32 1, i32* %elmptr9
  %elmptr10 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 7
  store i32 7, i32* %elmptr10
  %elmptr11 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 8
  store i32 8, i32* %elmptr11
  %elmptr12 = getelementptr [10 x i32], [10 x i32]* %data, i32 0, i32 9
  store i32 4, i32* %elmptr12
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %lhsptr = load i8*, i8** @a
  call void @increase_rc(i8* %tensor)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %tensor, i8** @a
  %a = load i8*, i8** @a
  %shape = call i8* @shape(i8* %a)
  %lhsptr13 = load i8*, i8** @l
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr13)
  store i8* %shape, i8** @l
  %a14 = load i8*, i8** @a
  %malloccall15 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor16 = bitcast i8* %malloccall15 to %tensor_t*
  %dtype17 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 0
  store i8 0, i8* %dtype17
  %ndims18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 1
  store i8 0, i8* %ndims18
  %malloccall19 = tail call i8* @malloc(i32 0)
  %dims20 = bitcast i8* %malloccall19 to [0 x i64]*
  %dims_as_i8ptr21 = bitcast [0 x i64]* %dims20 to i8*
  %malloccall22 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data23 = bitcast i8* %malloccall22 to [1 x i32]*
  %data_as_i8ptr24 = bitcast [1 x i32]* %data23 to i8*
  %elmptr25 = getelementptr [1 x i32], [1 x i32]* %data23, i32 0, i32 0
  store i32 0, i32* %elmptr25
  %dimsptr26 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 2
  store i8* %dims_as_i8ptr21, i8** %dimsptr26
  %dataptr27 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 3
```

```
  store i8* %data_as_i8ptr24, i8** %dataptr27
  %rc28 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor16, i32 0, i32 4
  store i8 0, i8* %rc28
  %tensor29 = bitcast %tensor_t* %raw_tensor16 to i8*
  %malloccall30 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor31 = bitcast i8* %malloccall30 to %tensor_t*
  %dtype32 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 0
  store i8 0, i8* %dtype32
  %ndims33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 1
  store i8 0, i8* %ndims33
  %malloccall34 = tail call i8* @malloc(i32 0)
  %dims35 = bitcast i8* %malloccall34 to [0 x i64]*
  %dims_as_i8ptr36 = bitcast [0 x i64]* %dims35 to i8*
  %malloccall37 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data38 = bitcast i8* %malloccall37 to [1 x i32]*
  %data_as_i8ptr39 = bitcast [1 x i32]* %data38 to i8*
  %elmptr40 = getelementptr [1 x i32], [1 x i32]* %data38, i32 0, i32 0
  store i32 0, i32* %elmptr40
  %dimsptr41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 2
  store i8* %dims_as_i8ptr36, i8** %dimsptr41
  %dataptr42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 3
  store i8* %data_as_i8ptr39, i8** %dataptr42
  %rc43 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor31, i32 0, i32 4
  store i8 0, i8* %rc43
  %tensor44 = bitcast %tensor_t* %raw_tensor31 to i8*
  %malloccall45 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor46 = bitcast i8* %malloccall45 to %tensor_t*
  %dtype47 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 0
  store i8 3, i8* %dtype47
  %ndims48 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 1
  store i8 1, i8* %ndims48
  %malloccall49 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims50 = bitcast i8* %malloccall49 to [1 x i64]*
  %dims_as_i8ptr51 = bitcast [1 x i64]* %dims50 to i8*
  %elmptr52 = getelementptr [1 x i64], [1 x i64]* %dims50, i64 0, i64 0
  store i64 1, i64* %elmptr52
  %malloccall53 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data54 = bitcast i8* %malloccall53 to [1 x i8*]*
  %data_as_i8ptr55 = bitcast [1 x i8*]* %data54 to i8*
  %elmptr56 = getelementptr [1 x i8*], [1 x i8*]* %data54, i64 0, i64 0
```

```
  store i8* %tensor44, i8** %elmptr56
  %dimsptr57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 2
  store i8* %dims_as_i8ptr51, i8** %dimsptr57
  %dataptr58 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 3
  store i8* %data_as_i8ptr55, i8** %dataptr58
  %rc59 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor46, i32 0, i32 4
  store i8 0, i8* %rc59
  %tensor60 = bitcast %tensor_t* %raw_tensor46 to i8*
  %l = load i8*, i8** @l
  %access_tensor = call i8* @index_get(i8* %l, i8* %tensor60)
  %malloccall61 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor62 = bitcast i8* %malloccall61 to %tensor_t*
  %dtype63 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 0
  store i8 0, i8* %dtype63
  %ndims64 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 1
  store i8 0, i8* %ndims64
  %malloccall65 = tail call i8* @malloc(i32 0)
  %dims66 = bitcast i8* %malloccall65 to [0 x i64]*
  %dims_as_i8ptr67 = bitcast [0 x i64]* %dims66 to i8*
  %malloccall68 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data69 = bitcast i8* %malloccall68 to [1 x i32]*
  %data_as_i8ptr70 = bitcast [1 x i32]* %data69 to i8*
  %elmptr71 = getelementptr [1 x i32], [1 x i32]* %data69, i32 0, i32 0
  store i32 1, i32* %elmptr71
  %dimsptr72 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 2
  store i8* %dims_as_i8ptr67, i8** %dimsptr72
  %dataptr73 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 3
  store i8* %data_as_i8ptr70, i8** %dataptr73
  %rc74 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor62, i32 0, i32 4
  store i8 0, i8* %rc74
  %tensor75 = bitcast %tensor_t* %raw_tensor62 to i8*
  %tmpOp = call i8* @subtract(i8* %access_tensor, i8* %tensor75)
  %ret = call i8* @quicksort(i8* %a14, i8* %tensor29, i8* %tmpOp)
  %a76 = load i8*, i8** @a
  call void @print(i8* %a76)
  ret i8 0
}

declare i8* @add(i8*, i8*)

declare i8* @subtract(i8*, i8*)
```

```
declare i8* @negative(i8*)

declare i8* @mult(i8*, i8*)

declare i8* @dotmul(i8*, i8*)

declare i8* @divide(i8*, i8*)

declare i8* @floordivide(i8*, i8*)

declare i8* @matpow(i8*, i8*)

declare i8* @dotpow(i8*, i8*)

declare i8* @mod(i8*, i8*)

declare i8* @transpose(i8*)

declare i8* @equal(i8*, i8*)

declare i8* @notequal(i8*, i8*)

declare i8* @greater(i8*, i8*)

declare i8* @greaterequal(i8*, i8*)

declare i8* @less(i8*, i8*)

declare i8* @lessequal(i8*, i8*)

declare i8* @range(i8*, i8*, i8*)

declare void @print(i8*)

declare i8* @logicaland(i8*, i8*)

declare i8* @logicalor(i8*, i8*)

declare i8* @logicalnot(i8*)

declare i32 @len(i8*)

declare i8* @zeros(i8*)
```

```
declare i8* @cat(i8*, i8*, i8*)

declare i8* @shape(i8*)

declare i8* @ones(i8*)

declare i8* @tensor_rand(i8*)

declare i8* @sum(i8*)

declare i8* @any(i8*)

declare i8* @all(i8*)

declare i8* @tensor_abs(i8*)

declare i8* @tensor_log(i8*)

declare i8* @tensor_floor(i8*)

declare i8* @tensor_ceil(i8*)

declare i8* @tensor_round(i8*)

declare i8* @int_of(i8*)

declare i8* @float_of(i8*)

declare i8* @inverse(i8*)

declare i1 @bool_of_zero(i8*)

declare i8* @pe_calc(i8* (i8*, i8*)**, i32, i8* (i8**)*, i8*, i8*)

declare i8* @index_get(i8*, i8*)

declare i8* @index_get_int(i8*, i32)

declare void @index_put(i8*, i8*, i8*)

declare void @increase_rc(i8*)

declare void @decrease_rc(i8*)
```

```
declare noalias i8* @malloc(i32)

define i8* @partition(i8* %0, i8* %1, i8* %2) {
entry:
  %A = alloca i8*
  store i8* %0, i8** %A
  %p = alloca i8*
  store i8* %1, i8** %p
  %r = alloca i8*
  store i8* %2, i8** %r
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 1, i8* %ndims
  %malloccall1 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims = bitcast i8* %malloccall1 to [1 x i64]*
  %dims_as_i8ptr = bitcast [1 x i64]* %dims to i8*
  %elmptr = getelementptr [1 x i64], [1 x i64]* %dims, i64 0, i64 0
  store i64 1, i64* %elmptr
  %malloccall2 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall2 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr3 = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 1, i32* %elmptr3
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %rand = call i8* @tensor_rand(i8* %tensor)
  %n = alloca i8*
  store i8* null, i8** %n
  %lhsptr = load i8*, i8** %n
  call void @increase_rc(i8* %rand)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %rand, i8** %n
  %malloccall4 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor5 = bitcast i8* %malloccall4 to %tensor_t*
```

```
  %dtype6 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 0
  store i8 0, i8* %dtype6
  %ndims7 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 1
  store i8 0, i8* %ndims7
  %malloccall8 = tail call i8* @malloc(i32 0)
  %dims9 = bitcast i8* %malloccall8 to [0 x i64]*
  %dims_as_i8ptr10 = bitcast [0 x i64]* %dims9 to i8*
  %malloccall11 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data12 = bitcast i8* %malloccall11 to [1 x i32]*
  %data_as_i8ptr13 = bitcast [1 x i32]* %data12 to i8*
  %elmptr14 = getelementptr [1 x i32], [1 x i32]* %data12, i32 0, i32 0
  store i32 0, i32* %elmptr14
  %dimsptr15 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 2
  store i8* %dims_as_i8ptr10, i8** %dimsptr15
  %dataptr16 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 3
  store i8* %data_as_i8ptr13, i8** %dataptr16
  %rc17 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 4
  store i8 0, i8* %rc17
  %tensor18 = bitcast %tensor_t* %raw_tensor5 to i8*
  %malloccall19 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor20 = bitcast i8* %malloccall19 to %tensor_t*
  %dtype21 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 0
  store i8 3, i8* %dtype21
  %ndims22 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 1
  store i8 1, i8* %ndims22
  %malloccall23 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims24 = bitcast i8* %malloccall23 to [1 x i64]*
  %dims_as_i8ptr25 = bitcast [1 x i64]* %dims24 to i8*
  %elmptr26 = getelementptr [1 x i64], [1 x i64]* %dims24, i64 0, i64 0
  store i64 1, i64* %elmptr26
  %malloccall27 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data28 = bitcast i8* %malloccall27 to [1 x i8*]*
  %data_as_i8ptr29 = bitcast [1 x i8*]* %data28 to i8*
  %elmptr30 = getelementptr [1 x i8*], [1 x i8*]* %data28, i64 0, i64 0
  store i8* %tensor18, i8** %elmptr30
  %dimsptr31 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 2
  store i8* %dims_as_i8ptr25, i8** %dimsptr31
  %dataptr32 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 3
  store i8* %data_as_i8ptr29, i8** %dataptr32
  %rc33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor20, i32 0, i32 4
  store i8 0, i8* %rc33
```

```
  %tensor34 = bitcast %tensor_t* %raw_tensor20 to i8*
  %n35 = load i8*, i8** %n
  %access_tensor = call i8* @index_get(i8* %n35, i8* %tensor34)
  %malloccall36 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor37 = bitcast i8* %malloccall36 to %tensor_t*
  %dtype38 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 0
  store i8 1, i8* %dtype38
  %ndims39 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 1
  store i8 0, i8* %ndims39
  %malloccall40 = tail call i8* @malloc(i32 0)
  %dims41 = bitcast i8* %malloccall40 to [0 x i64]*
  %dims_as_i8ptr42 = bitcast [0 x i64]* %dims41 to i8*
  %malloccall43 = tail call i8* @malloc(i32 ptrtoint (double* getelementptr (double, double* null,
i32 1) to i32))
  %data44 = bitcast i8* %malloccall43 to [1 x double]*
  %data_as_i8ptr45 = bitcast [1 x double]* %data44 to i8*
  %elmptr46 = getelementptr [1 x double], [1 x double]* %data44, i64 0, i64 0
  store double 1.000000e+00, double* %elmptr46
  %dimsptr47 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 2
  store i8* %dims_as_i8ptr42, i8** %dimsptr47
  %dataptr48 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 3
  store i8* %data_as_i8ptr45, i8** %dataptr48
  %rc49 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor37, i32 0, i32 4
  store i8 0, i8* %rc49
  %tensor50 = bitcast %tensor_t* %raw_tensor37 to i8*
  %tmpOp = call i8* @add(i8* %access_tensor, i8* %tensor50)
  %r51 = load i8*, i8** %r
  %float_of = call i8* @float_of(i8* %r51)
  %tmpOp52 = call i8* @dotmul(i8* %tmpOp, i8* %float_of)
  %malloccall53 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor54 = bitcast i8* %malloccall53 to %tensor_t*
  %dtype55 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 0
  store i8 1, i8* %dtype55
  %ndims56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 1
  store i8 0, i8* %ndims56
  %malloccall57 = tail call i8* @malloc(i32 0)
  %dims58 = bitcast i8* %malloccall57 to [0 x i64]*
  %dims_as_i8ptr59 = bitcast [0 x i64]* %dims58 to i8*
  %malloccall60 = tail call i8* @malloc(i32 ptrtoint (double* getelementptr (double, double* null,
i32 1) to i32))
  %data61 = bitcast i8* %malloccall60 to [1 x double]*
  %data_as_i8ptr62 = bitcast [1 x double]* %data61 to i8*
```

```
  %elmptr63 = getelementptr [1 x double], [1 x double]* %data61, i64 0, i64 0
  store double 1.000000e+01, double* %elmptr63
  %dimsptr64 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 2
  store i8* %dims_as_i8ptr59, i8** %dimsptr64
  %dataptr65 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 3
  store i8* %data_as_i8ptr62, i8** %dataptr65
  %rc66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 4
  store i8 0, i8* %rc66
  %tensor67 = bitcast %tensor_t* %raw_tensor54 to i8*
  %tmpOp68 = call i8* @dotmul(i8* %tmpOp52, i8* %tensor67)
  %int_of = call i8* @int_of(i8* %tmpOp68)
  %lhsptr69 = load i8*, i8** %n
  call void @increase_rc(i8* %int_of)
  call void @decrease_rc(i8* %lhsptr69)
  store i8* %int_of, i8** %n
  %n70 = load i8*, i8** %n
  %r71 = load i8*, i8** %r
  %p72 = load i8*, i8** %p
  %tmpOp73 = call i8* @subtract(i8* %r71, i8* %p72)
  %malloccall74 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor75 = bitcast i8* %malloccall74 to %tensor_t*
  %dtype76 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 0
  store i8 0, i8* %dtype76
  %ndims77 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 1
  store i8 0, i8* %ndims77
  %malloccall78 = tail call i8* @malloc(i32 0)
  %dims79 = bitcast i8* %malloccall78 to [0 x i64]*
  %dims_as_i8ptr80 = bitcast [0 x i64]* %dims79 to i8*
  %malloccall81 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data82 = bitcast i8* %malloccall81 to [1 x i32]*
  %data_as_i8ptr83 = bitcast [1 x i32]* %data82 to i8*
  %elmptr84 = getelementptr [1 x i32], [1 x i32]* %data82, i32 0, i32 0
  store i32 1, i32* %elmptr84
  %dimsptr85 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 2
  store i8* %dims_as_i8ptr80, i8** %dimsptr85
  %dataptr86 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 3
  store i8* %data_as_i8ptr83, i8** %dataptr86
  %rc87 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor75, i32 0, i32 4
  store i8 0, i8* %rc87
  %tensor88 = bitcast %tensor_t* %raw_tensor75 to i8*
  %tmpOp89 = call i8* @add(i8* %tmpOp73, i8* %tensor88)
  %tmpOp90 = call i8* @mod(i8* %n70, i8* %tmpOp89)
```

```
  %p91 = load i8*, i8** %p
  %tmpOp92 = call i8* @add(i8* %tmpOp90, i8* %p91)
  %lhsptr93 = load i8*, i8** %n
  call void @increase_rc(i8* %tmpOp92)
  call void @decrease_rc(i8* %lhsptr93)
  store i8* %tmpOp92, i8** %n
  %n94 = load i8*, i8** %n
  %malloccall95 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor96 = bitcast i8* %malloccall95 to %tensor_t*
  %dtype97 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 0
  store i8 3, i8* %dtype97
  %ndims98 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 1
  store i8 1, i8* %ndims98
  %malloccall99 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims100 = bitcast i8* %malloccall99 to [1 x i64]*
  %dims_as_i8ptr101 = bitcast [1 x i64]* %dims100 to i8*
  %elmptr102 = getelementptr [1 x i64], [1 x i64]* %dims100, i64 0, i64 0
  store i64 1, i64* %elmptr102
  %malloccall103 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data104 = bitcast i8* %malloccall103 to [1 x i8*]*
  %data_as_i8ptr105 = bitcast [1 x i8*]* %data104 to i8*
  %elmptr106 = getelementptr [1 x i8*], [1 x i8*]* %data104, i64 0, i64 0
  store i8* %n94, i8** %elmptr106
  %dimsptr107 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 2
  store i8* %dims_as_i8ptr101, i8** %dimsptr107
  %dataptr108 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 3
  store i8* %data_as_i8ptr105, i8** %dataptr108
  %rc109 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor96, i32 0, i32 4
  store i8 0, i8* %rc109
  %tensor110 = bitcast %tensor_t* %raw_tensor96 to i8*
  %A111 = load i8*, i8** %A
  %access_tensor112 = call i8* @index_get(i8* %A111, i8* %tensor110)
  %t = alloca i8*
  store i8* null, i8** %t
  %lhsptr113 = load i8*, i8** %t
  call void @increase_rc(i8* %access_tensor112)
  call void @decrease_rc(i8* %lhsptr113)
  store i8* %access_tensor112, i8** %t
  %r114 = load i8*, i8** %r
  %malloccall115 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor116 = bitcast i8* %malloccall115 to %tensor_t*
  %dtype117 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 0
  store i8 3, i8* %dtype117
  %ndims118 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 1
  store i8 1, i8* %ndims118
  %malloccall119 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims120 = bitcast i8* %malloccall119 to [1 x i64]*
  %dims_as_i8ptr121 = bitcast [1 x i64]* %dims120 to i8*
  %elmptr122 = getelementptr [1 x i64], [1 x i64]* %dims120, i64 0, i64 0
  store i64 1, i64* %elmptr122
  %malloccall123 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data124 = bitcast i8* %malloccall123 to [1 x i8*]*
  %data_as_i8ptr125 = bitcast [1 x i8*]* %data124 to i8*
  %elmptr126 = getelementptr [1 x i8*], [1 x i8*]* %data124, i64 0, i64 0
  store i8* %r114, i8** %elmptr126
  %dimsptr127 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 2
  store i8* %dims_as_i8ptr121, i8** %dimsptr127
  %dataptr128 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 3
  store i8* %data_as_i8ptr125, i8** %dataptr128
  %rc129 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor116, i32 0, i32 4
  store i8 0, i8* %rc129
  %tensor130 = bitcast %tensor_t* %raw_tensor116 to i8*
  %A131 = load i8*, i8** %A
  %access_tensor132 = call i8* @index_get(i8* %A131, i8* %tensor130)
  %n133 = load i8*, i8** %n
  %malloccall134 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor135 = bitcast i8* %malloccall134 to %tensor_t*
  %dtype136 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 0
  store i8 3, i8* %dtype136
  %ndims137 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 1
  store i8 1, i8* %ndims137
  %malloccall138 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims139 = bitcast i8* %malloccall138 to [1 x i64]*
  %dims_as_i8ptr140 = bitcast [1 x i64]* %dims139 to i8*
  %elmptr141 = getelementptr [1 x i64], [1 x i64]* %dims139, i64 0, i64 0
  store i64 1, i64* %elmptr141
  %malloccall142 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data143 = bitcast i8* %malloccall142 to [1 x i8*]*
  %data_as_i8ptr144 = bitcast [1 x i8*]* %data143 to i8*
```

```
  %elmptr145 = getelementptr [1 x i8*], [1 x i8*]* %data143, i64 0, i64 0
  store i8* %n133, i8** %elmptr145
  %dimsptr146 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 2
  store i8* %dims_as_i8ptr140, i8** %dimsptr146
  %dataptr147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 3
  store i8* %data_as_i8ptr144, i8** %dataptr147
  %rc148 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 4
  store i8 0, i8* %rc148
  %tensor149 = bitcast %tensor_t* %raw_tensor135 to i8*
  %A150 = load i8*, i8** %A
  call void @index_put(i8* %A150, i8* %tensor149, i8* %access_tensor132)
  %t151 = load i8*, i8** %t
  %r152 = load i8*, i8** %r
  %malloccall153 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor154 = bitcast i8* %malloccall153 to %tensor_t*
  %dtype155 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 0
  store i8 3, i8* %dtype155
  %ndims156 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 1
  store i8 1, i8* %ndims156
  %malloccall157 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims158 = bitcast i8* %malloccall157 to [1 x i64]*
  %dims_as_i8ptr159 = bitcast [1 x i64]* %dims158 to i8*
  %elmptr160 = getelementptr [1 x i64], [1 x i64]* %dims158, i64 0, i64 0
  store i64 1, i64* %elmptr160
  %malloccall161 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data162 = bitcast i8* %malloccall161 to [1 x i8*]*
  %data_as_i8ptr163 = bitcast [1 x i8*]* %data162 to i8*
  %elmptr164 = getelementptr [1 x i8*], [1 x i8*]* %data162, i64 0, i64 0
  store i8* %r152, i8** %elmptr164
  %dimsptr165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 2
  store i8* %dims_as_i8ptr159, i8** %dimsptr165
  %dataptr166 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 3
  store i8* %data_as_i8ptr163, i8** %dataptr166
  %rc167 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 4
  store i8 0, i8* %rc167
  %tensor168 = bitcast %tensor_t* %raw_tensor154 to i8*
  %A169 = load i8*, i8** %A
  call void @index_put(i8* %A169, i8* %tensor168, i8* %t151)
  %r170 = load i8*, i8** %r
  %malloccall171 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor172 = bitcast i8* %malloccall171 to %tensor_t*
  %dtype173 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 0
  store i8 3, i8* %dtype173
  %ndims174 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 1
  store i8 1, i8* %ndims174
  %malloccall175 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims176 = bitcast i8* %malloccall175 to [1 x i64]*
  %dims_as_i8ptr177 = bitcast [1 x i64]* %dims176 to i8*
  %elmptr178 = getelementptr [1 x i64], [1 x i64]* %dims176, i64 0, i64 0
  store i64 1, i64* %elmptr178
  %malloccall179 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data180 = bitcast i8* %malloccall179 to [1 x i8*]*
  %data_as_i8ptr181 = bitcast [1 x i8*]* %data180 to i8*
  %elmptr182 = getelementptr [1 x i8*], [1 x i8*]* %data180, i64 0, i64 0
  store i8* %r170, i8** %elmptr182
  %dimsptr183 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 2
  store i8* %dims_as_i8ptr177, i8** %dimsptr183
  %dataptr184 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 3
  store i8* %data_as_i8ptr181, i8** %dataptr184
  %rc185 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor172, i32 0, i32 4
  store i8 0, i8* %rc185
  %tensor186 = bitcast %tensor_t* %raw_tensor172 to i8*
  %A187 = load i8*, i8** %A
  %access_tensor188 = call i8* @index_get(i8* %A187, i8* %tensor186)
  %x = alloca i8*
  store i8* null, i8** %x
  %lhsptr189 = load i8*, i8** %x
  call void @increase_rc(i8* %access_tensor188)
  call void @decrease_rc(i8* %lhsptr189)
  store i8* %access_tensor188, i8** %x
  %p190 = load i8*, i8** %p
  %malloccall191 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor192 = bitcast i8* %malloccall191 to %tensor_t*
  %dtype193 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 0
  store i8 0, i8* %dtype193
  %ndims194 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 1
  store i8 0, i8* %ndims194
  %malloccall195 = tail call i8* @malloc(i32 0)
  %dims196 = bitcast i8* %malloccall195 to [0 x i64]*
  %dims_as_i8ptr197 = bitcast [0 x i64]* %dims196 to i8*
```

```
  %malloccall198 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data199 = bitcast i8* %malloccall198 to [1 x i32]*
  %data_as_i8ptr200 = bitcast [1 x i32]* %data199 to i8*
  %elmptr201 = getelementptr [1 x i32], [1 x i32]* %data199, i32 0, i32 0
  store i32 1, i32* %elmptr201
  %dimsptr202 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 2
  store i8* %dims_as_i8ptr197, i8** %dimsptr202
  %dataptr203 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 3
  store i8* %data_as_i8ptr200, i8** %dataptr203
  %rc204 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor192, i32 0, i32 4
  store i8 0, i8* %rc204
  %tensor205 = bitcast %tensor_t* %raw_tensor192 to i8*
  %tmpOp206 = call i8* @subtract(i8* %p190, i8* %tensor205)
  %i = alloca i8*
  store i8* null, i8** %i
  %lhsptr207 = load i8*, i8** %i
  call void @increase_rc(i8* %tmpOp206)
  call void @decrease_rc(i8* %lhsptr207)
  store i8* %tmpOp206, i8** %i
  %idxptr = alloca i32
  store i32 0, i32* %idxptr
  %p208 = load i8*, i8** %p
  %r209 = load i8*, i8** %r
  %malloccall210 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor211 = bitcast i8* %malloccall210 to %tensor_t*
  %dtype212 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor211, i32 0, i32 0
  store i8 0, i8* %dtype212
  %ndims213 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor211, i32 0, i32 1
  store i8 0, i8* %ndims213
  %malloccall214 = tail call i8* @malloc(i32 0)
  %dims215 = bitcast i8* %malloccall214 to [0 x i64]*
  %dims_as_i8ptr216 = bitcast [0 x i64]* %dims215 to i8*
  %malloccall217 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data218 = bitcast i8* %malloccall217 to [1 x i32]*
  %data_as_i8ptr219 = bitcast [1 x i32]* %data218 to i8*
  %elmptr220 = getelementptr [1 x i32], [1 x i32]* %data218, i32 0, i32 0
  store i32 1, i32* %elmptr220
  %dimsptr221 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor211, i32 0, i32 2
  store i8* %dims_as_i8ptr216, i8** %dimsptr221
  %dataptr222 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor211, i32 0, i32 3
  store i8* %data_as_i8ptr219, i8** %dataptr222
```

```
  %rc223 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor211, i32 0, i32 4
  store i8 0, i8* %rc223
  %tensor224 = bitcast %tensor_t* %raw_tensor211 to i8*
  %tmpOp225 = call i8* @range(i8* %p208, i8* %r209, i8* %tensor224)
  %length = call i32 @len(i8* %tmpOp225)
  br label %for

for:                                              ; preds = %merge, %entry
  %idx = load i32, i32* %idxptr
  %new_idx = add i32 %idx, 1
  store i32 %new_idx, i32* %idxptr
  %condition = icmp sgt i32 %new_idx, %length
  br i1 %condition, label %merge341, label %for_body

for_body:                                         ; preds = %for
  %j = alloca i8*
  store i8* null, i8** %j
  %indicator_as_tensor = call i8* @index_get_int(i8* %tmpOp225, i32 %idx)
  store i8* %indicator_as_tensor, i8** %j
  %j226 = load i8*, i8** %j
  %malloccall227 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor228 = bitcast i8* %malloccall227 to %tensor_t*
  %dtype229 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor228, i32 0, i32 0
  store i8 3, i8* %dtype229
  %ndims230 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor228, i32 0, i32 1
  store i8 1, i8* %ndims230
  %malloccall231 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims232 = bitcast i8* %malloccall231 to [1 x i64]*
  %dims_as_i8ptr233 = bitcast [1 x i64]* %dims232 to i8*
  %elmptr234 = getelementptr [1 x i64], [1 x i64]* %dims232, i64 0, i64 0
  store i64 1, i64* %elmptr234
  %malloccall235 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data236 = bitcast i8* %malloccall235 to [1 x i8*]*
  %data_as_i8ptr237 = bitcast [1 x i8*]* %data236 to i8*
  %elmptr238 = getelementptr [1 x i8*], [1 x i8*]* %data236, i64 0, i64 0
  store i8* %j226, i8** %elmptr238
  %dimsptr239 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor228, i32 0, i32 2
  store i8* %dims_as_i8ptr233, i8** %dimsptr239
  %dataptr240 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor228, i32 0, i32 3
  store i8* %data_as_i8ptr237, i8** %dataptr240
  %rc241 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor228, i32 0, i32 4
```

```
  store i8 0, i8* %rc241
  %tensor242 = bitcast %tensor_t* %raw_tensor228 to i8*
  %A243 = load i8*, i8** %A
  %access_tensor244 = call i8* @index_get(i8* %A243, i8* %tensor242)
  %x245 = load i8*, i8** %x
  %tmpOp246 = call i8* @lessequal(i8* %access_tensor244, i8* %x245)
  %bool = call i1 @bool_of_zero(i8* %tmpOp246)
  br i1 %bool, label %then, label %else

merge:                                            ; preds = %else, %then
  br label %for

then:                                             ; preds = %for_body
  %i247 = load i8*, i8** %i
  %malloccall248 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor249 = bitcast i8* %malloccall248 to %tensor_t*
  %dtype250 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor249, i32 0, i32 0
  store i8 0, i8* %dtype250
  %ndims251 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor249, i32 0, i32 1
  store i8 0, i8* %ndims251
  %malloccall252 = tail call i8* @malloc(i32 0)
  %dims253 = bitcast i8* %malloccall252 to [0 x i64]*
  %dims_as_i8ptr254 = bitcast [0 x i64]* %dims253 to i8*
  %malloccall255 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data256 = bitcast i8* %malloccall255 to [1 x i32]*
  %data_as_i8ptr257 = bitcast [1 x i32]* %data256 to i8*
  %elmptr258 = getelementptr [1 x i32], [1 x i32]* %data256, i32 0, i32 0
  store i32 1, i32* %elmptr258
  %dimsptr259 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor249, i32 0, i32 2
  store i8* %dims_as_i8ptr254, i8** %dimsptr259
  %dataptr260 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor249, i32 0, i32 3
  store i8* %data_as_i8ptr257, i8** %dataptr260
  %rc261 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor249, i32 0, i32 4
  store i8 0, i8* %rc261
  %tensor262 = bitcast %tensor_t* %raw_tensor249 to i8*
  %tmpOp263 = call i8* @add(i8* %i247, i8* %tensor262)
  %lhsptr264 = load i8*, i8** %i
  call void @increase_rc(i8* %tmpOp263)
  call void @decrease_rc(i8* %lhsptr264)
  store i8* %tmpOp263, i8** %i
  %i265 = load i8*, i8** %i
```

```
  %malloccall266 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor267 = bitcast i8* %malloccall266 to %tensor_t*
  %dtype268 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor267, i32 0, i32 0
  store i8 3, i8* %dtype268
  %ndims269 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor267, i32 0, i32 1
  store i8 1, i8* %ndims269
  %malloccall270 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims271 = bitcast i8* %malloccall270 to [1 x i64]*
  %dims_as_i8ptr272 = bitcast [1 x i64]* %dims271 to i8*
  %elmptr273 = getelementptr [1 x i64], [1 x i64]* %dims271, i64 0, i64 0
  store i64 1, i64* %elmptr273
  %malloccall274 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data275 = bitcast i8* %malloccall274 to [1 x i8*]*
  %data_as_i8ptr276 = bitcast [1 x i8*]* %data275 to i8*
  %elmptr277 = getelementptr [1 x i8*], [1 x i8*]* %data275, i64 0, i64 0
  store i8* %i265, i8** %elmptr277
  %dimsptr278 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor267, i32 0, i32 2
  store i8* %dims_as_i8ptr272, i8** %dimsptr278
  %dataptr279 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor267, i32 0, i32 3
  store i8* %data_as_i8ptr276, i8** %dataptr279
  %rc280 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor267, i32 0, i32 4
  store i8 0, i8* %rc280
  %tensor281 = bitcast %tensor_t* %raw_tensor267 to i8*
  %A282 = load i8*, i8** %A
  %access_tensor283 = call i8* @index_get(i8* %A282, i8* %tensor281)
  %lhsptr284 = load i8*, i8** %t
  call void @increase_rc(i8* %access_tensor283)
  call void @decrease_rc(i8* %lhsptr284)
  store i8* %access_tensor283, i8** %t
  %j285 = load i8*, i8** %j
  %malloccall286 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor287 = bitcast i8* %malloccall286 to %tensor_t*
  %dtype288 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 0
  store i8 3, i8* %dtype288
  %ndims289 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 1
  store i8 1, i8* %ndims289
  %malloccall290 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims291 = bitcast i8* %malloccall290 to [1 x i64]*
  %dims_as_i8ptr292 = bitcast [1 x i64]* %dims291 to i8*
```

```
  %elmptr293 = getelementptr [1 x i64], [1 x i64]* %dims291, i64 0, i64 0
  store i64 1, i64* %elmptr293
  %malloccall294 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data295 = bitcast i8* %malloccall294 to [1 x i8*]*
  %data_as_i8ptr296 = bitcast [1 x i8*]* %data295 to i8*
  %elmptr297 = getelementptr [1 x i8*], [1 x i8*]* %data295, i64 0, i64 0
  store i8* %j285, i8** %elmptr297
  %dimsptr298 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 2
  store i8* %dims_as_i8ptr292, i8** %dimsptr298
  %dataptr299 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 3
  store i8* %data_as_i8ptr296, i8** %dataptr299
  %rc300 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 4
  store i8 0, i8* %rc300
  %tensor301 = bitcast %tensor_t* %raw_tensor287 to i8*
  %A302 = load i8*, i8** %A
  %access_tensor303 = call i8* @index_get(i8* %A302, i8* %tensor301)
  %i304 = load i8*, i8** %i
  %malloccall305 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor306 = bitcast i8* %malloccall305 to %tensor_t*
  %dtype307 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor306, i32 0, i32 0
  store i8 3, i8* %dtype307
  %ndims308 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor306, i32 0, i32 1
  store i8 1, i8* %ndims308
  %malloccall309 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims310 = bitcast i8* %malloccall309 to [1 x i64]*
  %dims_as_i8ptr311 = bitcast [1 x i64]* %dims310 to i8*
  %elmptr312 = getelementptr [1 x i64], [1 x i64]* %dims310, i64 0, i64 0
  store i64 1, i64* %elmptr312
  %malloccall313 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data314 = bitcast i8* %malloccall313 to [1 x i8*]*
  %data_as_i8ptr315 = bitcast [1 x i8*]* %data314 to i8*
  %elmptr316 = getelementptr [1 x i8*], [1 x i8*]* %data314, i64 0, i64 0
  store i8* %i304, i8** %elmptr316
  %dimsptr317 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor306, i32 0, i32 2
  store i8* %dims_as_i8ptr311, i8** %dimsptr317
  %dataptr318 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor306, i32 0, i32 3
  store i8* %data_as_i8ptr315, i8** %dataptr318
  %rc319 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor306, i32 0, i32 4
  store i8 0, i8* %rc319
  %tensor320 = bitcast %tensor_t* %raw_tensor306 to i8*
```

```
  %A321 = load i8*, i8** %A
  call void @index_put(i8* %A321, i8* %tensor320, i8* %access_tensor303)
  %t322 = load i8*, i8** %t
  %j323 = load i8*, i8** %j
  %malloccall324 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor325 = bitcast i8* %malloccall324 to %tensor_t*
  %dtype326 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor325, i32 0, i32 0
  store i8 3, i8* %dtype326
  %ndims327 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor325, i32 0, i32 1
  store i8 1, i8* %ndims327
  %malloccall328 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims329 = bitcast i8* %malloccall328 to [1 x i64]*
  %dims_as_i8ptr330 = bitcast [1 x i64]* %dims329 to i8*
  %elmptr331 = getelementptr [1 x i64], [1 x i64]* %dims329, i64 0, i64 0
  store i64 1, i64* %elmptr331
  %malloccall332 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data333 = bitcast i8* %malloccall332 to [1 x i8*]*
  %data_as_i8ptr334 = bitcast [1 x i8*]* %data333 to i8*
  %elmptr335 = getelementptr [1 x i8*], [1 x i8*]* %data333, i64 0, i64 0
  store i8* %j323, i8** %elmptr335
  %dimsptr336 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor325, i32 0, i32 2
  store i8* %dims_as_i8ptr330, i8** %dimsptr336
  %dataptr337 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor325, i32 0, i32 3
  store i8* %data_as_i8ptr334, i8** %dataptr337
  %rc338 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor325, i32 0, i32 4
  store i8 0, i8* %rc338
  %tensor339 = bitcast %tensor_t* %raw_tensor325 to i8*
  %A340 = load i8*, i8** %A
  call void @index_put(i8* %A340, i8* %tensor339, i8* %t322)
  br label %merge

else:                                             ; preds = %for_body
  br label %merge

merge341:                                         ; preds = %for
  %i342 = load i8*, i8** %i
  %malloccall343 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor344 = bitcast i8* %malloccall343 to %tensor_t*
  %dtype345 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor344, i32 0, i32 0
  store i8 0, i8* %dtype345
```

```
  %ndims346 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor344, i32 0, i32 1
  store i8 0, i8* %ndims346
  %malloccall347 = tail call i8* @malloc(i32 0)
  %dims348 = bitcast i8* %malloccall347 to [0 x i64]*
  %dims_as_i8ptr349 = bitcast [0 x i64]* %dims348 to i8*
  %malloccall350 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data351 = bitcast i8* %malloccall350 to [1 x i32]*
  %data_as_i8ptr352 = bitcast [1 x i32]* %data351 to i8*
  %elmptr353 = getelementptr [1 x i32], [1 x i32]* %data351, i32 0, i32 0
  store i32 1, i32* %elmptr353
  %dimsptr354 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor344, i32 0, i32 2
  store i8* %dims_as_i8ptr349, i8** %dimsptr354
  %dataptr355 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor344, i32 0, i32 3
  store i8* %data_as_i8ptr352, i8** %dataptr355
  %rc356 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor344, i32 0, i32 4
  store i8 0, i8* %rc356
  %tensor357 = bitcast %tensor_t* %raw_tensor344 to i8*
  %tmpOp358 = call i8* @add(i8* %i342, i8* %tensor357)
  %malloccall359 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor360 = bitcast i8* %malloccall359 to %tensor_t*
  %dtype361 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor360, i32 0, i32 0
  store i8 3, i8* %dtype361
  %ndims362 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor360, i32 0, i32 1
  store i8 1, i8* %ndims362
  %malloccall363 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims364 = bitcast i8* %malloccall363 to [1 x i64]*
  %dims_as_i8ptr365 = bitcast [1 x i64]* %dims364 to i8*
  %elmptr366 = getelementptr [1 x i64], [1 x i64]* %dims364, i64 0, i64 0
  store i64 1, i64* %elmptr366
  %malloccall367 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data368 = bitcast i8* %malloccall367 to [1 x i8*]*
  %data_as_i8ptr369 = bitcast [1 x i8*]* %data368 to i8*
  %elmptr370 = getelementptr [1 x i8*], [1 x i8*]* %data368, i64 0, i64 0
  store i8* %tmpOp358, i8** %elmptr370
  %dimsptr371 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor360, i32 0, i32 2
  store i8* %dims_as_i8ptr365, i8** %dimsptr371
  %dataptr372 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor360, i32 0, i32 3
  store i8* %data_as_i8ptr369, i8** %dataptr372
  %rc373 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor360, i32 0, i32 4
  store i8 0, i8* %rc373
```

```
  %tensor374 = bitcast %tensor_t* %raw_tensor360 to i8*
  %A375 = load i8*, i8** %A
  %access_tensor376 = call i8* @index_get(i8* %A375, i8* %tensor374)
  %lhsptr377 = load i8*, i8** %t
  call void @increase_rc(i8* %access_tensor376)
  call void @decrease_rc(i8* %lhsptr377)
  store i8* %access_tensor376, i8** %t
  %r378 = load i8*, i8** %r
  %malloccall379 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor380 = bitcast i8* %malloccall379 to %tensor_t*
  %dtype381 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor380, i32 0, i32 0
  store i8 3, i8* %dtype381
  %ndims382 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor380, i32 0, i32 1
  store i8 1, i8* %ndims382
  %malloccall383 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims384 = bitcast i8* %malloccall383 to [1 x i64]*
  %dims_as_i8ptr385 = bitcast [1 x i64]* %dims384 to i8*
  %elmptr386 = getelementptr [1 x i64], [1 x i64]* %dims384, i64 0, i64 0
  store i64 1, i64* %elmptr386
  %malloccall387 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data388 = bitcast i8* %malloccall387 to [1 x i8*]*
  %data_as_i8ptr389 = bitcast [1 x i8*]* %data388 to i8*
  %elmptr390 = getelementptr [1 x i8*], [1 x i8*]* %data388, i64 0, i64 0
  store i8* %r378, i8** %elmptr390
  %dimsptr391 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor380, i32 0, i32 2
  store i8* %dims_as_i8ptr385, i8** %dimsptr391
  %dataptr392 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor380, i32 0, i32 3
  store i8* %data_as_i8ptr389, i8** %dataptr392
  %rc393 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor380, i32 0, i32 4
  store i8 0, i8* %rc393
  %tensor394 = bitcast %tensor_t* %raw_tensor380 to i8*
  %A395 = load i8*, i8** %A
  %access_tensor396 = call i8* @index_get(i8* %A395, i8* %tensor394)
  %i397 = load i8*, i8** %i
  %malloccall398 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor399 = bitcast i8* %malloccall398 to %tensor_t*
  %dtype400 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 0
  store i8 0, i8* %dtype400
  %ndims401 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 1
  store i8 0, i8* %ndims401
```

```
  %malloccall402 = tail call i8* @malloc(i32 0)
  %dims403 = bitcast i8* %malloccall402 to [0 x i64]*
  %dims_as_i8ptr404 = bitcast [0 x i64]* %dims403 to i8*
  %malloccall405 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data406 = bitcast i8* %malloccall405 to [1 x i32]*
  %data_as_i8ptr407 = bitcast [1 x i32]* %data406 to i8*
  %elmptr408 = getelementptr [1 x i32], [1 x i32]* %data406, i32 0, i32 0
  store i32 1, i32* %elmptr408
  %dimsptr409 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 2
  store i8* %dims_as_i8ptr404, i8** %dimsptr409
  %dataptr410 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 3
  store i8* %data_as_i8ptr407, i8** %dataptr410
  %rc411 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 4
  store i8 0, i8* %rc411
  %tensor412 = bitcast %tensor_t* %raw_tensor399 to i8*
  %tmpOp413 = call i8* @add(i8* %i397, i8* %tensor412)
  %malloccall414 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor415 = bitcast i8* %malloccall414 to %tensor_t*
  %dtype416 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor415, i32 0, i32 0
  store i8 3, i8* %dtype416
  %ndims417 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor415, i32 0, i32 1
  store i8 1, i8* %ndims417
  %malloccall418 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims419 = bitcast i8* %malloccall418 to [1 x i64]*
  %dims_as_i8ptr420 = bitcast [1 x i64]* %dims419 to i8*
  %elmptr421 = getelementptr [1 x i64], [1 x i64]* %dims419, i64 0, i64 0
  store i64 1, i64* %elmptr421
  %malloccall422 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data423 = bitcast i8* %malloccall422 to [1 x i8*]*
  %data_as_i8ptr424 = bitcast [1 x i8*]* %data423 to i8*
  %elmptr425 = getelementptr [1 x i8*], [1 x i8*]* %data423, i64 0, i64 0
  store i8* %tmpOp413, i8** %elmptr425
  %dimsptr426 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor415, i32 0, i32 2
  store i8* %dims_as_i8ptr420, i8** %dimsptr426
  %dataptr427 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor415, i32 0, i32 3
  store i8* %data_as_i8ptr424, i8** %dataptr427
  %rc428 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor415, i32 0, i32 4
  store i8 0, i8* %rc428
  %tensor429 = bitcast %tensor_t* %raw_tensor415 to i8*
  %A430 = load i8*, i8** %A
```

```
  call void @index_put(i8* %A430, i8* %tensor429, i8* %access_tensor396)
  %t431 = load i8*, i8** %t
  %r432 = load i8*, i8** %r
  %malloccall433 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor434 = bitcast i8* %malloccall433 to %tensor_t*
  %dtype435 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor434, i32 0, i32 0
  store i8 3, i8* %dtype435
  %ndims436 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor434, i32 0, i32 1
  store i8 1, i8* %ndims436
  %malloccall437 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims438 = bitcast i8* %malloccall437 to [1 x i64]*
  %dims_as_i8ptr439 = bitcast [1 x i64]* %dims438 to i8*
  %elmptr440 = getelementptr [1 x i64], [1 x i64]* %dims438, i64 0, i64 0
  store i64 1, i64* %elmptr440
  %malloccall441 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data442 = bitcast i8* %malloccall441 to [1 x i8*]*
  %data_as_i8ptr443 = bitcast [1 x i8*]* %data442 to i8*
  %elmptr444 = getelementptr [1 x i8*], [1 x i8*]* %data442, i64 0, i64 0
  store i8* %r432, i8** %elmptr444
  %dimsptr445 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor434, i32 0, i32 2
  store i8* %dims_as_i8ptr439, i8** %dimsptr445
  %dataptr446 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor434, i32 0, i32 3
  store i8* %data_as_i8ptr443, i8** %dataptr446
  %rc447 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor434, i32 0, i32 4
  store i8 0, i8* %rc447
  %tensor448 = bitcast %tensor_t* %raw_tensor434 to i8*
  %A449 = load i8*, i8** %A
  call void @index_put(i8* %A449, i8* %tensor448, i8* %t431)
  %i450 = load i8*, i8** %i
  %malloccall451 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor452 = bitcast i8* %malloccall451 to %tensor_t*
  %dtype453 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor452, i32 0, i32 0
  store i8 0, i8* %dtype453
  %ndims454 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor452, i32 0, i32 1
  store i8 0, i8* %ndims454
  %malloccall455 = tail call i8* @malloc(i32 0)
  %dims456 = bitcast i8* %malloccall455 to [0 x i64]*
  %dims_as_i8ptr457 = bitcast [0 x i64]* %dims456 to i8*
  %malloccall458 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
```

```
  %data459 = bitcast i8* %malloccall458 to [1 x i32]*
  %data_as_i8ptr460 = bitcast [1 x i32]* %data459 to i8*
  %elmptr461 = getelementptr [1 x i32], [1 x i32]* %data459, i32 0, i32 0
  store i32 1, i32* %elmptr461
  %dimsptr462 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor452, i32 0, i32 2
  store i8* %dims_as_i8ptr457, i8** %dimsptr462
  %dataptr463 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor452, i32 0, i32 3
  store i8* %data_as_i8ptr460, i8** %dataptr463
  %rc464 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor452, i32 0, i32 4
  store i8 0, i8* %rc464
  %tensor465 = bitcast %tensor_t* %raw_tensor452 to i8*
  %tmpOp466 = call i8* @add(i8* %i450, i8* %tensor465)
  ret i8* %tmpOp466
}

define i8* @quicksort(i8* %0, i8* %1, i8* %2) {
entry:
  %A = alloca i8*
  store i8* %0, i8** %A
  %p = alloca i8*
  store i8* %1, i8** %p
  %r = alloca i8*
  store i8* %2, i8** %r
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall1 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall1 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall2 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall2 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
```

```
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %q = alloca i8*
  store i8* null, i8** %q
  %lhsptr = load i8*, i8** %q
  call void @increase_rc(i8* %tensor)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %tensor, i8** %q
  %p3 = load i8*, i8** %p
  %r4 = load i8*, i8** %r
  %tmpOp = call i8* @less(i8* %p3, i8* %r4)
  %bool = call i1 @bool_of_zero(i8* %tmpOp)
  br i1 %bool, label %then, label %else

merge:                                            ; preds = %else, %then
  %malloccall49 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor50 = bitcast i8* %malloccall49 to %tensor_t*
  %dtype51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 0
  store i8 0, i8* %dtype51
  %ndims52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 1
  store i8 0, i8* %ndims52
  %malloccall53 = tail call i8* @malloc(i32 0)
  %dims54 = bitcast i8* %malloccall53 to [0 x i64]*
  %dims_as_i8ptr55 = bitcast [0 x i64]* %dims54 to i8*
  %malloccall56 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data57 = bitcast i8* %malloccall56 to [1 x i32]*
  %data_as_i8ptr58 = bitcast [1 x i32]* %data57 to i8*
  %elmptr59 = getelementptr [1 x i32], [1 x i32]* %data57, i32 0, i32 0
  store i32 1, i32* %elmptr59
  %dimsptr60 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 2
  store i8* %dims_as_i8ptr55, i8** %dimsptr60
  %dataptr61 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 3
  store i8* %data_as_i8ptr58, i8** %dataptr61
  %rc62 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor50, i32 0, i32 4
  store i8 0, i8* %rc62
  %tensor63 = bitcast %tensor_t* %raw_tensor50 to i8*
  ret i8* %tensor63

then:                                             ; preds = %entry
  %A5 = load i8*, i8** %A
  %p6 = load i8*, i8** %p
  %r7 = load i8*, i8** %r
  %ret = call i8* @partition(i8* %A5, i8* %p6, i8* %r7)
```

```
  %lhsptr8 = load i8*, i8** %q
  call void @increase_rc(i8* %ret)
  call void @decrease_rc(i8* %lhsptr8)
  store i8* %ret, i8** %q
  %A9 = load i8*, i8** %A
  %p10 = load i8*, i8** %p
  %q11 = load i8*, i8** %q
  %malloccall12 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor13 = bitcast i8* %malloccall12 to %tensor_t*
  %dtype14 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 0
  store i8 0, i8* %dtype14
  %ndims15 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 1
  store i8 0, i8* %ndims15
  %malloccall16 = tail call i8* @malloc(i32 0)
  %dims17 = bitcast i8* %malloccall16 to [0 x i64]*
  %dims_as_i8ptr18 = bitcast [0 x i64]* %dims17 to i8*
  %malloccall19 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data20 = bitcast i8* %malloccall19 to [1 x i32]*
  %data_as_i8ptr21 = bitcast [1 x i32]* %data20 to i8*
  %elmptr22 = getelementptr [1 x i32], [1 x i32]* %data20, i32 0, i32 0
  store i32 1, i32* %elmptr22
  %dimsptr23 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 2
  store i8* %dims_as_i8ptr18, i8** %dimsptr23
  %dataptr24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 3
  store i8* %data_as_i8ptr21, i8** %dataptr24
  %rc25 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor13, i32 0, i32 4
  store i8 0, i8* %rc25
  %tensor26 = bitcast %tensor_t* %raw_tensor13 to i8*
  %tmpOp27 = call i8* @subtract(i8* %q11, i8* %tensor26)
  %ret28 = call i8* @quicksort(i8* %A9, i8* %p10, i8* %tmpOp27)
  %A29 = load i8*, i8** %A
  %q30 = load i8*, i8** %q
  %malloccall31 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor32 = bitcast i8* %malloccall31 to %tensor_t*
  %dtype33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 0
  store i8 0, i8* %dtype33
  %ndims34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 1
  store i8 0, i8* %ndims34
  %malloccall35 = tail call i8* @malloc(i32 0)
  %dims36 = bitcast i8* %malloccall35 to [0 x i64]*
  %dims_as_i8ptr37 = bitcast [0 x i64]* %dims36 to i8*
```

```
  %malloccall38 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data39 = bitcast i8* %malloccall38 to [1 x i32]*
  %data_as_i8ptr40 = bitcast [1 x i32]* %data39 to i8*
  %elmptr41 = getelementptr [1 x i32], [1 x i32]* %data39, i32 0, i32 0
  store i32 1, i32* %elmptr41
  %dimsptr42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 2
  store i8* %dims_as_i8ptr37, i8** %dimsptr42
  %dataptr43 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 3
  store i8* %data_as_i8ptr40, i8** %dataptr43
  %rc44 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor32, i32 0, i32 4
  store i8 0, i8* %rc44
  %tensor45 = bitcast %tensor_t* %raw_tensor32 to i8*
  %tmpOp46 = call i8* @add(i8* %q30, i8* %tensor45)
  %r47 = load i8*, i8** %r
  %ret48 = call i8* @quicksort(i8* %A29, i8* %tmpOp46, i8* %r47)
  br label %merge

else:                                             ; preds = %entry
  br label %merge
}
```

6.4. Sample Test Three

```
parallel_define fast_mul {
   overload __*__ (x,y) {
          map f1 {
                l = shape(x);
                z = l[0]//4;
                return x[0:z:1,0:l[1]:1]*y;
          }
          map f2 {
                l = shape(x);
                z = l[0]//4;
                return x[z:z.*2:1,0:l[1]:1]*y;
          }
          map f3 {
                l = shape(x);
                z = l[0]//4;
                return x[z.*2:z.*3:1,0:l[1]:1]*y;
```

```
            }
            map f4 {
                  l = shape(x);
                  z = l[0]//4;
                  return x[z.*3:l[0]:1,0:l[1]:1]*y;
            }
            reduce {
                  return cat(cat(f1,f2,0),cat(f3,f4,0),0);
            }
      }
}



# create data
num_of_dimensions = 3000;
x_train = rand([4000, 3000]);
b_true = 1.;
w_true = rand([3000]) .* 3.;
y_train = x_train * w_true + b_true;

# random init
w = rand([3000]);
b = 0.;

# Mean Square Error loss function
def loss() {
      using fast_mul;
    y_test = x_train * w + b;
      end fast_mul;
    error = sum((y_test - y_train).^2 .* 0.5);

    return error / float_of(num_of_dimensions);
}

# Derivative of Loss w.r.t W
def dLdW() {
      using fast_mul;
    y_test = x_train * w + b;
      end fast_mul;
    dLdw = (y_test - y_train) * x_train / float_of(num_of_dimensions);
```

```
    return dLdw;
}

# Derivative of Loss w.r.t W
def dLdB() {
    y_test = x_train * w + b;
    dLdb = sum(y_test - y_train) / float_of(num_of_dimensions);

    return dLdb;
}

# training
lr = 0.0001;
for (epoch in 0:100:1) {
    print("*****************");
    print(loss());
    w = w - dLdW() .* lr;
    b = b - dLdB() .* lr;
}
```

```
; ModuleID = 'TENLab'
source_filename = "TENLab"

%tensor_t = type { i8, i8, i8*, i8*, i8 }

@fast_mulMULmaps = global [4 x i8* (i8*, i8*)*] zeroinitializer
@fast_mulMULreduce.1 = global i8* (i8**)* null
@num_of_dimensions = global i8* null
@x_train = global i8* null
@b_true = global i8* null
@w_true = global i8* null
@y_train = global i8* null
@w = global i8* null
@b = global i8* null
@lr = global i8* null
@epoch = global i8* null

define i8 @main() {
entry:
```

```
  store i8* (i8*, i8*)* @fast_mulMULf1, i8* (i8*, i8*)** getelementptr inbounds ([4 x i8* (i8*, i8*)*],
[4 x i8* (i8*, i8*)*]* @fast_mulMULmaps, i32 0, i32 0)
  store i8* (i8*, i8*)* @fast_mulMULf2, i8* (i8*, i8*)** getelementptr inbounds ([4 x i8* (i8*, i8*)*],
[4 x i8* (i8*, i8*)*]* @fast_mulMULmaps, i32 0, i32 1)
  store i8* (i8*, i8*)* @fast_mulMULf3, i8* (i8*, i8*)** getelementptr inbounds ([4 x i8* (i8*, i8*)*],
[4 x i8* (i8*, i8*)*]* @fast_mulMULmaps, i32 0, i32 2)
  store i8* (i8*, i8*)* @fast_mulMULf4, i8* (i8*, i8*)** getelementptr inbounds ([4 x i8* (i8*, i8*)*],
[4 x i8* (i8*, i8*)*]* @fast_mulMULmaps, i32 0, i32 3)
  store i8* (i8**)* @fast_mulMULreduce, i8* (i8**)** @fast_mulMULreduce.1
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall1 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall1 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall2 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall2 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 3000, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %lhsptr = load i8*, i8** @num_of_dimensions
  call void @increase_rc(i8* %tensor)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %tensor, i8** @num_of_dimensions
  %malloccall3 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor4 = bitcast i8* %malloccall3 to %tensor_t*
  %dtype5 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor4, i32 0, i32 0
  store i8 0, i8* %dtype5
  %ndims6 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor4, i32 0, i32 1
  store i8 1, i8* %ndims6
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims8 = bitcast i8* %malloccall7 to [1 x i64]*
```

```
  %dims_as_i8ptr9 = bitcast [1 x i64]* %dims8 to i8*
  %elmptr10 = getelementptr [1 x i64], [1 x i64]* %dims8, i64 0, i64 0
  store i64 2, i64* %elmptr10
  %malloccall11 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i32* getelementptr
(i32, i32* null, i32 1) to i64), i64 2) to i32))
  %data12 = bitcast i8* %malloccall11 to [2 x i32]*
  %data_as_i8ptr13 = bitcast [2 x i32]* %data12 to i8*
  %elmptr14 = getelementptr [2 x i32], [2 x i32]* %data12, i32 0, i32 0
  store i32 4000, i32* %elmptr14
  %elmptr15 = getelementptr [2 x i32], [2 x i32]* %data12, i32 0, i32 1
  store i32 3000, i32* %elmptr15
  %dimsptr16 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor4, i32 0, i32 2
  store i8* %dims_as_i8ptr9, i8** %dimsptr16
  %dataptr17 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor4, i32 0, i32 3
  store i8* %data_as_i8ptr13, i8** %dataptr17
  %rc18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor4, i32 0, i32 4
  store i8 0, i8* %rc18
  %tensor19 = bitcast %tensor_t* %raw_tensor4 to i8*
  %rand = call i8* @tensor_rand(i8* %tensor19)
  %lhsptr20 = load i8*, i8** @x_train
  call void @increase_rc(i8* %rand)
  call void @decrease_rc(i8* %lhsptr20)
  store i8* %rand, i8** @x_train
  %malloccall21 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor22 = bitcast i8* %malloccall21 to %tensor_t*
  %dtype23 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 0
  store i8 1, i8* %dtype23
  %ndims24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 1
  store i8 0, i8* %ndims24
  %malloccall25 = tail call i8* @malloc(i32 0)
  %dims26 = bitcast i8* %malloccall25 to [0 x i64]*
  %dims_as_i8ptr27 = bitcast [0 x i64]* %dims26 to i8*
  %malloccall28 = tail call i8* @malloc(i32 ptrtoint (double* getelementptr (double, double* null,
i32 1) to i32))
  %data29 = bitcast i8* %malloccall28 to [1 x double]*
  %data_as_i8ptr30 = bitcast [1 x double]* %data29 to i8*
  %elmptr31 = getelementptr [1 x double], [1 x double]* %data29, i64 0, i64 0
  store double 1.000000e+00, double* %elmptr31
  %dimsptr32 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 2
  store i8* %dims_as_i8ptr27, i8** %dimsptr32
  %dataptr33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 3
  store i8* %data_as_i8ptr30, i8** %dataptr33
  %rc34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 4
```

```
  store i8 0, i8* %rc34
  %tensor35 = bitcast %tensor_t* %raw_tensor22 to i8*
  %lhsptr36 = load i8*, i8** @b_true
  call void @increase_rc(i8* %tensor35)
  call void @decrease_rc(i8* %lhsptr36)
  store i8* %tensor35, i8** @b_true
  %malloccall37 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor38 = bitcast i8* %malloccall37 to %tensor_t*
  %dtype39 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 0
  store i8 0, i8* %dtype39
  %ndims40 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 1
  store i8 1, i8* %ndims40
  %malloccall41 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims42 = bitcast i8* %malloccall41 to [1 x i64]*
  %dims_as_i8ptr43 = bitcast [1 x i64]* %dims42 to i8*
  %elmptr44 = getelementptr [1 x i64], [1 x i64]* %dims42, i64 0, i64 0
  store i64 1, i64* %elmptr44
  %malloccall45 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data46 = bitcast i8* %malloccall45 to [1 x i32]*
  %data_as_i8ptr47 = bitcast [1 x i32]* %data46 to i8*
  %elmptr48 = getelementptr [1 x i32], [1 x i32]* %data46, i32 0, i32 0
  store i32 3000, i32* %elmptr48
  %dimsptr49 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 2
  store i8* %dims_as_i8ptr43, i8** %dimsptr49
  %dataptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 3
  store i8* %data_as_i8ptr47, i8** %dataptr50
  %rc51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 4
  store i8 0, i8* %rc51
  %tensor52 = bitcast %tensor_t* %raw_tensor38 to i8*
  %rand53 = call i8* @tensor_rand(i8* %tensor52)
  %malloccall54 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor55 = bitcast i8* %malloccall54 to %tensor_t*
  %dtype56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 0
  store i8 1, i8* %dtype56
  %ndims57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 1
  store i8 0, i8* %ndims57
  %malloccall58 = tail call i8* @malloc(i32 0)
  %dims59 = bitcast i8* %malloccall58 to [0 x i64]*
  %dims_as_i8ptr60 = bitcast [0 x i64]* %dims59 to i8*
```

```
  %malloccall61 = tail call i8* @malloc(i32 ptrtoint (double* getelementptr (double, double* null,
i32 1) to i32))
  %data62 = bitcast i8* %malloccall61 to [1 x double]*
  %data_as_i8ptr63 = bitcast [1 x double]* %data62 to i8*
  %elmptr64 = getelementptr [1 x double], [1 x double]* %data62, i64 0, i64 0
  store double 3.000000e+00, double* %elmptr64
  %dimsptr65 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 2
  store i8* %dims_as_i8ptr60, i8** %dimsptr65
  %dataptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 3
  store i8* %data_as_i8ptr63, i8** %dataptr66
  %rc67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 4
  store i8 0, i8* %rc67
  %tensor68 = bitcast %tensor_t* %raw_tensor55 to i8*
  %tmpOp = call i8* @dotmul(i8* %rand53, i8* %tensor68)
  %lhsptr69 = load i8*, i8** @w_true
  call void @increase_rc(i8* %tmpOp)
  call void @decrease_rc(i8* %lhsptr69)
  store i8* %tmpOp, i8** @w_true
  %x_train = load i8*, i8** @x_train
  %w_true = load i8*, i8** @w_true
  %tmpOp70 = call i8* @mult(i8* %x_train, i8* %w_true)
  %b_true = load i8*, i8** @b_true
  %tmpOp71 = call i8* @add(i8* %tmpOp70, i8* %b_true)
  %lhsptr72 = load i8*, i8** @y_train
  call void @increase_rc(i8* %tmpOp71)
  call void @decrease_rc(i8* %lhsptr72)
  store i8* %tmpOp71, i8** @y_train
  %malloccall73 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor74 = bitcast i8* %malloccall73 to %tensor_t*
  %dtype75 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor74, i32 0, i32 0
  store i8 0, i8* %dtype75
  %ndims76 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor74, i32 0, i32 1
  store i8 1, i8* %ndims76
  %malloccall77 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims78 = bitcast i8* %malloccall77 to [1 x i64]*
  %dims_as_i8ptr79 = bitcast [1 x i64]* %dims78 to i8*
  %elmptr80 = getelementptr [1 x i64], [1 x i64]* %dims78, i64 0, i64 0
  store i64 1, i64* %elmptr80
  %malloccall81 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data82 = bitcast i8* %malloccall81 to [1 x i32]*
  %data_as_i8ptr83 = bitcast [1 x i32]* %data82 to i8*
```

```
  %elmptr84 = getelementptr [1 x i32], [1 x i32]* %data82, i32 0, i32 0
  store i32 3000, i32* %elmptr84
  %dimsptr85 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor74, i32 0, i32 2
  store i8* %dims_as_i8ptr79, i8** %dimsptr85
  %dataptr86 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor74, i32 0, i32 3
  store i8* %data_as_i8ptr83, i8** %dataptr86
  %rc87 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor74, i32 0, i32 4
  store i8 0, i8* %rc87
  %tensor88 = bitcast %tensor_t* %raw_tensor74 to i8*
  %rand89 = call i8* @tensor_rand(i8* %tensor88)
  %lhsptr90 = load i8*, i8** @w
  call void @increase_rc(i8* %rand89)
  call void @decrease_rc(i8* %lhsptr90)
  store i8* %rand89, i8** @w
  %malloccall91 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor92 = bitcast i8* %malloccall91 to %tensor_t*
  %dtype93 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor92, i32 0, i32 0
  store i8 1, i8* %dtype93
  %ndims94 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor92, i32 0, i32 1
  store i8 0, i8* %ndims94
  %malloccall95 = tail call i8* @malloc(i32 0)
  %dims96 = bitcast i8* %malloccall95 to [0 x i64]*
  %dims_as_i8ptr97 = bitcast [0 x i64]* %dims96 to i8*
  %malloccall98 = tail call i8* @malloc(i32 ptrtoint (double* getelementptr (double, double* null,
i32 1) to i32))
  %data99 = bitcast i8* %malloccall98 to [1 x double]*
  %data_as_i8ptr100 = bitcast [1 x double]* %data99 to i8*
  %elmptr101 = getelementptr [1 x double], [1 x double]* %data99, i64 0, i64 0
  store double 0.000000e+00, double* %elmptr101
  %dimsptr102 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor92, i32 0, i32 2
  store i8* %dims_as_i8ptr97, i8** %dimsptr102
  %dataptr103 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor92, i32 0, i32 3
  store i8* %data_as_i8ptr100, i8** %dataptr103
  %rc104 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor92, i32 0, i32 4
  store i8 0, i8* %rc104
  %tensor105 = bitcast %tensor_t* %raw_tensor92 to i8*
  %lhsptr106 = load i8*, i8** @b
  call void @increase_rc(i8* %tensor105)
  call void @decrease_rc(i8* %lhsptr106)
  store i8* %tensor105, i8** @b
  %malloccall107 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor108 = bitcast i8* %malloccall107 to %tensor_t*
```

```
  %dtype109 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor108, i32 0, i32 0
  store i8 1, i8* %dtype109
  %ndims110 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor108, i32 0, i32 1
  store i8 0, i8* %ndims110
  %malloccall111 = tail call i8* @malloc(i32 0)
  %dims112 = bitcast i8* %malloccall111 to [0 x i64]*
  %dims_as_i8ptr113 = bitcast [0 x i64]* %dims112 to i8*
  %malloccall114 = tail call i8* @malloc(i32 ptrtoint (double* getelementptr (double, double* null,
i32 1) to i32))
  %data115 = bitcast i8* %malloccall114 to [1 x double]*
  %data_as_i8ptr116 = bitcast [1 x double]* %data115 to i8*
  %elmptr117 = getelementptr [1 x double], [1 x double]* %data115, i64 0, i64 0
  store double 1.000000e-04, double* %elmptr117
  %dimsptr118 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor108, i32 0, i32 2
  store i8* %dims_as_i8ptr113, i8** %dimsptr118
  %dataptr119 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor108, i32 0, i32 3
  store i8* %data_as_i8ptr116, i8** %dataptr119
  %rc120 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor108, i32 0, i32 4
  store i8 0, i8* %rc120
  %tensor121 = bitcast %tensor_t* %raw_tensor108 to i8*
  %lhsptr122 = load i8*, i8** @lr
  call void @increase_rc(i8* %tensor121)
  call void @decrease_rc(i8* %lhsptr122)
  store i8* %tensor121, i8** @lr
  %idxptr = alloca i32
  store i32 0, i32* %idxptr
  %malloccall123 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor124 = bitcast i8* %malloccall123 to %tensor_t*
  %dtype125 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor124, i32 0, i32 0
  store i8 0, i8* %dtype125
  %ndims126 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor124, i32 0, i32 1
  store i8 0, i8* %ndims126
  %malloccall127 = tail call i8* @malloc(i32 0)
  %dims128 = bitcast i8* %malloccall127 to [0 x i64]*
  %dims_as_i8ptr129 = bitcast [0 x i64]* %dims128 to i8*
  %malloccall130 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data131 = bitcast i8* %malloccall130 to [1 x i32]*
  %data_as_i8ptr132 = bitcast [1 x i32]* %data131 to i8*
  %elmptr133 = getelementptr [1 x i32], [1 x i32]* %data131, i32 0, i32 0
  store i32 0, i32* %elmptr133
  %dimsptr134 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor124, i32 0, i32 2
  store i8* %dims_as_i8ptr129, i8** %dimsptr134
```

```
  %dataptr135 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor124, i32 0, i32 3
  store i8* %data_as_i8ptr132, i8** %dataptr135
  %rc136 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor124, i32 0, i32 4
  store i8 0, i8* %rc136
  %tensor137 = bitcast %tensor_t* %raw_tensor124 to i8*
  %malloccall138 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor139 = bitcast i8* %malloccall138 to %tensor_t*
  %dtype140 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor139, i32 0, i32 0
  store i8 0, i8* %dtype140
  %ndims141 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor139, i32 0, i32 1
  store i8 0, i8* %ndims141
  %malloccall142 = tail call i8* @malloc(i32 0)
  %dims143 = bitcast i8* %malloccall142 to [0 x i64]*
  %dims_as_i8ptr144 = bitcast [0 x i64]* %dims143 to i8*
  %malloccall145 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data146 = bitcast i8* %malloccall145 to [1 x i32]*
  %data_as_i8ptr147 = bitcast [1 x i32]* %data146 to i8*
  %elmptr148 = getelementptr [1 x i32], [1 x i32]* %data146, i32 0, i32 0
  store i32 100, i32* %elmptr148
  %dimsptr149 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor139, i32 0, i32 2
  store i8* %dims_as_i8ptr144, i8** %dimsptr149
  %dataptr150 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor139, i32 0, i32 3
  store i8* %data_as_i8ptr147, i8** %dataptr150
  %rc151 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor139, i32 0, i32 4
  store i8 0, i8* %rc151
  %tensor152 = bitcast %tensor_t* %raw_tensor139 to i8*
  %malloccall153 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor154 = bitcast i8* %malloccall153 to %tensor_t*
  %dtype155 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 0
  store i8 0, i8* %dtype155
  %ndims156 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 1
  store i8 0, i8* %ndims156
  %malloccall157 = tail call i8* @malloc(i32 0)
  %dims158 = bitcast i8* %malloccall157 to [0 x i64]*
  %dims_as_i8ptr159 = bitcast [0 x i64]* %dims158 to i8*
  %malloccall160 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data161 = bitcast i8* %malloccall160 to [1 x i32]*
  %data_as_i8ptr162 = bitcast [1 x i32]* %data161 to i8*
  %elmptr163 = getelementptr [1 x i32], [1 x i32]* %data161, i32 0, i32 0
  store i32 1, i32* %elmptr163
```

```
  %dimsptr164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 2
  store i8* %dims_as_i8ptr159, i8** %dimsptr164
  %dataptr165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 3
  store i8* %data_as_i8ptr162, i8** %dataptr165
  %rc166 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor154, i32 0, i32 4
  store i8 0, i8* %rc166
  %tensor167 = bitcast %tensor_t* %raw_tensor154 to i8*
  %tmpOp168 = call i8* @range(i8* %tensor137, i8* %tensor152, i8* %tensor167)
  %length = call i32 @len(i8* %tmpOp168)
  br label %for

for:                                              ; preds = %for_body, %entry
  %idx = load i32, i32* %idxptr
  %new_idx = add i32 %idx, 1
  store i32 %new_idx, i32* %idxptr
  %condition = icmp sgt i32 %new_idx, %length
  br i1 %condition, label %merge, label %for_body

for_body:                                         ; preds = %for
  %indicator_as_tensor = call i8* @index_get_int(i8* %tmpOp168, i32 %idx)
  store i8* %indicator_as_tensor, i8** @epoch
  %malloccall169 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor170 = bitcast i8* %malloccall169 to %tensor_t*
  %dtype171 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor170, i32 0, i32 0
  store i8 2, i8* %dtype171
  %ndims172 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor170, i32 0, i32 1
  store i8 1, i8* %ndims172
  %malloccall173 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims174 = bitcast i8* %malloccall173 to [1 x i64]*
  %dims_as_i8ptr175 = bitcast [1 x i64]* %dims174 to i8*
  %elmptr176 = getelementptr [1 x i64], [1 x i64]* %dims174, i64 0, i64 0
  store i64 17, i64* %elmptr176
  %malloccall177 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i8* getelementptr
(i8, i8* null, i32 1) to i64), i64 18) to i32))
  %data178 = bitcast i8* %malloccall177 to [18 x i8]*
  %data_as_i8ptr179 = bitcast [18 x i8]* %data178 to i8*
  %elmptr180 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 0
  store i8 42, i8* %elmptr180
  %elmptr181 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 1
  store i8 42, i8* %elmptr181
  %elmptr182 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 2
  store i8 42, i8* %elmptr182
```

```
%elmptr183 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 3
store i8 42, i8* %elmptr183
%elmptr184 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 4
store i8 42, i8* %elmptr184
%elmptr185 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 5
store i8 42, i8* %elmptr185
%elmptr186 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 6
store i8 42, i8* %elmptr186
%elmptr187 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 7
store i8 42, i8* %elmptr187
%elmptr188 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 8
store i8 42, i8* %elmptr188
%elmptr189 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 9
store i8 42, i8* %elmptr189
%elmptr190 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 10
store i8 42, i8* %elmptr190
%elmptr191 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 11
store i8 42, i8* %elmptr191
%elmptr192 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 12
store i8 42, i8* %elmptr192
%elmptr193 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 13
store i8 42, i8* %elmptr193
%elmptr194 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 14
store i8 42, i8* %elmptr194
%elmptr195 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 15
store i8 42, i8* %elmptr195
%elmptr196 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 16
store i8 42, i8* %elmptr196
%elmptr197 = getelementptr [18 x i8], [18 x i8]* %data178, i8 0, i8 17
store i8 0, i8* %elmptr197
%dimsptr198 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor170, i32 0, i32 2
store i8* %dims_as_i8ptr175, i8** %dimsptr198
%dataptr199 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor170, i32 0, i32 3
store i8* %data_as_i8ptr179, i8** %dataptr199
%rc200 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor170, i32 0, i32 4
store i8 0, i8* %rc200
%tensor201 = bitcast %tensor_t* %raw_tensor170 to i8*
call void @print(i8* %tensor201)
%ret = call i8* @loss()
call void @print(i8* %ret)
%w = load i8*, i8** @w
%ret202 = call i8* @dLdW()
%lr = load i8*, i8** @lr
%tmpOp203 = call i8* @dotmul(i8* %ret202, i8* %lr)
```

```
  %tmpOp204 = call i8* @subtract(i8* %w, i8* %tmpOp203)
  %lhsptr205 = load i8*, i8** @w
  call void @increase_rc(i8* %tmpOp204)
  call void @decrease_rc(i8* %lhsptr205)
  store i8* %tmpOp204, i8** @w
  %b = load i8*, i8** @b
  %ret206 = call i8* @dLdB()
  %lr207 = load i8*, i8** @lr
  %tmpOp208 = call i8* @dotmul(i8* %ret206, i8* %lr207)
  %tmpOp209 = call i8* @subtract(i8* %b, i8* %tmpOp208)
  %lhsptr210 = load i8*, i8** @b
  call void @increase_rc(i8* %tmpOp209)
  call void @decrease_rc(i8* %lhsptr210)
  store i8* %tmpOp209, i8** @b
  br label %for

merge:                                            ; preds = %for
  ret i8 0
}

declare i8* @add(i8*, i8*)

declare i8* @subtract(i8*, i8*)

declare i8* @negative(i8*)

declare i8* @mult(i8*, i8*)

declare i8* @dotmul(i8*, i8*)

declare i8* @divide(i8*, i8*)

declare i8* @floordivide(i8*, i8*)

declare i8* @matpow(i8*, i8*)

declare i8* @dotpow(i8*, i8*)

declare i8* @mod(i8*, i8*)

declare i8* @transpose(i8*)

declare i8* @equal(i8*, i8*)
```

```
declare i8* @notequal(i8*, i8*)

declare i8* @greater(i8*, i8*)

declare i8* @greaterequal(i8*, i8*)

declare i8* @less(i8*, i8*)

declare i8* @lessequal(i8*, i8*)

declare i8* @range(i8*, i8*, i8*)

declare void @print(i8*)

declare i8* @logicaland(i8*, i8*)

declare i8* @logicalor(i8*, i8*)

declare i8* @logicalnot(i8*)

declare i32 @len(i8*)

declare i8* @zeros(i8*)

declare i8* @cat(i8*, i8*, i8*)

declare i8* @shape(i8*)

declare i8* @ones(i8*)

declare i8* @tensor_rand(i8*)

declare i8* @sum(i8*)

declare i8* @any(i8*)

declare i8* @all(i8*)

declare i8* @tensor_abs(i8*)

declare i8* @tensor_log(i8*)

declare i8* @tensor_floor(i8*)
```

```
declare i8* @tensor_ceil(i8*)

declare i8* @tensor_round(i8*)

declare i8* @int_of(i8*)

declare i8* @float_of(i8*)

declare i8* @inverse(i8*)

declare i1 @bool_of_zero(i8*)

declare i8* @pe_calc(i8* (i8*, i8*)**, i32, i8* (i8**)*, i8*, i8*)

declare i8* @index_get(i8*, i8*)

declare i8* @index_get_int(i8*, i32)

declare void @index_put(i8*, i8*, i8*)

declare void @increase_rc(i8*)

declare void @decrease_rc(i8*)

define i8* @fast_mulMULf1(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
  %l = alloca i8*
  store i8* null, i8** %l
  %lhsptr = load i8*, i8** %l
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
```

```
  store i8 0, i8* %ndims
  %malloccall2 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall2 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall3 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall3 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall4 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor5 = bitcast i8* %malloccall4 to %tensor_t*
  %dtype6 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 0
  store i8 3, i8* %dtype6
  %ndims7 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 1
  store i8 1, i8* %ndims7
  %malloccall8 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims9 = bitcast i8* %malloccall8 to [1 x i64]*
  %dims_as_i8ptr10 = bitcast [1 x i64]* %dims9 to i8*
  %elmptr11 = getelementptr [1 x i64], [1 x i64]* %dims9, i64 0, i64 0
  store i64 1, i64* %elmptr11
  %malloccall12 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data13 = bitcast i8* %malloccall12 to [1 x i8*]*
  %data_as_i8ptr14 = bitcast [1 x i8*]* %data13 to i8*
  %elmptr15 = getelementptr [1 x i8*], [1 x i8*]* %data13, i64 0, i64 0
  store i8* %tensor, i8** %elmptr15
  %dimsptr16 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 2
  store i8* %dims_as_i8ptr10, i8** %dimsptr16
  %dataptr17 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 3
  store i8* %data_as_i8ptr14, i8** %dataptr17
  %rc18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 4
  store i8 0, i8* %rc18
  %tensor19 = bitcast %tensor_t* %raw_tensor5 to i8*
  %l20 = load i8*, i8** %l
  %access_tensor = call i8* @index_get(i8* %l20, i8* %tensor19)
  %malloccall21 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor22 = bitcast i8* %malloccall21 to %tensor_t*
  %dtype23 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 0
  store i8 0, i8* %dtype23
  %ndims24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 1
  store i8 0, i8* %ndims24
  %malloccall25 = tail call i8* @malloc(i32 0)
  %dims26 = bitcast i8* %malloccall25 to [0 x i64]*
  %dims_as_i8ptr27 = bitcast [0 x i64]* %dims26 to i8*
  %malloccall28 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data29 = bitcast i8* %malloccall28 to [1 x i32]*
  %data_as_i8ptr30 = bitcast [1 x i32]* %data29 to i8*
  %elmptr31 = getelementptr [1 x i32], [1 x i32]* %data29, i32 0, i32 0
  store i32 4, i32* %elmptr31
  %dimsptr32 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 2
  store i8* %dims_as_i8ptr27, i8** %dimsptr32
  %dataptr33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 3
  store i8* %data_as_i8ptr30, i8** %dataptr33
  %rc34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 4
  store i8 0, i8* %rc34
  %tensor35 = bitcast %tensor_t* %raw_tensor22 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor35)
  %z = alloca i8*
  store i8* null, i8** %z
  %lhsptr36 = load i8*, i8** %z
  call void @increase_rc(i8* %tmpOp)
  call void @decrease_rc(i8* %lhsptr36)
  store i8* %tmpOp, i8** %z
  %malloccall37 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor38 = bitcast i8* %malloccall37 to %tensor_t*
  %dtype39 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 0
  store i8 0, i8* %dtype39
  %ndims40 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 1
  store i8 0, i8* %ndims40
  %malloccall41 = tail call i8* @malloc(i32 0)
  %dims42 = bitcast i8* %malloccall41 to [0 x i64]*
  %dims_as_i8ptr43 = bitcast [0 x i64]* %dims42 to i8*
  %malloccall44 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data45 = bitcast i8* %malloccall44 to [1 x i32]*
  %data_as_i8ptr46 = bitcast [1 x i32]* %data45 to i8*
  %elmptr47 = getelementptr [1 x i32], [1 x i32]* %data45, i32 0, i32 0
  store i32 0, i32* %elmptr47
```

```
  %dimsptr48 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 2
  store i8* %dims_as_i8ptr43, i8** %dimsptr48
  %dataptr49 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 3
  store i8* %data_as_i8ptr46, i8** %dataptr49
  %rc50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor38, i32 0, i32 4
  store i8 0, i8* %rc50
  %tensor51 = bitcast %tensor_t* %raw_tensor38 to i8*
  %z52 = load i8*, i8** %z
  %malloccall53 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor54 = bitcast i8* %malloccall53 to %tensor_t*
  %dtype55 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 0
  store i8 0, i8* %dtype55
  %ndims56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 1
  store i8 0, i8* %ndims56
  %malloccall57 = tail call i8* @malloc(i32 0)
  %dims58 = bitcast i8* %malloccall57 to [0 x i64]*
  %dims_as_i8ptr59 = bitcast [0 x i64]* %dims58 to i8*
  %malloccall60 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data61 = bitcast i8* %malloccall60 to [1 x i32]*
  %data_as_i8ptr62 = bitcast [1 x i32]* %data61 to i8*
  %elmptr63 = getelementptr [1 x i32], [1 x i32]* %data61, i32 0, i32 0
  store i32 1, i32* %elmptr63
  %dimsptr64 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 2
  store i8* %dims_as_i8ptr59, i8** %dimsptr64
  %dataptr65 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 3
  store i8* %data_as_i8ptr62, i8** %dataptr65
  %rc66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor54, i32 0, i32 4
  store i8 0, i8* %rc66
  %tensor67 = bitcast %tensor_t* %raw_tensor54 to i8*
  %tmpOp68 = call i8* @range(i8* %tensor51, i8* %z52, i8* %tensor67)
  %malloccall69 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor70 = bitcast i8* %malloccall69 to %tensor_t*
  %dtype71 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 0
  store i8 0, i8* %dtype71
  %ndims72 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 1
  store i8 0, i8* %ndims72
  %malloccall73 = tail call i8* @malloc(i32 0)
  %dims74 = bitcast i8* %malloccall73 to [0 x i64]*
  %dims_as_i8ptr75 = bitcast [0 x i64]* %dims74 to i8*
  %malloccall76 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
```

```
  %data77 = bitcast i8* %malloccall76 to [1 x i32]*
  %data_as_i8ptr78 = bitcast [1 x i32]* %data77 to i8*
  %elmptr79 = getelementptr [1 x i32], [1 x i32]* %data77, i32 0, i32 0
  store i32 0, i32* %elmptr79
  %dimsptr80 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 2
  store i8* %dims_as_i8ptr75, i8** %dimsptr80
  %dataptr81 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 3
  store i8* %data_as_i8ptr78, i8** %dataptr81
  %rc82 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 4
  store i8 0, i8* %rc82
  %tensor83 = bitcast %tensor_t* %raw_tensor70 to i8*
  %malloccall84 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor85 = bitcast i8* %malloccall84 to %tensor_t*
  %dtype86 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor85, i32 0, i32 0
  store i8 0, i8* %dtype86
  %ndims87 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor85, i32 0, i32 1
  store i8 0, i8* %ndims87
  %malloccall88 = tail call i8* @malloc(i32 0)
  %dims89 = bitcast i8* %malloccall88 to [0 x i64]*
  %dims_as_i8ptr90 = bitcast [0 x i64]* %dims89 to i8*
  %malloccall91 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data92 = bitcast i8* %malloccall91 to [1 x i32]*
  %data_as_i8ptr93 = bitcast [1 x i32]* %data92 to i8*
  %elmptr94 = getelementptr [1 x i32], [1 x i32]* %data92, i32 0, i32 0
  store i32 1, i32* %elmptr94
  %dimsptr95 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor85, i32 0, i32 2
  store i8* %dims_as_i8ptr90, i8** %dimsptr95
  %dataptr96 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor85, i32 0, i32 3
  store i8* %data_as_i8ptr93, i8** %dataptr96
  %rc97 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor85, i32 0, i32 4
  store i8 0, i8* %rc97
  %tensor98 = bitcast %tensor_t* %raw_tensor85 to i8*
  %malloccall99 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor100 = bitcast i8* %malloccall99 to %tensor_t*
  %dtype101 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor100, i32 0, i32 0
  store i8 3, i8* %dtype101
  %ndims102 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor100, i32 0, i32 1
  store i8 1, i8* %ndims102
  %malloccall103 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims104 = bitcast i8* %malloccall103 to [1 x i64]*
```

```
  %dims_as_i8ptr105 = bitcast [1 x i64]* %dims104 to i8*
  %elmptr106 = getelementptr [1 x i64], [1 x i64]* %dims104, i64 0, i64 0
  store i64 1, i64* %elmptr106
  %malloccall107 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data108 = bitcast i8* %malloccall107 to [1 x i8*]*
  %data_as_i8ptr109 = bitcast [1 x i8*]* %data108 to i8*
  %elmptr110 = getelementptr [1 x i8*], [1 x i8*]* %data108, i64 0, i64 0
  store i8* %tensor98, i8** %elmptr110
  %dimsptr111 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor100, i32 0, i32 2
  store i8* %dims_as_i8ptr105, i8** %dimsptr111
  %dataptr112 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor100, i32 0, i32 3
  store i8* %data_as_i8ptr109, i8** %dataptr112
  %rc113 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor100, i32 0, i32 4
  store i8 0, i8* %rc113
  %tensor114 = bitcast %tensor_t* %raw_tensor100 to i8*
  %l115 = load i8*, i8** %l
  %access_tensor116 = call i8* @index_get(i8* %l115, i8* %tensor114)
  %malloccall117 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor118 = bitcast i8* %malloccall117 to %tensor_t*
  %dtype119 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 0
  store i8 0, i8* %dtype119
  %ndims120 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 1
  store i8 0, i8* %ndims120
  %malloccall121 = tail call i8* @malloc(i32 0)
  %dims122 = bitcast i8* %malloccall121 to [0 x i64]*
  %dims_as_i8ptr123 = bitcast [0 x i64]* %dims122 to i8*
  %malloccall124 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data125 = bitcast i8* %malloccall124 to [1 x i32]*
  %data_as_i8ptr126 = bitcast [1 x i32]* %data125 to i8*
  %elmptr127 = getelementptr [1 x i32], [1 x i32]* %data125, i32 0, i32 0
  store i32 1, i32* %elmptr127
  %dimsptr128 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 2
  store i8* %dims_as_i8ptr123, i8** %dimsptr128
  %dataptr129 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 3
  store i8* %data_as_i8ptr126, i8** %dataptr129
  %rc130 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 4
  store i8 0, i8* %rc130
  %tensor131 = bitcast %tensor_t* %raw_tensor118 to i8*
  %tmpOp132 = call i8* @range(i8* %tensor83, i8* %access_tensor116, i8* %tensor131)
  %malloccall133 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor134 = bitcast i8* %malloccall133 to %tensor_t*
  %dtype135 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 0
  store i8 3, i8* %dtype135
  %ndims136 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 1
  store i8 1, i8* %ndims136
  %malloccall137 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims138 = bitcast i8* %malloccall137 to [1 x i64]*
  %dims_as_i8ptr139 = bitcast [1 x i64]* %dims138 to i8*
  %elmptr140 = getelementptr [1 x i64], [1 x i64]* %dims138, i64 0, i64 0
  store i64 2, i64* %elmptr140
  %malloccall141 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data142 = bitcast i8* %malloccall141 to [2 x i8*]*
  %data_as_i8ptr143 = bitcast [2 x i8*]* %data142 to i8*
  %elmptr144 = getelementptr [2 x i8*], [2 x i8*]* %data142, i64 0, i64 0
  store i8* %tmpOp68, i8** %elmptr144
  %elmptr145 = getelementptr [2 x i8*], [2 x i8*]* %data142, i64 0, i64 1
  store i8* %tmpOp132, i8** %elmptr145
  %dimsptr146 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 2
  store i8* %dims_as_i8ptr139, i8** %dimsptr146
  %dataptr147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 3
  store i8* %data_as_i8ptr143, i8** %dataptr147
  %rc148 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 4
  store i8 0, i8* %rc148
  %tensor149 = bitcast %tensor_t* %raw_tensor134 to i8*
  %x150 = load i8*, i8** %x
  %access_tensor151 = call i8* @index_get(i8* %x150, i8* %tensor149)
  %y152 = load i8*, i8** %y
  %tmpOp153 = call i8* @mult(i8* %access_tensor151, i8* %y152)
  ret i8* %tmpOp153
}

declare noalias i8* @malloc(i32)

define i8* @fast_mulMULf2(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
  %l = alloca i8*
```

```
  store i8* null, i8** %l
  %lhsptr = load i8*, i8** %l
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall2 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall2 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall3 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall3 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall4 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor5 = bitcast i8* %malloccall4 to %tensor_t*
  %dtype6 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 0
  store i8 3, i8* %dtype6
  %ndims7 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 1
  store i8 1, i8* %ndims7
  %malloccall8 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims9 = bitcast i8* %malloccall8 to [1 x i64]*
  %dims_as_i8ptr10 = bitcast [1 x i64]* %dims9 to i8*
  %elmptr11 = getelementptr [1 x i64], [1 x i64]* %dims9, i64 0, i64 0
  store i64 1, i64* %elmptr11
  %malloccall12 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data13 = bitcast i8* %malloccall12 to [1 x i8*]*
  %data_as_i8ptr14 = bitcast [1 x i8*]* %data13 to i8*
  %elmptr15 = getelementptr [1 x i8*], [1 x i8*]* %data13, i64 0, i64 0
  store i8* %tensor, i8** %elmptr15
```

```
  %dimsptr16 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 2
  store i8* %dims_as_i8ptr10, i8** %dimsptr16
  %dataptr17 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 3
  store i8* %data_as_i8ptr14, i8** %dataptr17
  %rc18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 4
  store i8 0, i8* %rc18
  %tensor19 = bitcast %tensor_t* %raw_tensor5 to i8*
  %l20 = load i8*, i8** %l
  %access_tensor = call i8* @index_get(i8* %l20, i8* %tensor19)
  %malloccall21 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor22 = bitcast i8* %malloccall21 to %tensor_t*
  %dtype23 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 0
  store i8 0, i8* %dtype23
  %ndims24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 1
  store i8 0, i8* %ndims24
  %malloccall25 = tail call i8* @malloc(i32 0)
  %dims26 = bitcast i8* %malloccall25 to [0 x i64]*
  %dims_as_i8ptr27 = bitcast [0 x i64]* %dims26 to i8*
  %malloccall28 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data29 = bitcast i8* %malloccall28 to [1 x i32]*
  %data_as_i8ptr30 = bitcast [1 x i32]* %data29 to i8*
  %elmptr31 = getelementptr [1 x i32], [1 x i32]* %data29, i32 0, i32 0
  store i32 4, i32* %elmptr31
  %dimsptr32 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 2
  store i8* %dims_as_i8ptr27, i8** %dimsptr32
  %dataptr33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 3
  store i8* %data_as_i8ptr30, i8** %dataptr33
  %rc34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 4
  store i8 0, i8* %rc34
  %tensor35 = bitcast %tensor_t* %raw_tensor22 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor35)
  %z = alloca i8*
  store i8* null, i8** %z
  %lhsptr36 = load i8*, i8** %z
  call void @increase_rc(i8* %tmpOp)
  call void @decrease_rc(i8* %lhsptr36)
  store i8* %tmpOp, i8** %z
  %z37 = load i8*, i8** %z
  %z38 = load i8*, i8** %z
  %malloccall39 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor40 = bitcast i8* %malloccall39 to %tensor_t*
```

```
  %dtype41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 0
  store i8 0, i8* %dtype41
  %ndims42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 1
  store i8 0, i8* %ndims42
  %malloccall43 = tail call i8* @malloc(i32 0)
  %dims44 = bitcast i8* %malloccall43 to [0 x i64]*
  %dims_as_i8ptr45 = bitcast [0 x i64]* %dims44 to i8*
  %malloccall46 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data47 = bitcast i8* %malloccall46 to [1 x i32]*
  %data_as_i8ptr48 = bitcast [1 x i32]* %data47 to i8*
  %elmptr49 = getelementptr [1 x i32], [1 x i32]* %data47, i32 0, i32 0
  store i32 2, i32* %elmptr49
  %dimsptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 2
  store i8* %dims_as_i8ptr45, i8** %dimsptr50
  %dataptr51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 3
  store i8* %data_as_i8ptr48, i8** %dataptr51
  %rc52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 4
  store i8 0, i8* %rc52
  %tensor53 = bitcast %tensor_t* %raw_tensor40 to i8*
  %tmpOp54 = call i8* @dotmul(i8* %z38, i8* %tensor53)
  %malloccall55 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor56 = bitcast i8* %malloccall55 to %tensor_t*
  %dtype57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 0
  store i8 0, i8* %dtype57
  %ndims58 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 1
  store i8 0, i8* %ndims58
  %malloccall59 = tail call i8* @malloc(i32 0)
  %dims60 = bitcast i8* %malloccall59 to [0 x i64]*
  %dims_as_i8ptr61 = bitcast [0 x i64]* %dims60 to i8*
  %malloccall62 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data63 = bitcast i8* %malloccall62 to [1 x i32]*
  %data_as_i8ptr64 = bitcast [1 x i32]* %data63 to i8*
  %elmptr65 = getelementptr [1 x i32], [1 x i32]* %data63, i32 0, i32 0
  store i32 1, i32* %elmptr65
  %dimsptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 2
  store i8* %dims_as_i8ptr61, i8** %dimsptr66
  %dataptr67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 3
  store i8* %data_as_i8ptr64, i8** %dataptr67
  %rc68 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 4
  store i8 0, i8* %rc68
  %tensor69 = bitcast %tensor_t* %raw_tensor56 to i8*
```

```
  %tmpOp70 = call i8* @range(i8* %z37, i8* %tmpOp54, i8* %tensor69)
  %malloccall71 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor72 = bitcast i8* %malloccall71 to %tensor_t*
  %dtype73 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 0
  store i8 0, i8* %dtype73
  %ndims74 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 1
  store i8 0, i8* %ndims74
  %malloccall75 = tail call i8* @malloc(i32 0)
  %dims76 = bitcast i8* %malloccall75 to [0 x i64]*
  %dims_as_i8ptr77 = bitcast [0 x i64]* %dims76 to i8*
  %malloccall78 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data79 = bitcast i8* %malloccall78 to [1 x i32]*
  %data_as_i8ptr80 = bitcast [1 x i32]* %data79 to i8*
  %elmptr81 = getelementptr [1 x i32], [1 x i32]* %data79, i32 0, i32 0
  store i32 0, i32* %elmptr81
  %dimsptr82 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 2
  store i8* %dims_as_i8ptr77, i8** %dimsptr82
  %dataptr83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 3
  store i8* %data_as_i8ptr80, i8** %dataptr83
  %rc84 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 4
  store i8 0, i8* %rc84
  %tensor85 = bitcast %tensor_t* %raw_tensor72 to i8*
  %malloccall86 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor87 = bitcast i8* %malloccall86 to %tensor_t*
  %dtype88 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor87, i32 0, i32 0
  store i8 0, i8* %dtype88
  %ndims89 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor87, i32 0, i32 1
  store i8 0, i8* %ndims89
  %malloccall90 = tail call i8* @malloc(i32 0)
  %dims91 = bitcast i8* %malloccall90 to [0 x i64]*
  %dims_as_i8ptr92 = bitcast [0 x i64]* %dims91 to i8*
  %malloccall93 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data94 = bitcast i8* %malloccall93 to [1 x i32]*
  %data_as_i8ptr95 = bitcast [1 x i32]* %data94 to i8*
  %elmptr96 = getelementptr [1 x i32], [1 x i32]* %data94, i32 0, i32 0
  store i32 1, i32* %elmptr96
  %dimsptr97 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor87, i32 0, i32 2
  store i8* %dims_as_i8ptr92, i8** %dimsptr97
  %dataptr98 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor87, i32 0, i32 3
  store i8* %data_as_i8ptr95, i8** %dataptr98
```

```
  %rc99 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor87, i32 0, i32 4
  store i8 0, i8* %rc99
  %tensor100 = bitcast %tensor_t* %raw_tensor87 to i8*
  %malloccall101 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor102 = bitcast i8* %malloccall101 to %tensor_t*
  %dtype103 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor102, i32 0, i32 0
  store i8 3, i8* %dtype103
  %ndims104 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor102, i32 0, i32 1
  store i8 1, i8* %ndims104
  %malloccall105 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims106 = bitcast i8* %malloccall105 to [1 x i64]*
  %dims_as_i8ptr107 = bitcast [1 x i64]* %dims106 to i8*
  %elmptr108 = getelementptr [1 x i64], [1 x i64]* %dims106, i64 0, i64 0
  store i64 1, i64* %elmptr108
  %malloccall109 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data110 = bitcast i8* %malloccall109 to [1 x i8*]*
  %data_as_i8ptr111 = bitcast [1 x i8*]* %data110 to i8*
  %elmptr112 = getelementptr [1 x i8*], [1 x i8*]* %data110, i64 0, i64 0
  store i8* %tensor100, i8** %elmptr112
  %dimsptr113 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor102, i32 0, i32 2
  store i8* %dims_as_i8ptr107, i8** %dimsptr113
  %dataptr114 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor102, i32 0, i32 3
  store i8* %data_as_i8ptr111, i8** %dataptr114
  %rc115 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor102, i32 0, i32 4
  store i8 0, i8* %rc115
  %tensor116 = bitcast %tensor_t* %raw_tensor102 to i8*
  %l117 = load i8*, i8** %l
  %access_tensor118 = call i8* @index_get(i8* %l117, i8* %tensor116)
  %malloccall119 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor120 = bitcast i8* %malloccall119 to %tensor_t*
  %dtype121 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor120, i32 0, i32 0
  store i8 0, i8* %dtype121
  %ndims122 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor120, i32 0, i32 1
  store i8 0, i8* %ndims122
  %malloccall123 = tail call i8* @malloc(i32 0)
  %dims124 = bitcast i8* %malloccall123 to [0 x i64]*
  %dims_as_i8ptr125 = bitcast [0 x i64]* %dims124 to i8*
  %malloccall126 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data127 = bitcast i8* %malloccall126 to [1 x i32]*
```

```
  %data_as_i8ptr128 = bitcast [1 x i32]* %data127 to i8*
  %elmptr129 = getelementptr [1 x i32], [1 x i32]* %data127, i32 0, i32 0
  store i32 1, i32* %elmptr129
  %dimsptr130 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor120, i32 0, i32 2
  store i8* %dims_as_i8ptr125, i8** %dimsptr130
  %dataptr131 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor120, i32 0, i32 3
  store i8* %data_as_i8ptr128, i8** %dataptr131
  %rc132 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor120, i32 0, i32 4
  store i8 0, i8* %rc132
  %tensor133 = bitcast %tensor_t* %raw_tensor120 to i8*
  %tmpOp134 = call i8* @range(i8* %tensor85, i8* %access_tensor118, i8* %tensor133)
  %malloccall135 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor136 = bitcast i8* %malloccall135 to %tensor_t*
  %dtype137 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 0
  store i8 3, i8* %dtype137
  %ndims138 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 1
  store i8 1, i8* %ndims138
  %malloccall139 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims140 = bitcast i8* %malloccall139 to [1 x i64]*
  %dims_as_i8ptr141 = bitcast [1 x i64]* %dims140 to i8*
  %elmptr142 = getelementptr [1 x i64], [1 x i64]* %dims140, i64 0, i64 0
  store i64 2, i64* %elmptr142
  %malloccall143 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data144 = bitcast i8* %malloccall143 to [2 x i8*]*
  %data_as_i8ptr145 = bitcast [2 x i8*]* %data144 to i8*
  %elmptr146 = getelementptr [2 x i8*], [2 x i8*]* %data144, i64 0, i64 0
  store i8* %tmpOp70, i8** %elmptr146
  %elmptr147 = getelementptr [2 x i8*], [2 x i8*]* %data144, i64 0, i64 1
  store i8* %tmpOp134, i8** %elmptr147
  %dimsptr148 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 2
  store i8* %dims_as_i8ptr141, i8** %dimsptr148
  %dataptr149 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 3
  store i8* %data_as_i8ptr145, i8** %dataptr149
  %rc150 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 4
  store i8 0, i8* %rc150
  %tensor151 = bitcast %tensor_t* %raw_tensor136 to i8*
  %x152 = load i8*, i8** %x
  %access_tensor153 = call i8* @index_get(i8* %x152, i8* %tensor151)
  %y154 = load i8*, i8** %y
  %tmpOp155 = call i8* @mult(i8* %access_tensor153, i8* %y154)
  ret i8* %tmpOp155
```

```
}

define i8* @fast_mulMULf3(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
  %l = alloca i8*
  store i8* null, i8** %l
  %lhsptr = load i8*, i8** %l
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall2 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall2 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall3 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall3 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall4 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor5 = bitcast i8* %malloccall4 to %tensor_t*
  %dtype6 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 0
  store i8 3, i8* %dtype6
  %ndims7 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 1
```

```
  store i8 1, i8* %ndims7
  %malloccall8 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims9 = bitcast i8* %malloccall8 to [1 x i64]*
  %dims_as_i8ptr10 = bitcast [1 x i64]* %dims9 to i8*
  %elmptr11 = getelementptr [1 x i64], [1 x i64]* %dims9, i64 0, i64 0
  store i64 1, i64* %elmptr11
  %malloccall12 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data13 = bitcast i8* %malloccall12 to [1 x i8*]*
  %data_as_i8ptr14 = bitcast [1 x i8*]* %data13 to i8*
  %elmptr15 = getelementptr [1 x i8*], [1 x i8*]* %data13, i64 0, i64 0
  store i8* %tensor, i8** %elmptr15
  %dimsptr16 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 2
  store i8* %dims_as_i8ptr10, i8** %dimsptr16
  %dataptr17 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 3
  store i8* %data_as_i8ptr14, i8** %dataptr17
  %rc18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 4
  store i8 0, i8* %rc18
  %tensor19 = bitcast %tensor_t* %raw_tensor5 to i8*
  %l20 = load i8*, i8** %l
  %access_tensor = call i8* @index_get(i8* %l20, i8* %tensor19)
  %malloccall21 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor22 = bitcast i8* %malloccall21 to %tensor_t*
  %dtype23 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 0
  store i8 0, i8* %dtype23
  %ndims24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 1
  store i8 0, i8* %ndims24
  %malloccall25 = tail call i8* @malloc(i32 0)
  %dims26 = bitcast i8* %malloccall25 to [0 x i64]*
  %dims_as_i8ptr27 = bitcast [0 x i64]* %dims26 to i8*
  %malloccall28 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data29 = bitcast i8* %malloccall28 to [1 x i32]*
  %data_as_i8ptr30 = bitcast [1 x i32]* %data29 to i8*
  %elmptr31 = getelementptr [1 x i32], [1 x i32]* %data29, i32 0, i32 0
  store i32 4, i32* %elmptr31
  %dimsptr32 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 2
  store i8* %dims_as_i8ptr27, i8** %dimsptr32
  %dataptr33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 3
  store i8* %data_as_i8ptr30, i8** %dataptr33
  %rc34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 4
  store i8 0, i8* %rc34
  %tensor35 = bitcast %tensor_t* %raw_tensor22 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor35)
```

```
  %z = alloca i8*
  store i8* null, i8** %z
  %lhsptr36 = load i8*, i8** %z
  call void @increase_rc(i8* %tmpOp)
  call void @decrease_rc(i8* %lhsptr36)
  store i8* %tmpOp, i8** %z
  %z37 = load i8*, i8** %z
  %malloccall38 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor39 = bitcast i8* %malloccall38 to %tensor_t*
  %dtype40 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 0
  store i8 0, i8* %dtype40
  %ndims41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 1
  store i8 0, i8* %ndims41
  %malloccall42 = tail call i8* @malloc(i32 0)
  %dims43 = bitcast i8* %malloccall42 to [0 x i64]*
  %dims_as_i8ptr44 = bitcast [0 x i64]* %dims43 to i8*
  %malloccall45 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data46 = bitcast i8* %malloccall45 to [1 x i32]*
  %data_as_i8ptr47 = bitcast [1 x i32]* %data46 to i8*
  %elmptr48 = getelementptr [1 x i32], [1 x i32]* %data46, i32 0, i32 0
  store i32 2, i32* %elmptr48
  %dimsptr49 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 2
  store i8* %dims_as_i8ptr44, i8** %dimsptr49
  %dataptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 3
  store i8* %data_as_i8ptr47, i8** %dataptr50
  %rc51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 4
  store i8 0, i8* %rc51
  %tensor52 = bitcast %tensor_t* %raw_tensor39 to i8*
  %tmpOp53 = call i8* @dotmul(i8* %z37, i8* %tensor52)
  %z54 = load i8*, i8** %z
  %malloccall55 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor56 = bitcast i8* %malloccall55 to %tensor_t*
  %dtype57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 0
  store i8 0, i8* %dtype57
  %ndims58 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 1
  store i8 0, i8* %ndims58
  %malloccall59 = tail call i8* @malloc(i32 0)
  %dims60 = bitcast i8* %malloccall59 to [0 x i64]*
  %dims_as_i8ptr61 = bitcast [0 x i64]* %dims60 to i8*
  %malloccall62 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
```

```
  %data63 = bitcast i8* %malloccall62 to [1 x i32]*
  %data_as_i8ptr64 = bitcast [1 x i32]* %data63 to i8*
  %elmptr65 = getelementptr [1 x i32], [1 x i32]* %data63, i32 0, i32 0
  store i32 3, i32* %elmptr65
  %dimsptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 2
  store i8* %dims_as_i8ptr61, i8** %dimsptr66
  %dataptr67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 3
  store i8* %data_as_i8ptr64, i8** %dataptr67
  %rc68 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor56, i32 0, i32 4
  store i8 0, i8* %rc68
  %tensor69 = bitcast %tensor_t* %raw_tensor56 to i8*
  %tmpOp70 = call i8* @dotmul(i8* %z54, i8* %tensor69)
  %malloccall71 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor72 = bitcast i8* %malloccall71 to %tensor_t*
  %dtype73 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 0
  store i8 0, i8* %dtype73
  %ndims74 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 1
  store i8 0, i8* %ndims74
  %malloccall75 = tail call i8* @malloc(i32 0)
  %dims76 = bitcast i8* %malloccall75 to [0 x i64]*
  %dims_as_i8ptr77 = bitcast [0 x i64]* %dims76 to i8*
  %malloccall78 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data79 = bitcast i8* %malloccall78 to [1 x i32]*
  %data_as_i8ptr80 = bitcast [1 x i32]* %data79 to i8*
  %elmptr81 = getelementptr [1 x i32], [1 x i32]* %data79, i32 0, i32 0
  store i32 1, i32* %elmptr81
  %dimsptr82 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 2
  store i8* %dims_as_i8ptr77, i8** %dimsptr82
  %dataptr83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 3
  store i8* %data_as_i8ptr80, i8** %dataptr83
  %rc84 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor72, i32 0, i32 4
  store i8 0, i8* %rc84
  %tensor85 = bitcast %tensor_t* %raw_tensor72 to i8*
  %tmpOp86 = call i8* @range(i8* %tmpOp53, i8* %tmpOp70, i8* %tensor85)
  %malloccall87 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor88 = bitcast i8* %malloccall87 to %tensor_t*
  %dtype89 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 0
  store i8 0, i8* %dtype89
  %ndims90 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 1
  store i8 0, i8* %ndims90
  %malloccall91 = tail call i8* @malloc(i32 0)
```

```
  %dims92 = bitcast i8* %malloccall91 to [0 x i64]*
  %dims_as_i8ptr93 = bitcast [0 x i64]* %dims92 to i8*
  %malloccall94 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data95 = bitcast i8* %malloccall94 to [1 x i32]*
  %data_as_i8ptr96 = bitcast [1 x i32]* %data95 to i8*
  %elmptr97 = getelementptr [1 x i32], [1 x i32]* %data95, i32 0, i32 0
  store i32 0, i32* %elmptr97
  %dimsptr98 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 2
  store i8* %dims_as_i8ptr93, i8** %dimsptr98
  %dataptr99 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 3
  store i8* %data_as_i8ptr96, i8** %dataptr99
  %rc100 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 4
  store i8 0, i8* %rc100
  %tensor101 = bitcast %tensor_t* %raw_tensor88 to i8*
  %malloccall102 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor103 = bitcast i8* %malloccall102 to %tensor_t*
  %dtype104 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor103, i32 0, i32 0
  store i8 0, i8* %dtype104
  %ndims105 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor103, i32 0, i32 1
  store i8 0, i8* %ndims105
  %malloccall106 = tail call i8* @malloc(i32 0)
  %dims107 = bitcast i8* %malloccall106 to [0 x i64]*
  %dims_as_i8ptr108 = bitcast [0 x i64]* %dims107 to i8*
  %malloccall109 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data110 = bitcast i8* %malloccall109 to [1 x i32]*
  %data_as_i8ptr111 = bitcast [1 x i32]* %data110 to i8*
  %elmptr112 = getelementptr [1 x i32], [1 x i32]* %data110, i32 0, i32 0
  store i32 1, i32* %elmptr112
  %dimsptr113 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor103, i32 0, i32 2
  store i8* %dims_as_i8ptr108, i8** %dimsptr113
  %dataptr114 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor103, i32 0, i32 3
  store i8* %data_as_i8ptr111, i8** %dataptr114
  %rc115 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor103, i32 0, i32 4
  store i8 0, i8* %rc115
  %tensor116 = bitcast %tensor_t* %raw_tensor103 to i8*
  %malloccall117 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor118 = bitcast i8* %malloccall117 to %tensor_t*
  %dtype119 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 0
  store i8 3, i8* %dtype119
  %ndims120 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 1
```

```
  store i8 1, i8* %ndims120
  %malloccall121 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims122 = bitcast i8* %malloccall121 to [1 x i64]*
  %dims_as_i8ptr123 = bitcast [1 x i64]* %dims122 to i8*
  %elmptr124 = getelementptr [1 x i64], [1 x i64]* %dims122, i64 0, i64 0
  store i64 1, i64* %elmptr124
  %malloccall125 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data126 = bitcast i8* %malloccall125 to [1 x i8*]*
  %data_as_i8ptr127 = bitcast [1 x i8*]* %data126 to i8*
  %elmptr128 = getelementptr [1 x i8*], [1 x i8*]* %data126, i64 0, i64 0
  store i8* %tensor116, i8** %elmptr128
  %dimsptr129 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 2
  store i8* %dims_as_i8ptr123, i8** %dimsptr129
  %dataptr130 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 3
  store i8* %data_as_i8ptr127, i8** %dataptr130
  %rc131 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor118, i32 0, i32 4
  store i8 0, i8* %rc131
  %tensor132 = bitcast %tensor_t* %raw_tensor118 to i8*
  %l133 = load i8*, i8** %l
  %access_tensor134 = call i8* @index_get(i8* %l133, i8* %tensor132)
  %malloccall135 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor136 = bitcast i8* %malloccall135 to %tensor_t*
  %dtype137 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 0
  store i8 0, i8* %dtype137
  %ndims138 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 1
  store i8 0, i8* %ndims138
  %malloccall139 = tail call i8* @malloc(i32 0)
  %dims140 = bitcast i8* %malloccall139 to [0 x i64]*
  %dims_as_i8ptr141 = bitcast [0 x i64]* %dims140 to i8*
  %malloccall142 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data143 = bitcast i8* %malloccall142 to [1 x i32]*
  %data_as_i8ptr144 = bitcast [1 x i32]* %data143 to i8*
  %elmptr145 = getelementptr [1 x i32], [1 x i32]* %data143, i32 0, i32 0
  store i32 1, i32* %elmptr145
  %dimsptr146 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 2
  store i8* %dims_as_i8ptr141, i8** %dimsptr146
  %dataptr147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 3
  store i8* %data_as_i8ptr144, i8** %dataptr147
  %rc148 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor136, i32 0, i32 4
  store i8 0, i8* %rc148
```

```
  %tensor149 = bitcast %tensor_t* %raw_tensor136 to i8*
  %tmpOp150 = call i8* @range(i8* %tensor101, i8* %access_tensor134, i8* %tensor149)
  %malloccall151 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor152 = bitcast i8* %malloccall151 to %tensor_t*
  %dtype153 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 0
  store i8 3, i8* %dtype153
  %ndims154 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 1
  store i8 1, i8* %ndims154
  %malloccall155 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims156 = bitcast i8* %malloccall155 to [1 x i64]*
  %dims_as_i8ptr157 = bitcast [1 x i64]* %dims156 to i8*
  %elmptr158 = getelementptr [1 x i64], [1 x i64]* %dims156, i64 0, i64 0
  store i64 2, i64* %elmptr158
  %malloccall159 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data160 = bitcast i8* %malloccall159 to [2 x i8*]*
  %data_as_i8ptr161 = bitcast [2 x i8*]* %data160 to i8*
  %elmptr162 = getelementptr [2 x i8*], [2 x i8*]* %data160, i64 0, i64 0
  store i8* %tmpOp86, i8** %elmptr162
  %elmptr163 = getelementptr [2 x i8*], [2 x i8*]* %data160, i64 0, i64 1
  store i8* %tmpOp150, i8** %elmptr163
  %dimsptr164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 2
  store i8* %dims_as_i8ptr157, i8** %dimsptr164
  %dataptr165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 3
  store i8* %data_as_i8ptr161, i8** %dataptr165
  %rc166 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 4
  store i8 0, i8* %rc166
  %tensor167 = bitcast %tensor_t* %raw_tensor152 to i8*
  %x168 = load i8*, i8** %x
  %access_tensor169 = call i8* @index_get(i8* %x168, i8* %tensor167)
  %y170 = load i8*, i8** %y
  %tmpOp171 = call i8* @mult(i8* %access_tensor169, i8* %y170)
  ret i8* %tmpOp171
}

define i8* @fast_mulMULf4(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
```

```
  %shape = call i8* @shape(i8* %x1)
  %l = alloca i8*
  store i8* null, i8** %l
  %lhsptr = load i8*, i8** %l
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall2 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall2 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall3 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall3 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall4 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor5 = bitcast i8* %malloccall4 to %tensor_t*
  %dtype6 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 0
  store i8 3, i8* %dtype6
  %ndims7 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 1
  store i8 1, i8* %ndims7
  %malloccall8 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims9 = bitcast i8* %malloccall8 to [1 x i64]*
  %dims_as_i8ptr10 = bitcast [1 x i64]* %dims9 to i8*
  %elmptr11 = getelementptr [1 x i64], [1 x i64]* %dims9, i64 0, i64 0
  store i64 1, i64* %elmptr11
  %malloccall12 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data13 = bitcast i8* %malloccall12 to [1 x i8*]*
  %data_as_i8ptr14 = bitcast [1 x i8*]* %data13 to i8*
```

```
  %elmptr15 = getelementptr [1 x i8*], [1 x i8*]* %data13, i64 0, i64 0
  store i8* %tensor, i8** %elmptr15
  %dimsptr16 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 2
  store i8* %dims_as_i8ptr10, i8** %dimsptr16
  %dataptr17 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 3
  store i8* %data_as_i8ptr14, i8** %dataptr17
  %rc18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor5, i32 0, i32 4
  store i8 0, i8* %rc18
  %tensor19 = bitcast %tensor_t* %raw_tensor5 to i8*
  %l20 = load i8*, i8** %l
  %access_tensor = call i8* @index_get(i8* %l20, i8* %tensor19)
  %malloccall21 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor22 = bitcast i8* %malloccall21 to %tensor_t*
  %dtype23 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 0
  store i8 0, i8* %dtype23
  %ndims24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 1
  store i8 0, i8* %ndims24
  %malloccall25 = tail call i8* @malloc(i32 0)
  %dims26 = bitcast i8* %malloccall25 to [0 x i64]*
  %dims_as_i8ptr27 = bitcast [0 x i64]* %dims26 to i8*
  %malloccall28 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data29 = bitcast i8* %malloccall28 to [1 x i32]*
  %data_as_i8ptr30 = bitcast [1 x i32]* %data29 to i8*
  %elmptr31 = getelementptr [1 x i32], [1 x i32]* %data29, i32 0, i32 0
  store i32 4, i32* %elmptr31
  %dimsptr32 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 2
  store i8* %dims_as_i8ptr27, i8** %dimsptr32
  %dataptr33 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 3
  store i8* %data_as_i8ptr30, i8** %dataptr33
  %rc34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor22, i32 0, i32 4
  store i8 0, i8* %rc34
  %tensor35 = bitcast %tensor_t* %raw_tensor22 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor35)
  %z = alloca i8*
  store i8* null, i8** %z
  %lhsptr36 = load i8*, i8** %z
  call void @increase_rc(i8* %tmpOp)
  call void @decrease_rc(i8* %lhsptr36)
  store i8* %tmpOp, i8** %z
  %z37 = load i8*, i8** %z
  %malloccall38 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor39 = bitcast i8* %malloccall38 to %tensor_t*
  %dtype40 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 0
  store i8 0, i8* %dtype40
  %ndims41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 1
  store i8 0, i8* %ndims41
  %malloccall42 = tail call i8* @malloc(i32 0)
  %dims43 = bitcast i8* %malloccall42 to [0 x i64]*
  %dims_as_i8ptr44 = bitcast [0 x i64]* %dims43 to i8*
  %malloccall45 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data46 = bitcast i8* %malloccall45 to [1 x i32]*
  %data_as_i8ptr47 = bitcast [1 x i32]* %data46 to i8*
  %elmptr48 = getelementptr [1 x i32], [1 x i32]* %data46, i32 0, i32 0
  store i32 3, i32* %elmptr48
  %dimsptr49 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 2
  store i8* %dims_as_i8ptr44, i8** %dimsptr49
  %dataptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 3
  store i8* %data_as_i8ptr47, i8** %dataptr50
  %rc51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor39, i32 0, i32 4
  store i8 0, i8* %rc51
  %tensor52 = bitcast %tensor_t* %raw_tensor39 to i8*
  %tmpOp53 = call i8* @dotmul(i8* %z37, i8* %tensor52)
  %malloccall54 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor55 = bitcast i8* %malloccall54 to %tensor_t*
  %dtype56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 0
  store i8 0, i8* %dtype56
  %ndims57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 1
  store i8 0, i8* %ndims57
  %malloccall58 = tail call i8* @malloc(i32 0)
  %dims59 = bitcast i8* %malloccall58 to [0 x i64]*
  %dims_as_i8ptr60 = bitcast [0 x i64]* %dims59 to i8*
  %malloccall61 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data62 = bitcast i8* %malloccall61 to [1 x i32]*
  %data_as_i8ptr63 = bitcast [1 x i32]* %data62 to i8*
  %elmptr64 = getelementptr [1 x i32], [1 x i32]* %data62, i32 0, i32 0
  store i32 0, i32* %elmptr64
  %dimsptr65 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 2
  store i8* %dims_as_i8ptr60, i8** %dimsptr65
  %dataptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 3
  store i8* %data_as_i8ptr63, i8** %dataptr66
  %rc67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 4
  store i8 0, i8* %rc67
```

```
  %tensor68 = bitcast %tensor_t* %raw_tensor55 to i8*
  %malloccall69 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor70 = bitcast i8* %malloccall69 to %tensor_t*
  %dtype71 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 0
  store i8 3, i8* %dtype71
  %ndims72 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 1
  store i8 1, i8* %ndims72
  %malloccall73 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims74 = bitcast i8* %malloccall73 to [1 x i64]*
  %dims_as_i8ptr75 = bitcast [1 x i64]* %dims74 to i8*
  %elmptr76 = getelementptr [1 x i64], [1 x i64]* %dims74, i64 0, i64 0
  store i64 1, i64* %elmptr76
  %malloccall77 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data78 = bitcast i8* %malloccall77 to [1 x i8*]*
  %data_as_i8ptr79 = bitcast [1 x i8*]* %data78 to i8*
  %elmptr80 = getelementptr [1 x i8*], [1 x i8*]* %data78, i64 0, i64 0
  store i8* %tensor68, i8** %elmptr80
  %dimsptr81 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 2
  store i8* %dims_as_i8ptr75, i8** %dimsptr81
  %dataptr82 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 3
  store i8* %data_as_i8ptr79, i8** %dataptr82
  %rc83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor70, i32 0, i32 4
  store i8 0, i8* %rc83
  %tensor84 = bitcast %tensor_t* %raw_tensor70 to i8*
  %l85 = load i8*, i8** %l
  %access_tensor86 = call i8* @index_get(i8* %l85, i8* %tensor84)
  %malloccall87 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor88 = bitcast i8* %malloccall87 to %tensor_t*
  %dtype89 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 0
  store i8 0, i8* %dtype89
  %ndims90 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 1
  store i8 0, i8* %ndims90
  %malloccall91 = tail call i8* @malloc(i32 0)
  %dims92 = bitcast i8* %malloccall91 to [0 x i64]*
  %dims_as_i8ptr93 = bitcast [0 x i64]* %dims92 to i8*
  %malloccall94 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data95 = bitcast i8* %malloccall94 to [1 x i32]*
  %data_as_i8ptr96 = bitcast [1 x i32]* %data95 to i8*
  %elmptr97 = getelementptr [1 x i32], [1 x i32]* %data95, i32 0, i32 0
  store i32 1, i32* %elmptr97
```

```
  %dimsptr98 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 2
  store i8* %dims_as_i8ptr93, i8** %dimsptr98
  %dataptr99 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 3
  store i8* %data_as_i8ptr96, i8** %dataptr99
  %rc100 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor88, i32 0, i32 4
  store i8 0, i8* %rc100
  %tensor101 = bitcast %tensor_t* %raw_tensor88 to i8*
  %tmpOp102 = call i8* @range(i8* %tmpOp53, i8* %access_tensor86, i8* %tensor101)
  %malloccall103 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor104 = bitcast i8* %malloccall103 to %tensor_t*
  %dtype105 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 0
  store i8 0, i8* %dtype105
  %ndims106 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 1
  store i8 0, i8* %ndims106
  %malloccall107 = tail call i8* @malloc(i32 0)
  %dims108 = bitcast i8* %malloccall107 to [0 x i64]*
  %dims_as_i8ptr109 = bitcast [0 x i64]* %dims108 to i8*
  %malloccall110 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data111 = bitcast i8* %malloccall110 to [1 x i32]*
  %data_as_i8ptr112 = bitcast [1 x i32]* %data111 to i8*
  %elmptr113 = getelementptr [1 x i32], [1 x i32]* %data111, i32 0, i32 0
  store i32 0, i32* %elmptr113
  %dimsptr114 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 2
  store i8* %dims_as_i8ptr109, i8** %dimsptr114
  %dataptr115 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 3
  store i8* %data_as_i8ptr112, i8** %dataptr115
  %rc116 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 4
  store i8 0, i8* %rc116
  %tensor117 = bitcast %tensor_t* %raw_tensor104 to i8*
  %malloccall118 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor119 = bitcast i8* %malloccall118 to %tensor_t*
  %dtype120 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 0
  store i8 0, i8* %dtype120
  %ndims121 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 1
  store i8 0, i8* %ndims121
  %malloccall122 = tail call i8* @malloc(i32 0)
  %dims123 = bitcast i8* %malloccall122 to [0 x i64]*
  %dims_as_i8ptr124 = bitcast [0 x i64]* %dims123 to i8*
  %malloccall125 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data126 = bitcast i8* %malloccall125 to [1 x i32]*
```

```
  %data_as_i8ptr127 = bitcast [1 x i32]* %data126 to i8*
  %elmptr128 = getelementptr [1 x i32], [1 x i32]* %data126, i32 0, i32 0
  store i32 1, i32* %elmptr128
  %dimsptr129 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 2
  store i8* %dims_as_i8ptr124, i8** %dimsptr129
  %dataptr130 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 3
  store i8* %data_as_i8ptr127, i8** %dataptr130
  %rc131 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 4
  store i8 0, i8* %rc131
  %tensor132 = bitcast %tensor_t* %raw_tensor119 to i8*
  %malloccall133 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor134 = bitcast i8* %malloccall133 to %tensor_t*
  %dtype135 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 0
  store i8 3, i8* %dtype135
  %ndims136 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 1
  store i8 1, i8* %ndims136
  %malloccall137 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims138 = bitcast i8* %malloccall137 to [1 x i64]*
  %dims_as_i8ptr139 = bitcast [1 x i64]* %dims138 to i8*
  %elmptr140 = getelementptr [1 x i64], [1 x i64]* %dims138, i64 0, i64 0
  store i64 1, i64* %elmptr140
  %malloccall141 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data142 = bitcast i8* %malloccall141 to [1 x i8*]*
  %data_as_i8ptr143 = bitcast [1 x i8*]* %data142 to i8*
  %elmptr144 = getelementptr [1 x i8*], [1 x i8*]* %data142, i64 0, i64 0
  store i8* %tensor132, i8** %elmptr144
  %dimsptr145 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 2
  store i8* %dims_as_i8ptr139, i8** %dimsptr145
  %dataptr146 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 3
  store i8* %data_as_i8ptr143, i8** %dataptr146
  %rc147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor134, i32 0, i32 4
  store i8 0, i8* %rc147
  %tensor148 = bitcast %tensor_t* %raw_tensor134 to i8*
  %l149 = load i8*, i8** %l
  %access_tensor150 = call i8* @index_get(i8* %l149, i8* %tensor148)
  %malloccall151 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor152 = bitcast i8* %malloccall151 to %tensor_t*
  %dtype153 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 0
  store i8 0, i8* %dtype153
  %ndims154 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 1
```

```
  store i8 0, i8* %ndims154
  %malloccall155 = tail call i8* @malloc(i32 0)
  %dims156 = bitcast i8* %malloccall155 to [0 x i64]*
  %dims_as_i8ptr157 = bitcast [0 x i64]* %dims156 to i8*
  %malloccall158 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data159 = bitcast i8* %malloccall158 to [1 x i32]*
  %data_as_i8ptr160 = bitcast [1 x i32]* %data159 to i8*
  %elmptr161 = getelementptr [1 x i32], [1 x i32]* %data159, i32 0, i32 0
  store i32 1, i32* %elmptr161
  %dimsptr162 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 2
  store i8* %dims_as_i8ptr157, i8** %dimsptr162
  %dataptr163 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 3
  store i8* %data_as_i8ptr160, i8** %dataptr163
  %rc164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor152, i32 0, i32 4
  store i8 0, i8* %rc164
  %tensor165 = bitcast %tensor_t* %raw_tensor152 to i8*
  %tmpOp166 = call i8* @range(i8* %tensor117, i8* %access_tensor150, i8* %tensor165)
  %malloccall167 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor168 = bitcast i8* %malloccall167 to %tensor_t*
  %dtype169 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor168, i32 0, i32 0
  store i8 3, i8* %dtype169
  %ndims170 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor168, i32 0, i32 1
  store i8 1, i8* %ndims170
  %malloccall171 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims172 = bitcast i8* %malloccall171 to [1 x i64]*
  %dims_as_i8ptr173 = bitcast [1 x i64]* %dims172 to i8*
  %elmptr174 = getelementptr [1 x i64], [1 x i64]* %dims172, i64 0, i64 0
  store i64 2, i64* %elmptr174
  %malloccall175 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data176 = bitcast i8* %malloccall175 to [2 x i8*]*
  %data_as_i8ptr177 = bitcast [2 x i8*]* %data176 to i8*
  %elmptr178 = getelementptr [2 x i8*], [2 x i8*]* %data176, i64 0, i64 0
  store i8* %tmpOp102, i8** %elmptr178
  %elmptr179 = getelementptr [2 x i8*], [2 x i8*]* %data176, i64 0, i64 1
  store i8* %tmpOp166, i8** %elmptr179
  %dimsptr180 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor168, i32 0, i32 2
  store i8* %dims_as_i8ptr173, i8** %dimsptr180
  %dataptr181 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor168, i32 0, i32 3
  store i8* %data_as_i8ptr177, i8** %dataptr181
  %rc182 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor168, i32 0, i32 4
```

```
  store i8 0, i8* %rc182
  %tensor183 = bitcast %tensor_t* %raw_tensor168 to i8*
  %x184 = load i8*, i8** %x
  %access_tensor185 = call i8* @index_get(i8* %x184, i8* %tensor183)
  %y186 = load i8*, i8** %y
  %tmpOp187 = call i8* @mult(i8* %access_tensor185, i8* %y186)
  ret i8* %tmpOp187
}

define i8* @fast_mulMULreduce(i8** %result) {
entry:
  %result1 = alloca i8**
  store i8** %result, i8*** %result1
  %f1 = alloca i8*
  %newa = load i8**, i8*** %result1
  %tmp = getelementptr i8*, i8** %newa, i32 0
  %newv = load i8*, i8** %tmp
  store i8* %newv, i8** %f1
  %f2 = alloca i8*
  %newa2 = load i8**, i8*** %result1
  %tmp3 = getelementptr i8*, i8** %newa2, i32 1
  %newv4 = load i8*, i8** %tmp3
  store i8* %newv4, i8** %f2
  %f3 = alloca i8*
  %newa5 = load i8**, i8*** %result1
  %tmp6 = getelementptr i8*, i8** %newa5, i32 2
  %newv7 = load i8*, i8** %tmp6
  store i8* %newv7, i8** %f3
  %f4 = alloca i8*
  %newa8 = load i8**, i8*** %result1
  %tmp9 = getelementptr i8*, i8** %newa8, i32 3
  %newv10 = load i8*, i8** %tmp9
  store i8* %newv10, i8** %f4
  %f111 = load i8*, i8** %f1
  %f212 = load i8*, i8** %f2
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall13 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall13 to [0 x i64]*
```

```
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall14 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data = bitcast i8* %malloccall14 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %cat = call i8* @cat(i8* %f111, i8* %f212, i8* %tensor)
  %f315 = load i8*, i8** %f3
  %f416 = load i8*, i8** %f4
  %malloccall17 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor18 = bitcast i8* %malloccall17 to %tensor_t*
  %dtype19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor18, i32 0, i32 0
  store i8 0, i8* %dtype19
  %ndims20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor18, i32 0, i32 1
  store i8 0, i8* %ndims20
  %malloccall21 = tail call i8* @malloc(i32 0)
  %dims22 = bitcast i8* %malloccall21 to [0 x i64]*
  %dims_as_i8ptr23 = bitcast [0 x i64]* %dims22 to i8*
  %malloccall24 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data25 = bitcast i8* %malloccall24 to [1 x i32]*
  %data_as_i8ptr26 = bitcast [1 x i32]* %data25 to i8*
  %elmptr27 = getelementptr [1 x i32], [1 x i32]* %data25, i32 0, i32 0
  store i32 0, i32* %elmptr27
  %dimsptr28 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor18, i32 0, i32 2
  store i8* %dims_as_i8ptr23, i8** %dimsptr28
  %dataptr29 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor18, i32 0, i32 3
  store i8* %data_as_i8ptr26, i8** %dataptr29
  %rc30 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor18, i32 0, i32 4
  store i8 0, i8* %rc30
  %tensor31 = bitcast %tensor_t* %raw_tensor18 to i8*
  %cat32 = call i8* @cat(i8* %f315, i8* %f416, i8* %tensor31)
  %malloccall33 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor34 = bitcast i8* %malloccall33 to %tensor_t*
```

```
  %dtype35 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor34, i32 0, i32 0
  store i8 0, i8* %dtype35
  %ndims36 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor34, i32 0, i32 1
  store i8 0, i8* %ndims36
  %malloccall37 = tail call i8* @malloc(i32 0)
  %dims38 = bitcast i8* %malloccall37 to [0 x i64]*
  %dims_as_i8ptr39 = bitcast [0 x i64]* %dims38 to i8*
  %malloccall40 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data41 = bitcast i8* %malloccall40 to [1 x i32]*
  %data_as_i8ptr42 = bitcast [1 x i32]* %data41 to i8*
  %elmptr43 = getelementptr [1 x i32], [1 x i32]* %data41, i32 0, i32 0
  store i32 0, i32* %elmptr43
  %dimsptr44 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor34, i32 0, i32 2
  store i8* %dims_as_i8ptr39, i8** %dimsptr44
  %dataptr45 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor34, i32 0, i32 3
  store i8* %data_as_i8ptr42, i8** %dataptr45
  %rc46 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor34, i32 0, i32 4
  store i8 0, i8* %rc46
  %tensor47 = bitcast %tensor_t* %raw_tensor34 to i8*
  %cat48 = call i8* @cat(i8* %cat, i8* %cat32, i8* %tensor47)
  ret i8* %cat48
}

define i8* @loss() {
entry:
  %x_train = load i8*, i8** @x_train
  %w = load i8*, i8** @w
  %newpr = load i8* (i8**)*, i8* (i8**)** @fast_mulMULreduce.1
  %tmpOp = call i8* @pe_calc(i8* (i8*, i8*)** getelementptr inbounds ([4 x i8* (i8*, i8*)*], [4 x i8*
(i8*, i8*)*]* @fast_mulMULmaps, i32 0, i32 0), i32 4, i8* (i8**)* %newpr, i8* %x_train, i8* %w)
  %b = load i8*, i8** @b
  %tmpOp1 = call i8* @add(i8* %tmpOp, i8* %b)
  %y_test = alloca i8*
  store i8* null, i8** %y_test
  %lhsptr = load i8*, i8** %y_test
  call void @increase_rc(i8* %tmpOp1)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %tmpOp1, i8** %y_test
  %y_test2 = load i8*, i8** %y_test
  %y_train = load i8*, i8** @y_train
  %tmpOp3 = call i8* @subtract(i8* %y_test2, i8* %y_train)
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall4 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall4 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall5 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall5 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 2, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %tmpOp6 = call i8* @dotpow(i8* %tmpOp3, i8* %tensor)
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor8 = bitcast i8* %malloccall7 to %tensor_t*
  %dtype9 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 0
  store i8 1, i8* %dtype9
  %ndims10 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 1
  store i8 0, i8* %ndims10
  %malloccall11 = tail call i8* @malloc(i32 0)
  %dims12 = bitcast i8* %malloccall11 to [0 x i64]*
  %dims_as_i8ptr13 = bitcast [0 x i64]* %dims12 to i8*
  %malloccall14 = tail call i8* @malloc(i32 ptrtoint (double* getelementptr (double, double* null,
i32 1) to i32))
  %data15 = bitcast i8* %malloccall14 to [1 x double]*
  %data_as_i8ptr16 = bitcast [1 x double]* %data15 to i8*
  %elmptr17 = getelementptr [1 x double], [1 x double]* %data15, i64 0, i64 0
  store double 5.000000e-01, double* %elmptr17
  %dimsptr18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 2
  store i8* %dims_as_i8ptr13, i8** %dimsptr18
  %dataptr19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 3
  store i8* %data_as_i8ptr16, i8** %dataptr19
  %rc20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 4
  store i8 0, i8* %rc20
  %tensor21 = bitcast %tensor_t* %raw_tensor8 to i8*
```

```
  %tmpOp22 = call i8* @dotmul(i8* %tmpOp6, i8* %tensor21)
  %sum = call i8* @sum(i8* %tmpOp22)
  %error = alloca i8*
  store i8* null, i8** %error
  %lhsptr23 = load i8*, i8** %error
  call void @increase_rc(i8* %sum)
  call void @decrease_rc(i8* %lhsptr23)
  store i8* %sum, i8** %error
  %error24 = load i8*, i8** %error
  %num_of_dimensions = load i8*, i8** @num_of_dimensions
  %float_of = call i8* @float_of(i8* %num_of_dimensions)
  %tmpOp25 = call i8* @divide(i8* %error24, i8* %float_of)
  ret i8* %tmpOp25
}

define i8* @dLdW() {
entry:
  %x_train = load i8*, i8** @x_train
  %w = load i8*, i8** @w
  %newpr = load i8* (i8**)*, i8* (i8**)** @fast_mulMULreduce.1
  %tmpOp = call i8* @pe_calc(i8* (i8*, i8*)** getelementptr inbounds ([4 x i8* (i8*, i8*)*], [4 x i8*
(i8*, i8*)*]* @fast_mulMULmaps, i32 0, i32 0), i32 4, i8* (i8**)* %newpr, i8* %x_train, i8* %w)
  %b = load i8*, i8** @b
  %tmpOp1 = call i8* @add(i8* %tmpOp, i8* %b)
  %y_test = alloca i8*
  store i8* null, i8** %y_test
  %lhsptr = load i8*, i8** %y_test
  call void @increase_rc(i8* %tmpOp1)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %tmpOp1, i8** %y_test
  %y_test2 = load i8*, i8** %y_test
  %y_train = load i8*, i8** @y_train
  %tmpOp3 = call i8* @subtract(i8* %y_test2, i8* %y_train)
  %x_train4 = load i8*, i8** @x_train
  %tmpOp5 = call i8* @mult(i8* %tmpOp3, i8* %x_train4)
  %num_of_dimensions = load i8*, i8** @num_of_dimensions
  %float_of = call i8* @float_of(i8* %num_of_dimensions)
  %tmpOp6 = call i8* @divide(i8* %tmpOp5, i8* %float_of)
  %dLdw = alloca i8*
  store i8* null, i8** %dLdw
  %lhsptr7 = load i8*, i8** %dLdw
  call void @increase_rc(i8* %tmpOp6)
  call void @decrease_rc(i8* %lhsptr7)
  store i8* %tmpOp6, i8** %dLdw
```

```
  %dLdw8 = load i8*, i8** %dLdw
  ret i8* %dLdw8
}

define i8* @dLdB() {
entry:
  %x_train = load i8*, i8** @x_train
  %w = load i8*, i8** @w
  %tmpOp = call i8* @mult(i8* %x_train, i8* %w)
  %b = load i8*, i8** @b
  %tmpOp1 = call i8* @add(i8* %tmpOp, i8* %b)
  %y_test = alloca i8*
  store i8* null, i8** %y_test
  %lhsptr = load i8*, i8** %y_test
  call void @increase_rc(i8* %tmpOp1)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %tmpOp1, i8** %y_test
  %y_test2 = load i8*, i8** %y_test
  %y_train = load i8*, i8** @y_train
  %tmpOp3 = call i8* @subtract(i8* %y_test2, i8* %y_train)
  %sum = call i8* @sum(i8* %tmpOp3)
  %num_of_dimensions = load i8*, i8** @num_of_dimensions
  %float_of = call i8* @float_of(i8* %num_of_dimensions)
  %tmpOp4 = call i8* @divide(i8* %sum, i8* %float_of)
  %dLdb = alloca i8*
  store i8* null, i8** %dLdb
  %lhsptr5 = load i8*, i8** %dLdb
  call void @increase_rc(i8* %tmpOp4)
  call void @decrease_rc(i8* %lhsptr5)
  store i8* %tmpOp4, i8** %dLdb
  %dLdb6 = load i8*, i8** %dLdb
  ret i8* %dLdb6
}
```

6.5. Sample Test Four

In this sample you can see how powerful our parallel functions are.

```
parallel_define Strassen {
    overload __*__ (x, y) {
        map f1 {
            l1 = shape(x);
```

```
    l2 = shape(y);
          x11 = x[0:l1[0]//2:1,0:l1[1]//2:1];
          x22 = x[l1[0]//2:l1[0]:1,l1[1]//2:l1[1]:1];
          y11 = y[0:l2[0]//2:1,0:l2[1]//2:1];
          y22 = y[l2[0]//2:l2[0]:1,l2[1]//2:l2[1]:1];
    return (x11+x22)*(y11+y22);
}
map f2 {
    l1 = shape(x);
    l2 = shape(y);
          x21 = x[l1[0]//2:l1[0]:1,0:l1[1]//2:1];
          x22 = x[l1[0]//2:l1[0]:1,l1[1]//2:l1[1]:1];
          y11 = y[0:l2[0]//2:1,0:l2[1]//2:1];
    return (x21+x22)*y11;
}
map f3 {
    l1 = shape(x);
    l2 = shape(y);
          x11 = x[0:l1[0]//2:1,0:l1[1]//2:1];
          y12 = y[0:l2[0]//2:1,l2[1]//2:l2[1]:1];
          y22 = y[l2[0]//2:l2[0]:1,l2[1]//2:l2[1]:1];
    return x11*(y12-y22);
}
map f4 {
    l1 = shape(x);
    l2 = shape(y);
          x22 = x[l1[0]//2:l1[0]:1,l1[1]//2:l1[1]:1];
          y11 = y[0:l2[0]//2:1,0:l2[1]//2:1];
          y21 = y[l2[0]//2:l2[0]:1,0:l2[1]//2:1];
    return x22*(y21-y11);
}
map f5 {
    l1 = shape(x);
    l2 = shape(y);
          x11 = x[0:l1[0]//2:1,0:l1[1]//2:1];
          y22 = y[l2[0]//2:l2[0]:1,l2[1]//2:l2[1]:1];
          x12 = x[0:l1[0]//2:1,l1[1]//2:l1[1]:1];
    return (x11+x12)*y22;
}
map f6 {
    l1 = shape(x);
    l2 = shape(y);
          x21 = x[l1[0]//2:l1[0]:1,0:l1[1]//2:1];
```

```
                y11 = y[0:l2[0]//2:1,0:l2[1]//2:1];
                x11 = x[0:l1[0]//2:1,0:l1[1]//2:1];
                y12 = y[0:l2[0]//2:1,l2[1]//2:l2[1]:1];
            return (x21-x11)*(y11+y12);
        }
        map f7 {
            l1 = shape(x);
            l2 = shape(y);
                x22 = x[l1[0]//2:l1[0]:1,l1[1]//2:l1[1]:1];
                y22 = y[l2[0]//2:l2[0]:1,l2[1]//2:l2[1]:1];
                y21 = y[l2[0]//2:l2[0]:1,0:l2[1]//2:1];
                x12 = x[0:l1[0]//2:1,l1[1]//2:l1[1]:1];
            return (x12-x22)*(y21+y22);
        }
        reduce {
            c11 = f1+f4-f5+f7;
            c12 = f3+f5;
            c21 = f2+f4;
            c22 = f1-f2+f3+f6;
            return cat(cat(c11,c12,0),cat(c21,c22,0),1);
        }
    }
}


a = zeros([4096, 4096]);
b = ones([4096, 4096]);
using Strassen;
c = a*b;
print("done");
```

```
; ModuleID = 'TENLab'
source_filename = "TENLab"

%tensor_t = type { i8, i8, i8*, i8*, i8 }

@StrassenMULmaps = global [7 x i8* (i8*, i8*)*] zeroinitializer
@StrassenMULreduce.1 = global i8* (i8**)* null
@a = global i8* null
@b = global i8* null
@c = global i8* null
```

```
define i8 @main() {
entry:
  store i8* (i8*, i8*)* @StrassenMULf1, i8* (i8*, i8*)** getelementptr inbounds ([7 x i8* (i8*, i8*)*],
[7 x i8* (i8*, i8*)*]* @StrassenMULmaps, i32 0, i32 0)
  store i8* (i8*, i8*)* @StrassenMULf2, i8* (i8*, i8*)** getelementptr inbounds ([7 x i8* (i8*, i8*)*],
[7 x i8* (i8*, i8*)*]* @StrassenMULmaps, i32 0, i32 1)
  store i8* (i8*, i8*)* @StrassenMULf3, i8* (i8*, i8*)** getelementptr inbounds ([7 x i8* (i8*, i8*)*],
[7 x i8* (i8*, i8*)*]* @StrassenMULmaps, i32 0, i32 2)
  store i8* (i8*, i8*)* @StrassenMULf4, i8* (i8*, i8*)** getelementptr inbounds ([7 x i8* (i8*, i8*)*],
[7 x i8* (i8*, i8*)*]* @StrassenMULmaps, i32 0, i32 3)
  store i8* (i8*, i8*)* @StrassenMULf5, i8* (i8*, i8*)** getelementptr inbounds ([7 x i8* (i8*, i8*)*],
[7 x i8* (i8*, i8*)*]* @StrassenMULmaps, i32 0, i32 4)
  store i8* (i8*, i8*)* @StrassenMULf6, i8* (i8*, i8*)** getelementptr inbounds ([7 x i8* (i8*, i8*)*],
[7 x i8* (i8*, i8*)*]* @StrassenMULmaps, i32 0, i32 5)
  store i8* (i8*, i8*)* @StrassenMULf7, i8* (i8*, i8*)** getelementptr inbounds ([7 x i8* (i8*, i8*)*],
[7 x i8* (i8*, i8*)*]* @StrassenMULmaps, i32 0, i32 6)
  store i8* (i8**)* @StrassenMULreduce, i8* (i8**)** @StrassenMULreduce.1
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 1, i8* %ndims
  %malloccall1 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims = bitcast i8* %malloccall1 to [1 x i64]*
  %dims_as_i8ptr = bitcast [1 x i64]* %dims to i8*
  %elmptr = getelementptr [1 x i64], [1 x i64]* %dims, i64 0, i64 0
  store i64 2, i64* %elmptr
  %malloccall2 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i32* getelementptr
(i32, i32* null, i32 1) to i64), i64 2) to i32))
  %data = bitcast i8* %malloccall2 to [2 x i32]*
  %data_as_i8ptr = bitcast [2 x i32]* %data to i8*
  %elmptr3 = getelementptr [2 x i32], [2 x i32]* %data, i32 0, i32 0
  store i32 4096, i32* %elmptr3
  %elmptr4 = getelementptr [2 x i32], [2 x i32]* %data, i32 0, i32 1
  store i32 4096, i32* %elmptr4
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
```

```
  %zeros = call i8* @zeros(i8* %tensor)
  %lhsptr = load i8*, i8** @a
  call void @increase_rc(i8* %zeros)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %zeros, i8** @a
  %malloccall5 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor6 = bitcast i8* %malloccall5 to %tensor_t*
  %dtype7 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor6, i32 0, i32 0
  store i8 0, i8* %dtype7
  %ndims8 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor6, i32 0, i32 1
  store i8 1, i8* %ndims8
  %malloccall9 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to i32))
  %dims10 = bitcast i8* %malloccall9 to [1 x i64]*
  %dims_as_i8ptr11 = bitcast [1 x i64]* %dims10 to i8*
  %elmptr12 = getelementptr [1 x i64], [1 x i64]* %dims10, i64 0, i64 0
  store i64 2, i64* %elmptr12
  %malloccall13 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i32* getelementptr
(i32, i32* null, i32 1) to i64), i64 2) to i32))
  %data14 = bitcast i8* %malloccall13 to [2 x i32]*
  %data_as_i8ptr15 = bitcast [2 x i32]* %data14 to i8*
  %elmptr16 = getelementptr [2 x i32], [2 x i32]* %data14, i32 0, i32 0
  store i32 4096, i32* %elmptr16
  %elmptr17 = getelementptr [2 x i32], [2 x i32]* %data14, i32 0, i32 1
  store i32 4096, i32* %elmptr17
  %dimsptr18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor6, i32 0, i32 2
  store i8* %dims_as_i8ptr11, i8** %dimsptr18
  %dataptr19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor6, i32 0, i32 3
  store i8* %data_as_i8ptr15, i8** %dataptr19
  %rc20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor6, i32 0, i32 4
  store i8 0, i8* %rc20
  %tensor21 = bitcast %tensor_t* %raw_tensor6 to i8*
  %ones = call i8* @ones(i8* %tensor21)
  %lhsptr22 = load i8*, i8** @b
  call void @increase_rc(i8* %ones)
  call void @decrease_rc(i8* %lhsptr22)
  store i8* %ones, i8** @b
  %a = load i8*, i8** @a
  %b = load i8*, i8** @b
  %newpr = load i8* (i8**)*, i8* (i8**)** @StrassenMULreduce.1
  %tmpOp = call i8* @pe_calc(i8* (i8*, i8*)** getelementptr inbounds ([7 x i8* (i8*, i8*)*], [7 x i8*
(i8*, i8*)*]* @StrassenMULmaps, i32 0, i32 0), i32 7, i8* (i8**)* %newpr, i8* %a, i8* %b)
  %lhsptr23 = load i8*, i8** @c
  call void @increase_rc(i8* %tmpOp)
```

```
  call void @decrease_rc(i8* %lhsptr23)
  store i8* %tmpOp, i8** @c
  %malloccall24 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor25 = bitcast i8* %malloccall24 to %tensor_t*
  %dtype26 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 0
  store i8 2, i8* %dtype26
  %ndims27 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 1
  store i8 1, i8* %ndims27
  %malloccall28 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims29 = bitcast i8* %malloccall28 to [1 x i64]*
  %dims_as_i8ptr30 = bitcast [1 x i64]* %dims29 to i8*
  %elmptr31 = getelementptr [1 x i64], [1 x i64]* %dims29, i64 0, i64 0
  store i64 4, i64* %elmptr31
  %malloccall32 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i8* getelementptr (i8,
i8* null, i32 1) to i64), i64 5) to i32))
  %data33 = bitcast i8* %malloccall32 to [5 x i8]*
  %data_as_i8ptr34 = bitcast [5 x i8]* %data33 to i8*
  %elmptr35 = getelementptr [5 x i8], [5 x i8]* %data33, i8 0, i8 0
  store i8 100, i8* %elmptr35
  %elmptr36 = getelementptr [5 x i8], [5 x i8]* %data33, i8 0, i8 1
  store i8 111, i8* %elmptr36
  %elmptr37 = getelementptr [5 x i8], [5 x i8]* %data33, i8 0, i8 2
  store i8 110, i8* %elmptr37
  %elmptr38 = getelementptr [5 x i8], [5 x i8]* %data33, i8 0, i8 3
  store i8 101, i8* %elmptr38
  %elmptr39 = getelementptr [5 x i8], [5 x i8]* %data33, i8 0, i8 4
  store i8 0, i8* %elmptr39
  %dimsptr40 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 2
  store i8* %dims_as_i8ptr30, i8** %dimsptr40
  %dataptr41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 3
  store i8* %data_as_i8ptr34, i8** %dataptr41
  %rc42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 4
  store i8 0, i8* %rc42
  %tensor43 = bitcast %tensor_t* %raw_tensor25 to i8*
  call void @print(i8* %tensor43)
  ret i8 0
}

declare i8* @add(i8*, i8*)

declare i8* @subtract(i8*, i8*)
```

```
declare i8* @negative(i8*)

declare i8* @mult(i8*, i8*)

declare i8* @dotmul(i8*, i8*)

declare i8* @divide(i8*, i8*)

declare i8* @floordivide(i8*, i8*)

declare i8* @matpow(i8*, i8*)

declare i8* @dotpow(i8*, i8*)

declare i8* @mod(i8*, i8*)

declare i8* @transpose(i8*)

declare i8* @equal(i8*, i8*)

declare i8* @notequal(i8*, i8*)

declare i8* @greater(i8*, i8*)

declare i8* @greaterequal(i8*, i8*)

declare i8* @less(i8*, i8*)

declare i8* @lessequal(i8*, i8*)

declare i8* @range(i8*, i8*, i8*)

declare void @print(i8*)

declare i8* @logicaland(i8*, i8*)

declare i8* @logicalor(i8*, i8*)

declare i8* @logicalnot(i8*)

declare i32 @len(i8*)

declare i8* @zeros(i8*)
```

```
declare i8* @cat(i8*, i8*, i8*)

declare i8* @shape(i8*)

declare i8* @ones(i8*)

declare i8* @tensor_rand(i8*)

declare i8* @sum(i8*)

declare i8* @any(i8*)

declare i8* @all(i8*)

declare i8* @tensor_abs(i8*)

declare i8* @tensor_log(i8*)

declare i8* @tensor_floor(i8*)

declare i8* @tensor_ceil(i8*)

declare i8* @tensor_round(i8*)

declare i8* @int_of(i8*)

declare i8* @float_of(i8*)

declare i8* @inverse(i8*)

declare i1 @bool_of_zero(i8*)

declare i8* @pe_calc(i8* (i8*, i8*)**, i32, i8* (i8**)*, i8*, i8*)

declare i8* @index_get(i8*, i8*)

declare i8* @index_get_int(i8*, i32)

declare void @index_put(i8*, i8*, i8*)

declare void @increase_rc(i8*)

declare void @decrease_rc(i8*)
```

```
define i8* @StrassenMULf1(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
  %l1 = alloca i8*
  store i8* null, i8** %l1
  %lhsptr = load i8*, i8** %l1
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l1
  %y2 = load i8*, i8** %y
  %shape3 = call i8* @shape(i8* %y2)
  %l2 = alloca i8*
  store i8* null, i8** %l2
  %lhsptr4 = load i8*, i8** %l2
  call void @increase_rc(i8* %shape3)
  call void @decrease_rc(i8* %lhsptr4)
  store i8* %shape3, i8** %l2
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall5 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall5 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall6 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall6 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
```

```
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor8 = bitcast i8* %malloccall7 to %tensor_t*
  %dtype9 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 0
  store i8 0, i8* %dtype9
  %ndims10 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 1
  store i8 0, i8* %ndims10
  %malloccall11 = tail call i8* @malloc(i32 0)
  %dims12 = bitcast i8* %malloccall11 to [0 x i64]*
  %dims_as_i8ptr13 = bitcast [0 x i64]* %dims12 to i8*
  %malloccall14 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data15 = bitcast i8* %malloccall14 to [1 x i32]*
  %data_as_i8ptr16 = bitcast [1 x i32]* %data15 to i8*
  %elmptr17 = getelementptr [1 x i32], [1 x i32]* %data15, i32 0, i32 0
  store i32 0, i32* %elmptr17
  %dimsptr18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 2
  store i8* %dims_as_i8ptr13, i8** %dimsptr18
  %dataptr19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 3
  store i8* %data_as_i8ptr16, i8** %dataptr19
  %rc20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 4
  store i8 0, i8* %rc20
  %tensor21 = bitcast %tensor_t* %raw_tensor8 to i8*
  %malloccall22 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor23 = bitcast i8* %malloccall22 to %tensor_t*
  %dtype24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 0
  store i8 3, i8* %dtype24
  %ndims25 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 1
  store i8 1, i8* %ndims25
  %malloccall26 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims27 = bitcast i8* %malloccall26 to [1 x i64]*
  %dims_as_i8ptr28 = bitcast [1 x i64]* %dims27 to i8*
  %elmptr29 = getelementptr [1 x i64], [1 x i64]* %dims27, i64 0, i64 0
  store i64 1, i64* %elmptr29
  %malloccall30 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data31 = bitcast i8* %malloccall30 to [1 x i8*]*
  %data_as_i8ptr32 = bitcast [1 x i8*]* %data31 to i8*
  %elmptr33 = getelementptr [1 x i8*], [1 x i8*]* %data31, i64 0, i64 0
  store i8* %tensor21, i8** %elmptr33
  %dimsptr34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 2
  store i8* %dims_as_i8ptr28, i8** %dimsptr34
  %dataptr35 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 3
```

```
  store i8* %data_as_i8ptr32, i8** %dataptr35
  %rc36 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 4
  store i8 0, i8* %rc36
  %tensor37 = bitcast %tensor_t* %raw_tensor23 to i8*
  %l138 = load i8*, i8** %l1
  %access_tensor = call i8* @index_get(i8* %l138, i8* %tensor37)
  %malloccall39 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor40 = bitcast i8* %malloccall39 to %tensor_t*
  %dtype41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 0
  store i8 0, i8* %dtype41
  %ndims42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 1
  store i8 0, i8* %ndims42
  %malloccall43 = tail call i8* @malloc(i32 0)
  %dims44 = bitcast i8* %malloccall43 to [0 x i64]*
  %dims_as_i8ptr45 = bitcast [0 x i64]* %dims44 to i8*
  %malloccall46 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data47 = bitcast i8* %malloccall46 to [1 x i32]*
  %data_as_i8ptr48 = bitcast [1 x i32]* %data47 to i8*
  %elmptr49 = getelementptr [1 x i32], [1 x i32]* %data47, i32 0, i32 0
  store i32 2, i32* %elmptr49
  %dimsptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 2
  store i8* %dims_as_i8ptr45, i8** %dimsptr50
  %dataptr51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 3
  store i8* %data_as_i8ptr48, i8** %dataptr51
  %rc52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 4
  store i8 0, i8* %rc52
  %tensor53 = bitcast %tensor_t* %raw_tensor40 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor53)
  %malloccall54 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor55 = bitcast i8* %malloccall54 to %tensor_t*
  %dtype56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 0
  store i8 0, i8* %dtype56
  %ndims57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 1
  store i8 0, i8* %ndims57
  %malloccall58 = tail call i8* @malloc(i32 0)
  %dims59 = bitcast i8* %malloccall58 to [0 x i64]*
  %dims_as_i8ptr60 = bitcast [0 x i64]* %dims59 to i8*
  %malloccall61 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data62 = bitcast i8* %malloccall61 to [1 x i32]*
  %data_as_i8ptr63 = bitcast [1 x i32]* %data62 to i8*
```

```
  %elmptr64 = getelementptr [1 x i32], [1 x i32]* %data62, i32 0, i32 0
  store i32 1, i32* %elmptr64
  %dimsptr65 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 2
  store i8* %dims_as_i8ptr60, i8** %dimsptr65
  %dataptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 3
  store i8* %data_as_i8ptr63, i8** %dataptr66
  %rc67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 4
  store i8 0, i8* %rc67
  %tensor68 = bitcast %tensor_t* %raw_tensor55 to i8*
  %tmpOp69 = call i8* @range(i8* %tensor, i8* %tmpOp, i8* %tensor68)
  %malloccall70 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor71 = bitcast i8* %malloccall70 to %tensor_t*
  %dtype72 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 0
  store i8 0, i8* %dtype72
  %ndims73 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 1
  store i8 0, i8* %ndims73
  %malloccall74 = tail call i8* @malloc(i32 0)
  %dims75 = bitcast i8* %malloccall74 to [0 x i64]*
  %dims_as_i8ptr76 = bitcast [0 x i64]* %dims75 to i8*
  %malloccall77 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data78 = bitcast i8* %malloccall77 to [1 x i32]*
  %data_as_i8ptr79 = bitcast [1 x i32]* %data78 to i8*
  %elmptr80 = getelementptr [1 x i32], [1 x i32]* %data78, i32 0, i32 0
  store i32 0, i32* %elmptr80
  %dimsptr81 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 2
  store i8* %dims_as_i8ptr76, i8** %dimsptr81
  %dataptr82 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 3
  store i8* %data_as_i8ptr79, i8** %dataptr82
  %rc83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 4
  store i8 0, i8* %rc83
  %tensor84 = bitcast %tensor_t* %raw_tensor71 to i8*
  %malloccall85 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor86 = bitcast i8* %malloccall85 to %tensor_t*
  %dtype87 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 0
  store i8 0, i8* %dtype87
  %ndims88 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 1
  store i8 0, i8* %ndims88
  %malloccall89 = tail call i8* @malloc(i32 0)
  %dims90 = bitcast i8* %malloccall89 to [0 x i64]*
  %dims_as_i8ptr91 = bitcast [0 x i64]* %dims90 to i8*
```

```
  %malloccall92 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data93 = bitcast i8* %malloccall92 to [1 x i32]*
  %data_as_i8ptr94 = bitcast [1 x i32]* %data93 to i8*
  %elmptr95 = getelementptr [1 x i32], [1 x i32]* %data93, i32 0, i32 0
  store i32 1, i32* %elmptr95
  %dimsptr96 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 2
  store i8* %dims_as_i8ptr91, i8** %dimsptr96
  %dataptr97 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 3
  store i8* %data_as_i8ptr94, i8** %dataptr97
  %rc98 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 4
  store i8 0, i8* %rc98
  %tensor99 = bitcast %tensor_t* %raw_tensor86 to i8*
  %malloccall100 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor101 = bitcast i8* %malloccall100 to %tensor_t*
  %dtype102 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 0
  store i8 3, i8* %dtype102
  %ndims103 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 1
  store i8 1, i8* %ndims103
  %malloccall104 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims105 = bitcast i8* %malloccall104 to [1 x i64]*
  %dims_as_i8ptr106 = bitcast [1 x i64]* %dims105 to i8*
  %elmptr107 = getelementptr [1 x i64], [1 x i64]* %dims105, i64 0, i64 0
  store i64 1, i64* %elmptr107
  %malloccall108 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data109 = bitcast i8* %malloccall108 to [1 x i8*]*
  %data_as_i8ptr110 = bitcast [1 x i8*]* %data109 to i8*
  %elmptr111 = getelementptr [1 x i8*], [1 x i8*]* %data109, i64 0, i64 0
  store i8* %tensor99, i8** %elmptr111
  %dimsptr112 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 2
  store i8* %dims_as_i8ptr106, i8** %dimsptr112
  %dataptr113 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 3
  store i8* %data_as_i8ptr110, i8** %dataptr113
  %rc114 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 4
  store i8 0, i8* %rc114
  %tensor115 = bitcast %tensor_t* %raw_tensor101 to i8*
  %l1116 = load i8*, i8** %l1
  %access_tensor117 = call i8* @index_get(i8* %l1116, i8* %tensor115)
  %malloccall118 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor119 = bitcast i8* %malloccall118 to %tensor_t*
```

```
  %dtype120 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 0
  store i8 0, i8* %dtype120
  %ndims121 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 1
  store i8 0, i8* %ndims121
  %malloccall122 = tail call i8* @malloc(i32 0)
  %dims123 = bitcast i8* %malloccall122 to [0 x i64]*
  %dims_as_i8ptr124 = bitcast [0 x i64]* %dims123 to i8*
  %malloccall125 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data126 = bitcast i8* %malloccall125 to [1 x i32]*
  %data_as_i8ptr127 = bitcast [1 x i32]* %data126 to i8*
  %elmptr128 = getelementptr [1 x i32], [1 x i32]* %data126, i32 0, i32 0
  store i32 2, i32* %elmptr128
  %dimsptr129 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 2
  store i8* %dims_as_i8ptr124, i8** %dimsptr129
  %dataptr130 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 3
  store i8* %data_as_i8ptr127, i8** %dataptr130
  %rc131 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 4
  store i8 0, i8* %rc131
  %tensor132 = bitcast %tensor_t* %raw_tensor119 to i8*
  %tmpOp133 = call i8* @floordivide(i8* %access_tensor117, i8* %tensor132)
  %malloccall134 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor135 = bitcast i8* %malloccall134 to %tensor_t*
  %dtype136 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 0
  store i8 0, i8* %dtype136
  %ndims137 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 1
  store i8 0, i8* %ndims137
  %malloccall138 = tail call i8* @malloc(i32 0)
  %dims139 = bitcast i8* %malloccall138 to [0 x i64]*
  %dims_as_i8ptr140 = bitcast [0 x i64]* %dims139 to i8*
  %malloccall141 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data142 = bitcast i8* %malloccall141 to [1 x i32]*
  %data_as_i8ptr143 = bitcast [1 x i32]* %data142 to i8*
  %elmptr144 = getelementptr [1 x i32], [1 x i32]* %data142, i32 0, i32 0
  store i32 1, i32* %elmptr144
  %dimsptr145 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 2
  store i8* %dims_as_i8ptr140, i8** %dimsptr145
  %dataptr146 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 3
  store i8* %data_as_i8ptr143, i8** %dataptr146
  %rc147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 4
  store i8 0, i8* %rc147
  %tensor148 = bitcast %tensor_t* %raw_tensor135 to i8*
```

```
  %tmpOp149 = call i8* @range(i8* %tensor84, i8* %tmpOp133, i8* %tensor148)
  %malloccall150 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor151 = bitcast i8* %malloccall150 to %tensor_t*
  %dtype152 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 0
  store i8 3, i8* %dtype152
  %ndims153 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 1
  store i8 1, i8* %ndims153
  %malloccall154 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims155 = bitcast i8* %malloccall154 to [1 x i64]*
  %dims_as_i8ptr156 = bitcast [1 x i64]* %dims155 to i8*
  %elmptr157 = getelementptr [1 x i64], [1 x i64]* %dims155, i64 0, i64 0
  store i64 2, i64* %elmptr157
  %malloccall158 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data159 = bitcast i8* %malloccall158 to [2 x i8*]*
  %data_as_i8ptr160 = bitcast [2 x i8*]* %data159 to i8*
  %elmptr161 = getelementptr [2 x i8*], [2 x i8*]* %data159, i64 0, i64 0
  store i8* %tmpOp69, i8** %elmptr161
  %elmptr162 = getelementptr [2 x i8*], [2 x i8*]* %data159, i64 0, i64 1
  store i8* %tmpOp149, i8** %elmptr162
  %dimsptr163 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 2
  store i8* %dims_as_i8ptr156, i8** %dimsptr163
  %dataptr164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 3
  store i8* %data_as_i8ptr160, i8** %dataptr164
  %rc165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 4
  store i8 0, i8* %rc165
  %tensor166 = bitcast %tensor_t* %raw_tensor151 to i8*
  %x167 = load i8*, i8** %x
  %access_tensor168 = call i8* @index_get(i8* %x167, i8* %tensor166)
  %x11 = alloca i8*
  store i8* null, i8** %x11
  %lhsptr169 = load i8*, i8** %x11
  call void @increase_rc(i8* %access_tensor168)
  call void @decrease_rc(i8* %lhsptr169)
  store i8* %access_tensor168, i8** %x11
  %malloccall170 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor171 = bitcast i8* %malloccall170 to %tensor_t*
  %dtype172 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 0
  store i8 0, i8* %dtype172
  %ndims173 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 1
  store i8 0, i8* %ndims173
```

```
  %malloccall174 = tail call i8* @malloc(i32 0)
  %dims175 = bitcast i8* %malloccall174 to [0 x i64]*
  %dims_as_i8ptr176 = bitcast [0 x i64]* %dims175 to i8*
  %malloccall177 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data178 = bitcast i8* %malloccall177 to [1 x i32]*
  %data_as_i8ptr179 = bitcast [1 x i32]* %data178 to i8*
  %elmptr180 = getelementptr [1 x i32], [1 x i32]* %data178, i32 0, i32 0
  store i32 0, i32* %elmptr180
  %dimsptr181 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 2
  store i8* %dims_as_i8ptr176, i8** %dimsptr181
  %dataptr182 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 3
  store i8* %data_as_i8ptr179, i8** %dataptr182
  %rc183 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 4
  store i8 0, i8* %rc183
  %tensor184 = bitcast %tensor_t* %raw_tensor171 to i8*
  %malloccall185 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor186 = bitcast i8* %malloccall185 to %tensor_t*
  %dtype187 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 0
  store i8 3, i8* %dtype187
  %ndims188 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 1
  store i8 1, i8* %ndims188
  %malloccall189 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims190 = bitcast i8* %malloccall189 to [1 x i64]*
  %dims_as_i8ptr191 = bitcast [1 x i64]* %dims190 to i8*
  %elmptr192 = getelementptr [1 x i64], [1 x i64]* %dims190, i64 0, i64 0
  store i64 1, i64* %elmptr192
  %malloccall193 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data194 = bitcast i8* %malloccall193 to [1 x i8*]*
  %data_as_i8ptr195 = bitcast [1 x i8*]* %data194 to i8*
  %elmptr196 = getelementptr [1 x i8*], [1 x i8*]* %data194, i64 0, i64 0
  store i8* %tensor184, i8** %elmptr196
  %dimsptr197 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 2
  store i8* %dims_as_i8ptr191, i8** %dimsptr197
  %dataptr198 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 3
  store i8* %data_as_i8ptr195, i8** %dataptr198
  %rc199 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 4
  store i8 0, i8* %rc199
  %tensor200 = bitcast %tensor_t* %raw_tensor186 to i8*
  %l1201 = load i8*, i8** %l1
  %access_tensor202 = call i8* @index_get(i8* %l1201, i8* %tensor200)
```

```
  %malloccall203 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor204 = bitcast i8* %malloccall203 to %tensor_t*
  %dtype205 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 0
  store i8 0, i8* %dtype205
  %ndims206 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 1
  store i8 0, i8* %ndims206
  %malloccall207 = tail call i8* @malloc(i32 0)
  %dims208 = bitcast i8* %malloccall207 to [0 x i64]*
  %dims_as_i8ptr209 = bitcast [0 x i64]* %dims208 to i8*
  %malloccall210 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data211 = bitcast i8* %malloccall210 to [1 x i32]*
  %data_as_i8ptr212 = bitcast [1 x i32]* %data211 to i8*
  %elmptr213 = getelementptr [1 x i32], [1 x i32]* %data211, i32 0, i32 0
  store i32 2, i32* %elmptr213
  %dimsptr214 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 2
  store i8* %dims_as_i8ptr209, i8** %dimsptr214
  %dataptr215 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 3
  store i8* %data_as_i8ptr212, i8** %dataptr215
  %rc216 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 4
  store i8 0, i8* %rc216
  %tensor217 = bitcast %tensor_t* %raw_tensor204 to i8*
  %tmpOp218 = call i8* @floordivide(i8* %access_tensor202, i8* %tensor217)
  %malloccall219 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor220 = bitcast i8* %malloccall219 to %tensor_t*
  %dtype221 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 0
  store i8 0, i8* %dtype221
  %ndims222 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 1
  store i8 0, i8* %ndims222
  %malloccall223 = tail call i8* @malloc(i32 0)
  %dims224 = bitcast i8* %malloccall223 to [0 x i64]*
  %dims_as_i8ptr225 = bitcast [0 x i64]* %dims224 to i8*
  %malloccall226 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data227 = bitcast i8* %malloccall226 to [1 x i32]*
  %data_as_i8ptr228 = bitcast [1 x i32]* %data227 to i8*
  %elmptr229 = getelementptr [1 x i32], [1 x i32]* %data227, i32 0, i32 0
  store i32 0, i32* %elmptr229
  %dimsptr230 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 2
  store i8* %dims_as_i8ptr225, i8** %dimsptr230
  %dataptr231 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 3
  store i8* %data_as_i8ptr228, i8** %dataptr231
```

```
  %rc232 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 4
  store i8 0, i8* %rc232
  %tensor233 = bitcast %tensor_t* %raw_tensor220 to i8*
  %malloccall234 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor235 = bitcast i8* %malloccall234 to %tensor_t*
  %dtype236 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 0
  store i8 3, i8* %dtype236
  %ndims237 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 1
  store i8 1, i8* %ndims237
  %malloccall238 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims239 = bitcast i8* %malloccall238 to [1 x i64]*
  %dims_as_i8ptr240 = bitcast [1 x i64]* %dims239 to i8*
  %elmptr241 = getelementptr [1 x i64], [1 x i64]* %dims239, i64 0, i64 0
  store i64 1, i64* %elmptr241
  %malloccall242 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data243 = bitcast i8* %malloccall242 to [1 x i8*]*
  %data_as_i8ptr244 = bitcast [1 x i8*]* %data243 to i8*
  %elmptr245 = getelementptr [1 x i8*], [1 x i8*]* %data243, i64 0, i64 0
  store i8* %tensor233, i8** %elmptr245
  %dimsptr246 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 2
  store i8* %dims_as_i8ptr240, i8** %dimsptr246
  %dataptr247 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 3
  store i8* %data_as_i8ptr244, i8** %dataptr247
  %rc248 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 4
  store i8 0, i8* %rc248
  %tensor249 = bitcast %tensor_t* %raw_tensor235 to i8*
  %l1250 = load i8*, i8** %l1
  %access_tensor251 = call i8* @index_get(i8* %l1250, i8* %tensor249)
  %malloccall252 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor253 = bitcast i8* %malloccall252 to %tensor_t*
  %dtype254 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 0
  store i8 0, i8* %dtype254
  %ndims255 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 1
  store i8 0, i8* %ndims255
  %malloccall256 = tail call i8* @malloc(i32 0)
  %dims257 = bitcast i8* %malloccall256 to [0 x i64]*
  %dims_as_i8ptr258 = bitcast [0 x i64]* %dims257 to i8*
  %malloccall259 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data260 = bitcast i8* %malloccall259 to [1 x i32]*
```

```
  %data_as_i8ptr261 = bitcast [1 x i32]* %data260 to i8*
  %elmptr262 = getelementptr [1 x i32], [1 x i32]* %data260, i32 0, i32 0
  store i32 1, i32* %elmptr262
  %dimsptr263 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 2
  store i8* %dims_as_i8ptr258, i8** %dimsptr263
  %dataptr264 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 3
  store i8* %data_as_i8ptr261, i8** %dataptr264
  %rc265 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 4
  store i8 0, i8* %rc265
  %tensor266 = bitcast %tensor_t* %raw_tensor253 to i8*
  %tmpOp267 = call i8* @range(i8* %tmpOp218, i8* %access_tensor251, i8* %tensor266)
  %malloccall268 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor269 = bitcast i8* %malloccall268 to %tensor_t*
  %dtype270 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 0
  store i8 0, i8* %dtype270
  %ndims271 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 1
  store i8 0, i8* %ndims271
  %malloccall272 = tail call i8* @malloc(i32 0)
  %dims273 = bitcast i8* %malloccall272 to [0 x i64]*
  %dims_as_i8ptr274 = bitcast [0 x i64]* %dims273 to i8*
  %malloccall275 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data276 = bitcast i8* %malloccall275 to [1 x i32]*
  %data_as_i8ptr277 = bitcast [1 x i32]* %data276 to i8*
  %elmptr278 = getelementptr [1 x i32], [1 x i32]* %data276, i32 0, i32 0
  store i32 1, i32* %elmptr278
  %dimsptr279 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 2
  store i8* %dims_as_i8ptr274, i8** %dimsptr279
  %dataptr280 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 3
  store i8* %data_as_i8ptr277, i8** %dataptr280
  %rc281 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 4
  store i8 0, i8* %rc281
  %tensor282 = bitcast %tensor_t* %raw_tensor269 to i8*
  %malloccall283 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor284 = bitcast i8* %malloccall283 to %tensor_t*
  %dtype285 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 0
  store i8 3, i8* %dtype285
  %ndims286 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 1
  store i8 1, i8* %ndims286
  %malloccall287 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims288 = bitcast i8* %malloccall287 to [1 x i64]*
```

```
  %dims_as_i8ptr289 = bitcast [1 x i64]* %dims288 to i8*
  %elmptr290 = getelementptr [1 x i64], [1 x i64]* %dims288, i64 0, i64 0
  store i64 1, i64* %elmptr290
  %malloccall291 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data292 = bitcast i8* %malloccall291 to [1 x i8*]*
  %data_as_i8ptr293 = bitcast [1 x i8*]* %data292 to i8*
  %elmptr294 = getelementptr [1 x i8*], [1 x i8*]* %data292, i64 0, i64 0
  store i8* %tensor282, i8** %elmptr294
  %dimsptr295 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 2
  store i8* %dims_as_i8ptr289, i8** %dimsptr295
  %dataptr296 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 3
  store i8* %data_as_i8ptr293, i8** %dataptr296
  %rc297 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 4
  store i8 0, i8* %rc297
  %tensor298 = bitcast %tensor_t* %raw_tensor284 to i8*
  %l1299 = load i8*, i8** %l1
  %access_tensor300 = call i8* @index_get(i8* %l1299, i8* %tensor298)
  %malloccall301 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor302 = bitcast i8* %malloccall301 to %tensor_t*
  %dtype303 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 0
  store i8 0, i8* %dtype303
  %ndims304 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 1
  store i8 0, i8* %ndims304
  %malloccall305 = tail call i8* @malloc(i32 0)
  %dims306 = bitcast i8* %malloccall305 to [0 x i64]*
  %dims_as_i8ptr307 = bitcast [0 x i64]* %dims306 to i8*
  %malloccall308 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data309 = bitcast i8* %malloccall308 to [1 x i32]*
  %data_as_i8ptr310 = bitcast [1 x i32]* %data309 to i8*
  %elmptr311 = getelementptr [1 x i32], [1 x i32]* %data309, i32 0, i32 0
  store i32 2, i32* %elmptr311
  %dimsptr312 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 2
  store i8* %dims_as_i8ptr307, i8** %dimsptr312
  %dataptr313 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 3
  store i8* %data_as_i8ptr310, i8** %dataptr313
  %rc314 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 4
  store i8 0, i8* %rc314
  %tensor315 = bitcast %tensor_t* %raw_tensor302 to i8*
  %tmpOp316 = call i8* @floordivide(i8* %access_tensor300, i8* %tensor315)
  %malloccall317 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor318 = bitcast i8* %malloccall317 to %tensor_t*
  %dtype319 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 0
  store i8 0, i8* %dtype319
  %ndims320 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 1
  store i8 0, i8* %ndims320
  %malloccall321 = tail call i8* @malloc(i32 0)
  %dims322 = bitcast i8* %malloccall321 to [0 x i64]*
  %dims_as_i8ptr323 = bitcast [0 x i64]* %dims322 to i8*
  %malloccall324 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data325 = bitcast i8* %malloccall324 to [1 x i32]*
  %data_as_i8ptr326 = bitcast [1 x i32]* %data325 to i8*
  %elmptr327 = getelementptr [1 x i32], [1 x i32]* %data325, i32 0, i32 0
  store i32 1, i32* %elmptr327
  %dimsptr328 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 2
  store i8* %dims_as_i8ptr323, i8** %dimsptr328
  %dataptr329 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 3
  store i8* %data_as_i8ptr326, i8** %dataptr329
  %rc330 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 4
  store i8 0, i8* %rc330
  %tensor331 = bitcast %tensor_t* %raw_tensor318 to i8*
  %malloccall332 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor333 = bitcast i8* %malloccall332 to %tensor_t*
  %dtype334 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 0
  store i8 3, i8* %dtype334
  %ndims335 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 1
  store i8 1, i8* %ndims335
  %malloccall336 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims337 = bitcast i8* %malloccall336 to [1 x i64]*
  %dims_as_i8ptr338 = bitcast [1 x i64]* %dims337 to i8*
  %elmptr339 = getelementptr [1 x i64], [1 x i64]* %dims337, i64 0, i64 0
  store i64 1, i64* %elmptr339
  %malloccall340 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data341 = bitcast i8* %malloccall340 to [1 x i8*]*
  %data_as_i8ptr342 = bitcast [1 x i8*]* %data341 to i8*
  %elmptr343 = getelementptr [1 x i8*], [1 x i8*]* %data341, i64 0, i64 0
  store i8* %tensor331, i8** %elmptr343
  %dimsptr344 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 2
  store i8* %dims_as_i8ptr338, i8** %dimsptr344
  %dataptr345 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 3
  store i8* %data_as_i8ptr342, i8** %dataptr345
```

```
  %rc346 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 4
  store i8 0, i8* %rc346
  %tensor347 = bitcast %tensor_t* %raw_tensor333 to i8*
  %l1348 = load i8*, i8** %l1
  %access_tensor349 = call i8* @index_get(i8* %l1348, i8* %tensor347)
  %malloccall350 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor351 = bitcast i8* %malloccall350 to %tensor_t*
  %dtype352 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 0
  store i8 0, i8* %dtype352
  %ndims353 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 1
  store i8 0, i8* %ndims353
  %malloccall354 = tail call i8* @malloc(i32 0)
  %dims355 = bitcast i8* %malloccall354 to [0 x i64]*
  %dims_as_i8ptr356 = bitcast [0 x i64]* %dims355 to i8*
  %malloccall357 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data358 = bitcast i8* %malloccall357 to [1 x i32]*
  %data_as_i8ptr359 = bitcast [1 x i32]* %data358 to i8*
  %elmptr360 = getelementptr [1 x i32], [1 x i32]* %data358, i32 0, i32 0
  store i32 1, i32* %elmptr360
  %dimsptr361 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 2
  store i8* %dims_as_i8ptr356, i8** %dimsptr361
  %dataptr362 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 3
  store i8* %data_as_i8ptr359, i8** %dataptr362
  %rc363 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 4
  store i8 0, i8* %rc363
  %tensor364 = bitcast %tensor_t* %raw_tensor351 to i8*
  %tmpOp365 = call i8* @range(i8* %tmpOp316, i8* %access_tensor349, i8* %tensor364)
  %malloccall366 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor367 = bitcast i8* %malloccall366 to %tensor_t*
  %dtype368 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 0
  store i8 3, i8* %dtype368
  %ndims369 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 1
  store i8 1, i8* %ndims369
  %malloccall370 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims371 = bitcast i8* %malloccall370 to [1 x i64]*
  %dims_as_i8ptr372 = bitcast [1 x i64]* %dims371 to i8*
  %elmptr373 = getelementptr [1 x i64], [1 x i64]* %dims371, i64 0, i64 0
  store i64 2, i64* %elmptr373
  %malloccall374 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
```

```
  %data375 = bitcast i8* %malloccall374 to [2 x i8*]*
  %data_as_i8ptr376 = bitcast [2 x i8*]* %data375 to i8*
  %elmptr377 = getelementptr [2 x i8*], [2 x i8*]* %data375, i64 0, i64 0
  store i8* %tmpOp267, i8** %elmptr377
  %elmptr378 = getelementptr [2 x i8*], [2 x i8*]* %data375, i64 0, i64 1
  store i8* %tmpOp365, i8** %elmptr378
  %dimsptr379 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 2
  store i8* %dims_as_i8ptr372, i8** %dimsptr379
  %dataptr380 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 3
  store i8* %data_as_i8ptr376, i8** %dataptr380
  %rc381 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 4
  store i8 0, i8* %rc381
  %tensor382 = bitcast %tensor_t* %raw_tensor367 to i8*
  %x383 = load i8*, i8** %x
  %access_tensor384 = call i8* @index_get(i8* %x383, i8* %tensor382)
  %x22 = alloca i8*
  store i8* null, i8** %x22
  %lhsptr385 = load i8*, i8** %x22
  call void @increase_rc(i8* %access_tensor384)
  call void @decrease_rc(i8* %lhsptr385)
  store i8* %access_tensor384, i8** %x22
  %malloccall386 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor387 = bitcast i8* %malloccall386 to %tensor_t*
  %dtype388 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 0
  store i8 0, i8* %dtype388
  %ndims389 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 1
  store i8 0, i8* %ndims389
  %malloccall390 = tail call i8* @malloc(i32 0)
  %dims391 = bitcast i8* %malloccall390 to [0 x i64]*
  %dims_as_i8ptr392 = bitcast [0 x i64]* %dims391 to i8*
  %malloccall393 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data394 = bitcast i8* %malloccall393 to [1 x i32]*
  %data_as_i8ptr395 = bitcast [1 x i32]* %data394 to i8*
  %elmptr396 = getelementptr [1 x i32], [1 x i32]* %data394, i32 0, i32 0
  store i32 0, i32* %elmptr396
  %dimsptr397 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 2
  store i8* %dims_as_i8ptr392, i8** %dimsptr397
  %dataptr398 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 3
  store i8* %data_as_i8ptr395, i8** %dataptr398
  %rc399 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 4
  store i8 0, i8* %rc399
  %tensor400 = bitcast %tensor_t* %raw_tensor387 to i8*
```

```
  %malloccall401 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor402 = bitcast i8* %malloccall401 to %tensor_t*
  %dtype403 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 0
  store i8 0, i8* %dtype403
  %ndims404 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 1
  store i8 0, i8* %ndims404
  %malloccall405 = tail call i8* @malloc(i32 0)
  %dims406 = bitcast i8* %malloccall405 to [0 x i64]*
  %dims_as_i8ptr407 = bitcast [0 x i64]* %dims406 to i8*
  %malloccall408 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data409 = bitcast i8* %malloccall408 to [1 x i32]*
  %data_as_i8ptr410 = bitcast [1 x i32]* %data409 to i8*
  %elmptr411 = getelementptr [1 x i32], [1 x i32]* %data409, i32 0, i32 0
  store i32 0, i32* %elmptr411
  %dimsptr412 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 2
  store i8* %dims_as_i8ptr407, i8** %dimsptr412
  %dataptr413 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 3
  store i8* %data_as_i8ptr410, i8** %dataptr413
  %rc414 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 4
  store i8 0, i8* %rc414
  %tensor415 = bitcast %tensor_t* %raw_tensor402 to i8*
  %malloccall416 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor417 = bitcast i8* %malloccall416 to %tensor_t*
  %dtype418 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 0
  store i8 3, i8* %dtype418
  %ndims419 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 1
  store i8 1, i8* %ndims419
  %malloccall420 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims421 = bitcast i8* %malloccall420 to [1 x i64]*
  %dims_as_i8ptr422 = bitcast [1 x i64]* %dims421 to i8*
  %elmptr423 = getelementptr [1 x i64], [1 x i64]* %dims421, i64 0, i64 0
  store i64 1, i64* %elmptr423
  %malloccall424 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data425 = bitcast i8* %malloccall424 to [1 x i8*]*
  %data_as_i8ptr426 = bitcast [1 x i8*]* %data425 to i8*
  %elmptr427 = getelementptr [1 x i8*], [1 x i8*]* %data425, i64 0, i64 0
  store i8* %tensor415, i8** %elmptr427
  %dimsptr428 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 2
  store i8* %dims_as_i8ptr422, i8** %dimsptr428
```

```
  %dataptr429 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 3
  store i8* %data_as_i8ptr426, i8** %dataptr429
  %rc430 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 4
  store i8 0, i8* %rc430
  %tensor431 = bitcast %tensor_t* %raw_tensor417 to i8*
  %l2432 = load i8*, i8** %l2
  %access_tensor433 = call i8* @index_get(i8* %l2432, i8* %tensor431)
  %malloccall434 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor435 = bitcast i8* %malloccall434 to %tensor_t*
  %dtype436 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 0
  store i8 0, i8* %dtype436
  %ndims437 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 1
  store i8 0, i8* %ndims437
  %malloccall438 = tail call i8* @malloc(i32 0)
  %dims439 = bitcast i8* %malloccall438 to [0 x i64]*
  %dims_as_i8ptr440 = bitcast [0 x i64]* %dims439 to i8*
  %malloccall441 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data442 = bitcast i8* %malloccall441 to [1 x i32]*
  %data_as_i8ptr443 = bitcast [1 x i32]* %data442 to i8*
  %elmptr444 = getelementptr [1 x i32], [1 x i32]* %data442, i32 0, i32 0
  store i32 2, i32* %elmptr444
  %dimsptr445 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 2
  store i8* %dims_as_i8ptr440, i8** %dimsptr445
  %dataptr446 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 3
  store i8* %data_as_i8ptr443, i8** %dataptr446
  %rc447 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 4
  store i8 0, i8* %rc447
  %tensor448 = bitcast %tensor_t* %raw_tensor435 to i8*
  %tmpOp449 = call i8* @floordivide(i8* %access_tensor433, i8* %tensor448)
  %malloccall450 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor451 = bitcast i8* %malloccall450 to %tensor_t*
  %dtype452 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 0
  store i8 0, i8* %dtype452
  %ndims453 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 1
  store i8 0, i8* %ndims453
  %malloccall454 = tail call i8* @malloc(i32 0)
  %dims455 = bitcast i8* %malloccall454 to [0 x i64]*
  %dims_as_i8ptr456 = bitcast [0 x i64]* %dims455 to i8*
  %malloccall457 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data458 = bitcast i8* %malloccall457 to [1 x i32]*
```

```
  %data_as_i8ptr459 = bitcast [1 x i32]* %data458 to i8*
  %elmptr460 = getelementptr [1 x i32], [1 x i32]* %data458, i32 0, i32 0
  store i32 1, i32* %elmptr460
  %dimsptr461 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 2
  store i8* %dims_as_i8ptr456, i8** %dimsptr461
  %dataptr462 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 3
  store i8* %data_as_i8ptr459, i8** %dataptr462
  %rc463 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 4
  store i8 0, i8* %rc463
  %tensor464 = bitcast %tensor_t* %raw_tensor451 to i8*
  %tmpOp465 = call i8* @range(i8* %tensor400, i8* %tmpOp449, i8* %tensor464)
  %malloccall466 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor467 = bitcast i8* %malloccall466 to %tensor_t*
  %dtype468 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 0
  store i8 0, i8* %dtype468
  %ndims469 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 1
  store i8 0, i8* %ndims469
  %malloccall470 = tail call i8* @malloc(i32 0)
  %dims471 = bitcast i8* %malloccall470 to [0 x i64]*
  %dims_as_i8ptr472 = bitcast [0 x i64]* %dims471 to i8*
  %malloccall473 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data474 = bitcast i8* %malloccall473 to [1 x i32]*
  %data_as_i8ptr475 = bitcast [1 x i32]* %data474 to i8*
  %elmptr476 = getelementptr [1 x i32], [1 x i32]* %data474, i32 0, i32 0
  store i32 0, i32* %elmptr476
  %dimsptr477 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 2
  store i8* %dims_as_i8ptr472, i8** %dimsptr477
  %dataptr478 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 3
  store i8* %data_as_i8ptr475, i8** %dataptr478
  %rc479 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 4
  store i8 0, i8* %rc479
  %tensor480 = bitcast %tensor_t* %raw_tensor467 to i8*
  %malloccall481 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor482 = bitcast i8* %malloccall481 to %tensor_t*
  %dtype483 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 0
  store i8 0, i8* %dtype483
  %ndims484 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 1
  store i8 0, i8* %ndims484
  %malloccall485 = tail call i8* @malloc(i32 0)
  %dims486 = bitcast i8* %malloccall485 to [0 x i64]*
  %dims_as_i8ptr487 = bitcast [0 x i64]* %dims486 to i8*
```

```
  %malloccall488 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data489 = bitcast i8* %malloccall488 to [1 x i32]*
  %data_as_i8ptr490 = bitcast [1 x i32]* %data489 to i8*
  %elmptr491 = getelementptr [1 x i32], [1 x i32]* %data489, i32 0, i32 0
  store i32 1, i32* %elmptr491
  %dimsptr492 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 2
  store i8* %dims_as_i8ptr487, i8** %dimsptr492
  %dataptr493 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 3
  store i8* %data_as_i8ptr490, i8** %dataptr493
  %rc494 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 4
  store i8 0, i8* %rc494
  %tensor495 = bitcast %tensor_t* %raw_tensor482 to i8*
  %malloccall496 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor497 = bitcast i8* %malloccall496 to %tensor_t*
  %dtype498 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 0
  store i8 3, i8* %dtype498
  %ndims499 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 1
  store i8 1, i8* %ndims499
  %malloccall500 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims501 = bitcast i8* %malloccall500 to [1 x i64]*
  %dims_as_i8ptr502 = bitcast [1 x i64]* %dims501 to i8*
  %elmptr503 = getelementptr [1 x i64], [1 x i64]* %dims501, i64 0, i64 0
  store i64 1, i64* %elmptr503
  %malloccall504 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data505 = bitcast i8* %malloccall504 to [1 x i8*]*
  %data_as_i8ptr506 = bitcast [1 x i8*]* %data505 to i8*
  %elmptr507 = getelementptr [1 x i8*], [1 x i8*]* %data505, i64 0, i64 0
  store i8* %tensor495, i8** %elmptr507
  %dimsptr508 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 2
  store i8* %dims_as_i8ptr502, i8** %dimsptr508
  %dataptr509 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 3
  store i8* %data_as_i8ptr506, i8** %dataptr509
  %rc510 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 4
  store i8 0, i8* %rc510
  %tensor511 = bitcast %tensor_t* %raw_tensor497 to i8*
  %l2512 = load i8*, i8** %l2
  %access_tensor513 = call i8* @index_get(i8* %l2512, i8* %tensor511)
  %malloccall514 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor515 = bitcast i8* %malloccall514 to %tensor_t*
```

```
  %dtype516 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 0
  store i8 0, i8* %dtype516
  %ndims517 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 1
  store i8 0, i8* %ndims517
  %malloccall518 = tail call i8* @malloc(i32 0)
  %dims519 = bitcast i8* %malloccall518 to [0 x i64]*
  %dims_as_i8ptr520 = bitcast [0 x i64]* %dims519 to i8*
  %malloccall521 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data522 = bitcast i8* %malloccall521 to [1 x i32]*
  %data_as_i8ptr523 = bitcast [1 x i32]* %data522 to i8*
  %elmptr524 = getelementptr [1 x i32], [1 x i32]* %data522, i32 0, i32 0
  store i32 2, i32* %elmptr524
  %dimsptr525 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 2
  store i8* %dims_as_i8ptr520, i8** %dimsptr525
  %dataptr526 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 3
  store i8* %data_as_i8ptr523, i8** %dataptr526
  %rc527 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 4
  store i8 0, i8* %rc527
  %tensor528 = bitcast %tensor_t* %raw_tensor515 to i8*
  %tmpOp529 = call i8* @floordivide(i8* %access_tensor513, i8* %tensor528)
  %malloccall530 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor531 = bitcast i8* %malloccall530 to %tensor_t*
  %dtype532 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 0
  store i8 0, i8* %dtype532
  %ndims533 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 1
  store i8 0, i8* %ndims533
  %malloccall534 = tail call i8* @malloc(i32 0)
  %dims535 = bitcast i8* %malloccall534 to [0 x i64]*
  %dims_as_i8ptr536 = bitcast [0 x i64]* %dims535 to i8*
  %malloccall537 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data538 = bitcast i8* %malloccall537 to [1 x i32]*
  %data_as_i8ptr539 = bitcast [1 x i32]* %data538 to i8*
  %elmptr540 = getelementptr [1 x i32], [1 x i32]* %data538, i32 0, i32 0
  store i32 1, i32* %elmptr540
  %dimsptr541 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 2
  store i8* %dims_as_i8ptr536, i8** %dimsptr541
  %dataptr542 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 3
  store i8* %data_as_i8ptr539, i8** %dataptr542
  %rc543 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 4
  store i8 0, i8* %rc543
  %tensor544 = bitcast %tensor_t* %raw_tensor531 to i8*
```

```
  %tmpOp545 = call i8* @range(i8* %tensor480, i8* %tmpOp529, i8* %tensor544)
  %malloccall546 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor547 = bitcast i8* %malloccall546 to %tensor_t*
  %dtype548 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor547, i32 0, i32 0
  store i8 3, i8* %dtype548
  %ndims549 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor547, i32 0, i32 1
  store i8 1, i8* %ndims549
  %malloccall550 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims551 = bitcast i8* %malloccall550 to [1 x i64]*
  %dims_as_i8ptr552 = bitcast [1 x i64]* %dims551 to i8*
  %elmptr553 = getelementptr [1 x i64], [1 x i64]* %dims551, i64 0, i64 0
  store i64 2, i64* %elmptr553
  %malloccall554 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data555 = bitcast i8* %malloccall554 to [2 x i8*]*
  %data_as_i8ptr556 = bitcast [2 x i8*]* %data555 to i8*
  %elmptr557 = getelementptr [2 x i8*], [2 x i8*]* %data555, i64 0, i64 0
  store i8* %tmpOp465, i8** %elmptr557
  %elmptr558 = getelementptr [2 x i8*], [2 x i8*]* %data555, i64 0, i64 1
  store i8* %tmpOp545, i8** %elmptr558
  %dimsptr559 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor547, i32 0, i32 2
  store i8* %dims_as_i8ptr552, i8** %dimsptr559
  %dataptr560 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor547, i32 0, i32 3
  store i8* %data_as_i8ptr556, i8** %dataptr560
  %rc561 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor547, i32 0, i32 4
  store i8 0, i8* %rc561
  %tensor562 = bitcast %tensor_t* %raw_tensor547 to i8*
  %y563 = load i8*, i8** %y
  %access_tensor564 = call i8* @index_get(i8* %y563, i8* %tensor562)
  %y11 = alloca i8*
  store i8* null, i8** %y11
  %lhsptr565 = load i8*, i8** %y11
  call void @increase_rc(i8* %access_tensor564)
  call void @decrease_rc(i8* %lhsptr565)
  store i8* %access_tensor564, i8** %y11
  %malloccall566 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor567 = bitcast i8* %malloccall566 to %tensor_t*
  %dtype568 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor567, i32 0, i32 0
  store i8 0, i8* %dtype568
  %ndims569 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor567, i32 0, i32 1
  store i8 0, i8* %ndims569
```

```
  %malloccall570 = tail call i8* @malloc(i32 0)
  %dims571 = bitcast i8* %malloccall570 to [0 x i64]*
  %dims_as_i8ptr572 = bitcast [0 x i64]* %dims571 to i8*
  %malloccall573 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data574 = bitcast i8* %malloccall573 to [1 x i32]*
  %data_as_i8ptr575 = bitcast [1 x i32]* %data574 to i8*
  %elmptr576 = getelementptr [1 x i32], [1 x i32]* %data574, i32 0, i32 0
  store i32 0, i32* %elmptr576
  %dimsptr577 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor567, i32 0, i32 2
  store i8* %dims_as_i8ptr572, i8** %dimsptr577
  %dataptr578 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor567, i32 0, i32 3
  store i8* %data_as_i8ptr575, i8** %dataptr578
  %rc579 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor567, i32 0, i32 4
  store i8 0, i8* %rc579
  %tensor580 = bitcast %tensor_t* %raw_tensor567 to i8*
  %malloccall581 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor582 = bitcast i8* %malloccall581 to %tensor_t*
  %dtype583 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor582, i32 0, i32 0
  store i8 3, i8* %dtype583
  %ndims584 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor582, i32 0, i32 1
  store i8 1, i8* %ndims584
  %malloccall585 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims586 = bitcast i8* %malloccall585 to [1 x i64]*
  %dims_as_i8ptr587 = bitcast [1 x i64]* %dims586 to i8*
  %elmptr588 = getelementptr [1 x i64], [1 x i64]* %dims586, i64 0, i64 0
  store i64 1, i64* %elmptr588
  %malloccall589 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data590 = bitcast i8* %malloccall589 to [1 x i8*]*
  %data_as_i8ptr591 = bitcast [1 x i8*]* %data590 to i8*
  %elmptr592 = getelementptr [1 x i8*], [1 x i8*]* %data590, i64 0, i64 0
  store i8* %tensor580, i8** %elmptr592
  %dimsptr593 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor582, i32 0, i32 2
  store i8* %dims_as_i8ptr587, i8** %dimsptr593
  %dataptr594 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor582, i32 0, i32 3
  store i8* %data_as_i8ptr591, i8** %dataptr594
  %rc595 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor582, i32 0, i32 4
  store i8 0, i8* %rc595
  %tensor596 = bitcast %tensor_t* %raw_tensor582 to i8*
  %l2597 = load i8*, i8** %l2
  %access_tensor598 = call i8* @index_get(i8* %l2597, i8* %tensor596)
```

```
  %malloccall599 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor600 = bitcast i8* %malloccall599 to %tensor_t*
  %dtype601 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor600, i32 0, i32 0
  store i8 0, i8* %dtype601
  %ndims602 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor600, i32 0, i32 1
  store i8 0, i8* %ndims602
  %malloccall603 = tail call i8* @malloc(i32 0)
  %dims604 = bitcast i8* %malloccall603 to [0 x i64]*
  %dims_as_i8ptr605 = bitcast [0 x i64]* %dims604 to i8*
  %malloccall606 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data607 = bitcast i8* %malloccall606 to [1 x i32]*
  %data_as_i8ptr608 = bitcast [1 x i32]* %data607 to i8*
  %elmptr609 = getelementptr [1 x i32], [1 x i32]* %data607, i32 0, i32 0
  store i32 2, i32* %elmptr609
  %dimsptr610 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor600, i32 0, i32 2
  store i8* %dims_as_i8ptr605, i8** %dimsptr610
  %dataptr611 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor600, i32 0, i32 3
  store i8* %data_as_i8ptr608, i8** %dataptr611
  %rc612 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor600, i32 0, i32 4
  store i8 0, i8* %rc612
  %tensor613 = bitcast %tensor_t* %raw_tensor600 to i8*
  %tmpOp614 = call i8* @floordivide(i8* %access_tensor598, i8* %tensor613)
  %malloccall615 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor616 = bitcast i8* %malloccall615 to %tensor_t*
  %dtype617 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor616, i32 0, i32 0
  store i8 0, i8* %dtype617
  %ndims618 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor616, i32 0, i32 1
  store i8 0, i8* %ndims618
  %malloccall619 = tail call i8* @malloc(i32 0)
  %dims620 = bitcast i8* %malloccall619 to [0 x i64]*
  %dims_as_i8ptr621 = bitcast [0 x i64]* %dims620 to i8*
  %malloccall622 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data623 = bitcast i8* %malloccall622 to [1 x i32]*
  %data_as_i8ptr624 = bitcast [1 x i32]* %data623 to i8*
  %elmptr625 = getelementptr [1 x i32], [1 x i32]* %data623, i32 0, i32 0
  store i32 0, i32* %elmptr625
  %dimsptr626 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor616, i32 0, i32 2
  store i8* %dims_as_i8ptr621, i8** %dimsptr626
  %dataptr627 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor616, i32 0, i32 3
  store i8* %data_as_i8ptr624, i8** %dataptr627
```

```
  %rc628 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor616, i32 0, i32 4
  store i8 0, i8* %rc628
  %tensor629 = bitcast %tensor_t* %raw_tensor616 to i8*
  %malloccall630 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor631 = bitcast i8* %malloccall630 to %tensor_t*
  %dtype632 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor631, i32 0, i32 0
  store i8 3, i8* %dtype632
  %ndims633 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor631, i32 0, i32 1
  store i8 1, i8* %ndims633
  %malloccall634 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims635 = bitcast i8* %malloccall634 to [1 x i64]*
  %dims_as_i8ptr636 = bitcast [1 x i64]* %dims635 to i8*
  %elmptr637 = getelementptr [1 x i64], [1 x i64]* %dims635, i64 0, i64 0
  store i64 1, i64* %elmptr637
  %malloccall638 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data639 = bitcast i8* %malloccall638 to [1 x i8*]*
  %data_as_i8ptr640 = bitcast [1 x i8*]* %data639 to i8*
  %elmptr641 = getelementptr [1 x i8*], [1 x i8*]* %data639, i64 0, i64 0
  store i8* %tensor629, i8** %elmptr641
  %dimsptr642 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor631, i32 0, i32 2
  store i8* %dims_as_i8ptr636, i8** %dimsptr642
  %dataptr643 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor631, i32 0, i32 3
  store i8* %data_as_i8ptr640, i8** %dataptr643
  %rc644 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor631, i32 0, i32 4
  store i8 0, i8* %rc644
  %tensor645 = bitcast %tensor_t* %raw_tensor631 to i8*
  %l2646 = load i8*, i8** %l2
  %access_tensor647 = call i8* @index_get(i8* %l2646, i8* %tensor645)
  %malloccall648 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor649 = bitcast i8* %malloccall648 to %tensor_t*
  %dtype650 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor649, i32 0, i32 0
  store i8 0, i8* %dtype650
  %ndims651 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor649, i32 0, i32 1
  store i8 0, i8* %ndims651
  %malloccall652 = tail call i8* @malloc(i32 0)
  %dims653 = bitcast i8* %malloccall652 to [0 x i64]*
  %dims_as_i8ptr654 = bitcast [0 x i64]* %dims653 to i8*
  %malloccall655 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data656 = bitcast i8* %malloccall655 to [1 x i32]*
```

```
  %data_as_i8ptr657 = bitcast [1 x i32]* %data656 to i8*
  %elmptr658 = getelementptr [1 x i32], [1 x i32]* %data656, i32 0, i32 0
  store i32 1, i32* %elmptr658
  %dimsptr659 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor649, i32 0, i32 2
  store i8* %dims_as_i8ptr654, i8** %dimsptr659
  %dataptr660 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor649, i32 0, i32 3
  store i8* %data_as_i8ptr657, i8** %dataptr660
  %rc661 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor649, i32 0, i32 4
  store i8 0, i8* %rc661
  %tensor662 = bitcast %tensor_t* %raw_tensor649 to i8*
  %tmpOp663 = call i8* @range(i8* %tmpOp614, i8* %access_tensor647, i8* %tensor662)
  %malloccall664 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor665 = bitcast i8* %malloccall664 to %tensor_t*
  %dtype666 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor665, i32 0, i32 0
  store i8 0, i8* %dtype666
  %ndims667 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor665, i32 0, i32 1
  store i8 0, i8* %ndims667
  %malloccall668 = tail call i8* @malloc(i32 0)
  %dims669 = bitcast i8* %malloccall668 to [0 x i64]*
  %dims_as_i8ptr670 = bitcast [0 x i64]* %dims669 to i8*
  %malloccall671 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data672 = bitcast i8* %malloccall671 to [1 x i32]*
  %data_as_i8ptr673 = bitcast [1 x i32]* %data672 to i8*
  %elmptr674 = getelementptr [1 x i32], [1 x i32]* %data672, i32 0, i32 0
  store i32 1, i32* %elmptr674
  %dimsptr675 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor665, i32 0, i32 2
  store i8* %dims_as_i8ptr670, i8** %dimsptr675
  %dataptr676 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor665, i32 0, i32 3
  store i8* %data_as_i8ptr673, i8** %dataptr676
  %rc677 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor665, i32 0, i32 4
  store i8 0, i8* %rc677
  %tensor678 = bitcast %tensor_t* %raw_tensor665 to i8*
  %malloccall679 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor680 = bitcast i8* %malloccall679 to %tensor_t*
  %dtype681 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor680, i32 0, i32 0
  store i8 3, i8* %dtype681
  %ndims682 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor680, i32 0, i32 1
  store i8 1, i8* %ndims682
  %malloccall683 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims684 = bitcast i8* %malloccall683 to [1 x i64]*
```

```
  %dims_as_i8ptr685 = bitcast [1 x i64]* %dims684 to i8*
  %elmptr686 = getelementptr [1 x i64], [1 x i64]* %dims684, i64 0, i64 0
  store i64 1, i64* %elmptr686
  %malloccall687 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data688 = bitcast i8* %malloccall687 to [1 x i8*]*
  %data_as_i8ptr689 = bitcast [1 x i8*]* %data688 to i8*
  %elmptr690 = getelementptr [1 x i8*], [1 x i8*]* %data688, i64 0, i64 0
  store i8* %tensor678, i8** %elmptr690
  %dimsptr691 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor680, i32 0, i32 2
  store i8* %dims_as_i8ptr685, i8** %dimsptr691
  %dataptr692 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor680, i32 0, i32 3
  store i8* %data_as_i8ptr689, i8** %dataptr692
  %rc693 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor680, i32 0, i32 4
  store i8 0, i8* %rc693
  %tensor694 = bitcast %tensor_t* %raw_tensor680 to i8*
  %l2695 = load i8*, i8** %l2
  %access_tensor696 = call i8* @index_get(i8* %l2695, i8* %tensor694)
  %malloccall697 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor698 = bitcast i8* %malloccall697 to %tensor_t*
  %dtype699 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor698, i32 0, i32 0
  store i8 0, i8* %dtype699
  %ndims700 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor698, i32 0, i32 1
  store i8 0, i8* %ndims700
  %malloccall701 = tail call i8* @malloc(i32 0)
  %dims702 = bitcast i8* %malloccall701 to [0 x i64]*
  %dims_as_i8ptr703 = bitcast [0 x i64]* %dims702 to i8*
  %malloccall704 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data705 = bitcast i8* %malloccall704 to [1 x i32]*
  %data_as_i8ptr706 = bitcast [1 x i32]* %data705 to i8*
  %elmptr707 = getelementptr [1 x i32], [1 x i32]* %data705, i32 0, i32 0
  store i32 2, i32* %elmptr707
  %dimsptr708 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor698, i32 0, i32 2
  store i8* %dims_as_i8ptr703, i8** %dimsptr708
  %dataptr709 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor698, i32 0, i32 3
  store i8* %data_as_i8ptr706, i8** %dataptr709
  %rc710 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor698, i32 0, i32 4
  store i8 0, i8* %rc710
  %tensor711 = bitcast %tensor_t* %raw_tensor698 to i8*
  %tmpOp712 = call i8* @floordivide(i8* %access_tensor696, i8* %tensor711)
  %malloccall713 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor714 = bitcast i8* %malloccall713 to %tensor_t*
  %dtype715 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor714, i32 0, i32 0
  store i8 0, i8* %dtype715
  %ndims716 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor714, i32 0, i32 1
  store i8 0, i8* %ndims716
  %malloccall717 = tail call i8* @malloc(i32 0)
  %dims718 = bitcast i8* %malloccall717 to [0 x i64]*
  %dims_as_i8ptr719 = bitcast [0 x i64]* %dims718 to i8*
  %malloccall720 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data721 = bitcast i8* %malloccall720 to [1 x i32]*
  %data_as_i8ptr722 = bitcast [1 x i32]* %data721 to i8*
  %elmptr723 = getelementptr [1 x i32], [1 x i32]* %data721, i32 0, i32 0
  store i32 1, i32* %elmptr723
  %dimsptr724 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor714, i32 0, i32 2
  store i8* %dims_as_i8ptr719, i8** %dimsptr724
  %dataptr725 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor714, i32 0, i32 3
  store i8* %data_as_i8ptr722, i8** %dataptr725
  %rc726 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor714, i32 0, i32 4
  store i8 0, i8* %rc726
  %tensor727 = bitcast %tensor_t* %raw_tensor714 to i8*
  %malloccall728 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor729 = bitcast i8* %malloccall728 to %tensor_t*
  %dtype730 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor729, i32 0, i32 0
  store i8 3, i8* %dtype730
  %ndims731 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor729, i32 0, i32 1
  store i8 1, i8* %ndims731
  %malloccall732 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims733 = bitcast i8* %malloccall732 to [1 x i64]*
  %dims_as_i8ptr734 = bitcast [1 x i64]* %dims733 to i8*
  %elmptr735 = getelementptr [1 x i64], [1 x i64]* %dims733, i64 0, i64 0
  store i64 1, i64* %elmptr735
  %malloccall736 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data737 = bitcast i8* %malloccall736 to [1 x i8*]*
  %data_as_i8ptr738 = bitcast [1 x i8*]* %data737 to i8*
  %elmptr739 = getelementptr [1 x i8*], [1 x i8*]* %data737, i64 0, i64 0
  store i8* %tensor727, i8** %elmptr739
  %dimsptr740 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor729, i32 0, i32 2
  store i8* %dims_as_i8ptr734, i8** %dimsptr740
  %dataptr741 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor729, i32 0, i32 3
  store i8* %data_as_i8ptr738, i8** %dataptr741
```

```
  %rc742 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor729, i32 0, i32 4
  store i8 0, i8* %rc742
  %tensor743 = bitcast %tensor_t* %raw_tensor729 to i8*
  %l2744 = load i8*, i8** %l2
  %access_tensor745 = call i8* @index_get(i8* %l2744, i8* %tensor743)
  %malloccall746 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor747 = bitcast i8* %malloccall746 to %tensor_t*
  %dtype748 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor747, i32 0, i32 0
  store i8 0, i8* %dtype748
  %ndims749 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor747, i32 0, i32 1
  store i8 0, i8* %ndims749
  %malloccall750 = tail call i8* @malloc(i32 0)
  %dims751 = bitcast i8* %malloccall750 to [0 x i64]*
  %dims_as_i8ptr752 = bitcast [0 x i64]* %dims751 to i8*
  %malloccall753 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data754 = bitcast i8* %malloccall753 to [1 x i32]*
  %data_as_i8ptr755 = bitcast [1 x i32]* %data754 to i8*
  %elmptr756 = getelementptr [1 x i32], [1 x i32]* %data754, i32 0, i32 0
  store i32 1, i32* %elmptr756
  %dimsptr757 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor747, i32 0, i32 2
  store i8* %dims_as_i8ptr752, i8** %dimsptr757
  %dataptr758 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor747, i32 0, i32 3
  store i8* %data_as_i8ptr755, i8** %dataptr758
  %rc759 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor747, i32 0, i32 4
  store i8 0, i8* %rc759
  %tensor760 = bitcast %tensor_t* %raw_tensor747 to i8*
  %tmpOp761 = call i8* @range(i8* %tmpOp712, i8* %access_tensor745, i8* %tensor760)
  %malloccall762 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor763 = bitcast i8* %malloccall762 to %tensor_t*
  %dtype764 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor763, i32 0, i32 0
  store i8 3, i8* %dtype764
  %ndims765 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor763, i32 0, i32 1
  store i8 1, i8* %ndims765
  %malloccall766 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims767 = bitcast i8* %malloccall766 to [1 x i64]*
  %dims_as_i8ptr768 = bitcast [1 x i64]* %dims767 to i8*
  %elmptr769 = getelementptr [1 x i64], [1 x i64]* %dims767, i64 0, i64 0
  store i64 2, i64* %elmptr769
  %malloccall770 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
```

```
  %data771 = bitcast i8* %malloccall770 to [2 x i8*]*
  %data_as_i8ptr772 = bitcast [2 x i8*]* %data771 to i8*
  %elmptr773 = getelementptr [2 x i8*], [2 x i8*]* %data771, i64 0, i64 0
  store i8* %tmpOp663, i8** %elmptr773
  %elmptr774 = getelementptr [2 x i8*], [2 x i8*]* %data771, i64 0, i64 1
  store i8* %tmpOp761, i8** %elmptr774
  %dimsptr775 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor763, i32 0, i32 2
  store i8* %dims_as_i8ptr768, i8** %dimsptr775
  %dataptr776 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor763, i32 0, i32 3
  store i8* %data_as_i8ptr772, i8** %dataptr776
  %rc777 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor763, i32 0, i32 4
  store i8 0, i8* %rc777
  %tensor778 = bitcast %tensor_t* %raw_tensor763 to i8*
  %y779 = load i8*, i8** %y
  %access_tensor780 = call i8* @index_get(i8* %y779, i8* %tensor778)
  %y22 = alloca i8*
  store i8* null, i8** %y22
  %lhsptr781 = load i8*, i8** %y22
  call void @increase_rc(i8* %access_tensor780)
  call void @decrease_rc(i8* %lhsptr781)
  store i8* %access_tensor780, i8** %y22
  %x11782 = load i8*, i8** %x11
  %x22783 = load i8*, i8** %x22
  %tmpOp784 = call i8* @add(i8* %x11782, i8* %x22783)
  %y11785 = load i8*, i8** %y11
  %y22786 = load i8*, i8** %y22
  %tmpOp787 = call i8* @add(i8* %y11785, i8* %y22786)
  %tmpOp788 = call i8* @mult(i8* %tmpOp784, i8* %tmpOp787)
  ret i8* %tmpOp788
}

declare noalias i8* @malloc(i32)

define i8* @StrassenMULf2(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
  %l1 = alloca i8*
  store i8* null, i8** %l1
  %lhsptr = load i8*, i8** %l1
```

```
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l1
  %y2 = load i8*, i8** %y
  %shape3 = call i8* @shape(i8* %y2)
  %l2 = alloca i8*
  store i8* null, i8** %l2
  %lhsptr4 = load i8*, i8** %l2
  call void @increase_rc(i8* %shape3)
  call void @decrease_rc(i8* %lhsptr4)
  store i8* %shape3, i8** %l2
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall5 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall5 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall6 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall6 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor8 = bitcast i8* %malloccall7 to %tensor_t*
  %dtype9 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 0
  store i8 3, i8* %dtype9
  %ndims10 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 1
  store i8 1, i8* %ndims10
  %malloccall11 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims12 = bitcast i8* %malloccall11 to [1 x i64]*
  %dims_as_i8ptr13 = bitcast [1 x i64]* %dims12 to i8*
```

```
  %elmptr14 = getelementptr [1 x i64], [1 x i64]* %dims12, i64 0, i64 0
  store i64 1, i64* %elmptr14
  %malloccall15 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data16 = bitcast i8* %malloccall15 to [1 x i8*]*
  %data_as_i8ptr17 = bitcast [1 x i8*]* %data16 to i8*
  %elmptr18 = getelementptr [1 x i8*], [1 x i8*]* %data16, i64 0, i64 0
  store i8* %tensor, i8** %elmptr18
  %dimsptr19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 2
  store i8* %dims_as_i8ptr13, i8** %dimsptr19
  %dataptr20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 3
  store i8* %data_as_i8ptr17, i8** %dataptr20
  %rc21 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 4
  store i8 0, i8* %rc21
  %tensor22 = bitcast %tensor_t* %raw_tensor8 to i8*
  %l123 = load i8*, i8** %l1
  %access_tensor = call i8* @index_get(i8* %l123, i8* %tensor22)
  %malloccall24 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor25 = bitcast i8* %malloccall24 to %tensor_t*
  %dtype26 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 0
  store i8 0, i8* %dtype26
  %ndims27 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 1
  store i8 0, i8* %ndims27
  %malloccall28 = tail call i8* @malloc(i32 0)
  %dims29 = bitcast i8* %malloccall28 to [0 x i64]*
  %dims_as_i8ptr30 = bitcast [0 x i64]* %dims29 to i8*
  %malloccall31 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data32 = bitcast i8* %malloccall31 to [1 x i32]*
  %data_as_i8ptr33 = bitcast [1 x i32]* %data32 to i8*
  %elmptr34 = getelementptr [1 x i32], [1 x i32]* %data32, i32 0, i32 0
  store i32 2, i32* %elmptr34
  %dimsptr35 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 2
  store i8* %dims_as_i8ptr30, i8** %dimsptr35
  %dataptr36 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 3
  store i8* %data_as_i8ptr33, i8** %dataptr36
  %rc37 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 4
  store i8 0, i8* %rc37
  %tensor38 = bitcast %tensor_t* %raw_tensor25 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor38)
  %malloccall39 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor40 = bitcast i8* %malloccall39 to %tensor_t*
  %dtype41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 0
```

```
  store i8 0, i8* %dtype41
  %ndims42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 1
  store i8 0, i8* %ndims42
  %malloccall43 = tail call i8* @malloc(i32 0)
  %dims44 = bitcast i8* %malloccall43 to [0 x i64]*
  %dims_as_i8ptr45 = bitcast [0 x i64]* %dims44 to i8*
  %malloccall46 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data47 = bitcast i8* %malloccall46 to [1 x i32]*
  %data_as_i8ptr48 = bitcast [1 x i32]* %data47 to i8*
  %elmptr49 = getelementptr [1 x i32], [1 x i32]* %data47, i32 0, i32 0
  store i32 0, i32* %elmptr49
  %dimsptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 2
  store i8* %dims_as_i8ptr45, i8** %dimsptr50
  %dataptr51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 3
  store i8* %data_as_i8ptr48, i8** %dataptr51
  %rc52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 4
  store i8 0, i8* %rc52
  %tensor53 = bitcast %tensor_t* %raw_tensor40 to i8*
  %malloccall54 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor55 = bitcast i8* %malloccall54 to %tensor_t*
  %dtype56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 0
  store i8 3, i8* %dtype56
  %ndims57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 1
  store i8 1, i8* %ndims57
  %malloccall58 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims59 = bitcast i8* %malloccall58 to [1 x i64]*
  %dims_as_i8ptr60 = bitcast [1 x i64]* %dims59 to i8*
  %elmptr61 = getelementptr [1 x i64], [1 x i64]* %dims59, i64 0, i64 0
  store i64 1, i64* %elmptr61
  %malloccall62 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data63 = bitcast i8* %malloccall62 to [1 x i8*]*
  %data_as_i8ptr64 = bitcast [1 x i8*]* %data63 to i8*
  %elmptr65 = getelementptr [1 x i8*], [1 x i8*]* %data63, i64 0, i64 0
  store i8* %tensor53, i8** %elmptr65
  %dimsptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 2
  store i8* %dims_as_i8ptr60, i8** %dimsptr66
  %dataptr67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 3
  store i8* %data_as_i8ptr64, i8** %dataptr67
  %rc68 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 4
  store i8 0, i8* %rc68
  %tensor69 = bitcast %tensor_t* %raw_tensor55 to i8*
```

```
  %l170 = load i8*, i8** %l1
  %access_tensor71 = call i8* @index_get(i8* %l170, i8* %tensor69)
  %malloccall72 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor73 = bitcast i8* %malloccall72 to %tensor_t*
  %dtype74 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 0
  store i8 0, i8* %dtype74
  %ndims75 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 1
  store i8 0, i8* %ndims75
  %malloccall76 = tail call i8* @malloc(i32 0)
  %dims77 = bitcast i8* %malloccall76 to [0 x i64]*
  %dims_as_i8ptr78 = bitcast [0 x i64]* %dims77 to i8*
  %malloccall79 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data80 = bitcast i8* %malloccall79 to [1 x i32]*
  %data_as_i8ptr81 = bitcast [1 x i32]* %data80 to i8*
  %elmptr82 = getelementptr [1 x i32], [1 x i32]* %data80, i32 0, i32 0
  store i32 1, i32* %elmptr82
  %dimsptr83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 2
  store i8* %dims_as_i8ptr78, i8** %dimsptr83
  %dataptr84 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 3
  store i8* %data_as_i8ptr81, i8** %dataptr84
  %rc85 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 4
  store i8 0, i8* %rc85
  %tensor86 = bitcast %tensor_t* %raw_tensor73 to i8*
  %tmpOp87 = call i8* @range(i8* %tmpOp, i8* %access_tensor71, i8* %tensor86)
  %malloccall88 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor89 = bitcast i8* %malloccall88 to %tensor_t*
  %dtype90 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 0
  store i8 0, i8* %dtype90
  %ndims91 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 1
  store i8 0, i8* %ndims91
  %malloccall92 = tail call i8* @malloc(i32 0)
  %dims93 = bitcast i8* %malloccall92 to [0 x i64]*
  %dims_as_i8ptr94 = bitcast [0 x i64]* %dims93 to i8*
  %malloccall95 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data96 = bitcast i8* %malloccall95 to [1 x i32]*
  %data_as_i8ptr97 = bitcast [1 x i32]* %data96 to i8*
  %elmptr98 = getelementptr [1 x i32], [1 x i32]* %data96, i32 0, i32 0
  store i32 0, i32* %elmptr98
  %dimsptr99 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 2
  store i8* %dims_as_i8ptr94, i8** %dimsptr99
```

```
  %dataptr100 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 3
  store i8* %data_as_i8ptr97, i8** %dataptr100
  %rc101 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 4
  store i8 0, i8* %rc101
  %tensor102 = bitcast %tensor_t* %raw_tensor89 to i8*
  %malloccall103 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor104 = bitcast i8* %malloccall103 to %tensor_t*
  %dtype105 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 0
  store i8 0, i8* %dtype105
  %ndims106 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 1
  store i8 0, i8* %ndims106
  %malloccall107 = tail call i8* @malloc(i32 0)
  %dims108 = bitcast i8* %malloccall107 to [0 x i64]*
  %dims_as_i8ptr109 = bitcast [0 x i64]* %dims108 to i8*
  %malloccall110 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data111 = bitcast i8* %malloccall110 to [1 x i32]*
  %data_as_i8ptr112 = bitcast [1 x i32]* %data111 to i8*
  %elmptr113 = getelementptr [1 x i32], [1 x i32]* %data111, i32 0, i32 0
  store i32 1, i32* %elmptr113
  %dimsptr114 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 2
  store i8* %dims_as_i8ptr109, i8** %dimsptr114
  %dataptr115 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 3
  store i8* %data_as_i8ptr112, i8** %dataptr115
  %rc116 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 4
  store i8 0, i8* %rc116
  %tensor117 = bitcast %tensor_t* %raw_tensor104 to i8*
  %malloccall118 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor119 = bitcast i8* %malloccall118 to %tensor_t*
  %dtype120 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 0
  store i8 3, i8* %dtype120
  %ndims121 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 1
  store i8 1, i8* %ndims121
  %malloccall122 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims123 = bitcast i8* %malloccall122 to [1 x i64]*
  %dims_as_i8ptr124 = bitcast [1 x i64]* %dims123 to i8*
  %elmptr125 = getelementptr [1 x i64], [1 x i64]* %dims123, i64 0, i64 0
  store i64 1, i64* %elmptr125
  %malloccall126 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data127 = bitcast i8* %malloccall126 to [1 x i8*]*
```

```
  %data_as_i8ptr128 = bitcast [1 x i8*]* %data127 to i8*
  %elmptr129 = getelementptr [1 x i8*], [1 x i8*]* %data127, i64 0, i64 0
  store i8* %tensor117, i8** %elmptr129
  %dimsptr130 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 2
  store i8* %dims_as_i8ptr124, i8** %dimsptr130
  %dataptr131 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 3
  store i8* %data_as_i8ptr128, i8** %dataptr131
  %rc132 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 4
  store i8 0, i8* %rc132
  %tensor133 = bitcast %tensor_t* %raw_tensor119 to i8*
  %l1134 = load i8*, i8** %l1
  %access_tensor135 = call i8* @index_get(i8* %l1134, i8* %tensor133)
  %malloccall136 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor137 = bitcast i8* %malloccall136 to %tensor_t*
  %dtype138 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 0
  store i8 0, i8* %dtype138
  %ndims139 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 1
  store i8 0, i8* %ndims139
  %malloccall140 = tail call i8* @malloc(i32 0)
  %dims141 = bitcast i8* %malloccall140 to [0 x i64]*
  %dims_as_i8ptr142 = bitcast [0 x i64]* %dims141 to i8*
  %malloccall143 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data144 = bitcast i8* %malloccall143 to [1 x i32]*
  %data_as_i8ptr145 = bitcast [1 x i32]* %data144 to i8*
  %elmptr146 = getelementptr [1 x i32], [1 x i32]* %data144, i32 0, i32 0
  store i32 2, i32* %elmptr146
  %dimsptr147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 2
  store i8* %dims_as_i8ptr142, i8** %dimsptr147
  %dataptr148 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 3
  store i8* %data_as_i8ptr145, i8** %dataptr148
  %rc149 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 4
  store i8 0, i8* %rc149
  %tensor150 = bitcast %tensor_t* %raw_tensor137 to i8*
  %tmpOp151 = call i8* @floordivide(i8* %access_tensor135, i8* %tensor150)
  %malloccall152 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor153 = bitcast i8* %malloccall152 to %tensor_t*
  %dtype154 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 0
  store i8 0, i8* %dtype154
  %ndims155 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 1
  store i8 0, i8* %ndims155
  %malloccall156 = tail call i8* @malloc(i32 0)
```

```
  %dims157 = bitcast i8* %malloccall156 to [0 x i64]*
  %dims_as_i8ptr158 = bitcast [0 x i64]* %dims157 to i8*
  %malloccall159 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data160 = bitcast i8* %malloccall159 to [1 x i32]*
  %data_as_i8ptr161 = bitcast [1 x i32]* %data160 to i8*
  %elmptr162 = getelementptr [1 x i32], [1 x i32]* %data160, i32 0, i32 0
  store i32 1, i32* %elmptr162
  %dimsptr163 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 2
  store i8* %dims_as_i8ptr158, i8** %dimsptr163
  %dataptr164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 3
  store i8* %data_as_i8ptr161, i8** %dataptr164
  %rc165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 4
  store i8 0, i8* %rc165
  %tensor166 = bitcast %tensor_t* %raw_tensor153 to i8*
  %tmpOp167 = call i8* @range(i8* %tensor102, i8* %tmpOp151, i8* %tensor166)
  %malloccall168 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor169 = bitcast i8* %malloccall168 to %tensor_t*
  %dtype170 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 0
  store i8 3, i8* %dtype170
  %ndims171 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 1
  store i8 1, i8* %ndims171
  %malloccall172 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims173 = bitcast i8* %malloccall172 to [1 x i64]*
  %dims_as_i8ptr174 = bitcast [1 x i64]* %dims173 to i8*
  %elmptr175 = getelementptr [1 x i64], [1 x i64]* %dims173, i64 0, i64 0
  store i64 2, i64* %elmptr175
  %malloccall176 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data177 = bitcast i8* %malloccall176 to [2 x i8*]*
  %data_as_i8ptr178 = bitcast [2 x i8*]* %data177 to i8*
  %elmptr179 = getelementptr [2 x i8*], [2 x i8*]* %data177, i64 0, i64 0
  store i8* %tmpOp87, i8** %elmptr179
  %elmptr180 = getelementptr [2 x i8*], [2 x i8*]* %data177, i64 0, i64 1
  store i8* %tmpOp167, i8** %elmptr180
  %dimsptr181 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 2
  store i8* %dims_as_i8ptr174, i8** %dimsptr181
  %dataptr182 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 3
  store i8* %data_as_i8ptr178, i8** %dataptr182
  %rc183 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 4
  store i8 0, i8* %rc183
  %tensor184 = bitcast %tensor_t* %raw_tensor169 to i8*
```

```
  %x185 = load i8*, i8** %x
  %access_tensor186 = call i8* @index_get(i8* %x185, i8* %tensor184)
  %x21 = alloca i8*
  store i8* null, i8** %x21
  %lhsptr187 = load i8*, i8** %x21
  call void @increase_rc(i8* %access_tensor186)
  call void @decrease_rc(i8* %lhsptr187)
  store i8* %access_tensor186, i8** %x21
  %malloccall188 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor189 = bitcast i8* %malloccall188 to %tensor_t*
  %dtype190 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 0
  store i8 0, i8* %dtype190
  %ndims191 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 1
  store i8 0, i8* %ndims191
  %malloccall192 = tail call i8* @malloc(i32 0)
  %dims193 = bitcast i8* %malloccall192 to [0 x i64]*
  %dims_as_i8ptr194 = bitcast [0 x i64]* %dims193 to i8*
  %malloccall195 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data196 = bitcast i8* %malloccall195 to [1 x i32]*
  %data_as_i8ptr197 = bitcast [1 x i32]* %data196 to i8*
  %elmptr198 = getelementptr [1 x i32], [1 x i32]* %data196, i32 0, i32 0
  store i32 0, i32* %elmptr198
  %dimsptr199 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 2
  store i8* %dims_as_i8ptr194, i8** %dimsptr199
  %dataptr200 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 3
  store i8* %data_as_i8ptr197, i8** %dataptr200
  %rc201 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 4
  store i8 0, i8* %rc201
  %tensor202 = bitcast %tensor_t* %raw_tensor189 to i8*
  %malloccall203 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor204 = bitcast i8* %malloccall203 to %tensor_t*
  %dtype205 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 0
  store i8 3, i8* %dtype205
  %ndims206 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 1
  store i8 1, i8* %ndims206
  %malloccall207 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims208 = bitcast i8* %malloccall207 to [1 x i64]*
  %dims_as_i8ptr209 = bitcast [1 x i64]* %dims208 to i8*
  %elmptr210 = getelementptr [1 x i64], [1 x i64]* %dims208, i64 0, i64 0
  store i64 1, i64* %elmptr210
```

```
  %malloccall211 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data212 = bitcast i8* %malloccall211 to [1 x i8*]*
  %data_as_i8ptr213 = bitcast [1 x i8*]* %data212 to i8*
  %elmptr214 = getelementptr [1 x i8*], [1 x i8*]* %data212, i64 0, i64 0
  store i8* %tensor202, i8** %elmptr214
  %dimsptr215 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 2
  store i8* %dims_as_i8ptr209, i8** %dimsptr215
  %dataptr216 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 3
  store i8* %data_as_i8ptr213, i8** %dataptr216
  %rc217 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 4
  store i8 0, i8* %rc217
  %tensor218 = bitcast %tensor_t* %raw_tensor204 to i8*
  %l1219 = load i8*, i8** %l1
  %access_tensor220 = call i8* @index_get(i8* %l1219, i8* %tensor218)
  %malloccall221 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor222 = bitcast i8* %malloccall221 to %tensor_t*
  %dtype223 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 0
  store i8 0, i8* %dtype223
  %ndims224 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 1
  store i8 0, i8* %ndims224
  %malloccall225 = tail call i8* @malloc(i32 0)
  %dims226 = bitcast i8* %malloccall225 to [0 x i64]*
  %dims_as_i8ptr227 = bitcast [0 x i64]* %dims226 to i8*
  %malloccall228 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data229 = bitcast i8* %malloccall228 to [1 x i32]*
  %data_as_i8ptr230 = bitcast [1 x i32]* %data229 to i8*
  %elmptr231 = getelementptr [1 x i32], [1 x i32]* %data229, i32 0, i32 0
  store i32 2, i32* %elmptr231
  %dimsptr232 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 2
  store i8* %dims_as_i8ptr227, i8** %dimsptr232
  %dataptr233 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 3
  store i8* %data_as_i8ptr230, i8** %dataptr233
  %rc234 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 4
  store i8 0, i8* %rc234
  %tensor235 = bitcast %tensor_t* %raw_tensor222 to i8*
  %tmpOp236 = call i8* @floordivide(i8* %access_tensor220, i8* %tensor235)
  %malloccall237 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor238 = bitcast i8* %malloccall237 to %tensor_t*
  %dtype239 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor238, i32 0, i32 0
  store i8 0, i8* %dtype239
```

```
  %ndims240 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor238, i32 0, i32 1
  store i8 0, i8* %ndims240
  %malloccall241 = tail call i8* @malloc(i32 0)
  %dims242 = bitcast i8* %malloccall241 to [0 x i64]*
  %dims_as_i8ptr243 = bitcast [0 x i64]* %dims242 to i8*
  %malloccall244 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data245 = bitcast i8* %malloccall244 to [1 x i32]*
  %data_as_i8ptr246 = bitcast [1 x i32]* %data245 to i8*
  %elmptr247 = getelementptr [1 x i32], [1 x i32]* %data245, i32 0, i32 0
  store i32 0, i32* %elmptr247
  %dimsptr248 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor238, i32 0, i32 2
  store i8* %dims_as_i8ptr243, i8** %dimsptr248
  %dataptr249 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor238, i32 0, i32 3
  store i8* %data_as_i8ptr246, i8** %dataptr249
  %rc250 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor238, i32 0, i32 4
  store i8 0, i8* %rc250
  %tensor251 = bitcast %tensor_t* %raw_tensor238 to i8*
  %malloccall252 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor253 = bitcast i8* %malloccall252 to %tensor_t*
  %dtype254 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 0
  store i8 3, i8* %dtype254
  %ndims255 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 1
  store i8 1, i8* %ndims255
  %malloccall256 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims257 = bitcast i8* %malloccall256 to [1 x i64]*
  %dims_as_i8ptr258 = bitcast [1 x i64]* %dims257 to i8*
  %elmptr259 = getelementptr [1 x i64], [1 x i64]* %dims257, i64 0, i64 0
  store i64 1, i64* %elmptr259
  %malloccall260 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data261 = bitcast i8* %malloccall260 to [1 x i8*]*
  %data_as_i8ptr262 = bitcast [1 x i8*]* %data261 to i8*
  %elmptr263 = getelementptr [1 x i8*], [1 x i8*]* %data261, i64 0, i64 0
  store i8* %tensor251, i8** %elmptr263
  %dimsptr264 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 2
  store i8* %dims_as_i8ptr258, i8** %dimsptr264
  %dataptr265 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 3
  store i8* %data_as_i8ptr262, i8** %dataptr265
  %rc266 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 4
  store i8 0, i8* %rc266
  %tensor267 = bitcast %tensor_t* %raw_tensor253 to i8*
```

```
  %l1268 = load i8*, i8** %l1
  %access_tensor269 = call i8* @index_get(i8* %l1268, i8* %tensor267)
  %malloccall270 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor271 = bitcast i8* %malloccall270 to %tensor_t*
  %dtype272 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 0
  store i8 0, i8* %dtype272
  %ndims273 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 1
  store i8 0, i8* %ndims273
  %malloccall274 = tail call i8* @malloc(i32 0)
  %dims275 = bitcast i8* %malloccall274 to [0 x i64]*
  %dims_as_i8ptr276 = bitcast [0 x i64]* %dims275 to i8*
  %malloccall277 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data278 = bitcast i8* %malloccall277 to [1 x i32]*
  %data_as_i8ptr279 = bitcast [1 x i32]* %data278 to i8*
  %elmptr280 = getelementptr [1 x i32], [1 x i32]* %data278, i32 0, i32 0
  store i32 1, i32* %elmptr280
  %dimsptr281 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 2
  store i8* %dims_as_i8ptr276, i8** %dimsptr281
  %dataptr282 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 3
  store i8* %data_as_i8ptr279, i8** %dataptr282
  %rc283 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 4
  store i8 0, i8* %rc283
  %tensor284 = bitcast %tensor_t* %raw_tensor271 to i8*
  %tmpOp285 = call i8* @range(i8* %tmpOp236, i8* %access_tensor269, i8* %tensor284)
  %malloccall286 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor287 = bitcast i8* %malloccall286 to %tensor_t*
  %dtype288 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 0
  store i8 0, i8* %dtype288
  %ndims289 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 1
  store i8 0, i8* %ndims289
  %malloccall290 = tail call i8* @malloc(i32 0)
  %dims291 = bitcast i8* %malloccall290 to [0 x i64]*
  %dims_as_i8ptr292 = bitcast [0 x i64]* %dims291 to i8*
  %malloccall293 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data294 = bitcast i8* %malloccall293 to [1 x i32]*
  %data_as_i8ptr295 = bitcast [1 x i32]* %data294 to i8*
  %elmptr296 = getelementptr [1 x i32], [1 x i32]* %data294, i32 0, i32 0
  store i32 1, i32* %elmptr296
  %dimsptr297 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 2
  store i8* %dims_as_i8ptr292, i8** %dimsptr297
```

```
  %dataptr298 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 3
  store i8* %data_as_i8ptr295, i8** %dataptr298
  %rc299 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 4
  store i8 0, i8* %rc299
  %tensor300 = bitcast %tensor_t* %raw_tensor287 to i8*
  %malloccall301 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor302 = bitcast i8* %malloccall301 to %tensor_t*
  %dtype303 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 0
  store i8 3, i8* %dtype303
  %ndims304 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 1
  store i8 1, i8* %ndims304
  %malloccall305 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims306 = bitcast i8* %malloccall305 to [1 x i64]*
  %dims_as_i8ptr307 = bitcast [1 x i64]* %dims306 to i8*
  %elmptr308 = getelementptr [1 x i64], [1 x i64]* %dims306, i64 0, i64 0
  store i64 1, i64* %elmptr308
  %malloccall309 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data310 = bitcast i8* %malloccall309 to [1 x i8*]*
  %data_as_i8ptr311 = bitcast [1 x i8*]* %data310 to i8*
  %elmptr312 = getelementptr [1 x i8*], [1 x i8*]* %data310, i64 0, i64 0
  store i8* %tensor300, i8** %elmptr312
  %dimsptr313 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 2
  store i8* %dims_as_i8ptr307, i8** %dimsptr313
  %dataptr314 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 3
  store i8* %data_as_i8ptr311, i8** %dataptr314
  %rc315 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 4
  store i8 0, i8* %rc315
  %tensor316 = bitcast %tensor_t* %raw_tensor302 to i8*
  %l1317 = load i8*, i8** %l1
  %access_tensor318 = call i8* @index_get(i8* %l1317, i8* %tensor316)
  %malloccall319 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor320 = bitcast i8* %malloccall319 to %tensor_t*
  %dtype321 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 0
  store i8 0, i8* %dtype321
  %ndims322 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 1
  store i8 0, i8* %ndims322
  %malloccall323 = tail call i8* @malloc(i32 0)
  %dims324 = bitcast i8* %malloccall323 to [0 x i64]*
  %dims_as_i8ptr325 = bitcast [0 x i64]* %dims324 to i8*
```

```
  %malloccall326 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data327 = bitcast i8* %malloccall326 to [1 x i32]*
  %data_as_i8ptr328 = bitcast [1 x i32]* %data327 to i8*
  %elmptr329 = getelementptr [1 x i32], [1 x i32]* %data327, i32 0, i32 0
  store i32 2, i32* %elmptr329
  %dimsptr330 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 2
  store i8* %dims_as_i8ptr325, i8** %dimsptr330
  %dataptr331 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 3
  store i8* %data_as_i8ptr328, i8** %dataptr331
  %rc332 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 4
  store i8 0, i8* %rc332
  %tensor333 = bitcast %tensor_t* %raw_tensor320 to i8*
  %tmpOp334 = call i8* @floordivide(i8* %access_tensor318, i8* %tensor333)
  %malloccall335 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor336 = bitcast i8* %malloccall335 to %tensor_t*
  %dtype337 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor336, i32 0, i32 0
  store i8 0, i8* %dtype337
  %ndims338 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor336, i32 0, i32 1
  store i8 0, i8* %ndims338
  %malloccall339 = tail call i8* @malloc(i32 0)
  %dims340 = bitcast i8* %malloccall339 to [0 x i64]*
  %dims_as_i8ptr341 = bitcast [0 x i64]* %dims340 to i8*
  %malloccall342 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data343 = bitcast i8* %malloccall342 to [1 x i32]*
  %data_as_i8ptr344 = bitcast [1 x i32]* %data343 to i8*
  %elmptr345 = getelementptr [1 x i32], [1 x i32]* %data343, i32 0, i32 0
  store i32 1, i32* %elmptr345
  %dimsptr346 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor336, i32 0, i32 2
  store i8* %dims_as_i8ptr341, i8** %dimsptr346
  %dataptr347 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor336, i32 0, i32 3
  store i8* %data_as_i8ptr344, i8** %dataptr347
  %rc348 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor336, i32 0, i32 4
  store i8 0, i8* %rc348
  %tensor349 = bitcast %tensor_t* %raw_tensor336 to i8*
  %malloccall350 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor351 = bitcast i8* %malloccall350 to %tensor_t*
  %dtype352 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 0
  store i8 3, i8* %dtype352
  %ndims353 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 1
  store i8 1, i8* %ndims353
```

```
  %malloccall354 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims355 = bitcast i8* %malloccall354 to [1 x i64]*
  %dims_as_i8ptr356 = bitcast [1 x i64]* %dims355 to i8*
  %elmptr357 = getelementptr [1 x i64], [1 x i64]* %dims355, i64 0, i64 0
  store i64 1, i64* %elmptr357
  %malloccall358 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data359 = bitcast i8* %malloccall358 to [1 x i8*]*
  %data_as_i8ptr360 = bitcast [1 x i8*]* %data359 to i8*
  %elmptr361 = getelementptr [1 x i8*], [1 x i8*]* %data359, i64 0, i64 0
  store i8* %tensor349, i8** %elmptr361
  %dimsptr362 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 2
  store i8* %dims_as_i8ptr356, i8** %dimsptr362
  %dataptr363 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 3
  store i8* %data_as_i8ptr360, i8** %dataptr363
  %rc364 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 4
  store i8 0, i8* %rc364
  %tensor365 = bitcast %tensor_t* %raw_tensor351 to i8*
  %l1366 = load i8*, i8** %l1
  %access_tensor367 = call i8* @index_get(i8* %l1366, i8* %tensor365)
  %malloccall368 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor369 = bitcast i8* %malloccall368 to %tensor_t*
  %dtype370 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 0
  store i8 0, i8* %dtype370
  %ndims371 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 1
  store i8 0, i8* %ndims371
  %malloccall372 = tail call i8* @malloc(i32 0)
  %dims373 = bitcast i8* %malloccall372 to [0 x i64]*
  %dims_as_i8ptr374 = bitcast [0 x i64]* %dims373 to i8*
  %malloccall375 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data376 = bitcast i8* %malloccall375 to [1 x i32]*
  %data_as_i8ptr377 = bitcast [1 x i32]* %data376 to i8*
  %elmptr378 = getelementptr [1 x i32], [1 x i32]* %data376, i32 0, i32 0
  store i32 1, i32* %elmptr378
  %dimsptr379 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 2
  store i8* %dims_as_i8ptr374, i8** %dimsptr379
  %dataptr380 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 3
  store i8* %data_as_i8ptr377, i8** %dataptr380
  %rc381 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 4
  store i8 0, i8* %rc381
  %tensor382 = bitcast %tensor_t* %raw_tensor369 to i8*
```

```
  %tmpOp383 = call i8* @range(i8* %tmpOp334, i8* %access_tensor367, i8* %tensor382)
  %malloccall384 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor385 = bitcast i8* %malloccall384 to %tensor_t*
  %dtype386 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor385, i32 0, i32 0
  store i8 3, i8* %dtype386
  %ndims387 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor385, i32 0, i32 1
  store i8 1, i8* %ndims387
  %malloccall388 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims389 = bitcast i8* %malloccall388 to [1 x i64]*
  %dims_as_i8ptr390 = bitcast [1 x i64]* %dims389 to i8*
  %elmptr391 = getelementptr [1 x i64], [1 x i64]* %dims389, i64 0, i64 0
  store i64 2, i64* %elmptr391
  %malloccall392 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data393 = bitcast i8* %malloccall392 to [2 x i8*]*
  %data_as_i8ptr394 = bitcast [2 x i8*]* %data393 to i8*
  %elmptr395 = getelementptr [2 x i8*], [2 x i8*]* %data393, i64 0, i64 0
  store i8* %tmpOp285, i8** %elmptr395
  %elmptr396 = getelementptr [2 x i8*], [2 x i8*]* %data393, i64 0, i64 1
  store i8* %tmpOp383, i8** %elmptr396
  %dimsptr397 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor385, i32 0, i32 2
  store i8* %dims_as_i8ptr390, i8** %dimsptr397
  %dataptr398 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor385, i32 0, i32 3
  store i8* %data_as_i8ptr394, i8** %dataptr398
  %rc399 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor385, i32 0, i32 4
  store i8 0, i8* %rc399
  %tensor400 = bitcast %tensor_t* %raw_tensor385 to i8*
  %x401 = load i8*, i8** %x
  %access_tensor402 = call i8* @index_get(i8* %x401, i8* %tensor400)
  %x22 = alloca i8*
  store i8* null, i8** %x22
  %lhsptr403 = load i8*, i8** %x22
  call void @increase_rc(i8* %access_tensor402)
  call void @decrease_rc(i8* %lhsptr403)
  store i8* %access_tensor402, i8** %x22
  %malloccall404 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor405 = bitcast i8* %malloccall404 to %tensor_t*
  %dtype406 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor405, i32 0, i32 0
  store i8 0, i8* %dtype406
  %ndims407 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor405, i32 0, i32 1
  store i8 0, i8* %ndims407
```

```
  %malloccall408 = tail call i8* @malloc(i32 0)
  %dims409 = bitcast i8* %malloccall408 to [0 x i64]*
  %dims_as_i8ptr410 = bitcast [0 x i64]* %dims409 to i8*
  %malloccall411 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data412 = bitcast i8* %malloccall411 to [1 x i32]*
  %data_as_i8ptr413 = bitcast [1 x i32]* %data412 to i8*
  %elmptr414 = getelementptr [1 x i32], [1 x i32]* %data412, i32 0, i32 0
  store i32 0, i32* %elmptr414
  %dimsptr415 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor405, i32 0, i32 2
  store i8* %dims_as_i8ptr410, i8** %dimsptr415
  %dataptr416 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor405, i32 0, i32 3
  store i8* %data_as_i8ptr413, i8** %dataptr416
  %rc417 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor405, i32 0, i32 4
  store i8 0, i8* %rc417
  %tensor418 = bitcast %tensor_t* %raw_tensor405 to i8*
  %malloccall419 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor420 = bitcast i8* %malloccall419 to %tensor_t*
  %dtype421 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 0
  store i8 0, i8* %dtype421
  %ndims422 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 1
  store i8 0, i8* %ndims422
  %malloccall423 = tail call i8* @malloc(i32 0)
  %dims424 = bitcast i8* %malloccall423 to [0 x i64]*
  %dims_as_i8ptr425 = bitcast [0 x i64]* %dims424 to i8*
  %malloccall426 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data427 = bitcast i8* %malloccall426 to [1 x i32]*
  %data_as_i8ptr428 = bitcast [1 x i32]* %data427 to i8*
  %elmptr429 = getelementptr [1 x i32], [1 x i32]* %data427, i32 0, i32 0
  store i32 0, i32* %elmptr429
  %dimsptr430 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 2
  store i8* %dims_as_i8ptr425, i8** %dimsptr430
  %dataptr431 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 3
  store i8* %data_as_i8ptr428, i8** %dataptr431
  %rc432 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 4
  store i8 0, i8* %rc432
  %tensor433 = bitcast %tensor_t* %raw_tensor420 to i8*
  %malloccall434 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor435 = bitcast i8* %malloccall434 to %tensor_t*
  %dtype436 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 0
  store i8 3, i8* %dtype436
```

```
  %ndims437 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 1
  store i8 1, i8* %ndims437
  %malloccall438 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims439 = bitcast i8* %malloccall438 to [1 x i64]*
  %dims_as_i8ptr440 = bitcast [1 x i64]* %dims439 to i8*
  %elmptr441 = getelementptr [1 x i64], [1 x i64]* %dims439, i64 0, i64 0
  store i64 1, i64* %elmptr441
  %malloccall442 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data443 = bitcast i8* %malloccall442 to [1 x i8*]*
  %data_as_i8ptr444 = bitcast [1 x i8*]* %data443 to i8*
  %elmptr445 = getelementptr [1 x i8*], [1 x i8*]* %data443, i64 0, i64 0
  store i8* %tensor433, i8** %elmptr445
  %dimsptr446 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 2
  store i8* %dims_as_i8ptr440, i8** %dimsptr446
  %dataptr447 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 3
  store i8* %data_as_i8ptr444, i8** %dataptr447
  %rc448 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 4
  store i8 0, i8* %rc448
  %tensor449 = bitcast %tensor_t* %raw_tensor435 to i8*
  %l2450 = load i8*, i8** %l2
  %access_tensor451 = call i8* @index_get(i8* %l2450, i8* %tensor449)
  %malloccall452 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor453 = bitcast i8* %malloccall452 to %tensor_t*
  %dtype454 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor453, i32 0, i32 0
  store i8 0, i8* %dtype454
  %ndims455 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor453, i32 0, i32 1
  store i8 0, i8* %ndims455
  %malloccall456 = tail call i8* @malloc(i32 0)
  %dims457 = bitcast i8* %malloccall456 to [0 x i64]*
  %dims_as_i8ptr458 = bitcast [0 x i64]* %dims457 to i8*
  %malloccall459 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data460 = bitcast i8* %malloccall459 to [1 x i32]*
  %data_as_i8ptr461 = bitcast [1 x i32]* %data460 to i8*
  %elmptr462 = getelementptr [1 x i32], [1 x i32]* %data460, i32 0, i32 0
  store i32 2, i32* %elmptr462
  %dimsptr463 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor453, i32 0, i32 2
  store i8* %dims_as_i8ptr458, i8** %dimsptr463
  %dataptr464 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor453, i32 0, i32 3
  store i8* %data_as_i8ptr461, i8** %dataptr464
  %rc465 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor453, i32 0, i32 4
```

```
  store i8 0, i8* %rc465
  %tensor466 = bitcast %tensor_t* %raw_tensor453 to i8*
  %tmpOp467 = call i8* @floordivide(i8* %access_tensor451, i8* %tensor466)
  %malloccall468 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor469 = bitcast i8* %malloccall468 to %tensor_t*
  %dtype470 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 0
  store i8 0, i8* %dtype470
  %ndims471 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 1
  store i8 0, i8* %ndims471
  %malloccall472 = tail call i8* @malloc(i32 0)
  %dims473 = bitcast i8* %malloccall472 to [0 x i64]*
  %dims_as_i8ptr474 = bitcast [0 x i64]* %dims473 to i8*
  %malloccall475 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data476 = bitcast i8* %malloccall475 to [1 x i32]*
  %data_as_i8ptr477 = bitcast [1 x i32]* %data476 to i8*
  %elmptr478 = getelementptr [1 x i32], [1 x i32]* %data476, i32 0, i32 0
  store i32 1, i32* %elmptr478
  %dimsptr479 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 2
  store i8* %dims_as_i8ptr474, i8** %dimsptr479
  %dataptr480 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 3
  store i8* %data_as_i8ptr477, i8** %dataptr480
  %rc481 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 4
  store i8 0, i8* %rc481
  %tensor482 = bitcast %tensor_t* %raw_tensor469 to i8*
  %tmpOp483 = call i8* @range(i8* %tensor418, i8* %tmpOp467, i8* %tensor482)
  %malloccall484 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor485 = bitcast i8* %malloccall484 to %tensor_t*
  %dtype486 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 0
  store i8 0, i8* %dtype486
  %ndims487 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 1
  store i8 0, i8* %ndims487
  %malloccall488 = tail call i8* @malloc(i32 0)
  %dims489 = bitcast i8* %malloccall488 to [0 x i64]*
  %dims_as_i8ptr490 = bitcast [0 x i64]* %dims489 to i8*
  %malloccall491 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data492 = bitcast i8* %malloccall491 to [1 x i32]*
  %data_as_i8ptr493 = bitcast [1 x i32]* %data492 to i8*
  %elmptr494 = getelementptr [1 x i32], [1 x i32]* %data492, i32 0, i32 0
  store i32 0, i32* %elmptr494
  %dimsptr495 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 2
```

```
  store i8* %dims_as_i8ptr490, i8** %dimsptr495
  %dataptr496 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 3
  store i8* %data_as_i8ptr493, i8** %dataptr496
  %rc497 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 4
  store i8 0, i8* %rc497
  %tensor498 = bitcast %tensor_t* %raw_tensor485 to i8*
  %malloccall499 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor500 = bitcast i8* %malloccall499 to %tensor_t*
  %dtype501 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 0
  store i8 0, i8* %dtype501
  %ndims502 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 1
  store i8 0, i8* %ndims502
  %malloccall503 = tail call i8* @malloc(i32 0)
  %dims504 = bitcast i8* %malloccall503 to [0 x i64]*
  %dims_as_i8ptr505 = bitcast [0 x i64]* %dims504 to i8*
  %malloccall506 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data507 = bitcast i8* %malloccall506 to [1 x i32]*
  %data_as_i8ptr508 = bitcast [1 x i32]* %data507 to i8*
  %elmptr509 = getelementptr [1 x i32], [1 x i32]* %data507, i32 0, i32 0
  store i32 1, i32* %elmptr509
  %dimsptr510 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 2
  store i8* %dims_as_i8ptr505, i8** %dimsptr510
  %dataptr511 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 3
  store i8* %data_as_i8ptr508, i8** %dataptr511
  %rc512 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 4
  store i8 0, i8* %rc512
  %tensor513 = bitcast %tensor_t* %raw_tensor500 to i8*
  %malloccall514 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor515 = bitcast i8* %malloccall514 to %tensor_t*
  %dtype516 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 0
  store i8 3, i8* %dtype516
  %ndims517 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 1
  store i8 1, i8* %ndims517
  %malloccall518 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims519 = bitcast i8* %malloccall518 to [1 x i64]*
  %dims_as_i8ptr520 = bitcast [1 x i64]* %dims519 to i8*
  %elmptr521 = getelementptr [1 x i64], [1 x i64]* %dims519, i64 0, i64 0
  store i64 1, i64* %elmptr521
  %malloccall522 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
```

```
  %data523 = bitcast i8* %malloccall522 to [1 x i8*]*
  %data_as_i8ptr524 = bitcast [1 x i8*]* %data523 to i8*
  %elmptr525 = getelementptr [1 x i8*], [1 x i8*]* %data523, i64 0, i64 0
  store i8* %tensor513, i8** %elmptr525
  %dimsptr526 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 2
  store i8* %dims_as_i8ptr520, i8** %dimsptr526
  %dataptr527 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 3
  store i8* %data_as_i8ptr524, i8** %dataptr527
  %rc528 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 4
  store i8 0, i8* %rc528
  %tensor529 = bitcast %tensor_t* %raw_tensor515 to i8*
  %l2530 = load i8*, i8** %l2
  %access_tensor531 = call i8* @index_get(i8* %l2530, i8* %tensor529)
  %malloccall532 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor533 = bitcast i8* %malloccall532 to %tensor_t*
  %dtype534 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 0
  store i8 0, i8* %dtype534
  %ndims535 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 1
  store i8 0, i8* %ndims535
  %malloccall536 = tail call i8* @malloc(i32 0)
  %dims537 = bitcast i8* %malloccall536 to [0 x i64]*
  %dims_as_i8ptr538 = bitcast [0 x i64]* %dims537 to i8*
  %malloccall539 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data540 = bitcast i8* %malloccall539 to [1 x i32]*
  %data_as_i8ptr541 = bitcast [1 x i32]* %data540 to i8*
  %elmptr542 = getelementptr [1 x i32], [1 x i32]* %data540, i32 0, i32 0
  store i32 2, i32* %elmptr542
  %dimsptr543 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 2
  store i8* %dims_as_i8ptr538, i8** %dimsptr543
  %dataptr544 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 3
  store i8* %data_as_i8ptr541, i8** %dataptr544
  %rc545 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 4
  store i8 0, i8* %rc545
  %tensor546 = bitcast %tensor_t* %raw_tensor533 to i8*
  %tmpOp547 = call i8* @floordivide(i8* %access_tensor531, i8* %tensor546)
  %malloccall548 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor549 = bitcast i8* %malloccall548 to %tensor_t*
  %dtype550 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 0
  store i8 0, i8* %dtype550
  %ndims551 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 1
  store i8 0, i8* %ndims551
```

```
  %malloccall552 = tail call i8* @malloc(i32 0)
  %dims553 = bitcast i8* %malloccall552 to [0 x i64]*
  %dims_as_i8ptr554 = bitcast [0 x i64]* %dims553 to i8*
  %malloccall555 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data556 = bitcast i8* %malloccall555 to [1 x i32]*
  %data_as_i8ptr557 = bitcast [1 x i32]* %data556 to i8*
  %elmptr558 = getelementptr [1 x i32], [1 x i32]* %data556, i32 0, i32 0
  store i32 1, i32* %elmptr558
  %dimsptr559 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 2
  store i8* %dims_as_i8ptr554, i8** %dimsptr559
  %dataptr560 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 3
  store i8* %data_as_i8ptr557, i8** %dataptr560
  %rc561 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 4
  store i8 0, i8* %rc561
  %tensor562 = bitcast %tensor_t* %raw_tensor549 to i8*
  %tmpOp563 = call i8* @range(i8* %tensor498, i8* %tmpOp547, i8* %tensor562)
  %malloccall564 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor565 = bitcast i8* %malloccall564 to %tensor_t*
  %dtype566 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 0
  store i8 3, i8* %dtype566
  %ndims567 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 1
  store i8 1, i8* %ndims567
  %malloccall568 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims569 = bitcast i8* %malloccall568 to [1 x i64]*
  %dims_as_i8ptr570 = bitcast [1 x i64]* %dims569 to i8*
  %elmptr571 = getelementptr [1 x i64], [1 x i64]* %dims569, i64 0, i64 0
  store i64 2, i64* %elmptr571
  %malloccall572 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data573 = bitcast i8* %malloccall572 to [2 x i8*]*
  %data_as_i8ptr574 = bitcast [2 x i8*]* %data573 to i8*
  %elmptr575 = getelementptr [2 x i8*], [2 x i8*]* %data573, i64 0, i64 0
  store i8* %tmpOp483, i8** %elmptr575
  %elmptr576 = getelementptr [2 x i8*], [2 x i8*]* %data573, i64 0, i64 1
  store i8* %tmpOp563, i8** %elmptr576
  %dimsptr577 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 2
  store i8* %dims_as_i8ptr570, i8** %dimsptr577
  %dataptr578 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 3
  store i8* %data_as_i8ptr574, i8** %dataptr578
  %rc579 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 4
  store i8 0, i8* %rc579
```

```
  %tensor580 = bitcast %tensor_t* %raw_tensor565 to i8*
  %y581 = load i8*, i8** %y
  %access_tensor582 = call i8* @index_get(i8* %y581, i8* %tensor580)
  %y11 = alloca i8*
  store i8* null, i8** %y11
  %lhsptr583 = load i8*, i8** %y11
  call void @increase_rc(i8* %access_tensor582)
  call void @decrease_rc(i8* %lhsptr583)
  store i8* %access_tensor582, i8** %y11
  %x21584 = load i8*, i8** %x21
  %x22585 = load i8*, i8** %x22
  %tmpOp586 = call i8* @add(i8* %x21584, i8* %x22585)
  %y11587 = load i8*, i8** %y11
  %tmpOp588 = call i8* @mult(i8* %tmpOp586, i8* %y11587)
  ret i8* %tmpOp588
}

define i8* @StrassenMULf3(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
  %l1 = alloca i8*
  store i8* null, i8** %l1
  %lhsptr = load i8*, i8** %l1
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l1
  %y2 = load i8*, i8** %y
  %shape3 = call i8* @shape(i8* %y2)
  %l2 = alloca i8*
  store i8* null, i8** %l2
  %lhsptr4 = load i8*, i8** %l2
  call void @increase_rc(i8* %shape3)
  call void @decrease_rc(i8* %lhsptr4)
  store i8* %shape3, i8** %l2
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
```

```
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall5 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall5 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall6 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall6 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor8 = bitcast i8* %malloccall7 to %tensor_t*
  %dtype9 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 0
  store i8 0, i8* %dtype9
  %ndims10 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 1
  store i8 0, i8* %ndims10
  %malloccall11 = tail call i8* @malloc(i32 0)
  %dims12 = bitcast i8* %malloccall11 to [0 x i64]*
  %dims_as_i8ptr13 = bitcast [0 x i64]* %dims12 to i8*
  %malloccall14 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data15 = bitcast i8* %malloccall14 to [1 x i32]*
  %data_as_i8ptr16 = bitcast [1 x i32]* %data15 to i8*
  %elmptr17 = getelementptr [1 x i32], [1 x i32]* %data15, i32 0, i32 0
  store i32 0, i32* %elmptr17
  %dimsptr18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 2
  store i8* %dims_as_i8ptr13, i8** %dimsptr18
  %dataptr19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 3
  store i8* %data_as_i8ptr16, i8** %dataptr19
  %rc20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 4
  store i8 0, i8* %rc20
  %tensor21 = bitcast %tensor_t* %raw_tensor8 to i8*
  %malloccall22 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor23 = bitcast i8* %malloccall22 to %tensor_t*
  %dtype24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 0
```

```
  store i8 3, i8* %dtype24
  %ndims25 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 1
  store i8 1, i8* %ndims25
  %malloccall26 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims27 = bitcast i8* %malloccall26 to [1 x i64]*
  %dims_as_i8ptr28 = bitcast [1 x i64]* %dims27 to i8*
  %elmptr29 = getelementptr [1 x i64], [1 x i64]* %dims27, i64 0, i64 0
  store i64 1, i64* %elmptr29
  %malloccall30 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data31 = bitcast i8* %malloccall30 to [1 x i8*]*
  %data_as_i8ptr32 = bitcast [1 x i8*]* %data31 to i8*
  %elmptr33 = getelementptr [1 x i8*], [1 x i8*]* %data31, i64 0, i64 0
  store i8* %tensor21, i8** %elmptr33
  %dimsptr34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 2
  store i8* %dims_as_i8ptr28, i8** %dimsptr34
  %dataptr35 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 3
  store i8* %data_as_i8ptr32, i8** %dataptr35
  %rc36 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 4
  store i8 0, i8* %rc36
  %tensor37 = bitcast %tensor_t* %raw_tensor23 to i8*
  %l138 = load i8*, i8** %l1
  %access_tensor = call i8* @index_get(i8* %l138, i8* %tensor37)
  %malloccall39 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor40 = bitcast i8* %malloccall39 to %tensor_t*
  %dtype41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 0
  store i8 0, i8* %dtype41
  %ndims42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 1
  store i8 0, i8* %ndims42
  %malloccall43 = tail call i8* @malloc(i32 0)
  %dims44 = bitcast i8* %malloccall43 to [0 x i64]*
  %dims_as_i8ptr45 = bitcast [0 x i64]* %dims44 to i8*
  %malloccall46 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data47 = bitcast i8* %malloccall46 to [1 x i32]*
  %data_as_i8ptr48 = bitcast [1 x i32]* %data47 to i8*
  %elmptr49 = getelementptr [1 x i32], [1 x i32]* %data47, i32 0, i32 0
  store i32 2, i32* %elmptr49
  %dimsptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 2
  store i8* %dims_as_i8ptr45, i8** %dimsptr50
  %dataptr51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 3
  store i8* %data_as_i8ptr48, i8** %dataptr51
  %rc52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 4
```

```
  store i8 0, i8* %rc52
  %tensor53 = bitcast %tensor_t* %raw_tensor40 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor53)
  %malloccall54 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor55 = bitcast i8* %malloccall54 to %tensor_t*
  %dtype56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 0
  store i8 0, i8* %dtype56
  %ndims57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 1
  store i8 0, i8* %ndims57
  %malloccall58 = tail call i8* @malloc(i32 0)
  %dims59 = bitcast i8* %malloccall58 to [0 x i64]*
  %dims_as_i8ptr60 = bitcast [0 x i64]* %dims59 to i8*
  %malloccall61 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data62 = bitcast i8* %malloccall61 to [1 x i32]*
  %data_as_i8ptr63 = bitcast [1 x i32]* %data62 to i8*
  %elmptr64 = getelementptr [1 x i32], [1 x i32]* %data62, i32 0, i32 0
  store i32 1, i32* %elmptr64
  %dimsptr65 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 2
  store i8* %dims_as_i8ptr60, i8** %dimsptr65
  %dataptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 3
  store i8* %data_as_i8ptr63, i8** %dataptr66
  %rc67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 4
  store i8 0, i8* %rc67
  %tensor68 = bitcast %tensor_t* %raw_tensor55 to i8*
  %tmpOp69 = call i8* @range(i8* %tensor, i8* %tmpOp, i8* %tensor68)
  %malloccall70 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor71 = bitcast i8* %malloccall70 to %tensor_t*
  %dtype72 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 0
  store i8 0, i8* %dtype72
  %ndims73 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 1
  store i8 0, i8* %ndims73
  %malloccall74 = tail call i8* @malloc(i32 0)
  %dims75 = bitcast i8* %malloccall74 to [0 x i64]*
  %dims_as_i8ptr76 = bitcast [0 x i64]* %dims75 to i8*
  %malloccall77 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data78 = bitcast i8* %malloccall77 to [1 x i32]*
  %data_as_i8ptr79 = bitcast [1 x i32]* %data78 to i8*
  %elmptr80 = getelementptr [1 x i32], [1 x i32]* %data78, i32 0, i32 0
  store i32 0, i32* %elmptr80
  %dimsptr81 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 2
```

```
  store i8* %dims_as_i8ptr76, i8** %dimsptr81
  %dataptr82 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 3
  store i8* %data_as_i8ptr79, i8** %dataptr82
  %rc83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 4
  store i8 0, i8* %rc83
  %tensor84 = bitcast %tensor_t* %raw_tensor71 to i8*
  %malloccall85 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor86 = bitcast i8* %malloccall85 to %tensor_t*
  %dtype87 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 0
  store i8 0, i8* %dtype87
  %ndims88 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 1
  store i8 0, i8* %ndims88
  %malloccall89 = tail call i8* @malloc(i32 0)
  %dims90 = bitcast i8* %malloccall89 to [0 x i64]*
  %dims_as_i8ptr91 = bitcast [0 x i64]* %dims90 to i8*
  %malloccall92 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data93 = bitcast i8* %malloccall92 to [1 x i32]*
  %data_as_i8ptr94 = bitcast [1 x i32]* %data93 to i8*
  %elmptr95 = getelementptr [1 x i32], [1 x i32]* %data93, i32 0, i32 0
  store i32 1, i32* %elmptr95
  %dimsptr96 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 2
  store i8* %dims_as_i8ptr91, i8** %dimsptr96
  %dataptr97 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 3
  store i8* %data_as_i8ptr94, i8** %dataptr97
  %rc98 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 4
  store i8 0, i8* %rc98
  %tensor99 = bitcast %tensor_t* %raw_tensor86 to i8*
  %malloccall100 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor101 = bitcast i8* %malloccall100 to %tensor_t*
  %dtype102 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 0
  store i8 3, i8* %dtype102
  %ndims103 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 1
  store i8 1, i8* %ndims103
  %malloccall104 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims105 = bitcast i8* %malloccall104 to [1 x i64]*
  %dims_as_i8ptr106 = bitcast [1 x i64]* %dims105 to i8*
  %elmptr107 = getelementptr [1 x i64], [1 x i64]* %dims105, i64 0, i64 0
  store i64 1, i64* %elmptr107
  %malloccall108 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
```

```
  %data109 = bitcast i8* %malloccall108 to [1 x i8*]*
  %data_as_i8ptr110 = bitcast [1 x i8*]* %data109 to i8*
  %elmptr111 = getelementptr [1 x i8*], [1 x i8*]* %data109, i64 0, i64 0
  store i8* %tensor99, i8** %elmptr111
  %dimsptr112 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 2
  store i8* %dims_as_i8ptr106, i8** %dimsptr112
  %dataptr113 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 3
  store i8* %data_as_i8ptr110, i8** %dataptr113
  %rc114 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 4
  store i8 0, i8* %rc114
  %tensor115 = bitcast %tensor_t* %raw_tensor101 to i8*
  %l1116 = load i8*, i8** %l1
  %access_tensor117 = call i8* @index_get(i8* %l1116, i8* %tensor115)
  %malloccall118 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor119 = bitcast i8* %malloccall118 to %tensor_t*
  %dtype120 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 0
  store i8 0, i8* %dtype120
  %ndims121 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 1
  store i8 0, i8* %ndims121
  %malloccall122 = tail call i8* @malloc(i32 0)
  %dims123 = bitcast i8* %malloccall122 to [0 x i64]*
  %dims_as_i8ptr124 = bitcast [0 x i64]* %dims123 to i8*
  %malloccall125 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data126 = bitcast i8* %malloccall125 to [1 x i32]*
  %data_as_i8ptr127 = bitcast [1 x i32]* %data126 to i8*
  %elmptr128 = getelementptr [1 x i32], [1 x i32]* %data126, i32 0, i32 0
  store i32 2, i32* %elmptr128
  %dimsptr129 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 2
  store i8* %dims_as_i8ptr124, i8** %dimsptr129
  %dataptr130 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 3
  store i8* %data_as_i8ptr127, i8** %dataptr130
  %rc131 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 4
  store i8 0, i8* %rc131
  %tensor132 = bitcast %tensor_t* %raw_tensor119 to i8*
  %tmpOp133 = call i8* @floordivide(i8* %access_tensor117, i8* %tensor132)
  %malloccall134 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor135 = bitcast i8* %malloccall134 to %tensor_t*
  %dtype136 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 0
  store i8 0, i8* %dtype136
  %ndims137 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 1
  store i8 0, i8* %ndims137
```

```
  %malloccall138 = tail call i8* @malloc(i32 0)
  %dims139 = bitcast i8* %malloccall138 to [0 x i64]*
  %dims_as_i8ptr140 = bitcast [0 x i64]* %dims139 to i8*
  %malloccall141 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data142 = bitcast i8* %malloccall141 to [1 x i32]*
  %data_as_i8ptr143 = bitcast [1 x i32]* %data142 to i8*
  %elmptr144 = getelementptr [1 x i32], [1 x i32]* %data142, i32 0, i32 0
  store i32 1, i32* %elmptr144
  %dimsptr145 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 2
  store i8* %dims_as_i8ptr140, i8** %dimsptr145
  %dataptr146 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 3
  store i8* %data_as_i8ptr143, i8** %dataptr146
  %rc147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 4
  store i8 0, i8* %rc147
  %tensor148 = bitcast %tensor_t* %raw_tensor135 to i8*
  %tmpOp149 = call i8* @range(i8* %tensor84, i8* %tmpOp133, i8* %tensor148)
  %malloccall150 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor151 = bitcast i8* %malloccall150 to %tensor_t*
  %dtype152 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 0
  store i8 3, i8* %dtype152
  %ndims153 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 1
  store i8 1, i8* %ndims153
  %malloccall154 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims155 = bitcast i8* %malloccall154 to [1 x i64]*
  %dims_as_i8ptr156 = bitcast [1 x i64]* %dims155 to i8*
  %elmptr157 = getelementptr [1 x i64], [1 x i64]* %dims155, i64 0, i64 0
  store i64 2, i64* %elmptr157
  %malloccall158 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data159 = bitcast i8* %malloccall158 to [2 x i8*]*
  %data_as_i8ptr160 = bitcast [2 x i8*]* %data159 to i8*
  %elmptr161 = getelementptr [2 x i8*], [2 x i8*]* %data159, i64 0, i64 0
  store i8* %tmpOp69, i8** %elmptr161
  %elmptr162 = getelementptr [2 x i8*], [2 x i8*]* %data159, i64 0, i64 1
  store i8* %tmpOp149, i8** %elmptr162
  %dimsptr163 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 2
  store i8* %dims_as_i8ptr156, i8** %dimsptr163
  %dataptr164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 3
  store i8* %data_as_i8ptr160, i8** %dataptr164
  %rc165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 4
  store i8 0, i8* %rc165
```

```
  %tensor166 = bitcast %tensor_t* %raw_tensor151 to i8*
  %x167 = load i8*, i8** %x
  %access_tensor168 = call i8* @index_get(i8* %x167, i8* %tensor166)
  %x11 = alloca i8*
  store i8* null, i8** %x11
  %lhsptr169 = load i8*, i8** %x11
  call void @increase_rc(i8* %access_tensor168)
  call void @decrease_rc(i8* %lhsptr169)
  store i8* %access_tensor168, i8** %x11
  %malloccall170 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor171 = bitcast i8* %malloccall170 to %tensor_t*
  %dtype172 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 0
  store i8 0, i8* %dtype172
  %ndims173 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 1
  store i8 0, i8* %ndims173
  %malloccall174 = tail call i8* @malloc(i32 0)
  %dims175 = bitcast i8* %malloccall174 to [0 x i64]*
  %dims_as_i8ptr176 = bitcast [0 x i64]* %dims175 to i8*
  %malloccall177 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data178 = bitcast i8* %malloccall177 to [1 x i32]*
  %data_as_i8ptr179 = bitcast [1 x i32]* %data178 to i8*
  %elmptr180 = getelementptr [1 x i32], [1 x i32]* %data178, i32 0, i32 0
  store i32 0, i32* %elmptr180
  %dimsptr181 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 2
  store i8* %dims_as_i8ptr176, i8** %dimsptr181
  %dataptr182 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 3
  store i8* %data_as_i8ptr179, i8** %dataptr182
  %rc183 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 4
  store i8 0, i8* %rc183
  %tensor184 = bitcast %tensor_t* %raw_tensor171 to i8*
  %malloccall185 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor186 = bitcast i8* %malloccall185 to %tensor_t*
  %dtype187 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 0
  store i8 0, i8* %dtype187
  %ndims188 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 1
  store i8 0, i8* %ndims188
  %malloccall189 = tail call i8* @malloc(i32 0)
  %dims190 = bitcast i8* %malloccall189 to [0 x i64]*
  %dims_as_i8ptr191 = bitcast [0 x i64]* %dims190 to i8*
  %malloccall192 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
```

```
  %data193 = bitcast i8* %malloccall192 to [1 x i32]*
  %data_as_i8ptr194 = bitcast [1 x i32]* %data193 to i8*
  %elmptr195 = getelementptr [1 x i32], [1 x i32]* %data193, i32 0, i32 0
  store i32 0, i32* %elmptr195
  %dimsptr196 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 2
  store i8* %dims_as_i8ptr191, i8** %dimsptr196
  %dataptr197 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 3
  store i8* %data_as_i8ptr194, i8** %dataptr197
  %rc198 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 4
  store i8 0, i8* %rc198
  %tensor199 = bitcast %tensor_t* %raw_tensor186 to i8*
  %malloccall200 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor201 = bitcast i8* %malloccall200 to %tensor_t*
  %dtype202 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor201, i32 0, i32 0
  store i8 3, i8* %dtype202
  %ndims203 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor201, i32 0, i32 1
  store i8 1, i8* %ndims203
  %malloccall204 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims205 = bitcast i8* %malloccall204 to [1 x i64]*
  %dims_as_i8ptr206 = bitcast [1 x i64]* %dims205 to i8*
  %elmptr207 = getelementptr [1 x i64], [1 x i64]* %dims205, i64 0, i64 0
  store i64 1, i64* %elmptr207
  %malloccall208 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data209 = bitcast i8* %malloccall208 to [1 x i8*]*
  %data_as_i8ptr210 = bitcast [1 x i8*]* %data209 to i8*
  %elmptr211 = getelementptr [1 x i8*], [1 x i8*]* %data209, i64 0, i64 0
  store i8* %tensor199, i8** %elmptr211
  %dimsptr212 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor201, i32 0, i32 2
  store i8* %dims_as_i8ptr206, i8** %dimsptr212
  %dataptr213 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor201, i32 0, i32 3
  store i8* %data_as_i8ptr210, i8** %dataptr213
  %rc214 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor201, i32 0, i32 4
  store i8 0, i8* %rc214
  %tensor215 = bitcast %tensor_t* %raw_tensor201 to i8*
  %l2216 = load i8*, i8** %l2
  %access_tensor217 = call i8* @index_get(i8* %l2216, i8* %tensor215)
  %malloccall218 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor219 = bitcast i8* %malloccall218 to %tensor_t*
  %dtype220 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 0
  store i8 0, i8* %dtype220
```

```
  %ndims221 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 1
  store i8 0, i8* %ndims221
  %malloccall222 = tail call i8* @malloc(i32 0)
  %dims223 = bitcast i8* %malloccall222 to [0 x i64]*
  %dims_as_i8ptr224 = bitcast [0 x i64]* %dims223 to i8*
  %malloccall225 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data226 = bitcast i8* %malloccall225 to [1 x i32]*
  %data_as_i8ptr227 = bitcast [1 x i32]* %data226 to i8*
  %elmptr228 = getelementptr [1 x i32], [1 x i32]* %data226, i32 0, i32 0
  store i32 2, i32* %elmptr228
  %dimsptr229 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 2
  store i8* %dims_as_i8ptr224, i8** %dimsptr229
  %dataptr230 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 3
  store i8* %data_as_i8ptr227, i8** %dataptr230
  %rc231 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 4
  store i8 0, i8* %rc231
  %tensor232 = bitcast %tensor_t* %raw_tensor219 to i8*
  %tmpOp233 = call i8* @floordivide(i8* %access_tensor217, i8* %tensor232)
  %malloccall234 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor235 = bitcast i8* %malloccall234 to %tensor_t*
  %dtype236 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 0
  store i8 0, i8* %dtype236
  %ndims237 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 1
  store i8 0, i8* %ndims237
  %malloccall238 = tail call i8* @malloc(i32 0)
  %dims239 = bitcast i8* %malloccall238 to [0 x i64]*
  %dims_as_i8ptr240 = bitcast [0 x i64]* %dims239 to i8*
  %malloccall241 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data242 = bitcast i8* %malloccall241 to [1 x i32]*
  %data_as_i8ptr243 = bitcast [1 x i32]* %data242 to i8*
  %elmptr244 = getelementptr [1 x i32], [1 x i32]* %data242, i32 0, i32 0
  store i32 1, i32* %elmptr244
  %dimsptr245 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 2
  store i8* %dims_as_i8ptr240, i8** %dimsptr245
  %dataptr246 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 3
  store i8* %data_as_i8ptr243, i8** %dataptr246
  %rc247 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 4
  store i8 0, i8* %rc247
  %tensor248 = bitcast %tensor_t* %raw_tensor235 to i8*
  %tmpOp249 = call i8* @range(i8* %tensor184, i8* %tmpOp233, i8* %tensor248)
```

```
  %malloccall250 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor251 = bitcast i8* %malloccall250 to %tensor_t*
  %dtype252 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor251, i32 0, i32 0
  store i8 0, i8* %dtype252
  %ndims253 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor251, i32 0, i32 1
  store i8 0, i8* %ndims253
  %malloccall254 = tail call i8* @malloc(i32 0)
  %dims255 = bitcast i8* %malloccall254 to [0 x i64]*
  %dims_as_i8ptr256 = bitcast [0 x i64]* %dims255 to i8*
  %malloccall257 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data258 = bitcast i8* %malloccall257 to [1 x i32]*
  %data_as_i8ptr259 = bitcast [1 x i32]* %data258 to i8*
  %elmptr260 = getelementptr [1 x i32], [1 x i32]* %data258, i32 0, i32 0
  store i32 1, i32* %elmptr260
  %dimsptr261 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor251, i32 0, i32 2
  store i8* %dims_as_i8ptr256, i8** %dimsptr261
  %dataptr262 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor251, i32 0, i32 3
  store i8* %data_as_i8ptr259, i8** %dataptr262
  %rc263 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor251, i32 0, i32 4
  store i8 0, i8* %rc263
  %tensor264 = bitcast %tensor_t* %raw_tensor251 to i8*
  %malloccall265 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor266 = bitcast i8* %malloccall265 to %tensor_t*
  %dtype267 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor266, i32 0, i32 0
  store i8 3, i8* %dtype267
  %ndims268 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor266, i32 0, i32 1
  store i8 1, i8* %ndims268
  %malloccall269 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims270 = bitcast i8* %malloccall269 to [1 x i64]*
  %dims_as_i8ptr271 = bitcast [1 x i64]* %dims270 to i8*
  %elmptr272 = getelementptr [1 x i64], [1 x i64]* %dims270, i64 0, i64 0
  store i64 1, i64* %elmptr272
  %malloccall273 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data274 = bitcast i8* %malloccall273 to [1 x i8*]*
  %data_as_i8ptr275 = bitcast [1 x i8*]* %data274 to i8*
  %elmptr276 = getelementptr [1 x i8*], [1 x i8*]* %data274, i64 0, i64 0
  store i8* %tensor264, i8** %elmptr276
  %dimsptr277 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor266, i32 0, i32 2
  store i8* %dims_as_i8ptr271, i8** %dimsptr277
```

```
  %dataptr278 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor266, i32 0, i32 3
  store i8* %data_as_i8ptr275, i8** %dataptr278
  %rc279 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor266, i32 0, i32 4
  store i8 0, i8* %rc279
  %tensor280 = bitcast %tensor_t* %raw_tensor266 to i8*
  %l2281 = load i8*, i8** %l2
  %access_tensor282 = call i8* @index_get(i8* %l2281, i8* %tensor280)
  %malloccall283 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor284 = bitcast i8* %malloccall283 to %tensor_t*
  %dtype285 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 0
  store i8 0, i8* %dtype285
  %ndims286 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 1
  store i8 0, i8* %ndims286
  %malloccall287 = tail call i8* @malloc(i32 0)
  %dims288 = bitcast i8* %malloccall287 to [0 x i64]*
  %dims_as_i8ptr289 = bitcast [0 x i64]* %dims288 to i8*
  %malloccall290 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data291 = bitcast i8* %malloccall290 to [1 x i32]*
  %data_as_i8ptr292 = bitcast [1 x i32]* %data291 to i8*
  %elmptr293 = getelementptr [1 x i32], [1 x i32]* %data291, i32 0, i32 0
  store i32 2, i32* %elmptr293
  %dimsptr294 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 2
  store i8* %dims_as_i8ptr289, i8** %dimsptr294
  %dataptr295 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 3
  store i8* %data_as_i8ptr292, i8** %dataptr295
  %rc296 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 4
  store i8 0, i8* %rc296
  %tensor297 = bitcast %tensor_t* %raw_tensor284 to i8*
  %tmpOp298 = call i8* @floordivide(i8* %access_tensor282, i8* %tensor297)
  %malloccall299 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor300 = bitcast i8* %malloccall299 to %tensor_t*
  %dtype301 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor300, i32 0, i32 0
  store i8 0, i8* %dtype301
  %ndims302 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor300, i32 0, i32 1
  store i8 0, i8* %ndims302
  %malloccall303 = tail call i8* @malloc(i32 0)
  %dims304 = bitcast i8* %malloccall303 to [0 x i64]*
  %dims_as_i8ptr305 = bitcast [0 x i64]* %dims304 to i8*
  %malloccall306 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data307 = bitcast i8* %malloccall306 to [1 x i32]*
```

```
  %data_as_i8ptr308 = bitcast [1 x i32]* %data307 to i8*
  %elmptr309 = getelementptr [1 x i32], [1 x i32]* %data307, i32 0, i32 0
  store i32 1, i32* %elmptr309
  %dimsptr310 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor300, i32 0, i32 2
  store i8* %dims_as_i8ptr305, i8** %dimsptr310
  %dataptr311 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor300, i32 0, i32 3
  store i8* %data_as_i8ptr308, i8** %dataptr311
  %rc312 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor300, i32 0, i32 4
  store i8 0, i8* %rc312
  %tensor313 = bitcast %tensor_t* %raw_tensor300 to i8*
  %malloccall314 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor315 = bitcast i8* %malloccall314 to %tensor_t*
  %dtype316 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor315, i32 0, i32 0
  store i8 3, i8* %dtype316
  %ndims317 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor315, i32 0, i32 1
  store i8 1, i8* %ndims317
  %malloccall318 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims319 = bitcast i8* %malloccall318 to [1 x i64]*
  %dims_as_i8ptr320 = bitcast [1 x i64]* %dims319 to i8*
  %elmptr321 = getelementptr [1 x i64], [1 x i64]* %dims319, i64 0, i64 0
  store i64 1, i64* %elmptr321
  %malloccall322 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data323 = bitcast i8* %malloccall322 to [1 x i8*]*
  %data_as_i8ptr324 = bitcast [1 x i8*]* %data323 to i8*
  %elmptr325 = getelementptr [1 x i8*], [1 x i8*]* %data323, i64 0, i64 0
  store i8* %tensor313, i8** %elmptr325
  %dimsptr326 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor315, i32 0, i32 2
  store i8* %dims_as_i8ptr320, i8** %dimsptr326
  %dataptr327 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor315, i32 0, i32 3
  store i8* %data_as_i8ptr324, i8** %dataptr327
  %rc328 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor315, i32 0, i32 4
  store i8 0, i8* %rc328
  %tensor329 = bitcast %tensor_t* %raw_tensor315 to i8*
  %l2330 = load i8*, i8** %l2
  %access_tensor331 = call i8* @index_get(i8* %l2330, i8* %tensor329)
  %malloccall332 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor333 = bitcast i8* %malloccall332 to %tensor_t*
  %dtype334 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 0
  store i8 0, i8* %dtype334
  %ndims335 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 1
```

```
  store i8 0, i8* %ndims335
  %malloccall336 = tail call i8* @malloc(i32 0)
  %dims337 = bitcast i8* %malloccall336 to [0 x i64]*
  %dims_as_i8ptr338 = bitcast [0 x i64]* %dims337 to i8*
  %malloccall339 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data340 = bitcast i8* %malloccall339 to [1 x i32]*
  %data_as_i8ptr341 = bitcast [1 x i32]* %data340 to i8*
  %elmptr342 = getelementptr [1 x i32], [1 x i32]* %data340, i32 0, i32 0
  store i32 1, i32* %elmptr342
  %dimsptr343 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 2
  store i8* %dims_as_i8ptr338, i8** %dimsptr343
  %dataptr344 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 3
  store i8* %data_as_i8ptr341, i8** %dataptr344
  %rc345 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 4
  store i8 0, i8* %rc345
  %tensor346 = bitcast %tensor_t* %raw_tensor333 to i8*
  %tmpOp347 = call i8* @range(i8* %tmpOp298, i8* %access_tensor331, i8* %tensor346)
  %malloccall348 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor349 = bitcast i8* %malloccall348 to %tensor_t*
  %dtype350 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 0
  store i8 3, i8* %dtype350
  %ndims351 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 1
  store i8 1, i8* %ndims351
  %malloccall352 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims353 = bitcast i8* %malloccall352 to [1 x i64]*
  %dims_as_i8ptr354 = bitcast [1 x i64]* %dims353 to i8*
  %elmptr355 = getelementptr [1 x i64], [1 x i64]* %dims353, i64 0, i64 0
  store i64 2, i64* %elmptr355
  %malloccall356 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data357 = bitcast i8* %malloccall356 to [2 x i8*]*
  %data_as_i8ptr358 = bitcast [2 x i8*]* %data357 to i8*
  %elmptr359 = getelementptr [2 x i8*], [2 x i8*]* %data357, i64 0, i64 0
  store i8* %tmpOp249, i8** %elmptr359
  %elmptr360 = getelementptr [2 x i8*], [2 x i8*]* %data357, i64 0, i64 1
  store i8* %tmpOp347, i8** %elmptr360
  %dimsptr361 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 2
  store i8* %dims_as_i8ptr354, i8** %dimsptr361
  %dataptr362 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 3
  store i8* %data_as_i8ptr358, i8** %dataptr362
  %rc363 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 4
```

```
  store i8 0, i8* %rc363
  %tensor364 = bitcast %tensor_t* %raw_tensor349 to i8*
  %y365 = load i8*, i8** %y
  %access_tensor366 = call i8* @index_get(i8* %y365, i8* %tensor364)
  %y12 = alloca i8*
  store i8* null, i8** %y12
  %lhsptr367 = load i8*, i8** %y12
  call void @increase_rc(i8* %access_tensor366)
  call void @decrease_rc(i8* %lhsptr367)
  store i8* %access_tensor366, i8** %y12
  %malloccall368 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor369 = bitcast i8* %malloccall368 to %tensor_t*
  %dtype370 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 0
  store i8 0, i8* %dtype370
  %ndims371 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 1
  store i8 0, i8* %ndims371
  %malloccall372 = tail call i8* @malloc(i32 0)
  %dims373 = bitcast i8* %malloccall372 to [0 x i64]*
  %dims_as_i8ptr374 = bitcast [0 x i64]* %dims373 to i8*
  %malloccall375 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data376 = bitcast i8* %malloccall375 to [1 x i32]*
  %data_as_i8ptr377 = bitcast [1 x i32]* %data376 to i8*
  %elmptr378 = getelementptr [1 x i32], [1 x i32]* %data376, i32 0, i32 0
  store i32 0, i32* %elmptr378
  %dimsptr379 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 2
  store i8* %dims_as_i8ptr374, i8** %dimsptr379
  %dataptr380 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 3
  store i8* %data_as_i8ptr377, i8** %dataptr380
  %rc381 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 4
  store i8 0, i8* %rc381
  %tensor382 = bitcast %tensor_t* %raw_tensor369 to i8*
  %malloccall383 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor384 = bitcast i8* %malloccall383 to %tensor_t*
  %dtype385 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 0
  store i8 3, i8* %dtype385
  %ndims386 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 1
  store i8 1, i8* %ndims386
  %malloccall387 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims388 = bitcast i8* %malloccall387 to [1 x i64]*
  %dims_as_i8ptr389 = bitcast [1 x i64]* %dims388 to i8*
```

```
  %elmptr390 = getelementptr [1 x i64], [1 x i64]* %dims388, i64 0, i64 0
  store i64 1, i64* %elmptr390
  %malloccall391 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data392 = bitcast i8* %malloccall391 to [1 x i8*]*
  %data_as_i8ptr393 = bitcast [1 x i8*]* %data392 to i8*
  %elmptr394 = getelementptr [1 x i8*], [1 x i8*]* %data392, i64 0, i64 0
  store i8* %tensor382, i8** %elmptr394
  %dimsptr395 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 2
  store i8* %dims_as_i8ptr389, i8** %dimsptr395
  %dataptr396 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 3
  store i8* %data_as_i8ptr393, i8** %dataptr396
  %rc397 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 4
  store i8 0, i8* %rc397
  %tensor398 = bitcast %tensor_t* %raw_tensor384 to i8*
  %l2399 = load i8*, i8** %l2
  %access_tensor400 = call i8* @index_get(i8* %l2399, i8* %tensor398)
  %malloccall401 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor402 = bitcast i8* %malloccall401 to %tensor_t*
  %dtype403 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 0
  store i8 0, i8* %dtype403
  %ndims404 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 1
  store i8 0, i8* %ndims404
  %malloccall405 = tail call i8* @malloc(i32 0)
  %dims406 = bitcast i8* %malloccall405 to [0 x i64]*
  %dims_as_i8ptr407 = bitcast [0 x i64]* %dims406 to i8*
  %malloccall408 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data409 = bitcast i8* %malloccall408 to [1 x i32]*
  %data_as_i8ptr410 = bitcast [1 x i32]* %data409 to i8*
  %elmptr411 = getelementptr [1 x i32], [1 x i32]* %data409, i32 0, i32 0
  store i32 2, i32* %elmptr411
  %dimsptr412 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 2
  store i8* %dims_as_i8ptr407, i8** %dimsptr412
  %dataptr413 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 3
  store i8* %data_as_i8ptr410, i8** %dataptr413
  %rc414 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 4
  store i8 0, i8* %rc414
  %tensor415 = bitcast %tensor_t* %raw_tensor402 to i8*
  %tmpOp416 = call i8* @floordivide(i8* %access_tensor400, i8* %tensor415)
  %malloccall417 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor418 = bitcast i8* %malloccall417 to %tensor_t*
```

```
  %dtype419 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor418, i32 0, i32 0
  store i8 0, i8* %dtype419
  %ndims420 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor418, i32 0, i32 1
  store i8 0, i8* %ndims420
  %malloccall421 = tail call i8* @malloc(i32 0)
  %dims422 = bitcast i8* %malloccall421 to [0 x i64]*
  %dims_as_i8ptr423 = bitcast [0 x i64]* %dims422 to i8*
  %malloccall424 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data425 = bitcast i8* %malloccall424 to [1 x i32]*
  %data_as_i8ptr426 = bitcast [1 x i32]* %data425 to i8*
  %elmptr427 = getelementptr [1 x i32], [1 x i32]* %data425, i32 0, i32 0
  store i32 0, i32* %elmptr427
  %dimsptr428 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor418, i32 0, i32 2
  store i8* %dims_as_i8ptr423, i8** %dimsptr428
  %dataptr429 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor418, i32 0, i32 3
  store i8* %data_as_i8ptr426, i8** %dataptr429
  %rc430 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor418, i32 0, i32 4
  store i8 0, i8* %rc430
  %tensor431 = bitcast %tensor_t* %raw_tensor418 to i8*
  %malloccall432 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor433 = bitcast i8* %malloccall432 to %tensor_t*
  %dtype434 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 0
  store i8 3, i8* %dtype434
  %ndims435 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 1
  store i8 1, i8* %ndims435
  %malloccall436 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims437 = bitcast i8* %malloccall436 to [1 x i64]*
  %dims_as_i8ptr438 = bitcast [1 x i64]* %dims437 to i8*
  %elmptr439 = getelementptr [1 x i64], [1 x i64]* %dims437, i64 0, i64 0
  store i64 1, i64* %elmptr439
  %malloccall440 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data441 = bitcast i8* %malloccall440 to [1 x i8*]*
  %data_as_i8ptr442 = bitcast [1 x i8*]* %data441 to i8*
  %elmptr443 = getelementptr [1 x i8*], [1 x i8*]* %data441, i64 0, i64 0
  store i8* %tensor431, i8** %elmptr443
  %dimsptr444 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 2
  store i8* %dims_as_i8ptr438, i8** %dimsptr444
  %dataptr445 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 3
  store i8* %data_as_i8ptr442, i8** %dataptr445
  %rc446 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 4
```

```
  store i8 0, i8* %rc446
  %tensor447 = bitcast %tensor_t* %raw_tensor433 to i8*
  %l2448 = load i8*, i8** %l2
  %access_tensor449 = call i8* @index_get(i8* %l2448, i8* %tensor447)
  %malloccall450 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor451 = bitcast i8* %malloccall450 to %tensor_t*
  %dtype452 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 0
  store i8 0, i8* %dtype452
  %ndims453 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 1
  store i8 0, i8* %ndims453
  %malloccall454 = tail call i8* @malloc(i32 0)
  %dims455 = bitcast i8* %malloccall454 to [0 x i64]*
  %dims_as_i8ptr456 = bitcast [0 x i64]* %dims455 to i8*
  %malloccall457 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data458 = bitcast i8* %malloccall457 to [1 x i32]*
  %data_as_i8ptr459 = bitcast [1 x i32]* %data458 to i8*
  %elmptr460 = getelementptr [1 x i32], [1 x i32]* %data458, i32 0, i32 0
  store i32 1, i32* %elmptr460
  %dimsptr461 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 2
  store i8* %dims_as_i8ptr456, i8** %dimsptr461
  %dataptr462 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 3
  store i8* %data_as_i8ptr459, i8** %dataptr462
  %rc463 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 4
  store i8 0, i8* %rc463
  %tensor464 = bitcast %tensor_t* %raw_tensor451 to i8*
  %tmpOp465 = call i8* @range(i8* %tmpOp416, i8* %access_tensor449, i8* %tensor464)
  %malloccall466 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor467 = bitcast i8* %malloccall466 to %tensor_t*
  %dtype468 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 0
  store i8 0, i8* %dtype468
  %ndims469 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 1
  store i8 0, i8* %ndims469
  %malloccall470 = tail call i8* @malloc(i32 0)
  %dims471 = bitcast i8* %malloccall470 to [0 x i64]*
  %dims_as_i8ptr472 = bitcast [0 x i64]* %dims471 to i8*
  %malloccall473 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data474 = bitcast i8* %malloccall473 to [1 x i32]*
  %data_as_i8ptr475 = bitcast [1 x i32]* %data474 to i8*
  %elmptr476 = getelementptr [1 x i32], [1 x i32]* %data474, i32 0, i32 0
  store i32 1, i32* %elmptr476
```

```
  %dimsptr477 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 2
  store i8* %dims_as_i8ptr472, i8** %dimsptr477
  %dataptr478 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 3
  store i8* %data_as_i8ptr475, i8** %dataptr478
  %rc479 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 4
  store i8 0, i8* %rc479
  %tensor480 = bitcast %tensor_t* %raw_tensor467 to i8*
  %malloccall481 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor482 = bitcast i8* %malloccall481 to %tensor_t*
  %dtype483 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 0
  store i8 3, i8* %dtype483
  %ndims484 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 1
  store i8 1, i8* %ndims484
  %malloccall485 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims486 = bitcast i8* %malloccall485 to [1 x i64]*
  %dims_as_i8ptr487 = bitcast [1 x i64]* %dims486 to i8*
  %elmptr488 = getelementptr [1 x i64], [1 x i64]* %dims486, i64 0, i64 0
  store i64 1, i64* %elmptr488
  %malloccall489 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data490 = bitcast i8* %malloccall489 to [1 x i8*]*
  %data_as_i8ptr491 = bitcast [1 x i8*]* %data490 to i8*
  %elmptr492 = getelementptr [1 x i8*], [1 x i8*]* %data490, i64 0, i64 0
  store i8* %tensor480, i8** %elmptr492
  %dimsptr493 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 2
  store i8* %dims_as_i8ptr487, i8** %dimsptr493
  %dataptr494 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 3
  store i8* %data_as_i8ptr491, i8** %dataptr494
  %rc495 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 4
  store i8 0, i8* %rc495
  %tensor496 = bitcast %tensor_t* %raw_tensor482 to i8*
  %l2497 = load i8*, i8** %l2
  %access_tensor498 = call i8* @index_get(i8* %l2497, i8* %tensor496)
  %malloccall499 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor500 = bitcast i8* %malloccall499 to %tensor_t*
  %dtype501 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 0
  store i8 0, i8* %dtype501
  %ndims502 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 1
  store i8 0, i8* %ndims502
  %malloccall503 = tail call i8* @malloc(i32 0)
  %dims504 = bitcast i8* %malloccall503 to [0 x i64]*
```

```
  %dims_as_i8ptr505 = bitcast [0 x i64]* %dims504 to i8*
  %malloccall506 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data507 = bitcast i8* %malloccall506 to [1 x i32]*
  %data_as_i8ptr508 = bitcast [1 x i32]* %data507 to i8*
  %elmptr509 = getelementptr [1 x i32], [1 x i32]* %data507, i32 0, i32 0
  store i32 2, i32* %elmptr509
  %dimsptr510 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 2
  store i8* %dims_as_i8ptr505, i8** %dimsptr510
  %dataptr511 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 3
  store i8* %data_as_i8ptr508, i8** %dataptr511
  %rc512 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 4
  store i8 0, i8* %rc512
  %tensor513 = bitcast %tensor_t* %raw_tensor500 to i8*
  %tmpOp514 = call i8* @floordivide(i8* %access_tensor498, i8* %tensor513)
  %malloccall515 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor516 = bitcast i8* %malloccall515 to %tensor_t*
  %dtype517 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 0
  store i8 0, i8* %dtype517
  %ndims518 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 1
  store i8 0, i8* %ndims518
  %malloccall519 = tail call i8* @malloc(i32 0)
  %dims520 = bitcast i8* %malloccall519 to [0 x i64]*
  %dims_as_i8ptr521 = bitcast [0 x i64]* %dims520 to i8*
  %malloccall522 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data523 = bitcast i8* %malloccall522 to [1 x i32]*
  %data_as_i8ptr524 = bitcast [1 x i32]* %data523 to i8*
  %elmptr525 = getelementptr [1 x i32], [1 x i32]* %data523, i32 0, i32 0
  store i32 1, i32* %elmptr525
  %dimsptr526 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 2
  store i8* %dims_as_i8ptr521, i8** %dimsptr526
  %dataptr527 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 3
  store i8* %data_as_i8ptr524, i8** %dataptr527
  %rc528 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 4
  store i8 0, i8* %rc528
  %tensor529 = bitcast %tensor_t* %raw_tensor516 to i8*
  %malloccall530 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor531 = bitcast i8* %malloccall530 to %tensor_t*
  %dtype532 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 0
  store i8 3, i8* %dtype532
  %ndims533 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 1
```

```
  store i8 1, i8* %ndims533
  %malloccall534 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims535 = bitcast i8* %malloccall534 to [1 x i64]*
  %dims_as_i8ptr536 = bitcast [1 x i64]* %dims535 to i8*
  %elmptr537 = getelementptr [1 x i64], [1 x i64]* %dims535, i64 0, i64 0
  store i64 1, i64* %elmptr537
  %malloccall538 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data539 = bitcast i8* %malloccall538 to [1 x i8*]*
  %data_as_i8ptr540 = bitcast [1 x i8*]* %data539 to i8*
  %elmptr541 = getelementptr [1 x i8*], [1 x i8*]* %data539, i64 0, i64 0
  store i8* %tensor529, i8** %elmptr541
  %dimsptr542 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 2
  store i8* %dims_as_i8ptr536, i8** %dimsptr542
  %dataptr543 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 3
  store i8* %data_as_i8ptr540, i8** %dataptr543
  %rc544 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 4
  store i8 0, i8* %rc544
  %tensor545 = bitcast %tensor_t* %raw_tensor531 to i8*
  %l2546 = load i8*, i8** %l2
  %access_tensor547 = call i8* @index_get(i8* %l2546, i8* %tensor545)
  %malloccall548 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor549 = bitcast i8* %malloccall548 to %tensor_t*
  %dtype550 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 0
  store i8 0, i8* %dtype550
  %ndims551 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 1
  store i8 0, i8* %ndims551
  %malloccall552 = tail call i8* @malloc(i32 0)
  %dims553 = bitcast i8* %malloccall552 to [0 x i64]*
  %dims_as_i8ptr554 = bitcast [0 x i64]* %dims553 to i8*
  %malloccall555 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data556 = bitcast i8* %malloccall555 to [1 x i32]*
  %data_as_i8ptr557 = bitcast [1 x i32]* %data556 to i8*
  %elmptr558 = getelementptr [1 x i32], [1 x i32]* %data556, i32 0, i32 0
  store i32 1, i32* %elmptr558
  %dimsptr559 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 2
  store i8* %dims_as_i8ptr554, i8** %dimsptr559
  %dataptr560 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 3
  store i8* %data_as_i8ptr557, i8** %dataptr560
  %rc561 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 4
  store i8 0, i8* %rc561
```

```
  %tensor562 = bitcast %tensor_t* %raw_tensor549 to i8*
  %tmpOp563 = call i8* @range(i8* %tmpOp514, i8* %access_tensor547, i8* %tensor562)
  %malloccall564 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor565 = bitcast i8* %malloccall564 to %tensor_t*
  %dtype566 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 0
  store i8 3, i8* %dtype566
  %ndims567 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 1
  store i8 1, i8* %ndims567
  %malloccall568 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims569 = bitcast i8* %malloccall568 to [1 x i64]*
  %dims_as_i8ptr570 = bitcast [1 x i64]* %dims569 to i8*
  %elmptr571 = getelementptr [1 x i64], [1 x i64]* %dims569, i64 0, i64 0
  store i64 2, i64* %elmptr571
  %malloccall572 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data573 = bitcast i8* %malloccall572 to [2 x i8*]*
  %data_as_i8ptr574 = bitcast [2 x i8*]* %data573 to i8*
  %elmptr575 = getelementptr [2 x i8*], [2 x i8*]* %data573, i64 0, i64 0
  store i8* %tmpOp465, i8** %elmptr575
  %elmptr576 = getelementptr [2 x i8*], [2 x i8*]* %data573, i64 0, i64 1
  store i8* %tmpOp563, i8** %elmptr576
  %dimsptr577 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 2
  store i8* %dims_as_i8ptr570, i8** %dimsptr577
  %dataptr578 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 3
  store i8* %data_as_i8ptr574, i8** %dataptr578
  %rc579 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 4
  store i8 0, i8* %rc579
  %tensor580 = bitcast %tensor_t* %raw_tensor565 to i8*
  %y581 = load i8*, i8** %y
  %access_tensor582 = call i8* @index_get(i8* %y581, i8* %tensor580)
  %y22 = alloca i8*
  store i8* null, i8** %y22
  %lhsptr583 = load i8*, i8** %y22
  call void @increase_rc(i8* %access_tensor582)
  call void @decrease_rc(i8* %lhsptr583)
  store i8* %access_tensor582, i8** %y22
  %x11584 = load i8*, i8** %x11
  %y12585 = load i8*, i8** %y12
  %y22586 = load i8*, i8** %y22
  %tmpOp587 = call i8* @subtract(i8* %y12585, i8* %y22586)
  %tmpOp588 = call i8* @mult(i8* %x11584, i8* %tmpOp587)
  ret i8* %tmpOp588
```

```
}

define i8* @StrassenMULf4(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
  %l1 = alloca i8*
  store i8* null, i8** %l1
  %lhsptr = load i8*, i8** %l1
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l1
  %y2 = load i8*, i8** %y
  %shape3 = call i8* @shape(i8* %y2)
  %l2 = alloca i8*
  store i8* null, i8** %l2
  %lhsptr4 = load i8*, i8** %l2
  call void @increase_rc(i8* %shape3)
  call void @decrease_rc(i8* %lhsptr4)
  store i8* %shape3, i8** %l2
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall5 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall5 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall6 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall6 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
```

```
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor8 = bitcast i8* %malloccall7 to %tensor_t*
  %dtype9 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 0
  store i8 3, i8* %dtype9
  %ndims10 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 1
  store i8 1, i8* %ndims10
  %malloccall11 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims12 = bitcast i8* %malloccall11 to [1 x i64]*
  %dims_as_i8ptr13 = bitcast [1 x i64]* %dims12 to i8*
  %elmptr14 = getelementptr [1 x i64], [1 x i64]* %dims12, i64 0, i64 0
  store i64 1, i64* %elmptr14
  %malloccall15 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data16 = bitcast i8* %malloccall15 to [1 x i8*]*
  %data_as_i8ptr17 = bitcast [1 x i8*]* %data16 to i8*
  %elmptr18 = getelementptr [1 x i8*], [1 x i8*]* %data16, i64 0, i64 0
  store i8* %tensor, i8** %elmptr18
  %dimsptr19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 2
  store i8* %dims_as_i8ptr13, i8** %dimsptr19
  %dataptr20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 3
  store i8* %data_as_i8ptr17, i8** %dataptr20
  %rc21 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 4
  store i8 0, i8* %rc21
  %tensor22 = bitcast %tensor_t* %raw_tensor8 to i8*
  %l123 = load i8*, i8** %l1
  %access_tensor = call i8* @index_get(i8* %l123, i8* %tensor22)
  %malloccall24 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor25 = bitcast i8* %malloccall24 to %tensor_t*
  %dtype26 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 0
  store i8 0, i8* %dtype26
  %ndims27 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 1
  store i8 0, i8* %ndims27
  %malloccall28 = tail call i8* @malloc(i32 0)
  %dims29 = bitcast i8* %malloccall28 to [0 x i64]*
  %dims_as_i8ptr30 = bitcast [0 x i64]* %dims29 to i8*
  %malloccall31 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data32 = bitcast i8* %malloccall31 to [1 x i32]*
  %data_as_i8ptr33 = bitcast [1 x i32]* %data32 to i8*
  %elmptr34 = getelementptr [1 x i32], [1 x i32]* %data32, i32 0, i32 0
```

```
  store i32 2, i32* %elmptr34
  %dimsptr35 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 2
  store i8* %dims_as_i8ptr30, i8** %dimsptr35
  %dataptr36 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 3
  store i8* %data_as_i8ptr33, i8** %dataptr36
  %rc37 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 4
  store i8 0, i8* %rc37
  %tensor38 = bitcast %tensor_t* %raw_tensor25 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor38)
  %malloccall39 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor40 = bitcast i8* %malloccall39 to %tensor_t*
  %dtype41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 0
  store i8 0, i8* %dtype41
  %ndims42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 1
  store i8 0, i8* %ndims42
  %malloccall43 = tail call i8* @malloc(i32 0)
  %dims44 = bitcast i8* %malloccall43 to [0 x i64]*
  %dims_as_i8ptr45 = bitcast [0 x i64]* %dims44 to i8*
  %malloccall46 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data47 = bitcast i8* %malloccall46 to [1 x i32]*
  %data_as_i8ptr48 = bitcast [1 x i32]* %data47 to i8*
  %elmptr49 = getelementptr [1 x i32], [1 x i32]* %data47, i32 0, i32 0
  store i32 0, i32* %elmptr49
  %dimsptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 2
  store i8* %dims_as_i8ptr45, i8** %dimsptr50
  %dataptr51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 3
  store i8* %data_as_i8ptr48, i8** %dataptr51
  %rc52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 4
  store i8 0, i8* %rc52
  %tensor53 = bitcast %tensor_t* %raw_tensor40 to i8*
  %malloccall54 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor55 = bitcast i8* %malloccall54 to %tensor_t*
  %dtype56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 0
  store i8 3, i8* %dtype56
  %ndims57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 1
  store i8 1, i8* %ndims57
  %malloccall58 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims59 = bitcast i8* %malloccall58 to [1 x i64]*
  %dims_as_i8ptr60 = bitcast [1 x i64]* %dims59 to i8*
  %elmptr61 = getelementptr [1 x i64], [1 x i64]* %dims59, i64 0, i64 0
```

```
  store i64 1, i64* %elmptr61
  %malloccall62 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data63 = bitcast i8* %malloccall62 to [1 x i8*]*
  %data_as_i8ptr64 = bitcast [1 x i8*]* %data63 to i8*
  %elmptr65 = getelementptr [1 x i8*], [1 x i8*]* %data63, i64 0, i64 0
  store i8* %tensor53, i8** %elmptr65
  %dimsptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 2
  store i8* %dims_as_i8ptr60, i8** %dimsptr66
  %dataptr67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 3
  store i8* %data_as_i8ptr64, i8** %dataptr67
  %rc68 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 4
  store i8 0, i8* %rc68
  %tensor69 = bitcast %tensor_t* %raw_tensor55 to i8*
  %l170 = load i8*, i8** %l1
  %access_tensor71 = call i8* @index_get(i8* %l170, i8* %tensor69)
  %malloccall72 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor73 = bitcast i8* %malloccall72 to %tensor_t*
  %dtype74 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 0
  store i8 0, i8* %dtype74
  %ndims75 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 1
  store i8 0, i8* %ndims75
  %malloccall76 = tail call i8* @malloc(i32 0)
  %dims77 = bitcast i8* %malloccall76 to [0 x i64]*
  %dims_as_i8ptr78 = bitcast [0 x i64]* %dims77 to i8*
  %malloccall79 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data80 = bitcast i8* %malloccall79 to [1 x i32]*
  %data_as_i8ptr81 = bitcast [1 x i32]* %data80 to i8*
  %elmptr82 = getelementptr [1 x i32], [1 x i32]* %data80, i32 0, i32 0
  store i32 1, i32* %elmptr82
  %dimsptr83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 2
  store i8* %dims_as_i8ptr78, i8** %dimsptr83
  %dataptr84 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 3
  store i8* %data_as_i8ptr81, i8** %dataptr84
  %rc85 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 4
  store i8 0, i8* %rc85
  %tensor86 = bitcast %tensor_t* %raw_tensor73 to i8*
  %tmpOp87 = call i8* @range(i8* %tmpOp, i8* %access_tensor71, i8* %tensor86)
  %malloccall88 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor89 = bitcast i8* %malloccall88 to %tensor_t*
  %dtype90 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 0
  store i8 0, i8* %dtype90
```

```
  %ndims91 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 1
  store i8 0, i8* %ndims91
  %malloccall92 = tail call i8* @malloc(i32 0)
  %dims93 = bitcast i8* %malloccall92 to [0 x i64]*
  %dims_as_i8ptr94 = bitcast [0 x i64]* %dims93 to i8*
  %malloccall95 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data96 = bitcast i8* %malloccall95 to [1 x i32]*
  %data_as_i8ptr97 = bitcast [1 x i32]* %data96 to i8*
  %elmptr98 = getelementptr [1 x i32], [1 x i32]* %data96, i32 0, i32 0
  store i32 1, i32* %elmptr98
  %dimsptr99 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 2
  store i8* %dims_as_i8ptr94, i8** %dimsptr99
  %dataptr100 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 3
  store i8* %data_as_i8ptr97, i8** %dataptr100
  %rc101 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 4
  store i8 0, i8* %rc101
  %tensor102 = bitcast %tensor_t* %raw_tensor89 to i8*
  %malloccall103 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor104 = bitcast i8* %malloccall103 to %tensor_t*
  %dtype105 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 0
  store i8 3, i8* %dtype105
  %ndims106 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 1
  store i8 1, i8* %ndims106
  %malloccall107 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims108 = bitcast i8* %malloccall107 to [1 x i64]*
  %dims_as_i8ptr109 = bitcast [1 x i64]* %dims108 to i8*
  %elmptr110 = getelementptr [1 x i64], [1 x i64]* %dims108, i64 0, i64 0
  store i64 1, i64* %elmptr110
  %malloccall111 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data112 = bitcast i8* %malloccall111 to [1 x i8*]*
  %data_as_i8ptr113 = bitcast [1 x i8*]* %data112 to i8*
  %elmptr114 = getelementptr [1 x i8*], [1 x i8*]* %data112, i64 0, i64 0
  store i8* %tensor102, i8** %elmptr114
  %dimsptr115 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 2
  store i8* %dims_as_i8ptr109, i8** %dimsptr115
  %dataptr116 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 3
  store i8* %data_as_i8ptr113, i8** %dataptr116
  %rc117 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 4
  store i8 0, i8* %rc117
  %tensor118 = bitcast %tensor_t* %raw_tensor104 to i8*
```

```
  %l1119 = load i8*, i8** %l1
  %access_tensor120 = call i8* @index_get(i8* %l1119, i8* %tensor118)
  %malloccall121 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor122 = bitcast i8* %malloccall121 to %tensor_t*
  %dtype123 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 0
  store i8 0, i8* %dtype123
  %ndims124 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 1
  store i8 0, i8* %ndims124
  %malloccall125 = tail call i8* @malloc(i32 0)
  %dims126 = bitcast i8* %malloccall125 to [0 x i64]*
  %dims_as_i8ptr127 = bitcast [0 x i64]* %dims126 to i8*
  %malloccall128 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data129 = bitcast i8* %malloccall128 to [1 x i32]*
  %data_as_i8ptr130 = bitcast [1 x i32]* %data129 to i8*
  %elmptr131 = getelementptr [1 x i32], [1 x i32]* %data129, i32 0, i32 0
  store i32 2, i32* %elmptr131
  %dimsptr132 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 2
  store i8* %dims_as_i8ptr127, i8** %dimsptr132
  %dataptr133 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 3
  store i8* %data_as_i8ptr130, i8** %dataptr133
  %rc134 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 4
  store i8 0, i8* %rc134
  %tensor135 = bitcast %tensor_t* %raw_tensor122 to i8*
  %tmpOp136 = call i8* @floordivide(i8* %access_tensor120, i8* %tensor135)
  %malloccall137 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor138 = bitcast i8* %malloccall137 to %tensor_t*
  %dtype139 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 0
  store i8 0, i8* %dtype139
  %ndims140 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 1
  store i8 0, i8* %ndims140
  %malloccall141 = tail call i8* @malloc(i32 0)
  %dims142 = bitcast i8* %malloccall141 to [0 x i64]*
  %dims_as_i8ptr143 = bitcast [0 x i64]* %dims142 to i8*
  %malloccall144 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data145 = bitcast i8* %malloccall144 to [1 x i32]*
  %data_as_i8ptr146 = bitcast [1 x i32]* %data145 to i8*
  %elmptr147 = getelementptr [1 x i32], [1 x i32]* %data145, i32 0, i32 0
  store i32 1, i32* %elmptr147
  %dimsptr148 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 2
  store i8* %dims_as_i8ptr143, i8** %dimsptr148
```

```
  %dataptr149 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 3
  store i8* %data_as_i8ptr146, i8** %dataptr149
  %rc150 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 4
  store i8 0, i8* %rc150
  %tensor151 = bitcast %tensor_t* %raw_tensor138 to i8*
  %malloccall152 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor153 = bitcast i8* %malloccall152 to %tensor_t*
  %dtype154 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 0
  store i8 3, i8* %dtype154
  %ndims155 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 1
  store i8 1, i8* %ndims155
  %malloccall156 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims157 = bitcast i8* %malloccall156 to [1 x i64]*
  %dims_as_i8ptr158 = bitcast [1 x i64]* %dims157 to i8*
  %elmptr159 = getelementptr [1 x i64], [1 x i64]* %dims157, i64 0, i64 0
  store i64 1, i64* %elmptr159
  %malloccall160 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data161 = bitcast i8* %malloccall160 to [1 x i8*]*
  %data_as_i8ptr162 = bitcast [1 x i8*]* %data161 to i8*
  %elmptr163 = getelementptr [1 x i8*], [1 x i8*]* %data161, i64 0, i64 0
  store i8* %tensor151, i8** %elmptr163
  %dimsptr164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 2
  store i8* %dims_as_i8ptr158, i8** %dimsptr164
  %dataptr165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 3
  store i8* %data_as_i8ptr162, i8** %dataptr165
  %rc166 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 4
  store i8 0, i8* %rc166
  %tensor167 = bitcast %tensor_t* %raw_tensor153 to i8*
  %l1168 = load i8*, i8** %l1
  %access_tensor169 = call i8* @index_get(i8* %l1168, i8* %tensor167)
  %malloccall170 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor171 = bitcast i8* %malloccall170 to %tensor_t*
  %dtype172 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 0
  store i8 0, i8* %dtype172
  %ndims173 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 1
  store i8 0, i8* %ndims173
  %malloccall174 = tail call i8* @malloc(i32 0)
  %dims175 = bitcast i8* %malloccall174 to [0 x i64]*
  %dims_as_i8ptr176 = bitcast [0 x i64]* %dims175 to i8*
```

```
  %malloccall177 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data178 = bitcast i8* %malloccall177 to [1 x i32]*
  %data_as_i8ptr179 = bitcast [1 x i32]* %data178 to i8*
  %elmptr180 = getelementptr [1 x i32], [1 x i32]* %data178, i32 0, i32 0
  store i32 1, i32* %elmptr180
  %dimsptr181 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 2
  store i8* %dims_as_i8ptr176, i8** %dimsptr181
  %dataptr182 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 3
  store i8* %data_as_i8ptr179, i8** %dataptr182
  %rc183 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 4
  store i8 0, i8* %rc183
  %tensor184 = bitcast %tensor_t* %raw_tensor171 to i8*
  %tmpOp185 = call i8* @range(i8* %tmpOp136, i8* %access_tensor169, i8* %tensor184)
  %malloccall186 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor187 = bitcast i8* %malloccall186 to %tensor_t*
  %dtype188 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 0
  store i8 3, i8* %dtype188
  %ndims189 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 1
  store i8 1, i8* %ndims189
  %malloccall190 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims191 = bitcast i8* %malloccall190 to [1 x i64]*
  %dims_as_i8ptr192 = bitcast [1 x i64]* %dims191 to i8*
  %elmptr193 = getelementptr [1 x i64], [1 x i64]* %dims191, i64 0, i64 0
  store i64 2, i64* %elmptr193
  %malloccall194 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data195 = bitcast i8* %malloccall194 to [2 x i8*]*
  %data_as_i8ptr196 = bitcast [2 x i8*]* %data195 to i8*
  %elmptr197 = getelementptr [2 x i8*], [2 x i8*]* %data195, i64 0, i64 0
  store i8* %tmpOp87, i8** %elmptr197
  %elmptr198 = getelementptr [2 x i8*], [2 x i8*]* %data195, i64 0, i64 1
  store i8* %tmpOp185, i8** %elmptr198
  %dimsptr199 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 2
  store i8* %dims_as_i8ptr192, i8** %dimsptr199
  %dataptr200 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 3
  store i8* %data_as_i8ptr196, i8** %dataptr200
  %rc201 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 4
  store i8 0, i8* %rc201
  %tensor202 = bitcast %tensor_t* %raw_tensor187 to i8*
  %x203 = load i8*, i8** %x
  %access_tensor204 = call i8* @index_get(i8* %x203, i8* %tensor202)
```

```
  %x22 = alloca i8*
  store i8* null, i8** %x22
  %lhsptr205 = load i8*, i8** %x22
  call void @increase_rc(i8* %access_tensor204)
  call void @decrease_rc(i8* %lhsptr205)
  store i8* %access_tensor204, i8** %x22
  %malloccall206 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor207 = bitcast i8* %malloccall206 to %tensor_t*
  %dtype208 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 0
  store i8 0, i8* %dtype208
  %ndims209 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 1
  store i8 0, i8* %ndims209
  %malloccall210 = tail call i8* @malloc(i32 0)
  %dims211 = bitcast i8* %malloccall210 to [0 x i64]*
  %dims_as_i8ptr212 = bitcast [0 x i64]* %dims211 to i8*
  %malloccall213 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data214 = bitcast i8* %malloccall213 to [1 x i32]*
  %data_as_i8ptr215 = bitcast [1 x i32]* %data214 to i8*
  %elmptr216 = getelementptr [1 x i32], [1 x i32]* %data214, i32 0, i32 0
  store i32 0, i32* %elmptr216
  %dimsptr217 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 2
  store i8* %dims_as_i8ptr212, i8** %dimsptr217
  %dataptr218 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 3
  store i8* %data_as_i8ptr215, i8** %dataptr218
  %rc219 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 4
  store i8 0, i8* %rc219
  %tensor220 = bitcast %tensor_t* %raw_tensor207 to i8*
  %malloccall221 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor222 = bitcast i8* %malloccall221 to %tensor_t*
  %dtype223 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 0
  store i8 0, i8* %dtype223
  %ndims224 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 1
  store i8 0, i8* %ndims224
  %malloccall225 = tail call i8* @malloc(i32 0)
  %dims226 = bitcast i8* %malloccall225 to [0 x i64]*
  %dims_as_i8ptr227 = bitcast [0 x i64]* %dims226 to i8*
  %malloccall228 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data229 = bitcast i8* %malloccall228 to [1 x i32]*
  %data_as_i8ptr230 = bitcast [1 x i32]* %data229 to i8*
  %elmptr231 = getelementptr [1 x i32], [1 x i32]* %data229, i32 0, i32 0
```

```
  store i32 0, i32* %elmptr231
  %dimsptr232 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 2
  store i8* %dims_as_i8ptr227, i8** %dimsptr232
  %dataptr233 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 3
  store i8* %data_as_i8ptr230, i8** %dataptr233
  %rc234 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 4
  store i8 0, i8* %rc234
  %tensor235 = bitcast %tensor_t* %raw_tensor222 to i8*
  %malloccall236 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor237 = bitcast i8* %malloccall236 to %tensor_t*
  %dtype238 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 0
  store i8 3, i8* %dtype238
  %ndims239 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 1
  store i8 1, i8* %ndims239
  %malloccall240 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims241 = bitcast i8* %malloccall240 to [1 x i64]*
  %dims_as_i8ptr242 = bitcast [1 x i64]* %dims241 to i8*
  %elmptr243 = getelementptr [1 x i64], [1 x i64]* %dims241, i64 0, i64 0
  store i64 1, i64* %elmptr243
  %malloccall244 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data245 = bitcast i8* %malloccall244 to [1 x i8*]*
  %data_as_i8ptr246 = bitcast [1 x i8*]* %data245 to i8*
  %elmptr247 = getelementptr [1 x i8*], [1 x i8*]* %data245, i64 0, i64 0
  store i8* %tensor235, i8** %elmptr247
  %dimsptr248 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 2
  store i8* %dims_as_i8ptr242, i8** %dimsptr248
  %dataptr249 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 3
  store i8* %data_as_i8ptr246, i8** %dataptr249
  %rc250 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 4
  store i8 0, i8* %rc250
  %tensor251 = bitcast %tensor_t* %raw_tensor237 to i8*
  %l2252 = load i8*, i8** %l2
  %access_tensor253 = call i8* @index_get(i8* %l2252, i8* %tensor251)
  %malloccall254 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor255 = bitcast i8* %malloccall254 to %tensor_t*
  %dtype256 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor255, i32 0, i32 0
  store i8 0, i8* %dtype256
  %ndims257 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor255, i32 0, i32 1
  store i8 0, i8* %ndims257
  %malloccall258 = tail call i8* @malloc(i32 0)
```

```
  %dims259 = bitcast i8* %malloccall258 to [0 x i64]*
  %dims_as_i8ptr260 = bitcast [0 x i64]* %dims259 to i8*
  %malloccall261 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data262 = bitcast i8* %malloccall261 to [1 x i32]*
  %data_as_i8ptr263 = bitcast [1 x i32]* %data262 to i8*
  %elmptr264 = getelementptr [1 x i32], [1 x i32]* %data262, i32 0, i32 0
  store i32 2, i32* %elmptr264
  %dimsptr265 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor255, i32 0, i32 2
  store i8* %dims_as_i8ptr260, i8** %dimsptr265
  %dataptr266 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor255, i32 0, i32 3
  store i8* %data_as_i8ptr263, i8** %dataptr266
  %rc267 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor255, i32 0, i32 4
  store i8 0, i8* %rc267
  %tensor268 = bitcast %tensor_t* %raw_tensor255 to i8*
  %tmpOp269 = call i8* @floordivide(i8* %access_tensor253, i8* %tensor268)
  %malloccall270 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor271 = bitcast i8* %malloccall270 to %tensor_t*
  %dtype272 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 0
  store i8 0, i8* %dtype272
  %ndims273 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 1
  store i8 0, i8* %ndims273
  %malloccall274 = tail call i8* @malloc(i32 0)
  %dims275 = bitcast i8* %malloccall274 to [0 x i64]*
  %dims_as_i8ptr276 = bitcast [0 x i64]* %dims275 to i8*
  %malloccall277 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data278 = bitcast i8* %malloccall277 to [1 x i32]*
  %data_as_i8ptr279 = bitcast [1 x i32]* %data278 to i8*
  %elmptr280 = getelementptr [1 x i32], [1 x i32]* %data278, i32 0, i32 0
  store i32 1, i32* %elmptr280
  %dimsptr281 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 2
  store i8* %dims_as_i8ptr276, i8** %dimsptr281
  %dataptr282 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 3
  store i8* %data_as_i8ptr279, i8** %dataptr282
  %rc283 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 4
  store i8 0, i8* %rc283
  %tensor284 = bitcast %tensor_t* %raw_tensor271 to i8*
  %tmpOp285 = call i8* @range(i8* %tensor220, i8* %tmpOp269, i8* %tensor284)
  %malloccall286 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor287 = bitcast i8* %malloccall286 to %tensor_t*
  %dtype288 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 0
```

```
  store i8 0, i8* %dtype288
  %ndims289 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 1
  store i8 0, i8* %ndims289
  %malloccall290 = tail call i8* @malloc(i32 0)
  %dims291 = bitcast i8* %malloccall290 to [0 x i64]*
  %dims_as_i8ptr292 = bitcast [0 x i64]* %dims291 to i8*
  %malloccall293 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data294 = bitcast i8* %malloccall293 to [1 x i32]*
  %data_as_i8ptr295 = bitcast [1 x i32]* %data294 to i8*
  %elmptr296 = getelementptr [1 x i32], [1 x i32]* %data294, i32 0, i32 0
  store i32 0, i32* %elmptr296
  %dimsptr297 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 2
  store i8* %dims_as_i8ptr292, i8** %dimsptr297
  %dataptr298 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 3
  store i8* %data_as_i8ptr295, i8** %dataptr298
  %rc299 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor287, i32 0, i32 4
  store i8 0, i8* %rc299
  %tensor300 = bitcast %tensor_t* %raw_tensor287 to i8*
  %malloccall301 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor302 = bitcast i8* %malloccall301 to %tensor_t*
  %dtype303 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 0
  store i8 0, i8* %dtype303
  %ndims304 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 1
  store i8 0, i8* %ndims304
  %malloccall305 = tail call i8* @malloc(i32 0)
  %dims306 = bitcast i8* %malloccall305 to [0 x i64]*
  %dims_as_i8ptr307 = bitcast [0 x i64]* %dims306 to i8*
  %malloccall308 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data309 = bitcast i8* %malloccall308 to [1 x i32]*
  %data_as_i8ptr310 = bitcast [1 x i32]* %data309 to i8*
  %elmptr311 = getelementptr [1 x i32], [1 x i32]* %data309, i32 0, i32 0
  store i32 1, i32* %elmptr311
  %dimsptr312 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 2
  store i8* %dims_as_i8ptr307, i8** %dimsptr312
  %dataptr313 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 3
  store i8* %data_as_i8ptr310, i8** %dataptr313
  %rc314 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 4
  store i8 0, i8* %rc314
  %tensor315 = bitcast %tensor_t* %raw_tensor302 to i8*
  %malloccall316 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor317 = bitcast i8* %malloccall316 to %tensor_t*
  %dtype318 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 0
  store i8 3, i8* %dtype318
  %ndims319 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 1
  store i8 1, i8* %ndims319
  %malloccall320 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims321 = bitcast i8* %malloccall320 to [1 x i64]*
  %dims_as_i8ptr322 = bitcast [1 x i64]* %dims321 to i8*
  %elmptr323 = getelementptr [1 x i64], [1 x i64]* %dims321, i64 0, i64 0
  store i64 1, i64* %elmptr323
  %malloccall324 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data325 = bitcast i8* %malloccall324 to [1 x i8*]*
  %data_as_i8ptr326 = bitcast [1 x i8*]* %data325 to i8*
  %elmptr327 = getelementptr [1 x i8*], [1 x i8*]* %data325, i64 0, i64 0
  store i8* %tensor315, i8** %elmptr327
  %dimsptr328 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 2
  store i8* %dims_as_i8ptr322, i8** %dimsptr328
  %dataptr329 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 3
  store i8* %data_as_i8ptr326, i8** %dataptr329
  %rc330 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 4
  store i8 0, i8* %rc330
  %tensor331 = bitcast %tensor_t* %raw_tensor317 to i8*
  %l2332 = load i8*, i8** %l2
  %access_tensor333 = call i8* @index_get(i8* %l2332, i8* %tensor331)
  %malloccall334 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor335 = bitcast i8* %malloccall334 to %tensor_t*
  %dtype336 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor335, i32 0, i32 0
  store i8 0, i8* %dtype336
  %ndims337 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor335, i32 0, i32 1
  store i8 0, i8* %ndims337
  %malloccall338 = tail call i8* @malloc(i32 0)
  %dims339 = bitcast i8* %malloccall338 to [0 x i64]*
  %dims_as_i8ptr340 = bitcast [0 x i64]* %dims339 to i8*
  %malloccall341 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data342 = bitcast i8* %malloccall341 to [1 x i32]*
  %data_as_i8ptr343 = bitcast [1 x i32]* %data342 to i8*
  %elmptr344 = getelementptr [1 x i32], [1 x i32]* %data342, i32 0, i32 0
  store i32 2, i32* %elmptr344
  %dimsptr345 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor335, i32 0, i32 2
  store i8* %dims_as_i8ptr340, i8** %dimsptr345
```

```
  %dataptr346 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor335, i32 0, i32 3
  store i8* %data_as_i8ptr343, i8** %dataptr346
  %rc347 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor335, i32 0, i32 4
  store i8 0, i8* %rc347
  %tensor348 = bitcast %tensor_t* %raw_tensor335 to i8*
  %tmpOp349 = call i8* @floordivide(i8* %access_tensor333, i8* %tensor348)
  %malloccall350 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor351 = bitcast i8* %malloccall350 to %tensor_t*
  %dtype352 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 0
  store i8 0, i8* %dtype352
  %ndims353 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 1
  store i8 0, i8* %ndims353
  %malloccall354 = tail call i8* @malloc(i32 0)
  %dims355 = bitcast i8* %malloccall354 to [0 x i64]*
  %dims_as_i8ptr356 = bitcast [0 x i64]* %dims355 to i8*
  %malloccall357 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data358 = bitcast i8* %malloccall357 to [1 x i32]*
  %data_as_i8ptr359 = bitcast [1 x i32]* %data358 to i8*
  %elmptr360 = getelementptr [1 x i32], [1 x i32]* %data358, i32 0, i32 0
  store i32 1, i32* %elmptr360
  %dimsptr361 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 2
  store i8* %dims_as_i8ptr356, i8** %dimsptr361
  %dataptr362 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 3
  store i8* %data_as_i8ptr359, i8** %dataptr362
  %rc363 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 4
  store i8 0, i8* %rc363
  %tensor364 = bitcast %tensor_t* %raw_tensor351 to i8*
  %tmpOp365 = call i8* @range(i8* %tensor300, i8* %tmpOp349, i8* %tensor364)
  %malloccall366 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor367 = bitcast i8* %malloccall366 to %tensor_t*
  %dtype368 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 0
  store i8 3, i8* %dtype368
  %ndims369 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 1
  store i8 1, i8* %ndims369
  %malloccall370 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims371 = bitcast i8* %malloccall370 to [1 x i64]*
  %dims_as_i8ptr372 = bitcast [1 x i64]* %dims371 to i8*
  %elmptr373 = getelementptr [1 x i64], [1 x i64]* %dims371, i64 0, i64 0
  store i64 2, i64* %elmptr373
```

```
  %malloccall374 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data375 = bitcast i8* %malloccall374 to [2 x i8*]*
  %data_as_i8ptr376 = bitcast [2 x i8*]* %data375 to i8*
  %elmptr377 = getelementptr [2 x i8*], [2 x i8*]* %data375, i64 0, i64 0
  store i8* %tmpOp285, i8** %elmptr377
  %elmptr378 = getelementptr [2 x i8*], [2 x i8*]* %data375, i64 0, i64 1
  store i8* %tmpOp365, i8** %elmptr378
  %dimsptr379 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 2
  store i8* %dims_as_i8ptr372, i8** %dimsptr379
  %dataptr380 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 3
  store i8* %data_as_i8ptr376, i8** %dataptr380
  %rc381 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 4
  store i8 0, i8* %rc381
  %tensor382 = bitcast %tensor_t* %raw_tensor367 to i8*
  %y383 = load i8*, i8** %y
  %access_tensor384 = call i8* @index_get(i8* %y383, i8* %tensor382)
  %y11 = alloca i8*
  store i8* null, i8** %y11
  %lhsptr385 = load i8*, i8** %y11
  call void @increase_rc(i8* %access_tensor384)
  call void @decrease_rc(i8* %lhsptr385)
  store i8* %access_tensor384, i8** %y11
  %malloccall386 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr (%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor387 = bitcast i8* %malloccall386 to %tensor_t*
  %dtype388 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 0
  store i8 0, i8* %dtype388
  %ndims389 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 1
  store i8 0, i8* %ndims389
  %malloccall390 = tail call i8* @malloc(i32 0)
  %dims391 = bitcast i8* %malloccall390 to [0 x i64]*
  %dims_as_i8ptr392 = bitcast [0 x i64]* %dims391 to i8*
  %malloccall393 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data394 = bitcast i8* %malloccall393 to [1 x i32]*
  %data_as_i8ptr395 = bitcast [1 x i32]* %data394 to i8*
  %elmptr396 = getelementptr [1 x i32], [1 x i32]* %data394, i32 0, i32 0
  store i32 0, i32* %elmptr396
  %dimsptr397 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 2
  store i8* %dims_as_i8ptr392, i8** %dimsptr397
  %dataptr398 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 3
  store i8* %data_as_i8ptr395, i8** %dataptr398
  %rc399 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 4
```

```
  store i8 0, i8* %rc399
  %tensor400 = bitcast %tensor_t* %raw_tensor387 to i8*
  %malloccall401 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor402 = bitcast i8* %malloccall401 to %tensor_t*
  %dtype403 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 0
  store i8 3, i8* %dtype403
  %ndims404 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 1
  store i8 1, i8* %ndims404
  %malloccall405 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims406 = bitcast i8* %malloccall405 to [1 x i64]*
  %dims_as_i8ptr407 = bitcast [1 x i64]* %dims406 to i8*
  %elmptr408 = getelementptr [1 x i64], [1 x i64]* %dims406, i64 0, i64 0
  store i64 1, i64* %elmptr408
  %malloccall409 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data410 = bitcast i8* %malloccall409 to [1 x i8*]*
  %data_as_i8ptr411 = bitcast [1 x i8*]* %data410 to i8*
  %elmptr412 = getelementptr [1 x i8*], [1 x i8*]* %data410, i64 0, i64 0
  store i8* %tensor400, i8** %elmptr412
  %dimsptr413 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 2
  store i8* %dims_as_i8ptr407, i8** %dimsptr413
  %dataptr414 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 3
  store i8* %data_as_i8ptr411, i8** %dataptr414
  %rc415 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 4
  store i8 0, i8* %rc415
  %tensor416 = bitcast %tensor_t* %raw_tensor402 to i8*
  %l2417 = load i8*, i8** %l2
  %access_tensor418 = call i8* @index_get(i8* %l2417, i8* %tensor416)
  %malloccall419 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor420 = bitcast i8* %malloccall419 to %tensor_t*
  %dtype421 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 0
  store i8 0, i8* %dtype421
  %ndims422 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 1
  store i8 0, i8* %ndims422
  %malloccall423 = tail call i8* @malloc(i32 0)
  %dims424 = bitcast i8* %malloccall423 to [0 x i64]*
  %dims_as_i8ptr425 = bitcast [0 x i64]* %dims424 to i8*
  %malloccall426 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data427 = bitcast i8* %malloccall426 to [1 x i32]*
  %data_as_i8ptr428 = bitcast [1 x i32]* %data427 to i8*
```

```
  %elmptr429 = getelementptr [1 x i32], [1 x i32]* %data427, i32 0, i32 0
  store i32 2, i32* %elmptr429
  %dimsptr430 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 2
  store i8* %dims_as_i8ptr425, i8** %dimsptr430
  %dataptr431 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 3
  store i8* %data_as_i8ptr428, i8** %dataptr431
  %rc432 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor420, i32 0, i32 4
  store i8 0, i8* %rc432
  %tensor433 = bitcast %tensor_t* %raw_tensor420 to i8*
  %tmpOp434 = call i8* @floordivide(i8* %access_tensor418, i8* %tensor433)
  %malloccall435 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor436 = bitcast i8* %malloccall435 to %tensor_t*
  %dtype437 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor436, i32 0, i32 0
  store i8 0, i8* %dtype437
  %ndims438 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor436, i32 0, i32 1
  store i8 0, i8* %ndims438
  %malloccall439 = tail call i8* @malloc(i32 0)
  %dims440 = bitcast i8* %malloccall439 to [0 x i64]*
  %dims_as_i8ptr441 = bitcast [0 x i64]* %dims440 to i8*
  %malloccall442 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data443 = bitcast i8* %malloccall442 to [1 x i32]*
  %data_as_i8ptr444 = bitcast [1 x i32]* %data443 to i8*
  %elmptr445 = getelementptr [1 x i32], [1 x i32]* %data443, i32 0, i32 0
  store i32 0, i32* %elmptr445
  %dimsptr446 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor436, i32 0, i32 2
  store i8* %dims_as_i8ptr441, i8** %dimsptr446
  %dataptr447 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor436, i32 0, i32 3
  store i8* %data_as_i8ptr444, i8** %dataptr447
  %rc448 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor436, i32 0, i32 4
  store i8 0, i8* %rc448
  %tensor449 = bitcast %tensor_t* %raw_tensor436 to i8*
  %malloccall450 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor451 = bitcast i8* %malloccall450 to %tensor_t*
  %dtype452 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 0
  store i8 3, i8* %dtype452
  %ndims453 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 1
  store i8 1, i8* %ndims453
  %malloccall454 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims455 = bitcast i8* %malloccall454 to [1 x i64]*
  %dims_as_i8ptr456 = bitcast [1 x i64]* %dims455 to i8*
```

```
  %elmptr457 = getelementptr [1 x i64], [1 x i64]* %dims455, i64 0, i64 0
  store i64 1, i64* %elmptr457
  %malloccall458 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data459 = bitcast i8* %malloccall458 to [1 x i8*]*
  %data_as_i8ptr460 = bitcast [1 x i8*]* %data459 to i8*
  %elmptr461 = getelementptr [1 x i8*], [1 x i8*]* %data459, i64 0, i64 0
  store i8* %tensor449, i8** %elmptr461
  %dimsptr462 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 2
  store i8* %dims_as_i8ptr456, i8** %dimsptr462
  %dataptr463 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 3
  store i8* %data_as_i8ptr460, i8** %dataptr463
  %rc464 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 4
  store i8 0, i8* %rc464
  %tensor465 = bitcast %tensor_t* %raw_tensor451 to i8*
  %l2466 = load i8*, i8** %l2
  %access_tensor467 = call i8* @index_get(i8* %l2466, i8* %tensor465)
  %malloccall468 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor469 = bitcast i8* %malloccall468 to %tensor_t*
  %dtype470 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 0
  store i8 0, i8* %dtype470
  %ndims471 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 1
  store i8 0, i8* %ndims471
  %malloccall472 = tail call i8* @malloc(i32 0)
  %dims473 = bitcast i8* %malloccall472 to [0 x i64]*
  %dims_as_i8ptr474 = bitcast [0 x i64]* %dims473 to i8*
  %malloccall475 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data476 = bitcast i8* %malloccall475 to [1 x i32]*
  %data_as_i8ptr477 = bitcast [1 x i32]* %data476 to i8*
  %elmptr478 = getelementptr [1 x i32], [1 x i32]* %data476, i32 0, i32 0
  store i32 1, i32* %elmptr478
  %dimsptr479 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 2
  store i8* %dims_as_i8ptr474, i8** %dimsptr479
  %dataptr480 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 3
  store i8* %data_as_i8ptr477, i8** %dataptr480
  %rc481 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor469, i32 0, i32 4
  store i8 0, i8* %rc481
  %tensor482 = bitcast %tensor_t* %raw_tensor469 to i8*
  %tmpOp483 = call i8* @range(i8* %tmpOp434, i8* %access_tensor467, i8* %tensor482)
  %malloccall484 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor485 = bitcast i8* %malloccall484 to %tensor_t*
```

```
  %dtype486 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 0
  store i8 0, i8* %dtype486
  %ndims487 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 1
  store i8 0, i8* %ndims487
  %malloccall488 = tail call i8* @malloc(i32 0)
  %dims489 = bitcast i8* %malloccall488 to [0 x i64]*
  %dims_as_i8ptr490 = bitcast [0 x i64]* %dims489 to i8*
  %malloccall491 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data492 = bitcast i8* %malloccall491 to [1 x i32]*
  %data_as_i8ptr493 = bitcast [1 x i32]* %data492 to i8*
  %elmptr494 = getelementptr [1 x i32], [1 x i32]* %data492, i32 0, i32 0
  store i32 0, i32* %elmptr494
  %dimsptr495 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 2
  store i8* %dims_as_i8ptr490, i8** %dimsptr495
  %dataptr496 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 3
  store i8* %data_as_i8ptr493, i8** %dataptr496
  %rc497 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor485, i32 0, i32 4
  store i8 0, i8* %rc497
  %tensor498 = bitcast %tensor_t* %raw_tensor485 to i8*
  %malloccall499 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor500 = bitcast i8* %malloccall499 to %tensor_t*
  %dtype501 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 0
  store i8 0, i8* %dtype501
  %ndims502 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 1
  store i8 0, i8* %ndims502
  %malloccall503 = tail call i8* @malloc(i32 0)
  %dims504 = bitcast i8* %malloccall503 to [0 x i64]*
  %dims_as_i8ptr505 = bitcast [0 x i64]* %dims504 to i8*
  %malloccall506 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data507 = bitcast i8* %malloccall506 to [1 x i32]*
  %data_as_i8ptr508 = bitcast [1 x i32]* %data507 to i8*
  %elmptr509 = getelementptr [1 x i32], [1 x i32]* %data507, i32 0, i32 0
  store i32 1, i32* %elmptr509
  %dimsptr510 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 2
  store i8* %dims_as_i8ptr505, i8** %dimsptr510
  %dataptr511 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 3
  store i8* %data_as_i8ptr508, i8** %dataptr511
  %rc512 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 4
  store i8 0, i8* %rc512
  %tensor513 = bitcast %tensor_t* %raw_tensor500 to i8*
```

```
  %malloccall514 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor515 = bitcast i8* %malloccall514 to %tensor_t*
  %dtype516 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 0
  store i8 3, i8* %dtype516
  %ndims517 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 1
  store i8 1, i8* %ndims517
  %malloccall518 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims519 = bitcast i8* %malloccall518 to [1 x i64]*
  %dims_as_i8ptr520 = bitcast [1 x i64]* %dims519 to i8*
  %elmptr521 = getelementptr [1 x i64], [1 x i64]* %dims519, i64 0, i64 0
  store i64 1, i64* %elmptr521
  %malloccall522 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data523 = bitcast i8* %malloccall522 to [1 x i8*]*
  %data_as_i8ptr524 = bitcast [1 x i8*]* %data523 to i8*
  %elmptr525 = getelementptr [1 x i8*], [1 x i8*]* %data523, i64 0, i64 0
  store i8* %tensor513, i8** %elmptr525
  %dimsptr526 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 2
  store i8* %dims_as_i8ptr520, i8** %dimsptr526
  %dataptr527 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 3
  store i8* %data_as_i8ptr524, i8** %dataptr527
  %rc528 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor515, i32 0, i32 4
  store i8 0, i8* %rc528
  %tensor529 = bitcast %tensor_t* %raw_tensor515 to i8*
  %l2530 = load i8*, i8** %l2
  %access_tensor531 = call i8* @index_get(i8* %l2530, i8* %tensor529)
  %malloccall532 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor533 = bitcast i8* %malloccall532 to %tensor_t*
  %dtype534 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 0
  store i8 0, i8* %dtype534
  %ndims535 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 1
  store i8 0, i8* %ndims535
  %malloccall536 = tail call i8* @malloc(i32 0)
  %dims537 = bitcast i8* %malloccall536 to [0 x i64]*
  %dims_as_i8ptr538 = bitcast [0 x i64]* %dims537 to i8*
  %malloccall539 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data540 = bitcast i8* %malloccall539 to [1 x i32]*
  %data_as_i8ptr541 = bitcast [1 x i32]* %data540 to i8*
  %elmptr542 = getelementptr [1 x i32], [1 x i32]* %data540, i32 0, i32 0
  store i32 2, i32* %elmptr542
```

```
  %dimsptr543 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 2
  store i8* %dims_as_i8ptr538, i8** %dimsptr543
  %dataptr544 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 3
  store i8* %data_as_i8ptr541, i8** %dataptr544
  %rc545 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor533, i32 0, i32 4
  store i8 0, i8* %rc545
  %tensor546 = bitcast %tensor_t* %raw_tensor533 to i8*
  %tmpOp547 = call i8* @floordivide(i8* %access_tensor531, i8* %tensor546)
  %malloccall548 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor549 = bitcast i8* %malloccall548 to %tensor_t*
  %dtype550 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 0
  store i8 0, i8* %dtype550
  %ndims551 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 1
  store i8 0, i8* %ndims551
  %malloccall552 = tail call i8* @malloc(i32 0)
  %dims553 = bitcast i8* %malloccall552 to [0 x i64]*
  %dims_as_i8ptr554 = bitcast [0 x i64]* %dims553 to i8*
  %malloccall555 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data556 = bitcast i8* %malloccall555 to [1 x i32]*
  %data_as_i8ptr557 = bitcast [1 x i32]* %data556 to i8*
  %elmptr558 = getelementptr [1 x i32], [1 x i32]* %data556, i32 0, i32 0
  store i32 1, i32* %elmptr558
  %dimsptr559 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 2
  store i8* %dims_as_i8ptr554, i8** %dimsptr559
  %dataptr560 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 3
  store i8* %data_as_i8ptr557, i8** %dataptr560
  %rc561 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 4
  store i8 0, i8* %rc561
  %tensor562 = bitcast %tensor_t* %raw_tensor549 to i8*
  %tmpOp563 = call i8* @range(i8* %tensor498, i8* %tmpOp547, i8* %tensor562)
  %malloccall564 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor565 = bitcast i8* %malloccall564 to %tensor_t*
  %dtype566 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 0
  store i8 3, i8* %dtype566
  %ndims567 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 1
  store i8 1, i8* %ndims567
  %malloccall568 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims569 = bitcast i8* %malloccall568 to [1 x i64]*
  %dims_as_i8ptr570 = bitcast [1 x i64]* %dims569 to i8*
  %elmptr571 = getelementptr [1 x i64], [1 x i64]* %dims569, i64 0, i64 0
```

```
  store i64 2, i64* %elmptr571
  %malloccall572 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data573 = bitcast i8* %malloccall572 to [2 x i8*]*
  %data_as_i8ptr574 = bitcast [2 x i8*]* %data573 to i8*
  %elmptr575 = getelementptr [2 x i8*], [2 x i8*]* %data573, i64 0, i64 0
  store i8* %tmpOp483, i8** %elmptr575
  %elmptr576 = getelementptr [2 x i8*], [2 x i8*]* %data573, i64 0, i64 1
  store i8* %tmpOp563, i8** %elmptr576
  %dimsptr577 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 2
  store i8* %dims_as_i8ptr570, i8** %dimsptr577
  %dataptr578 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 3
  store i8* %data_as_i8ptr574, i8** %dataptr578
  %rc579 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 4
  store i8 0, i8* %rc579
  %tensor580 = bitcast %tensor_t* %raw_tensor565 to i8*
  %y581 = load i8*, i8** %y
  %access_tensor582 = call i8* @index_get(i8* %y581, i8* %tensor580)
  %y21 = alloca i8*
  store i8* null, i8** %y21
  %lhsptr583 = load i8*, i8** %y21
  call void @increase_rc(i8* %access_tensor582)
  call void @decrease_rc(i8* %lhsptr583)
  store i8* %access_tensor582, i8** %y21
  %x22584 = load i8*, i8** %x22
  %y21585 = load i8*, i8** %y21
  %y11586 = load i8*, i8** %y11
  %tmpOp587 = call i8* @subtract(i8* %y21585, i8* %y11586)
  %tmpOp588 = call i8* @mult(i8* %x22584, i8* %tmpOp587)
  ret i8* %tmpOp588
}

define i8* @StrassenMULf5(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
  %l1 = alloca i8*
  store i8* null, i8** %l1
  %lhsptr = load i8*, i8** %l1
  call void @increase_rc(i8* %shape)
```

```
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l1
  %y2 = load i8*, i8** %y
  %shape3 = call i8* @shape(i8* %y2)
  %l2 = alloca i8*
  store i8* null, i8** %l2
  %lhsptr4 = load i8*, i8** %l2
  call void @increase_rc(i8* %shape3)
  call void @decrease_rc(i8* %lhsptr4)
  store i8* %shape3, i8** %l2
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall5 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall5 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall6 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall6 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor8 = bitcast i8* %malloccall7 to %tensor_t*
  %dtype9 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 0
  store i8 0, i8* %dtype9
  %ndims10 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 1
  store i8 0, i8* %ndims10
  %malloccall11 = tail call i8* @malloc(i32 0)
  %dims12 = bitcast i8* %malloccall11 to [0 x i64]*
  %dims_as_i8ptr13 = bitcast [0 x i64]* %dims12 to i8*
  %malloccall14 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
```

```
  %data15 = bitcast i8* %malloccall14 to [1 x i32]*
  %data_as_i8ptr16 = bitcast [1 x i32]* %data15 to i8*
  %elmptr17 = getelementptr [1 x i32], [1 x i32]* %data15, i32 0, i32 0
  store i32 0, i32* %elmptr17
  %dimsptr18 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 2
  store i8* %dims_as_i8ptr13, i8** %dimsptr18
  %dataptr19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 3
  store i8* %data_as_i8ptr16, i8** %dataptr19
  %rc20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 4
  store i8 0, i8* %rc20
  %tensor21 = bitcast %tensor_t* %raw_tensor8 to i8*
  %malloccall22 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor23 = bitcast i8* %malloccall22 to %tensor_t*
  %dtype24 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 0
  store i8 3, i8* %dtype24
  %ndims25 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 1
  store i8 1, i8* %ndims25
  %malloccall26 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims27 = bitcast i8* %malloccall26 to [1 x i64]*
  %dims_as_i8ptr28 = bitcast [1 x i64]* %dims27 to i8*
  %elmptr29 = getelementptr [1 x i64], [1 x i64]* %dims27, i64 0, i64 0
  store i64 1, i64* %elmptr29
  %malloccall30 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data31 = bitcast i8* %malloccall30 to [1 x i8*]*
  %data_as_i8ptr32 = bitcast [1 x i8*]* %data31 to i8*
  %elmptr33 = getelementptr [1 x i8*], [1 x i8*]* %data31, i64 0, i64 0
  store i8* %tensor21, i8** %elmptr33
  %dimsptr34 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 2
  store i8* %dims_as_i8ptr28, i8** %dimsptr34
  %dataptr35 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 3
  store i8* %data_as_i8ptr32, i8** %dataptr35
  %rc36 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor23, i32 0, i32 4
  store i8 0, i8* %rc36
  %tensor37 = bitcast %tensor_t* %raw_tensor23 to i8*
  %l138 = load i8*, i8** %l1
  %access_tensor = call i8* @index_get(i8* %l138, i8* %tensor37)
  %malloccall39 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor40 = bitcast i8* %malloccall39 to %tensor_t*
  %dtype41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 0
  store i8 0, i8* %dtype41
  %ndims42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 1
```

```
  store i8 0, i8* %ndims42
  %malloccall43 = tail call i8* @malloc(i32 0)
  %dims44 = bitcast i8* %malloccall43 to [0 x i64]*
  %dims_as_i8ptr45 = bitcast [0 x i64]* %dims44 to i8*
  %malloccall46 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data47 = bitcast i8* %malloccall46 to [1 x i32]*
  %data_as_i8ptr48 = bitcast [1 x i32]* %data47 to i8*
  %elmptr49 = getelementptr [1 x i32], [1 x i32]* %data47, i32 0, i32 0
  store i32 2, i32* %elmptr49
  %dimsptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 2
  store i8* %dims_as_i8ptr45, i8** %dimsptr50
  %dataptr51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 3
  store i8* %data_as_i8ptr48, i8** %dataptr51
  %rc52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 4
  store i8 0, i8* %rc52
  %tensor53 = bitcast %tensor_t* %raw_tensor40 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor53)
  %malloccall54 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor55 = bitcast i8* %malloccall54 to %tensor_t*
  %dtype56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 0
  store i8 0, i8* %dtype56
  %ndims57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 1
  store i8 0, i8* %ndims57
  %malloccall58 = tail call i8* @malloc(i32 0)
  %dims59 = bitcast i8* %malloccall58 to [0 x i64]*
  %dims_as_i8ptr60 = bitcast [0 x i64]* %dims59 to i8*
  %malloccall61 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data62 = bitcast i8* %malloccall61 to [1 x i32]*
  %data_as_i8ptr63 = bitcast [1 x i32]* %data62 to i8*
  %elmptr64 = getelementptr [1 x i32], [1 x i32]* %data62, i32 0, i32 0
  store i32 1, i32* %elmptr64
  %dimsptr65 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 2
  store i8* %dims_as_i8ptr60, i8** %dimsptr65
  %dataptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 3
  store i8* %data_as_i8ptr63, i8** %dataptr66
  %rc67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 4
  store i8 0, i8* %rc67
  %tensor68 = bitcast %tensor_t* %raw_tensor55 to i8*
  %tmpOp69 = call i8* @range(i8* %tensor, i8* %tmpOp, i8* %tensor68)
  %malloccall70 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor71 = bitcast i8* %malloccall70 to %tensor_t*
  %dtype72 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 0
  store i8 0, i8* %dtype72
  %ndims73 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 1
  store i8 0, i8* %ndims73
  %malloccall74 = tail call i8* @malloc(i32 0)
  %dims75 = bitcast i8* %malloccall74 to [0 x i64]*
  %dims_as_i8ptr76 = bitcast [0 x i64]* %dims75 to i8*
  %malloccall77 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data78 = bitcast i8* %malloccall77 to [1 x i32]*
  %data_as_i8ptr79 = bitcast [1 x i32]* %data78 to i8*
  %elmptr80 = getelementptr [1 x i32], [1 x i32]* %data78, i32 0, i32 0
  store i32 0, i32* %elmptr80
  %dimsptr81 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 2
  store i8* %dims_as_i8ptr76, i8** %dimsptr81
  %dataptr82 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 3
  store i8* %data_as_i8ptr79, i8** %dataptr82
  %rc83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor71, i32 0, i32 4
  store i8 0, i8* %rc83
  %tensor84 = bitcast %tensor_t* %raw_tensor71 to i8*
  %malloccall85 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor86 = bitcast i8* %malloccall85 to %tensor_t*
  %dtype87 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 0
  store i8 0, i8* %dtype87
  %ndims88 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 1
  store i8 0, i8* %ndims88
  %malloccall89 = tail call i8* @malloc(i32 0)
  %dims90 = bitcast i8* %malloccall89 to [0 x i64]*
  %dims_as_i8ptr91 = bitcast [0 x i64]* %dims90 to i8*
  %malloccall92 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data93 = bitcast i8* %malloccall92 to [1 x i32]*
  %data_as_i8ptr94 = bitcast [1 x i32]* %data93 to i8*
  %elmptr95 = getelementptr [1 x i32], [1 x i32]* %data93, i32 0, i32 0
  store i32 1, i32* %elmptr95
  %dimsptr96 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 2
  store i8* %dims_as_i8ptr91, i8** %dimsptr96
  %dataptr97 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 3
  store i8* %data_as_i8ptr94, i8** %dataptr97
  %rc98 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor86, i32 0, i32 4
  store i8 0, i8* %rc98
  %tensor99 = bitcast %tensor_t* %raw_tensor86 to i8*
```

```
  %malloccall100 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor101 = bitcast i8* %malloccall100 to %tensor_t*
  %dtype102 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 0
  store i8 3, i8* %dtype102
  %ndims103 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 1
  store i8 1, i8* %ndims103
  %malloccall104 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims105 = bitcast i8* %malloccall104 to [1 x i64]*
  %dims_as_i8ptr106 = bitcast [1 x i64]* %dims105 to i8*
  %elmptr107 = getelementptr [1 x i64], [1 x i64]* %dims105, i64 0, i64 0
  store i64 1, i64* %elmptr107
  %malloccall108 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data109 = bitcast i8* %malloccall108 to [1 x i8*]*
  %data_as_i8ptr110 = bitcast [1 x i8*]* %data109 to i8*
  %elmptr111 = getelementptr [1 x i8*], [1 x i8*]* %data109, i64 0, i64 0
  store i8* %tensor99, i8** %elmptr111
  %dimsptr112 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 2
  store i8* %dims_as_i8ptr106, i8** %dimsptr112
  %dataptr113 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 3
  store i8* %data_as_i8ptr110, i8** %dataptr113
  %rc114 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor101, i32 0, i32 4
  store i8 0, i8* %rc114
  %tensor115 = bitcast %tensor_t* %raw_tensor101 to i8*
  %l1116 = load i8*, i8** %l1
  %access_tensor117 = call i8* @index_get(i8* %l1116, i8* %tensor115)
  %malloccall118 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor119 = bitcast i8* %malloccall118 to %tensor_t*
  %dtype120 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 0
  store i8 0, i8* %dtype120
  %ndims121 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 1
  store i8 0, i8* %ndims121
  %malloccall122 = tail call i8* @malloc(i32 0)
  %dims123 = bitcast i8* %malloccall122 to [0 x i64]*
  %dims_as_i8ptr124 = bitcast [0 x i64]* %dims123 to i8*
  %malloccall125 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data126 = bitcast i8* %malloccall125 to [1 x i32]*
  %data_as_i8ptr127 = bitcast [1 x i32]* %data126 to i8*
  %elmptr128 = getelementptr [1 x i32], [1 x i32]* %data126, i32 0, i32 0
  store i32 2, i32* %elmptr128
```

```
  %dimsptr129 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 2
  store i8* %dims_as_i8ptr124, i8** %dimsptr129
  %dataptr130 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 3
  store i8* %data_as_i8ptr127, i8** %dataptr130
  %rc131 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 4
  store i8 0, i8* %rc131
  %tensor132 = bitcast %tensor_t* %raw_tensor119 to i8*
  %tmpOp133 = call i8* @floordivide(i8* %access_tensor117, i8* %tensor132)
  %malloccall134 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor135 = bitcast i8* %malloccall134 to %tensor_t*
  %dtype136 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 0
  store i8 0, i8* %dtype136
  %ndims137 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 1
  store i8 0, i8* %ndims137
  %malloccall138 = tail call i8* @malloc(i32 0)
  %dims139 = bitcast i8* %malloccall138 to [0 x i64]*
  %dims_as_i8ptr140 = bitcast [0 x i64]* %dims139 to i8*
  %malloccall141 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data142 = bitcast i8* %malloccall141 to [1 x i32]*
  %data_as_i8ptr143 = bitcast [1 x i32]* %data142 to i8*
  %elmptr144 = getelementptr [1 x i32], [1 x i32]* %data142, i32 0, i32 0
  store i32 1, i32* %elmptr144
  %dimsptr145 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 2
  store i8* %dims_as_i8ptr140, i8** %dimsptr145
  %dataptr146 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 3
  store i8* %data_as_i8ptr143, i8** %dataptr146
  %rc147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor135, i32 0, i32 4
  store i8 0, i8* %rc147
  %tensor148 = bitcast %tensor_t* %raw_tensor135 to i8*
  %tmpOp149 = call i8* @range(i8* %tensor84, i8* %tmpOp133, i8* %tensor148)
  %malloccall150 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor151 = bitcast i8* %malloccall150 to %tensor_t*
  %dtype152 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 0
  store i8 3, i8* %dtype152
  %ndims153 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 1
  store i8 1, i8* %ndims153
  %malloccall154 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims155 = bitcast i8* %malloccall154 to [1 x i64]*
  %dims_as_i8ptr156 = bitcast [1 x i64]* %dims155 to i8*
  %elmptr157 = getelementptr [1 x i64], [1 x i64]* %dims155, i64 0, i64 0
```

```
  store i64 2, i64* %elmptr157
  %malloccall158 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data159 = bitcast i8* %malloccall158 to [2 x i8*]*
  %data_as_i8ptr160 = bitcast [2 x i8*]* %data159 to i8*
  %elmptr161 = getelementptr [2 x i8*], [2 x i8*]* %data159, i64 0, i64 0
  store i8* %tmpOp69, i8** %elmptr161
  %elmptr162 = getelementptr [2 x i8*], [2 x i8*]* %data159, i64 0, i64 1
  store i8* %tmpOp149, i8** %elmptr162
  %dimsptr163 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 2
  store i8* %dims_as_i8ptr156, i8** %dimsptr163
  %dataptr164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 3
  store i8* %data_as_i8ptr160, i8** %dataptr164
  %rc165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor151, i32 0, i32 4
  store i8 0, i8* %rc165
  %tensor166 = bitcast %tensor_t* %raw_tensor151 to i8*
  %x167 = load i8*, i8** %x
  %access_tensor168 = call i8* @index_get(i8* %x167, i8* %tensor166)
  %x11 = alloca i8*
  store i8* null, i8** %x11
  %lhsptr169 = load i8*, i8** %x11
  call void @increase_rc(i8* %access_tensor168)
  call void @decrease_rc(i8* %lhsptr169)
  store i8* %access_tensor168, i8** %x11
  %malloccall170 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr (%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor171 = bitcast i8* %malloccall170 to %tensor_t*
  %dtype172 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 0
  store i8 0, i8* %dtype172
  %ndims173 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 1
  store i8 0, i8* %ndims173
  %malloccall174 = tail call i8* @malloc(i32 0)
  %dims175 = bitcast i8* %malloccall174 to [0 x i64]*
  %dims_as_i8ptr176 = bitcast [0 x i64]* %dims175 to i8*
  %malloccall177 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data178 = bitcast i8* %malloccall177 to [1 x i32]*
  %data_as_i8ptr179 = bitcast [1 x i32]* %data178 to i8*
  %elmptr180 = getelementptr [1 x i32], [1 x i32]* %data178, i32 0, i32 0
  store i32 0, i32* %elmptr180
  %dimsptr181 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 2
  store i8* %dims_as_i8ptr176, i8** %dimsptr181
  %dataptr182 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 3
  store i8* %data_as_i8ptr179, i8** %dataptr182
```

```
  %rc183 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 4
  store i8 0, i8* %rc183
  %tensor184 = bitcast %tensor_t* %raw_tensor171 to i8*
  %malloccall185 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor186 = bitcast i8* %malloccall185 to %tensor_t*
  %dtype187 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 0
  store i8 3, i8* %dtype187
  %ndims188 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 1
  store i8 1, i8* %ndims188
  %malloccall189 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims190 = bitcast i8* %malloccall189 to [1 x i64]*
  %dims_as_i8ptr191 = bitcast [1 x i64]* %dims190 to i8*
  %elmptr192 = getelementptr [1 x i64], [1 x i64]* %dims190, i64 0, i64 0
  store i64 1, i64* %elmptr192
  %malloccall193 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data194 = bitcast i8* %malloccall193 to [1 x i8*]*
  %data_as_i8ptr195 = bitcast [1 x i8*]* %data194 to i8*
  %elmptr196 = getelementptr [1 x i8*], [1 x i8*]* %data194, i64 0, i64 0
  store i8* %tensor184, i8** %elmptr196
  %dimsptr197 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 2
  store i8* %dims_as_i8ptr191, i8** %dimsptr197
  %dataptr198 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 3
  store i8* %data_as_i8ptr195, i8** %dataptr198
  %rc199 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor186, i32 0, i32 4
  store i8 0, i8* %rc199
  %tensor200 = bitcast %tensor_t* %raw_tensor186 to i8*
  %l2201 = load i8*, i8** %l2
  %access_tensor202 = call i8* @index_get(i8* %l2201, i8* %tensor200)
  %malloccall203 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor204 = bitcast i8* %malloccall203 to %tensor_t*
  %dtype205 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 0
  store i8 0, i8* %dtype205
  %ndims206 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 1
  store i8 0, i8* %ndims206
  %malloccall207 = tail call i8* @malloc(i32 0)
  %dims208 = bitcast i8* %malloccall207 to [0 x i64]*
  %dims_as_i8ptr209 = bitcast [0 x i64]* %dims208 to i8*
  %malloccall210 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data211 = bitcast i8* %malloccall210 to [1 x i32]*
```

```
  %data_as_i8ptr212 = bitcast [1 x i32]* %data211 to i8*
  %elmptr213 = getelementptr [1 x i32], [1 x i32]* %data211, i32 0, i32 0
  store i32 2, i32* %elmptr213
  %dimsptr214 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 2
  store i8* %dims_as_i8ptr209, i8** %dimsptr214
  %dataptr215 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 3
  store i8* %data_as_i8ptr212, i8** %dataptr215
  %rc216 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 4
  store i8 0, i8* %rc216
  %tensor217 = bitcast %tensor_t* %raw_tensor204 to i8*
  %tmpOp218 = call i8* @floordivide(i8* %access_tensor202, i8* %tensor217)
  %malloccall219 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor220 = bitcast i8* %malloccall219 to %tensor_t*
  %dtype221 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 0
  store i8 0, i8* %dtype221
  %ndims222 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 1
  store i8 0, i8* %ndims222
  %malloccall223 = tail call i8* @malloc(i32 0)
  %dims224 = bitcast i8* %malloccall223 to [0 x i64]*
  %dims_as_i8ptr225 = bitcast [0 x i64]* %dims224 to i8*
  %malloccall226 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data227 = bitcast i8* %malloccall226 to [1 x i32]*
  %data_as_i8ptr228 = bitcast [1 x i32]* %data227 to i8*
  %elmptr229 = getelementptr [1 x i32], [1 x i32]* %data227, i32 0, i32 0
  store i32 0, i32* %elmptr229
  %dimsptr230 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 2
  store i8* %dims_as_i8ptr225, i8** %dimsptr230
  %dataptr231 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 3
  store i8* %data_as_i8ptr228, i8** %dataptr231
  %rc232 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor220, i32 0, i32 4
  store i8 0, i8* %rc232
  %tensor233 = bitcast %tensor_t* %raw_tensor220 to i8*
  %malloccall234 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor235 = bitcast i8* %malloccall234 to %tensor_t*
  %dtype236 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 0
  store i8 3, i8* %dtype236
  %ndims237 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 1
  store i8 1, i8* %ndims237
  %malloccall238 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims239 = bitcast i8* %malloccall238 to [1 x i64]*
```

```
  %dims_as_i8ptr240 = bitcast [1 x i64]* %dims239 to i8*
  %elmptr241 = getelementptr [1 x i64], [1 x i64]* %dims239, i64 0, i64 0
  store i64 1, i64* %elmptr241
  %malloccall242 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data243 = bitcast i8* %malloccall242 to [1 x i8*]*
  %data_as_i8ptr244 = bitcast [1 x i8*]* %data243 to i8*
  %elmptr245 = getelementptr [1 x i8*], [1 x i8*]* %data243, i64 0, i64 0
  store i8* %tensor233, i8** %elmptr245
  %dimsptr246 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 2
  store i8* %dims_as_i8ptr240, i8** %dimsptr246
  %dataptr247 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 3
  store i8* %data_as_i8ptr244, i8** %dataptr247
  %rc248 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor235, i32 0, i32 4
  store i8 0, i8* %rc248
  %tensor249 = bitcast %tensor_t* %raw_tensor235 to i8*
  %l2250 = load i8*, i8** %l2
  %access_tensor251 = call i8* @index_get(i8* %l2250, i8* %tensor249)
  %malloccall252 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor253 = bitcast i8* %malloccall252 to %tensor_t*
  %dtype254 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 0
  store i8 0, i8* %dtype254
  %ndims255 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 1
  store i8 0, i8* %ndims255
  %malloccall256 = tail call i8* @malloc(i32 0)
  %dims257 = bitcast i8* %malloccall256 to [0 x i64]*
  %dims_as_i8ptr258 = bitcast [0 x i64]* %dims257 to i8*
  %malloccall259 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data260 = bitcast i8* %malloccall259 to [1 x i32]*
  %data_as_i8ptr261 = bitcast [1 x i32]* %data260 to i8*
  %elmptr262 = getelementptr [1 x i32], [1 x i32]* %data260, i32 0, i32 0
  store i32 1, i32* %elmptr262
  %dimsptr263 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 2
  store i8* %dims_as_i8ptr258, i8** %dimsptr263
  %dataptr264 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 3
  store i8* %data_as_i8ptr261, i8** %dataptr264
  %rc265 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 4
  store i8 0, i8* %rc265
  %tensor266 = bitcast %tensor_t* %raw_tensor253 to i8*
  %tmpOp267 = call i8* @range(i8* %tmpOp218, i8* %access_tensor251, i8* %tensor266)
  %malloccall268 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor269 = bitcast i8* %malloccall268 to %tensor_t*
  %dtype270 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 0
  store i8 0, i8* %dtype270
  %ndims271 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 1
  store i8 0, i8* %ndims271
  %malloccall272 = tail call i8* @malloc(i32 0)
  %dims273 = bitcast i8* %malloccall272 to [0 x i64]*
  %dims_as_i8ptr274 = bitcast [0 x i64]* %dims273 to i8*
  %malloccall275 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data276 = bitcast i8* %malloccall275 to [1 x i32]*
  %data_as_i8ptr277 = bitcast [1 x i32]* %data276 to i8*
  %elmptr278 = getelementptr [1 x i32], [1 x i32]* %data276, i32 0, i32 0
  store i32 1, i32* %elmptr278
  %dimsptr279 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 2
  store i8* %dims_as_i8ptr274, i8** %dimsptr279
  %dataptr280 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 3
  store i8* %data_as_i8ptr277, i8** %dataptr280
  %rc281 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 4
  store i8 0, i8* %rc281
  %tensor282 = bitcast %tensor_t* %raw_tensor269 to i8*
  %malloccall283 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor284 = bitcast i8* %malloccall283 to %tensor_t*
  %dtype285 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 0
  store i8 3, i8* %dtype285
  %ndims286 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 1
  store i8 1, i8* %ndims286
  %malloccall287 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims288 = bitcast i8* %malloccall287 to [1 x i64]*
  %dims_as_i8ptr289 = bitcast [1 x i64]* %dims288 to i8*
  %elmptr290 = getelementptr [1 x i64], [1 x i64]* %dims288, i64 0, i64 0
  store i64 1, i64* %elmptr290
  %malloccall291 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data292 = bitcast i8* %malloccall291 to [1 x i8*]*
  %data_as_i8ptr293 = bitcast [1 x i8*]* %data292 to i8*
  %elmptr294 = getelementptr [1 x i8*], [1 x i8*]* %data292, i64 0, i64 0
  store i8* %tensor282, i8** %elmptr294
  %dimsptr295 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 2
  store i8* %dims_as_i8ptr289, i8** %dimsptr295
  %dataptr296 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 3
  store i8* %data_as_i8ptr293, i8** %dataptr296
```

```
  %rc297 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 4
  store i8 0, i8* %rc297
  %tensor298 = bitcast %tensor_t* %raw_tensor284 to i8*
  %l2299 = load i8*, i8** %l2
  %access_tensor300 = call i8* @index_get(i8* %l2299, i8* %tensor298)
  %malloccall301 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor302 = bitcast i8* %malloccall301 to %tensor_t*
  %dtype303 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 0
  store i8 0, i8* %dtype303
  %ndims304 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 1
  store i8 0, i8* %ndims304
  %malloccall305 = tail call i8* @malloc(i32 0)
  %dims306 = bitcast i8* %malloccall305 to [0 x i64]*
  %dims_as_i8ptr307 = bitcast [0 x i64]* %dims306 to i8*
  %malloccall308 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data309 = bitcast i8* %malloccall308 to [1 x i32]*
  %data_as_i8ptr310 = bitcast [1 x i32]* %data309 to i8*
  %elmptr311 = getelementptr [1 x i32], [1 x i32]* %data309, i32 0, i32 0
  store i32 2, i32* %elmptr311
  %dimsptr312 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 2
  store i8* %dims_as_i8ptr307, i8** %dimsptr312
  %dataptr313 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 3
  store i8* %data_as_i8ptr310, i8** %dataptr313
  %rc314 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor302, i32 0, i32 4
  store i8 0, i8* %rc314
  %tensor315 = bitcast %tensor_t* %raw_tensor302 to i8*
  %tmpOp316 = call i8* @floordivide(i8* %access_tensor300, i8* %tensor315)
  %malloccall317 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor318 = bitcast i8* %malloccall317 to %tensor_t*
  %dtype319 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 0
  store i8 0, i8* %dtype319
  %ndims320 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 1
  store i8 0, i8* %ndims320
  %malloccall321 = tail call i8* @malloc(i32 0)
  %dims322 = bitcast i8* %malloccall321 to [0 x i64]*
  %dims_as_i8ptr323 = bitcast [0 x i64]* %dims322 to i8*
  %malloccall324 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data325 = bitcast i8* %malloccall324 to [1 x i32]*
  %data_as_i8ptr326 = bitcast [1 x i32]* %data325 to i8*
  %elmptr327 = getelementptr [1 x i32], [1 x i32]* %data325, i32 0, i32 0
```

```
  store i32 1, i32* %elmptr327
  %dimsptr328 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 2
  store i8* %dims_as_i8ptr323, i8** %dimsptr328
  %dataptr329 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 3
  store i8* %data_as_i8ptr326, i8** %dataptr329
  %rc330 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor318, i32 0, i32 4
  store i8 0, i8* %rc330
  %tensor331 = bitcast %tensor_t* %raw_tensor318 to i8*
  %malloccall332 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor333 = bitcast i8* %malloccall332 to %tensor_t*
  %dtype334 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 0
  store i8 3, i8* %dtype334
  %ndims335 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 1
  store i8 1, i8* %ndims335
  %malloccall336 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims337 = bitcast i8* %malloccall336 to [1 x i64]*
  %dims_as_i8ptr338 = bitcast [1 x i64]* %dims337 to i8*
  %elmptr339 = getelementptr [1 x i64], [1 x i64]* %dims337, i64 0, i64 0
  store i64 1, i64* %elmptr339
  %malloccall340 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data341 = bitcast i8* %malloccall340 to [1 x i8*]*
  %data_as_i8ptr342 = bitcast [1 x i8*]* %data341 to i8*
  %elmptr343 = getelementptr [1 x i8*], [1 x i8*]* %data341, i64 0, i64 0
  store i8* %tensor331, i8** %elmptr343
  %dimsptr344 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 2
  store i8* %dims_as_i8ptr338, i8** %dimsptr344
  %dataptr345 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 3
  store i8* %data_as_i8ptr342, i8** %dataptr345
  %rc346 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 4
  store i8 0, i8* %rc346
  %tensor347 = bitcast %tensor_t* %raw_tensor333 to i8*
  %l2348 = load i8*, i8** %l2
  %access_tensor349 = call i8* @index_get(i8* %l2348, i8* %tensor347)
  %malloccall350 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor351 = bitcast i8* %malloccall350 to %tensor_t*
  %dtype352 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 0
  store i8 0, i8* %dtype352
  %ndims353 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 1
  store i8 0, i8* %ndims353
  %malloccall354 = tail call i8* @malloc(i32 0)
```

```
  %dims355 = bitcast i8* %malloccall354 to [0 x i64]*
  %dims_as_i8ptr356 = bitcast [0 x i64]* %dims355 to i8*
  %malloccall357 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data358 = bitcast i8* %malloccall357 to [1 x i32]*
  %data_as_i8ptr359 = bitcast [1 x i32]* %data358 to i8*
  %elmptr360 = getelementptr [1 x i32], [1 x i32]* %data358, i32 0, i32 0
  store i32 1, i32* %elmptr360
  %dimsptr361 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 2
  store i8* %dims_as_i8ptr356, i8** %dimsptr361
  %dataptr362 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 3
  store i8* %data_as_i8ptr359, i8** %dataptr362
  %rc363 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor351, i32 0, i32 4
  store i8 0, i8* %rc363
  %tensor364 = bitcast %tensor_t* %raw_tensor351 to i8*
  %tmpOp365 = call i8* @range(i8* %tmpOp316, i8* %access_tensor349, i8* %tensor364)
  %malloccall366 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor367 = bitcast i8* %malloccall366 to %tensor_t*
  %dtype368 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 0
  store i8 3, i8* %dtype368
  %ndims369 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 1
  store i8 1, i8* %ndims369
  %malloccall370 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims371 = bitcast i8* %malloccall370 to [1 x i64]*
  %dims_as_i8ptr372 = bitcast [1 x i64]* %dims371 to i8*
  %elmptr373 = getelementptr [1 x i64], [1 x i64]* %dims371, i64 0, i64 0
  store i64 2, i64* %elmptr373
  %malloccall374 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data375 = bitcast i8* %malloccall374 to [2 x i8*]*
  %data_as_i8ptr376 = bitcast [2 x i8*]* %data375 to i8*
  %elmptr377 = getelementptr [2 x i8*], [2 x i8*]* %data375, i64 0, i64 0
  store i8* %tmpOp267, i8** %elmptr377
  %elmptr378 = getelementptr [2 x i8*], [2 x i8*]* %data375, i64 0, i64 1
  store i8* %tmpOp365, i8** %elmptr378
  %dimsptr379 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 2
  store i8* %dims_as_i8ptr372, i8** %dimsptr379
  %dataptr380 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 3
  store i8* %data_as_i8ptr376, i8** %dataptr380
  %rc381 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor367, i32 0, i32 4
  store i8 0, i8* %rc381
  %tensor382 = bitcast %tensor_t* %raw_tensor367 to i8*
```

```
  %y383 = load i8*, i8** %y
  %access_tensor384 = call i8* @index_get(i8* %y383, i8* %tensor382)
  %y22 = alloca i8*
  store i8* null, i8** %y22
  %lhsptr385 = load i8*, i8** %y22
  call void @increase_rc(i8* %access_tensor384)
  call void @decrease_rc(i8* %lhsptr385)
  store i8* %access_tensor384, i8** %y22
  %malloccall386 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor387 = bitcast i8* %malloccall386 to %tensor_t*
  %dtype388 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 0
  store i8 0, i8* %dtype388
  %ndims389 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 1
  store i8 0, i8* %ndims389
  %malloccall390 = tail call i8* @malloc(i32 0)
  %dims391 = bitcast i8* %malloccall390 to [0 x i64]*
  %dims_as_i8ptr392 = bitcast [0 x i64]* %dims391 to i8*
  %malloccall393 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data394 = bitcast i8* %malloccall393 to [1 x i32]*
  %data_as_i8ptr395 = bitcast [1 x i32]* %data394 to i8*
  %elmptr396 = getelementptr [1 x i32], [1 x i32]* %data394, i32 0, i32 0
  store i32 0, i32* %elmptr396
  %dimsptr397 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 2
  store i8* %dims_as_i8ptr392, i8** %dimsptr397
  %dataptr398 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 3
  store i8* %data_as_i8ptr395, i8** %dataptr398
  %rc399 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 4
  store i8 0, i8* %rc399
  %tensor400 = bitcast %tensor_t* %raw_tensor387 to i8*
  %malloccall401 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor402 = bitcast i8* %malloccall401 to %tensor_t*
  %dtype403 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 0
  store i8 0, i8* %dtype403
  %ndims404 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 1
  store i8 0, i8* %ndims404
  %malloccall405 = tail call i8* @malloc(i32 0)
  %dims406 = bitcast i8* %malloccall405 to [0 x i64]*
  %dims_as_i8ptr407 = bitcast [0 x i64]* %dims406 to i8*
  %malloccall408 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data409 = bitcast i8* %malloccall408 to [1 x i32]*
```

```
  %data_as_i8ptr410 = bitcast [1 x i32]* %data409 to i8*
  %elmptr411 = getelementptr [1 x i32], [1 x i32]* %data409, i32 0, i32 0
  store i32 0, i32* %elmptr411
  %dimsptr412 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 2
  store i8* %dims_as_i8ptr407, i8** %dimsptr412
  %dataptr413 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 3
  store i8* %data_as_i8ptr410, i8** %dataptr413
  %rc414 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor402, i32 0, i32 4
  store i8 0, i8* %rc414
  %tensor415 = bitcast %tensor_t* %raw_tensor402 to i8*
  %malloccall416 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor417 = bitcast i8* %malloccall416 to %tensor_t*
  %dtype418 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 0
  store i8 3, i8* %dtype418
  %ndims419 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 1
  store i8 1, i8* %ndims419
  %malloccall420 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims421 = bitcast i8* %malloccall420 to [1 x i64]*
  %dims_as_i8ptr422 = bitcast [1 x i64]* %dims421 to i8*
  %elmptr423 = getelementptr [1 x i64], [1 x i64]* %dims421, i64 0, i64 0
  store i64 1, i64* %elmptr423
  %malloccall424 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data425 = bitcast i8* %malloccall424 to [1 x i8*]*
  %data_as_i8ptr426 = bitcast [1 x i8*]* %data425 to i8*
  %elmptr427 = getelementptr [1 x i8*], [1 x i8*]* %data425, i64 0, i64 0
  store i8* %tensor415, i8** %elmptr427
  %dimsptr428 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 2
  store i8* %dims_as_i8ptr422, i8** %dimsptr428
  %dataptr429 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 3
  store i8* %data_as_i8ptr426, i8** %dataptr429
  %rc430 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 4
  store i8 0, i8* %rc430
  %tensor431 = bitcast %tensor_t* %raw_tensor417 to i8*
  %l1432 = load i8*, i8** %l1
  %access_tensor433 = call i8* @index_get(i8* %l1432, i8* %tensor431)
  %malloccall434 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor435 = bitcast i8* %malloccall434 to %tensor_t*
  %dtype436 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 0
  store i8 0, i8* %dtype436
  %ndims437 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 1
```

```
  store i8 0, i8* %ndims437
  %malloccall438 = tail call i8* @malloc(i32 0)
  %dims439 = bitcast i8* %malloccall438 to [0 x i64]*
  %dims_as_i8ptr440 = bitcast [0 x i64]* %dims439 to i8*
  %malloccall441 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data442 = bitcast i8* %malloccall441 to [1 x i32]*
  %data_as_i8ptr443 = bitcast [1 x i32]* %data442 to i8*
  %elmptr444 = getelementptr [1 x i32], [1 x i32]* %data442, i32 0, i32 0
  store i32 2, i32* %elmptr444
  %dimsptr445 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 2
  store i8* %dims_as_i8ptr440, i8** %dimsptr445
  %dataptr446 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 3
  store i8* %data_as_i8ptr443, i8** %dataptr446
  %rc447 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor435, i32 0, i32 4
  store i8 0, i8* %rc447
  %tensor448 = bitcast %tensor_t* %raw_tensor435 to i8*
  %tmpOp449 = call i8* @floordivide(i8* %access_tensor433, i8* %tensor448)
  %malloccall450 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor451 = bitcast i8* %malloccall450 to %tensor_t*
  %dtype452 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 0
  store i8 0, i8* %dtype452
  %ndims453 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 1
  store i8 0, i8* %ndims453
  %malloccall454 = tail call i8* @malloc(i32 0)
  %dims455 = bitcast i8* %malloccall454 to [0 x i64]*
  %dims_as_i8ptr456 = bitcast [0 x i64]* %dims455 to i8*
  %malloccall457 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data458 = bitcast i8* %malloccall457 to [1 x i32]*
  %data_as_i8ptr459 = bitcast [1 x i32]* %data458 to i8*
  %elmptr460 = getelementptr [1 x i32], [1 x i32]* %data458, i32 0, i32 0
  store i32 1, i32* %elmptr460
  %dimsptr461 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 2
  store i8* %dims_as_i8ptr456, i8** %dimsptr461
  %dataptr462 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 3
  store i8* %data_as_i8ptr459, i8** %dataptr462
  %rc463 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor451, i32 0, i32 4
  store i8 0, i8* %rc463
  %tensor464 = bitcast %tensor_t* %raw_tensor451 to i8*
  %tmpOp465 = call i8* @range(i8* %tensor400, i8* %tmpOp449, i8* %tensor464)
  %malloccall466 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor467 = bitcast i8* %malloccall466 to %tensor_t*
  %dtype468 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 0
  store i8 0, i8* %dtype468
  %ndims469 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 1
  store i8 0, i8* %ndims469
  %malloccall470 = tail call i8* @malloc(i32 0)
  %dims471 = bitcast i8* %malloccall470 to [0 x i64]*
  %dims_as_i8ptr472 = bitcast [0 x i64]* %dims471 to i8*
  %malloccall473 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data474 = bitcast i8* %malloccall473 to [1 x i32]*
  %data_as_i8ptr475 = bitcast [1 x i32]* %data474 to i8*
  %elmptr476 = getelementptr [1 x i32], [1 x i32]* %data474, i32 0, i32 0
  store i32 1, i32* %elmptr476
  %dimsptr477 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 2
  store i8* %dims_as_i8ptr472, i8** %dimsptr477
  %dataptr478 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 3
  store i8* %data_as_i8ptr475, i8** %dataptr478
  %rc479 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor467, i32 0, i32 4
  store i8 0, i8* %rc479
  %tensor480 = bitcast %tensor_t* %raw_tensor467 to i8*
  %malloccall481 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor482 = bitcast i8* %malloccall481 to %tensor_t*
  %dtype483 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 0
  store i8 3, i8* %dtype483
  %ndims484 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 1
  store i8 1, i8* %ndims484
  %malloccall485 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims486 = bitcast i8* %malloccall485 to [1 x i64]*
  %dims_as_i8ptr487 = bitcast [1 x i64]* %dims486 to i8*
  %elmptr488 = getelementptr [1 x i64], [1 x i64]* %dims486, i64 0, i64 0
  store i64 1, i64* %elmptr488
  %malloccall489 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data490 = bitcast i8* %malloccall489 to [1 x i8*]*
  %data_as_i8ptr491 = bitcast [1 x i8*]* %data490 to i8*
  %elmptr492 = getelementptr [1 x i8*], [1 x i8*]* %data490, i64 0, i64 0
  store i8* %tensor480, i8** %elmptr492
  %dimsptr493 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 2
  store i8* %dims_as_i8ptr487, i8** %dimsptr493
  %dataptr494 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 3
  store i8* %data_as_i8ptr491, i8** %dataptr494
```

```
  %rc495 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor482, i32 0, i32 4
  store i8 0, i8* %rc495
  %tensor496 = bitcast %tensor_t* %raw_tensor482 to i8*
  %l1497 = load i8*, i8** %l1
  %access_tensor498 = call i8* @index_get(i8* %l1497, i8* %tensor496)
  %malloccall499 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor500 = bitcast i8* %malloccall499 to %tensor_t*
  %dtype501 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 0
  store i8 0, i8* %dtype501
  %ndims502 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 1
  store i8 0, i8* %ndims502
  %malloccall503 = tail call i8* @malloc(i32 0)
  %dims504 = bitcast i8* %malloccall503 to [0 x i64]*
  %dims_as_i8ptr505 = bitcast [0 x i64]* %dims504 to i8*
  %malloccall506 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data507 = bitcast i8* %malloccall506 to [1 x i32]*
  %data_as_i8ptr508 = bitcast [1 x i32]* %data507 to i8*
  %elmptr509 = getelementptr [1 x i32], [1 x i32]* %data507, i32 0, i32 0
  store i32 2, i32* %elmptr509
  %dimsptr510 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 2
  store i8* %dims_as_i8ptr505, i8** %dimsptr510
  %dataptr511 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 3
  store i8* %data_as_i8ptr508, i8** %dataptr511
  %rc512 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor500, i32 0, i32 4
  store i8 0, i8* %rc512
  %tensor513 = bitcast %tensor_t* %raw_tensor500 to i8*
  %tmpOp514 = call i8* @floordivide(i8* %access_tensor498, i8* %tensor513)
  %malloccall515 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor516 = bitcast i8* %malloccall515 to %tensor_t*
  %dtype517 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 0
  store i8 0, i8* %dtype517
  %ndims518 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 1
  store i8 0, i8* %ndims518
  %malloccall519 = tail call i8* @malloc(i32 0)
  %dims520 = bitcast i8* %malloccall519 to [0 x i64]*
  %dims_as_i8ptr521 = bitcast [0 x i64]* %dims520 to i8*
  %malloccall522 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data523 = bitcast i8* %malloccall522 to [1 x i32]*
  %data_as_i8ptr524 = bitcast [1 x i32]* %data523 to i8*
  %elmptr525 = getelementptr [1 x i32], [1 x i32]* %data523, i32 0, i32 0
```

```
  store i32 1, i32* %elmptr525
  %dimsptr526 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 2
  store i8* %dims_as_i8ptr521, i8** %dimsptr526
  %dataptr527 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 3
  store i8* %data_as_i8ptr524, i8** %dataptr527
  %rc528 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor516, i32 0, i32 4
  store i8 0, i8* %rc528
  %tensor529 = bitcast %tensor_t* %raw_tensor516 to i8*
  %malloccall530 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor531 = bitcast i8* %malloccall530 to %tensor_t*
  %dtype532 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 0
  store i8 3, i8* %dtype532
  %ndims533 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 1
  store i8 1, i8* %ndims533
  %malloccall534 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims535 = bitcast i8* %malloccall534 to [1 x i64]*
  %dims_as_i8ptr536 = bitcast [1 x i64]* %dims535 to i8*
  %elmptr537 = getelementptr [1 x i64], [1 x i64]* %dims535, i64 0, i64 0
  store i64 1, i64* %elmptr537
  %malloccall538 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data539 = bitcast i8* %malloccall538 to [1 x i8*]*
  %data_as_i8ptr540 = bitcast [1 x i8*]* %data539 to i8*
  %elmptr541 = getelementptr [1 x i8*], [1 x i8*]* %data539, i64 0, i64 0
  store i8* %tensor529, i8** %elmptr541
  %dimsptr542 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 2
  store i8* %dims_as_i8ptr536, i8** %dimsptr542
  %dataptr543 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 3
  store i8* %data_as_i8ptr540, i8** %dataptr543
  %rc544 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor531, i32 0, i32 4
  store i8 0, i8* %rc544
  %tensor545 = bitcast %tensor_t* %raw_tensor531 to i8*
  %l1546 = load i8*, i8** %l1
  %access_tensor547 = call i8* @index_get(i8* %l1546, i8* %tensor545)
  %malloccall548 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor549 = bitcast i8* %malloccall548 to %tensor_t*
  %dtype550 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 0
  store i8 0, i8* %dtype550
  %ndims551 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 1
  store i8 0, i8* %ndims551
  %malloccall552 = tail call i8* @malloc(i32 0)
```

```
  %dims553 = bitcast i8* %malloccall552 to [0 x i64]*
  %dims_as_i8ptr554 = bitcast [0 x i64]* %dims553 to i8*
  %malloccall555 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data556 = bitcast i8* %malloccall555 to [1 x i32]*
  %data_as_i8ptr557 = bitcast [1 x i32]* %data556 to i8*
  %elmptr558 = getelementptr [1 x i32], [1 x i32]* %data556, i32 0, i32 0
  store i32 1, i32* %elmptr558
  %dimsptr559 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 2
  store i8* %dims_as_i8ptr554, i8** %dimsptr559
  %dataptr560 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 3
  store i8* %data_as_i8ptr557, i8** %dataptr560
  %rc561 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 4
  store i8 0, i8* %rc561
  %tensor562 = bitcast %tensor_t* %raw_tensor549 to i8*
  %tmpOp563 = call i8* @range(i8* %tmpOp514, i8* %access_tensor547, i8* %tensor562)
  %malloccall564 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor565 = bitcast i8* %malloccall564 to %tensor_t*
  %dtype566 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 0
  store i8 3, i8* %dtype566
  %ndims567 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 1
  store i8 1, i8* %ndims567
  %malloccall568 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims569 = bitcast i8* %malloccall568 to [1 x i64]*
  %dims_as_i8ptr570 = bitcast [1 x i64]* %dims569 to i8*
  %elmptr571 = getelementptr [1 x i64], [1 x i64]* %dims569, i64 0, i64 0
  store i64 2, i64* %elmptr571
  %malloccall572 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data573 = bitcast i8* %malloccall572 to [2 x i8*]*
  %data_as_i8ptr574 = bitcast [2 x i8*]* %data573 to i8*
  %elmptr575 = getelementptr [2 x i8*], [2 x i8*]* %data573, i64 0, i64 0
  store i8* %tmpOp465, i8** %elmptr575
  %elmptr576 = getelementptr [2 x i8*], [2 x i8*]* %data573, i64 0, i64 1
  store i8* %tmpOp563, i8** %elmptr576
  %dimsptr577 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 2
  store i8* %dims_as_i8ptr570, i8** %dimsptr577
  %dataptr578 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 3
  store i8* %data_as_i8ptr574, i8** %dataptr578
  %rc579 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor565, i32 0, i32 4
  store i8 0, i8* %rc579
  %tensor580 = bitcast %tensor_t* %raw_tensor565 to i8*
```

```
  %x581 = load i8*, i8** %x
  %access_tensor582 = call i8* @index_get(i8* %x581, i8* %tensor580)
  %x12 = alloca i8*
  store i8* null, i8** %x12
  %lhsptr583 = load i8*, i8** %x12
  call void @increase_rc(i8* %access_tensor582)
  call void @decrease_rc(i8* %lhsptr583)
  store i8* %access_tensor582, i8** %x12
  %x11584 = load i8*, i8** %x11
  %x12585 = load i8*, i8** %x12
  %tmpOp586 = call i8* @add(i8* %x11584, i8* %x12585)
  %y22587 = load i8*, i8** %y22
  %tmpOp588 = call i8* @mult(i8* %tmpOp586, i8* %y22587)
  ret i8* %tmpOp588
}

define i8* @StrassenMULf6(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
  %l1 = alloca i8*
  store i8* null, i8** %l1
  %lhsptr = load i8*, i8** %l1
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l1
  %y2 = load i8*, i8** %y
  %shape3 = call i8* @shape(i8* %y2)
  %l2 = alloca i8*
  store i8* null, i8** %l2
  %lhsptr4 = load i8*, i8** %l2
  call void @increase_rc(i8* %shape3)
  call void @decrease_rc(i8* %lhsptr4)
  store i8* %shape3, i8** %l2
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
```

```
  store i8 0, i8* %ndims
  %malloccall5 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall5 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall6 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall6 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor8 = bitcast i8* %malloccall7 to %tensor_t*
  %dtype9 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 0
  store i8 3, i8* %dtype9
  %ndims10 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 1
  store i8 1, i8* %ndims10
  %malloccall11 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims12 = bitcast i8* %malloccall11 to [1 x i64]*
  %dims_as_i8ptr13 = bitcast [1 x i64]* %dims12 to i8*
  %elmptr14 = getelementptr [1 x i64], [1 x i64]* %dims12, i64 0, i64 0
  store i64 1, i64* %elmptr14
  %malloccall15 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data16 = bitcast i8* %malloccall15 to [1 x i8*]*
  %data_as_i8ptr17 = bitcast [1 x i8*]* %data16 to i8*
  %elmptr18 = getelementptr [1 x i8*], [1 x i8*]* %data16, i64 0, i64 0
  store i8* %tensor, i8** %elmptr18
  %dimsptr19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 2
  store i8* %dims_as_i8ptr13, i8** %dimsptr19
  %dataptr20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 3
  store i8* %data_as_i8ptr17, i8** %dataptr20
  %rc21 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 4
  store i8 0, i8* %rc21
  %tensor22 = bitcast %tensor_t* %raw_tensor8 to i8*
  %l123 = load i8*, i8** %l1
  %access_tensor = call i8* @index_get(i8* %l123, i8* %tensor22)
```

```
  %malloccall24 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor25 = bitcast i8* %malloccall24 to %tensor_t*
  %dtype26 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 0
  store i8 0, i8* %dtype26
  %ndims27 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 1
  store i8 0, i8* %ndims27
  %malloccall28 = tail call i8* @malloc(i32 0)
  %dims29 = bitcast i8* %malloccall28 to [0 x i64]*
  %dims_as_i8ptr30 = bitcast [0 x i64]* %dims29 to i8*
  %malloccall31 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data32 = bitcast i8* %malloccall31 to [1 x i32]*
  %data_as_i8ptr33 = bitcast [1 x i32]* %data32 to i8*
  %elmptr34 = getelementptr [1 x i32], [1 x i32]* %data32, i32 0, i32 0
  store i32 2, i32* %elmptr34
  %dimsptr35 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 2
  store i8* %dims_as_i8ptr30, i8** %dimsptr35
  %dataptr36 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 3
  store i8* %data_as_i8ptr33, i8** %dataptr36
  %rc37 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 4
  store i8 0, i8* %rc37
  %tensor38 = bitcast %tensor_t* %raw_tensor25 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor38)
  %malloccall39 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor40 = bitcast i8* %malloccall39 to %tensor_t*
  %dtype41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 0
  store i8 0, i8* %dtype41
  %ndims42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 1
  store i8 0, i8* %ndims42
  %malloccall43 = tail call i8* @malloc(i32 0)
  %dims44 = bitcast i8* %malloccall43 to [0 x i64]*
  %dims_as_i8ptr45 = bitcast [0 x i64]* %dims44 to i8*
  %malloccall46 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data47 = bitcast i8* %malloccall46 to [1 x i32]*
  %data_as_i8ptr48 = bitcast [1 x i32]* %data47 to i8*
  %elmptr49 = getelementptr [1 x i32], [1 x i32]* %data47, i32 0, i32 0
  store i32 0, i32* %elmptr49
  %dimsptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 2
  store i8* %dims_as_i8ptr45, i8** %dimsptr50
  %dataptr51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 3
  store i8* %data_as_i8ptr48, i8** %dataptr51
```

```
  %rc52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 4
  store i8 0, i8* %rc52
  %tensor53 = bitcast %tensor_t* %raw_tensor40 to i8*
  %malloccall54 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor55 = bitcast i8* %malloccall54 to %tensor_t*
  %dtype56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 0
  store i8 3, i8* %dtype56
  %ndims57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 1
  store i8 1, i8* %ndims57
  %malloccall58 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims59 = bitcast i8* %malloccall58 to [1 x i64]*
  %dims_as_i8ptr60 = bitcast [1 x i64]* %dims59 to i8*
  %elmptr61 = getelementptr [1 x i64], [1 x i64]* %dims59, i64 0, i64 0
  store i64 1, i64* %elmptr61
  %malloccall62 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data63 = bitcast i8* %malloccall62 to [1 x i8*]*
  %data_as_i8ptr64 = bitcast [1 x i8*]* %data63 to i8*
  %elmptr65 = getelementptr [1 x i8*], [1 x i8*]* %data63, i64 0, i64 0
  store i8* %tensor53, i8** %elmptr65
  %dimsptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 2
  store i8* %dims_as_i8ptr60, i8** %dimsptr66
  %dataptr67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 3
  store i8* %data_as_i8ptr64, i8** %dataptr67
  %rc68 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 4
  store i8 0, i8* %rc68
  %tensor69 = bitcast %tensor_t* %raw_tensor55 to i8*
  %l170 = load i8*, i8** %l1
  %access_tensor71 = call i8* @index_get(i8* %l170, i8* %tensor69)
  %malloccall72 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor73 = bitcast i8* %malloccall72 to %tensor_t*
  %dtype74 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 0
  store i8 0, i8* %dtype74
  %ndims75 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 1
  store i8 0, i8* %ndims75
  %malloccall76 = tail call i8* @malloc(i32 0)
  %dims77 = bitcast i8* %malloccall76 to [0 x i64]*
  %dims_as_i8ptr78 = bitcast [0 x i64]* %dims77 to i8*
  %malloccall79 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data80 = bitcast i8* %malloccall79 to [1 x i32]*
  %data_as_i8ptr81 = bitcast [1 x i32]* %data80 to i8*
```

```
  %elmptr82 = getelementptr [1 x i32], [1 x i32]* %data80, i32 0, i32 0
  store i32 1, i32* %elmptr82
  %dimsptr83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 2
  store i8* %dims_as_i8ptr78, i8** %dimsptr83
  %dataptr84 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 3
  store i8* %data_as_i8ptr81, i8** %dataptr84
  %rc85 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 4
  store i8 0, i8* %rc85
  %tensor86 = bitcast %tensor_t* %raw_tensor73 to i8*
  %tmpOp87 = call i8* @range(i8* %tmpOp, i8* %access_tensor71, i8* %tensor86)
  %malloccall88 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor89 = bitcast i8* %malloccall88 to %tensor_t*
  %dtype90 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 0
  store i8 0, i8* %dtype90
  %ndims91 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 1
  store i8 0, i8* %ndims91
  %malloccall92 = tail call i8* @malloc(i32 0)
  %dims93 = bitcast i8* %malloccall92 to [0 x i64]*
  %dims_as_i8ptr94 = bitcast [0 x i64]* %dims93 to i8*
  %malloccall95 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data96 = bitcast i8* %malloccall95 to [1 x i32]*
  %data_as_i8ptr97 = bitcast [1 x i32]* %data96 to i8*
  %elmptr98 = getelementptr [1 x i32], [1 x i32]* %data96, i32 0, i32 0
  store i32 0, i32* %elmptr98
  %dimsptr99 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 2
  store i8* %dims_as_i8ptr94, i8** %dimsptr99
  %dataptr100 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 3
  store i8* %data_as_i8ptr97, i8** %dataptr100
  %rc101 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 4
  store i8 0, i8* %rc101
  %tensor102 = bitcast %tensor_t* %raw_tensor89 to i8*
  %malloccall103 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor104 = bitcast i8* %malloccall103 to %tensor_t*
  %dtype105 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 0
  store i8 0, i8* %dtype105
  %ndims106 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 1
  store i8 0, i8* %ndims106
  %malloccall107 = tail call i8* @malloc(i32 0)
  %dims108 = bitcast i8* %malloccall107 to [0 x i64]*
  %dims_as_i8ptr109 = bitcast [0 x i64]* %dims108 to i8*
```

```
  %malloccall110 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data111 = bitcast i8* %malloccall110 to [1 x i32]*
  %data_as_i8ptr112 = bitcast [1 x i32]* %data111 to i8*
  %elmptr113 = getelementptr [1 x i32], [1 x i32]* %data111, i32 0, i32 0
  store i32 1, i32* %elmptr113
  %dimsptr114 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 2
  store i8* %dims_as_i8ptr109, i8** %dimsptr114
  %dataptr115 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 3
  store i8* %data_as_i8ptr112, i8** %dataptr115
  %rc116 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 4
  store i8 0, i8* %rc116
  %tensor117 = bitcast %tensor_t* %raw_tensor104 to i8*
  %malloccall118 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor119 = bitcast i8* %malloccall118 to %tensor_t*
  %dtype120 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 0
  store i8 3, i8* %dtype120
  %ndims121 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 1
  store i8 1, i8* %ndims121
  %malloccall122 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims123 = bitcast i8* %malloccall122 to [1 x i64]*
  %dims_as_i8ptr124 = bitcast [1 x i64]* %dims123 to i8*
  %elmptr125 = getelementptr [1 x i64], [1 x i64]* %dims123, i64 0, i64 0
  store i64 1, i64* %elmptr125
  %malloccall126 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data127 = bitcast i8* %malloccall126 to [1 x i8*]*
  %data_as_i8ptr128 = bitcast [1 x i8*]* %data127 to i8*
  %elmptr129 = getelementptr [1 x i8*], [1 x i8*]* %data127, i64 0, i64 0
  store i8* %tensor117, i8** %elmptr129
  %dimsptr130 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 2
  store i8* %dims_as_i8ptr124, i8** %dimsptr130
  %dataptr131 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 3
  store i8* %data_as_i8ptr128, i8** %dataptr131
  %rc132 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor119, i32 0, i32 4
  store i8 0, i8* %rc132
  %tensor133 = bitcast %tensor_t* %raw_tensor119 to i8*
  %l1134 = load i8*, i8** %l1
  %access_tensor135 = call i8* @index_get(i8* %l1134, i8* %tensor133)
  %malloccall136 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor137 = bitcast i8* %malloccall136 to %tensor_t*
```

```
  %dtype138 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 0
  store i8 0, i8* %dtype138
  %ndims139 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 1
  store i8 0, i8* %ndims139
  %malloccall140 = tail call i8* @malloc(i32 0)
  %dims141 = bitcast i8* %malloccall140 to [0 x i64]*
  %dims_as_i8ptr142 = bitcast [0 x i64]* %dims141 to i8*
  %malloccall143 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data144 = bitcast i8* %malloccall143 to [1 x i32]*
  %data_as_i8ptr145 = bitcast [1 x i32]* %data144 to i8*
  %elmptr146 = getelementptr [1 x i32], [1 x i32]* %data144, i32 0, i32 0
  store i32 2, i32* %elmptr146
  %dimsptr147 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 2
  store i8* %dims_as_i8ptr142, i8** %dimsptr147
  %dataptr148 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 3
  store i8* %data_as_i8ptr145, i8** %dataptr148
  %rc149 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor137, i32 0, i32 4
  store i8 0, i8* %rc149
  %tensor150 = bitcast %tensor_t* %raw_tensor137 to i8*
  %tmpOp151 = call i8* @floordivide(i8* %access_tensor135, i8* %tensor150)
  %malloccall152 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor153 = bitcast i8* %malloccall152 to %tensor_t*
  %dtype154 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 0
  store i8 0, i8* %dtype154
  %ndims155 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 1
  store i8 0, i8* %ndims155
  %malloccall156 = tail call i8* @malloc(i32 0)
  %dims157 = bitcast i8* %malloccall156 to [0 x i64]*
  %dims_as_i8ptr158 = bitcast [0 x i64]* %dims157 to i8*
  %malloccall159 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data160 = bitcast i8* %malloccall159 to [1 x i32]*
  %data_as_i8ptr161 = bitcast [1 x i32]* %data160 to i8*
  %elmptr162 = getelementptr [1 x i32], [1 x i32]* %data160, i32 0, i32 0
  store i32 1, i32* %elmptr162
  %dimsptr163 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 2
  store i8* %dims_as_i8ptr158, i8** %dimsptr163
  %dataptr164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 3
  store i8* %data_as_i8ptr161, i8** %dataptr164
  %rc165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 4
  store i8 0, i8* %rc165
  %tensor166 = bitcast %tensor_t* %raw_tensor153 to i8*
```

```
  %tmpOp167 = call i8* @range(i8* %tensor102, i8* %tmpOp151, i8* %tensor166)
  %malloccall168 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor169 = bitcast i8* %malloccall168 to %tensor_t*
  %dtype170 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 0
  store i8 3, i8* %dtype170
  %ndims171 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 1
  store i8 1, i8* %ndims171
  %malloccall172 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims173 = bitcast i8* %malloccall172 to [1 x i64]*
  %dims_as_i8ptr174 = bitcast [1 x i64]* %dims173 to i8*
  %elmptr175 = getelementptr [1 x i64], [1 x i64]* %dims173, i64 0, i64 0
  store i64 2, i64* %elmptr175
  %malloccall176 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data177 = bitcast i8* %malloccall176 to [2 x i8*]*
  %data_as_i8ptr178 = bitcast [2 x i8*]* %data177 to i8*
  %elmptr179 = getelementptr [2 x i8*], [2 x i8*]* %data177, i64 0, i64 0
  store i8* %tmpOp87, i8** %elmptr179
  %elmptr180 = getelementptr [2 x i8*], [2 x i8*]* %data177, i64 0, i64 1
  store i8* %tmpOp167, i8** %elmptr180
  %dimsptr181 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 2
  store i8* %dims_as_i8ptr174, i8** %dimsptr181
  %dataptr182 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 3
  store i8* %data_as_i8ptr178, i8** %dataptr182
  %rc183 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor169, i32 0, i32 4
  store i8 0, i8* %rc183
  %tensor184 = bitcast %tensor_t* %raw_tensor169 to i8*
  %x185 = load i8*, i8** %x
  %access_tensor186 = call i8* @index_get(i8* %x185, i8* %tensor184)
  %x21 = alloca i8*
  store i8* null, i8** %x21
  %lhsptr187 = load i8*, i8** %x21
  call void @increase_rc(i8* %access_tensor186)
  call void @decrease_rc(i8* %lhsptr187)
  store i8* %access_tensor186, i8** %x21
  %malloccall188 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor189 = bitcast i8* %malloccall188 to %tensor_t*
  %dtype190 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 0
  store i8 0, i8* %dtype190
  %ndims191 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 1
  store i8 0, i8* %ndims191
```

```
  %malloccall192 = tail call i8* @malloc(i32 0)
  %dims193 = bitcast i8* %malloccall192 to [0 x i64]*
  %dims_as_i8ptr194 = bitcast [0 x i64]* %dims193 to i8*
  %malloccall195 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data196 = bitcast i8* %malloccall195 to [1 x i32]*
  %data_as_i8ptr197 = bitcast [1 x i32]* %data196 to i8*
  %elmptr198 = getelementptr [1 x i32], [1 x i32]* %data196, i32 0, i32 0
  store i32 0, i32* %elmptr198
  %dimsptr199 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 2
  store i8* %dims_as_i8ptr194, i8** %dimsptr199
  %dataptr200 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 3
  store i8* %data_as_i8ptr197, i8** %dataptr200
  %rc201 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor189, i32 0, i32 4
  store i8 0, i8* %rc201
  %tensor202 = bitcast %tensor_t* %raw_tensor189 to i8*
  %malloccall203 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor204 = bitcast i8* %malloccall203 to %tensor_t*
  %dtype205 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 0
  store i8 0, i8* %dtype205
  %ndims206 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 1
  store i8 0, i8* %ndims206
  %malloccall207 = tail call i8* @malloc(i32 0)
  %dims208 = bitcast i8* %malloccall207 to [0 x i64]*
  %dims_as_i8ptr209 = bitcast [0 x i64]* %dims208 to i8*
  %malloccall210 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data211 = bitcast i8* %malloccall210 to [1 x i32]*
  %data_as_i8ptr212 = bitcast [1 x i32]* %data211 to i8*
  %elmptr213 = getelementptr [1 x i32], [1 x i32]* %data211, i32 0, i32 0
  store i32 0, i32* %elmptr213
  %dimsptr214 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 2
  store i8* %dims_as_i8ptr209, i8** %dimsptr214
  %dataptr215 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 3
  store i8* %data_as_i8ptr212, i8** %dataptr215
  %rc216 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor204, i32 0, i32 4
  store i8 0, i8* %rc216
  %tensor217 = bitcast %tensor_t* %raw_tensor204 to i8*
  %malloccall218 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor219 = bitcast i8* %malloccall218 to %tensor_t*
  %dtype220 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 0
  store i8 3, i8* %dtype220
```

```
  %ndims221 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 1
  store i8 1, i8* %ndims221
  %malloccall222 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims223 = bitcast i8* %malloccall222 to [1 x i64]*
  %dims_as_i8ptr224 = bitcast [1 x i64]* %dims223 to i8*
  %elmptr225 = getelementptr [1 x i64], [1 x i64]* %dims223, i64 0, i64 0
  store i64 1, i64* %elmptr225
  %malloccall226 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data227 = bitcast i8* %malloccall226 to [1 x i8*]*
  %data_as_i8ptr228 = bitcast [1 x i8*]* %data227 to i8*
  %elmptr229 = getelementptr [1 x i8*], [1 x i8*]* %data227, i64 0, i64 0
  store i8* %tensor217, i8** %elmptr229
  %dimsptr230 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 2
  store i8* %dims_as_i8ptr224, i8** %dimsptr230
  %dataptr231 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 3
  store i8* %data_as_i8ptr228, i8** %dataptr231
  %rc232 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor219, i32 0, i32 4
  store i8 0, i8* %rc232
  %tensor233 = bitcast %tensor_t* %raw_tensor219 to i8*
  %l2234 = load i8*, i8** %l2
  %access_tensor235 = call i8* @index_get(i8* %l2234, i8* %tensor233)
  %malloccall236 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor237 = bitcast i8* %malloccall236 to %tensor_t*
  %dtype238 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 0
  store i8 0, i8* %dtype238
  %ndims239 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 1
  store i8 0, i8* %ndims239
  %malloccall240 = tail call i8* @malloc(i32 0)
  %dims241 = bitcast i8* %malloccall240 to [0 x i64]*
  %dims_as_i8ptr242 = bitcast [0 x i64]* %dims241 to i8*
  %malloccall243 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data244 = bitcast i8* %malloccall243 to [1 x i32]*
  %data_as_i8ptr245 = bitcast [1 x i32]* %data244 to i8*
  %elmptr246 = getelementptr [1 x i32], [1 x i32]* %data244, i32 0, i32 0
  store i32 2, i32* %elmptr246
  %dimsptr247 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 2
  store i8* %dims_as_i8ptr242, i8** %dimsptr247
  %dataptr248 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 3
  store i8* %data_as_i8ptr245, i8** %dataptr248
  %rc249 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor237, i32 0, i32 4
```

```
  store i8 0, i8* %rc249
  %tensor250 = bitcast %tensor_t* %raw_tensor237 to i8*
  %tmpOp251 = call i8* @floordivide(i8* %access_tensor235, i8* %tensor250)
  %malloccall252 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor253 = bitcast i8* %malloccall252 to %tensor_t*
  %dtype254 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 0
  store i8 0, i8* %dtype254
  %ndims255 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 1
  store i8 0, i8* %ndims255
  %malloccall256 = tail call i8* @malloc(i32 0)
  %dims257 = bitcast i8* %malloccall256 to [0 x i64]*
  %dims_as_i8ptr258 = bitcast [0 x i64]* %dims257 to i8*
  %malloccall259 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data260 = bitcast i8* %malloccall259 to [1 x i32]*
  %data_as_i8ptr261 = bitcast [1 x i32]* %data260 to i8*
  %elmptr262 = getelementptr [1 x i32], [1 x i32]* %data260, i32 0, i32 0
  store i32 1, i32* %elmptr262
  %dimsptr263 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 2
  store i8* %dims_as_i8ptr258, i8** %dimsptr263
  %dataptr264 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 3
  store i8* %data_as_i8ptr261, i8** %dataptr264
  %rc265 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor253, i32 0, i32 4
  store i8 0, i8* %rc265
  %tensor266 = bitcast %tensor_t* %raw_tensor253 to i8*
  %tmpOp267 = call i8* @range(i8* %tensor202, i8* %tmpOp251, i8* %tensor266)
  %malloccall268 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor269 = bitcast i8* %malloccall268 to %tensor_t*
  %dtype270 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 0
  store i8 0, i8* %dtype270
  %ndims271 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 1
  store i8 0, i8* %ndims271
  %malloccall272 = tail call i8* @malloc(i32 0)
  %dims273 = bitcast i8* %malloccall272 to [0 x i64]*
  %dims_as_i8ptr274 = bitcast [0 x i64]* %dims273 to i8*
  %malloccall275 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data276 = bitcast i8* %malloccall275 to [1 x i32]*
  %data_as_i8ptr277 = bitcast [1 x i32]* %data276 to i8*
  %elmptr278 = getelementptr [1 x i32], [1 x i32]* %data276, i32 0, i32 0
  store i32 0, i32* %elmptr278
  %dimsptr279 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 2
```

```
  store i8* %dims_as_i8ptr274, i8** %dimsptr279
  %dataptr280 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 3
  store i8* %data_as_i8ptr277, i8** %dataptr280
  %rc281 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor269, i32 0, i32 4
  store i8 0, i8* %rc281
  %tensor282 = bitcast %tensor_t* %raw_tensor269 to i8*
  %malloccall283 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor284 = bitcast i8* %malloccall283 to %tensor_t*
  %dtype285 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 0
  store i8 0, i8* %dtype285
  %ndims286 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 1
  store i8 0, i8* %ndims286
  %malloccall287 = tail call i8* @malloc(i32 0)
  %dims288 = bitcast i8* %malloccall287 to [0 x i64]*
  %dims_as_i8ptr289 = bitcast [0 x i64]* %dims288 to i8*
  %malloccall290 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data291 = bitcast i8* %malloccall290 to [1 x i32]*
  %data_as_i8ptr292 = bitcast [1 x i32]* %data291 to i8*
  %elmptr293 = getelementptr [1 x i32], [1 x i32]* %data291, i32 0, i32 0
  store i32 1, i32* %elmptr293
  %dimsptr294 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 2
  store i8* %dims_as_i8ptr289, i8** %dimsptr294
  %dataptr295 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 3
  store i8* %data_as_i8ptr292, i8** %dataptr295
  %rc296 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor284, i32 0, i32 4
  store i8 0, i8* %rc296
  %tensor297 = bitcast %tensor_t* %raw_tensor284 to i8*
  %malloccall298 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor299 = bitcast i8* %malloccall298 to %tensor_t*
  %dtype300 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor299, i32 0, i32 0
  store i8 3, i8* %dtype300
  %ndims301 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor299, i32 0, i32 1
  store i8 1, i8* %ndims301
  %malloccall302 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims303 = bitcast i8* %malloccall302 to [1 x i64]*
  %dims_as_i8ptr304 = bitcast [1 x i64]* %dims303 to i8*
  %elmptr305 = getelementptr [1 x i64], [1 x i64]* %dims303, i64 0, i64 0
  store i64 1, i64* %elmptr305
  %malloccall306 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
```

```
  %data307 = bitcast i8* %malloccall306 to [1 x i8*]*
  %data_as_i8ptr308 = bitcast [1 x i8*]* %data307 to i8*
  %elmptr309 = getelementptr [1 x i8*], [1 x i8*]* %data307, i64 0, i64 0
  store i8* %tensor297, i8** %elmptr309
  %dimsptr310 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor299, i32 0, i32 2
  store i8* %dims_as_i8ptr304, i8** %dimsptr310
  %dataptr311 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor299, i32 0, i32 3
  store i8* %data_as_i8ptr308, i8** %dataptr311
  %rc312 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor299, i32 0, i32 4
  store i8 0, i8* %rc312
  %tensor313 = bitcast %tensor_t* %raw_tensor299 to i8*
  %l2314 = load i8*, i8** %l2
  %access_tensor315 = call i8* @index_get(i8* %l2314, i8* %tensor313)
  %malloccall316 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor317 = bitcast i8* %malloccall316 to %tensor_t*
  %dtype318 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 0
  store i8 0, i8* %dtype318
  %ndims319 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 1
  store i8 0, i8* %ndims319
  %malloccall320 = tail call i8* @malloc(i32 0)
  %dims321 = bitcast i8* %malloccall320 to [0 x i64]*
  %dims_as_i8ptr322 = bitcast [0 x i64]* %dims321 to i8*
  %malloccall323 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data324 = bitcast i8* %malloccall323 to [1 x i32]*
  %data_as_i8ptr325 = bitcast [1 x i32]* %data324 to i8*
  %elmptr326 = getelementptr [1 x i32], [1 x i32]* %data324, i32 0, i32 0
  store i32 2, i32* %elmptr326
  %dimsptr327 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 2
  store i8* %dims_as_i8ptr322, i8** %dimsptr327
  %dataptr328 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 3
  store i8* %data_as_i8ptr325, i8** %dataptr328
  %rc329 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor317, i32 0, i32 4
  store i8 0, i8* %rc329
  %tensor330 = bitcast %tensor_t* %raw_tensor317 to i8*
  %tmpOp331 = call i8* @floordivide(i8* %access_tensor315, i8* %tensor330)
  %malloccall332 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor333 = bitcast i8* %malloccall332 to %tensor_t*
  %dtype334 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 0
  store i8 0, i8* %dtype334
  %ndims335 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 1
  store i8 0, i8* %ndims335
```

```
  %malloccall336 = tail call i8* @malloc(i32 0)
  %dims337 = bitcast i8* %malloccall336 to [0 x i64]*
  %dims_as_i8ptr338 = bitcast [0 x i64]* %dims337 to i8*
  %malloccall339 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data340 = bitcast i8* %malloccall339 to [1 x i32]*
  %data_as_i8ptr341 = bitcast [1 x i32]* %data340 to i8*
  %elmptr342 = getelementptr [1 x i32], [1 x i32]* %data340, i32 0, i32 0
  store i32 1, i32* %elmptr342
  %dimsptr343 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 2
  store i8* %dims_as_i8ptr338, i8** %dimsptr343
  %dataptr344 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 3
  store i8* %data_as_i8ptr341, i8** %dataptr344
  %rc345 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor333, i32 0, i32 4
  store i8 0, i8* %rc345
  %tensor346 = bitcast %tensor_t* %raw_tensor333 to i8*
  %tmpOp347 = call i8* @range(i8* %tensor282, i8* %tmpOp331, i8* %tensor346)
  %malloccall348 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor349 = bitcast i8* %malloccall348 to %tensor_t*
  %dtype350 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 0
  store i8 3, i8* %dtype350
  %ndims351 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 1
  store i8 1, i8* %ndims351
  %malloccall352 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims353 = bitcast i8* %malloccall352 to [1 x i64]*
  %dims_as_i8ptr354 = bitcast [1 x i64]* %dims353 to i8*
  %elmptr355 = getelementptr [1 x i64], [1 x i64]* %dims353, i64 0, i64 0
  store i64 2, i64* %elmptr355
  %malloccall356 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data357 = bitcast i8* %malloccall356 to [2 x i8*]*
  %data_as_i8ptr358 = bitcast [2 x i8*]* %data357 to i8*
  %elmptr359 = getelementptr [2 x i8*], [2 x i8*]* %data357, i64 0, i64 0
  store i8* %tmpOp267, i8** %elmptr359
  %elmptr360 = getelementptr [2 x i8*], [2 x i8*]* %data357, i64 0, i64 1
  store i8* %tmpOp347, i8** %elmptr360
  %dimsptr361 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 2
  store i8* %dims_as_i8ptr354, i8** %dimsptr361
  %dataptr362 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 3
  store i8* %data_as_i8ptr358, i8** %dataptr362
  %rc363 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor349, i32 0, i32 4
  store i8 0, i8* %rc363
```

```
  %tensor364 = bitcast %tensor_t* %raw_tensor349 to i8*
  %y365 = load i8*, i8** %y
  %access_tensor366 = call i8* @index_get(i8* %y365, i8* %tensor364)
  %y11 = alloca i8*
  store i8* null, i8** %y11
  %lhsptr367 = load i8*, i8** %y11
  call void @increase_rc(i8* %access_tensor366)
  call void @decrease_rc(i8* %lhsptr367)
  store i8* %access_tensor366, i8** %y11
  %malloccall368 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor369 = bitcast i8* %malloccall368 to %tensor_t*
  %dtype370 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 0
  store i8 0, i8* %dtype370
  %ndims371 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 1
  store i8 0, i8* %ndims371
  %malloccall372 = tail call i8* @malloc(i32 0)
  %dims373 = bitcast i8* %malloccall372 to [0 x i64]*
  %dims_as_i8ptr374 = bitcast [0 x i64]* %dims373 to i8*
  %malloccall375 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data376 = bitcast i8* %malloccall375 to [1 x i32]*
  %data_as_i8ptr377 = bitcast [1 x i32]* %data376 to i8*
  %elmptr378 = getelementptr [1 x i32], [1 x i32]* %data376, i32 0, i32 0
  store i32 0, i32* %elmptr378
  %dimsptr379 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 2
  store i8* %dims_as_i8ptr374, i8** %dimsptr379
  %dataptr380 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 3
  store i8* %data_as_i8ptr377, i8** %dataptr380
  %rc381 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 4
  store i8 0, i8* %rc381
  %tensor382 = bitcast %tensor_t* %raw_tensor369 to i8*
  %malloccall383 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor384 = bitcast i8* %malloccall383 to %tensor_t*
  %dtype385 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 0
  store i8 0, i8* %dtype385
  %ndims386 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 1
  store i8 0, i8* %ndims386
  %malloccall387 = tail call i8* @malloc(i32 0)
  %dims388 = bitcast i8* %malloccall387 to [0 x i64]*
  %dims_as_i8ptr389 = bitcast [0 x i64]* %dims388 to i8*
  %malloccall390 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
```

```
  %data391 = bitcast i8* %malloccall390 to [1 x i32]*
  %data_as_i8ptr392 = bitcast [1 x i32]* %data391 to i8*
  %elmptr393 = getelementptr [1 x i32], [1 x i32]* %data391, i32 0, i32 0
  store i32 0, i32* %elmptr393
  %dimsptr394 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 2
  store i8* %dims_as_i8ptr389, i8** %dimsptr394
  %dataptr395 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 3
  store i8* %data_as_i8ptr392, i8** %dataptr395
  %rc396 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor384, i32 0, i32 4
  store i8 0, i8* %rc396
  %tensor397 = bitcast %tensor_t* %raw_tensor384 to i8*
  %malloccall398 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor399 = bitcast i8* %malloccall398 to %tensor_t*
  %dtype400 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 0
  store i8 3, i8* %dtype400
  %ndims401 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 1
  store i8 1, i8* %ndims401
  %malloccall402 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims403 = bitcast i8* %malloccall402 to [1 x i64]*
  %dims_as_i8ptr404 = bitcast [1 x i64]* %dims403 to i8*
  %elmptr405 = getelementptr [1 x i64], [1 x i64]* %dims403, i64 0, i64 0
  store i64 1, i64* %elmptr405
  %malloccall406 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data407 = bitcast i8* %malloccall406 to [1 x i8*]*
  %data_as_i8ptr408 = bitcast [1 x i8*]* %data407 to i8*
  %elmptr409 = getelementptr [1 x i8*], [1 x i8*]* %data407, i64 0, i64 0
  store i8* %tensor397, i8** %elmptr409
  %dimsptr410 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 2
  store i8* %dims_as_i8ptr404, i8** %dimsptr410
  %dataptr411 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 3
  store i8* %data_as_i8ptr408, i8** %dataptr411
  %rc412 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor399, i32 0, i32 4
  store i8 0, i8* %rc412
  %tensor413 = bitcast %tensor_t* %raw_tensor399 to i8*
  %l1414 = load i8*, i8** %l1
  %access_tensor415 = call i8* @index_get(i8* %l1414, i8* %tensor413)
  %malloccall416 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor417 = bitcast i8* %malloccall416 to %tensor_t*
  %dtype418 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 0
  store i8 0, i8* %dtype418
```

```
  %ndims419 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 1
  store i8 0, i8* %ndims419
  %malloccall420 = tail call i8* @malloc(i32 0)
  %dims421 = bitcast i8* %malloccall420 to [0 x i64]*
  %dims_as_i8ptr422 = bitcast [0 x i64]* %dims421 to i8*
  %malloccall423 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data424 = bitcast i8* %malloccall423 to [1 x i32]*
  %data_as_i8ptr425 = bitcast [1 x i32]* %data424 to i8*
  %elmptr426 = getelementptr [1 x i32], [1 x i32]* %data424, i32 0, i32 0
  store i32 2, i32* %elmptr426
  %dimsptr427 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 2
  store i8* %dims_as_i8ptr422, i8** %dimsptr427
  %dataptr428 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 3
  store i8* %data_as_i8ptr425, i8** %dataptr428
  %rc429 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor417, i32 0, i32 4
  store i8 0, i8* %rc429
  %tensor430 = bitcast %tensor_t* %raw_tensor417 to i8*
  %tmpOp431 = call i8* @floordivide(i8* %access_tensor415, i8* %tensor430)
  %malloccall432 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor433 = bitcast i8* %malloccall432 to %tensor_t*
  %dtype434 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 0
  store i8 0, i8* %dtype434
  %ndims435 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 1
  store i8 0, i8* %ndims435
  %malloccall436 = tail call i8* @malloc(i32 0)
  %dims437 = bitcast i8* %malloccall436 to [0 x i64]*
  %dims_as_i8ptr438 = bitcast [0 x i64]* %dims437 to i8*
  %malloccall439 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data440 = bitcast i8* %malloccall439 to [1 x i32]*
  %data_as_i8ptr441 = bitcast [1 x i32]* %data440 to i8*
  %elmptr442 = getelementptr [1 x i32], [1 x i32]* %data440, i32 0, i32 0
  store i32 1, i32* %elmptr442
  %dimsptr443 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 2
  store i8* %dims_as_i8ptr438, i8** %dimsptr443
  %dataptr444 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 3
  store i8* %data_as_i8ptr441, i8** %dataptr444
  %rc445 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor433, i32 0, i32 4
  store i8 0, i8* %rc445
  %tensor446 = bitcast %tensor_t* %raw_tensor433 to i8*
  %tmpOp447 = call i8* @range(i8* %tensor382, i8* %tmpOp431, i8* %tensor446)
```

```
  %malloccall448 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor449 = bitcast i8* %malloccall448 to %tensor_t*
  %dtype450 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor449, i32 0, i32 0
  store i8 0, i8* %dtype450
  %ndims451 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor449, i32 0, i32 1
  store i8 0, i8* %ndims451
  %malloccall452 = tail call i8* @malloc(i32 0)
  %dims453 = bitcast i8* %malloccall452 to [0 x i64]*
  %dims_as_i8ptr454 = bitcast [0 x i64]* %dims453 to i8*
  %malloccall455 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data456 = bitcast i8* %malloccall455 to [1 x i32]*
  %data_as_i8ptr457 = bitcast [1 x i32]* %data456 to i8*
  %elmptr458 = getelementptr [1 x i32], [1 x i32]* %data456, i32 0, i32 0
  store i32 0, i32* %elmptr458
  %dimsptr459 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor449, i32 0, i32 2
  store i8* %dims_as_i8ptr454, i8** %dimsptr459
  %dataptr460 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor449, i32 0, i32 3
  store i8* %data_as_i8ptr457, i8** %dataptr460
  %rc461 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor449, i32 0, i32 4
  store i8 0, i8* %rc461
  %tensor462 = bitcast %tensor_t* %raw_tensor449 to i8*
  %malloccall463 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor464 = bitcast i8* %malloccall463 to %tensor_t*
  %dtype465 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor464, i32 0, i32 0
  store i8 0, i8* %dtype465
  %ndims466 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor464, i32 0, i32 1
  store i8 0, i8* %ndims466
  %malloccall467 = tail call i8* @malloc(i32 0)
  %dims468 = bitcast i8* %malloccall467 to [0 x i64]*
  %dims_as_i8ptr469 = bitcast [0 x i64]* %dims468 to i8*
  %malloccall470 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data471 = bitcast i8* %malloccall470 to [1 x i32]*
  %data_as_i8ptr472 = bitcast [1 x i32]* %data471 to i8*
  %elmptr473 = getelementptr [1 x i32], [1 x i32]* %data471, i32 0, i32 0
  store i32 1, i32* %elmptr473
  %dimsptr474 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor464, i32 0, i32 2
  store i8* %dims_as_i8ptr469, i8** %dimsptr474
  %dataptr475 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor464, i32 0, i32 3
  store i8* %data_as_i8ptr472, i8** %dataptr475
  %rc476 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor464, i32 0, i32 4
```

```
  store i8 0, i8* %rc476
  %tensor477 = bitcast %tensor_t* %raw_tensor464 to i8*
  %malloccall478 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor479 = bitcast i8* %malloccall478 to %tensor_t*
  %dtype480 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor479, i32 0, i32 0
  store i8 3, i8* %dtype480
  %ndims481 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor479, i32 0, i32 1
  store i8 1, i8* %ndims481
  %malloccall482 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims483 = bitcast i8* %malloccall482 to [1 x i64]*
  %dims_as_i8ptr484 = bitcast [1 x i64]* %dims483 to i8*
  %elmptr485 = getelementptr [1 x i64], [1 x i64]* %dims483, i64 0, i64 0
  store i64 1, i64* %elmptr485
  %malloccall486 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data487 = bitcast i8* %malloccall486 to [1 x i8*]*
  %data_as_i8ptr488 = bitcast [1 x i8*]* %data487 to i8*
  %elmptr489 = getelementptr [1 x i8*], [1 x i8*]* %data487, i64 0, i64 0
  store i8* %tensor477, i8** %elmptr489
  %dimsptr490 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor479, i32 0, i32 2
  store i8* %dims_as_i8ptr484, i8** %dimsptr490
  %dataptr491 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor479, i32 0, i32 3
  store i8* %data_as_i8ptr488, i8** %dataptr491
  %rc492 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor479, i32 0, i32 4
  store i8 0, i8* %rc492
  %tensor493 = bitcast %tensor_t* %raw_tensor479 to i8*
  %l1494 = load i8*, i8** %l1
  %access_tensor495 = call i8* @index_get(i8* %l1494, i8* %tensor493)
  %malloccall496 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor497 = bitcast i8* %malloccall496 to %tensor_t*
  %dtype498 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 0
  store i8 0, i8* %dtype498
  %ndims499 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 1
  store i8 0, i8* %ndims499
  %malloccall500 = tail call i8* @malloc(i32 0)
  %dims501 = bitcast i8* %malloccall500 to [0 x i64]*
  %dims_as_i8ptr502 = bitcast [0 x i64]* %dims501 to i8*
  %malloccall503 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data504 = bitcast i8* %malloccall503 to [1 x i32]*
  %data_as_i8ptr505 = bitcast [1 x i32]* %data504 to i8*
```

```
  %elmptr506 = getelementptr [1 x i32], [1 x i32]* %data504, i32 0, i32 0
  store i32 2, i32* %elmptr506
  %dimsptr507 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 2
  store i8* %dims_as_i8ptr502, i8** %dimsptr507
  %dataptr508 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 3
  store i8* %data_as_i8ptr505, i8** %dataptr508
  %rc509 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor497, i32 0, i32 4
  store i8 0, i8* %rc509
  %tensor510 = bitcast %tensor_t* %raw_tensor497 to i8*
  %tmpOp511 = call i8* @floordivide(i8* %access_tensor495, i8* %tensor510)
  %malloccall512 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor513 = bitcast i8* %malloccall512 to %tensor_t*
  %dtype514 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor513, i32 0, i32 0
  store i8 0, i8* %dtype514
  %ndims515 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor513, i32 0, i32 1
  store i8 0, i8* %ndims515
  %malloccall516 = tail call i8* @malloc(i32 0)
  %dims517 = bitcast i8* %malloccall516 to [0 x i64]*
  %dims_as_i8ptr518 = bitcast [0 x i64]* %dims517 to i8*
  %malloccall519 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data520 = bitcast i8* %malloccall519 to [1 x i32]*
  %data_as_i8ptr521 = bitcast [1 x i32]* %data520 to i8*
  %elmptr522 = getelementptr [1 x i32], [1 x i32]* %data520, i32 0, i32 0
  store i32 1, i32* %elmptr522
  %dimsptr523 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor513, i32 0, i32 2
  store i8* %dims_as_i8ptr518, i8** %dimsptr523
  %dataptr524 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor513, i32 0, i32 3
  store i8* %data_as_i8ptr521, i8** %dataptr524
  %rc525 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor513, i32 0, i32 4
  store i8 0, i8* %rc525
  %tensor526 = bitcast %tensor_t* %raw_tensor513 to i8*
  %tmpOp527 = call i8* @range(i8* %tensor462, i8* %tmpOp511, i8* %tensor526)
  %malloccall528 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor529 = bitcast i8* %malloccall528 to %tensor_t*
  %dtype530 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor529, i32 0, i32 0
  store i8 3, i8* %dtype530
  %ndims531 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor529, i32 0, i32 1
  store i8 1, i8* %ndims531
  %malloccall532 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims533 = bitcast i8* %malloccall532 to [1 x i64]*
```

```
  %dims_as_i8ptr534 = bitcast [1 x i64]* %dims533 to i8*
  %elmptr535 = getelementptr [1 x i64], [1 x i64]* %dims533, i64 0, i64 0
  store i64 2, i64* %elmptr535
  %malloccall536 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data537 = bitcast i8* %malloccall536 to [2 x i8*]*
  %data_as_i8ptr538 = bitcast [2 x i8*]* %data537 to i8*
  %elmptr539 = getelementptr [2 x i8*], [2 x i8*]* %data537, i64 0, i64 0
  store i8* %tmpOp447, i8** %elmptr539
  %elmptr540 = getelementptr [2 x i8*], [2 x i8*]* %data537, i64 0, i64 1
  store i8* %tmpOp527, i8** %elmptr540
  %dimsptr541 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor529, i32 0, i32 2
  store i8* %dims_as_i8ptr534, i8** %dimsptr541
  %dataptr542 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor529, i32 0, i32 3
  store i8* %data_as_i8ptr538, i8** %dataptr542
  %rc543 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor529, i32 0, i32 4
  store i8 0, i8* %rc543
  %tensor544 = bitcast %tensor_t* %raw_tensor529 to i8*
  %x545 = load i8*, i8** %x
  %access_tensor546 = call i8* @index_get(i8* %x545, i8* %tensor544)
  %x11 = alloca i8*
  store i8* null, i8** %x11
  %lhsptr547 = load i8*, i8** %x11
  call void @increase_rc(i8* %access_tensor546)
  call void @decrease_rc(i8* %lhsptr547)
  store i8* %access_tensor546, i8** %x11
  %malloccall548 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor549 = bitcast i8* %malloccall548 to %tensor_t*
  %dtype550 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 0
  store i8 0, i8* %dtype550
  %ndims551 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 1
  store i8 0, i8* %ndims551
  %malloccall552 = tail call i8* @malloc(i32 0)
  %dims553 = bitcast i8* %malloccall552 to [0 x i64]*
  %dims_as_i8ptr554 = bitcast [0 x i64]* %dims553 to i8*
  %malloccall555 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data556 = bitcast i8* %malloccall555 to [1 x i32]*
  %data_as_i8ptr557 = bitcast [1 x i32]* %data556 to i8*
  %elmptr558 = getelementptr [1 x i32], [1 x i32]* %data556, i32 0, i32 0
  store i32 0, i32* %elmptr558
  %dimsptr559 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 2
  store i8* %dims_as_i8ptr554, i8** %dimsptr559
```

```
  %dataptr560 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 3
  store i8* %data_as_i8ptr557, i8** %dataptr560
  %rc561 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor549, i32 0, i32 4
  store i8 0, i8* %rc561
  %tensor562 = bitcast %tensor_t* %raw_tensor549 to i8*
  %malloccall563 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor564 = bitcast i8* %malloccall563 to %tensor_t*
  %dtype565 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor564, i32 0, i32 0
  store i8 0, i8* %dtype565
  %ndims566 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor564, i32 0, i32 1
  store i8 0, i8* %ndims566
  %malloccall567 = tail call i8* @malloc(i32 0)
  %dims568 = bitcast i8* %malloccall567 to [0 x i64]*
  %dims_as_i8ptr569 = bitcast [0 x i64]* %dims568 to i8*
  %malloccall570 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data571 = bitcast i8* %malloccall570 to [1 x i32]*
  %data_as_i8ptr572 = bitcast [1 x i32]* %data571 to i8*
  %elmptr573 = getelementptr [1 x i32], [1 x i32]* %data571, i32 0, i32 0
  store i32 0, i32* %elmptr573
  %dimsptr574 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor564, i32 0, i32 2
  store i8* %dims_as_i8ptr569, i8** %dimsptr574
  %dataptr575 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor564, i32 0, i32 3
  store i8* %data_as_i8ptr572, i8** %dataptr575
  %rc576 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor564, i32 0, i32 4
  store i8 0, i8* %rc576
  %tensor577 = bitcast %tensor_t* %raw_tensor564 to i8*
  %malloccall578 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor579 = bitcast i8* %malloccall578 to %tensor_t*
  %dtype580 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor579, i32 0, i32 0
  store i8 3, i8* %dtype580
  %ndims581 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor579, i32 0, i32 1
  store i8 1, i8* %ndims581
  %malloccall582 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims583 = bitcast i8* %malloccall582 to [1 x i64]*
  %dims_as_i8ptr584 = bitcast [1 x i64]* %dims583 to i8*
  %elmptr585 = getelementptr [1 x i64], [1 x i64]* %dims583, i64 0, i64 0
  store i64 1, i64* %elmptr585
  %malloccall586 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data587 = bitcast i8* %malloccall586 to [1 x i8*]*
```

```
  %data_as_i8ptr588 = bitcast [1 x i8*]* %data587 to i8*
  %elmptr589 = getelementptr [1 x i8*], [1 x i8*]* %data587, i64 0, i64 0
  store i8* %tensor577, i8** %elmptr589
  %dimsptr590 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor579, i32 0, i32 2
  store i8* %dims_as_i8ptr584, i8** %dimsptr590
  %dataptr591 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor579, i32 0, i32 3
  store i8* %data_as_i8ptr588, i8** %dataptr591
  %rc592 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor579, i32 0, i32 4
  store i8 0, i8* %rc592
  %tensor593 = bitcast %tensor_t* %raw_tensor579 to i8*
  %l2594 = load i8*, i8** %l2
  %access_tensor595 = call i8* @index_get(i8* %l2594, i8* %tensor593)
  %malloccall596 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor597 = bitcast i8* %malloccall596 to %tensor_t*
  %dtype598 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor597, i32 0, i32 0
  store i8 0, i8* %dtype598
  %ndims599 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor597, i32 0, i32 1
  store i8 0, i8* %ndims599
  %malloccall600 = tail call i8* @malloc(i32 0)
  %dims601 = bitcast i8* %malloccall600 to [0 x i64]*
  %dims_as_i8ptr602 = bitcast [0 x i64]* %dims601 to i8*
  %malloccall603 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data604 = bitcast i8* %malloccall603 to [1 x i32]*
  %data_as_i8ptr605 = bitcast [1 x i32]* %data604 to i8*
  %elmptr606 = getelementptr [1 x i32], [1 x i32]* %data604, i32 0, i32 0
  store i32 2, i32* %elmptr606
  %dimsptr607 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor597, i32 0, i32 2
  store i8* %dims_as_i8ptr602, i8** %dimsptr607
  %dataptr608 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor597, i32 0, i32 3
  store i8* %data_as_i8ptr605, i8** %dataptr608
  %rc609 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor597, i32 0, i32 4
  store i8 0, i8* %rc609
  %tensor610 = bitcast %tensor_t* %raw_tensor597 to i8*
  %tmpOp611 = call i8* @floordivide(i8* %access_tensor595, i8* %tensor610)
  %malloccall612 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor613 = bitcast i8* %malloccall612 to %tensor_t*
  %dtype614 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor613, i32 0, i32 0
  store i8 0, i8* %dtype614
  %ndims615 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor613, i32 0, i32 1
  store i8 0, i8* %ndims615
  %malloccall616 = tail call i8* @malloc(i32 0)
```

```
  %dims617 = bitcast i8* %malloccall616 to [0 x i64]*
  %dims_as_i8ptr618 = bitcast [0 x i64]* %dims617 to i8*
  %malloccall619 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data620 = bitcast i8* %malloccall619 to [1 x i32]*
  %data_as_i8ptr621 = bitcast [1 x i32]* %data620 to i8*
  %elmptr622 = getelementptr [1 x i32], [1 x i32]* %data620, i32 0, i32 0
  store i32 1, i32* %elmptr622
  %dimsptr623 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor613, i32 0, i32 2
  store i8* %dims_as_i8ptr618, i8** %dimsptr623
  %dataptr624 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor613, i32 0, i32 3
  store i8* %data_as_i8ptr621, i8** %dataptr624
  %rc625 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor613, i32 0, i32 4
  store i8 0, i8* %rc625
  %tensor626 = bitcast %tensor_t* %raw_tensor613 to i8*
  %tmpOp627 = call i8* @range(i8* %tensor562, i8* %tmpOp611, i8* %tensor626)
  %malloccall628 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor629 = bitcast i8* %malloccall628 to %tensor_t*
  %dtype630 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor629, i32 0, i32 0
  store i8 0, i8* %dtype630
  %ndims631 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor629, i32 0, i32 1
  store i8 0, i8* %ndims631
  %malloccall632 = tail call i8* @malloc(i32 0)
  %dims633 = bitcast i8* %malloccall632 to [0 x i64]*
  %dims_as_i8ptr634 = bitcast [0 x i64]* %dims633 to i8*
  %malloccall635 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data636 = bitcast i8* %malloccall635 to [1 x i32]*
  %data_as_i8ptr637 = bitcast [1 x i32]* %data636 to i8*
  %elmptr638 = getelementptr [1 x i32], [1 x i32]* %data636, i32 0, i32 0
  store i32 1, i32* %elmptr638
  %dimsptr639 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor629, i32 0, i32 2
  store i8* %dims_as_i8ptr634, i8** %dimsptr639
  %dataptr640 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor629, i32 0, i32 3
  store i8* %data_as_i8ptr637, i8** %dataptr640
  %rc641 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor629, i32 0, i32 4
  store i8 0, i8* %rc641
  %tensor642 = bitcast %tensor_t* %raw_tensor629 to i8*
  %malloccall643 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor644 = bitcast i8* %malloccall643 to %tensor_t*
  %dtype645 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor644, i32 0, i32 0
  store i8 3, i8* %dtype645
```

```
  %ndims646 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor644, i32 0, i32 1
  store i8 1, i8* %ndims646
  %malloccall647 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims648 = bitcast i8* %malloccall647 to [1 x i64]*
  %dims_as_i8ptr649 = bitcast [1 x i64]* %dims648 to i8*
  %elmptr650 = getelementptr [1 x i64], [1 x i64]* %dims648, i64 0, i64 0
  store i64 1, i64* %elmptr650
  %malloccall651 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data652 = bitcast i8* %malloccall651 to [1 x i8*]*
  %data_as_i8ptr653 = bitcast [1 x i8*]* %data652 to i8*
  %elmptr654 = getelementptr [1 x i8*], [1 x i8*]* %data652, i64 0, i64 0
  store i8* %tensor642, i8** %elmptr654
  %dimsptr655 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor644, i32 0, i32 2
  store i8* %dims_as_i8ptr649, i8** %dimsptr655
  %dataptr656 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor644, i32 0, i32 3
  store i8* %data_as_i8ptr653, i8** %dataptr656
  %rc657 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor644, i32 0, i32 4
  store i8 0, i8* %rc657
  %tensor658 = bitcast %tensor_t* %raw_tensor644 to i8*
  %l2659 = load i8*, i8** %l2
  %access_tensor660 = call i8* @index_get(i8* %l2659, i8* %tensor658)
  %malloccall661 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor662 = bitcast i8* %malloccall661 to %tensor_t*
  %dtype663 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor662, i32 0, i32 0
  store i8 0, i8* %dtype663
  %ndims664 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor662, i32 0, i32 1
  store i8 0, i8* %ndims664
  %malloccall665 = tail call i8* @malloc(i32 0)
  %dims666 = bitcast i8* %malloccall665 to [0 x i64]*
  %dims_as_i8ptr667 = bitcast [0 x i64]* %dims666 to i8*
  %malloccall668 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data669 = bitcast i8* %malloccall668 to [1 x i32]*
  %data_as_i8ptr670 = bitcast [1 x i32]* %data669 to i8*
  %elmptr671 = getelementptr [1 x i32], [1 x i32]* %data669, i32 0, i32 0
  store i32 2, i32* %elmptr671
  %dimsptr672 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor662, i32 0, i32 2
  store i8* %dims_as_i8ptr667, i8** %dimsptr672
  %dataptr673 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor662, i32 0, i32 3
  store i8* %data_as_i8ptr670, i8** %dataptr673
  %rc674 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor662, i32 0, i32 4
```

```
  store i8 0, i8* %rc674
  %tensor675 = bitcast %tensor_t* %raw_tensor662 to i8*
  %tmpOp676 = call i8* @floordivide(i8* %access_tensor660, i8* %tensor675)
  %malloccall677 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor678 = bitcast i8* %malloccall677 to %tensor_t*
  %dtype679 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor678, i32 0, i32 0
  store i8 0, i8* %dtype679
  %ndims680 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor678, i32 0, i32 1
  store i8 0, i8* %ndims680
  %malloccall681 = tail call i8* @malloc(i32 0)
  %dims682 = bitcast i8* %malloccall681 to [0 x i64]*
  %dims_as_i8ptr683 = bitcast [0 x i64]* %dims682 to i8*
  %malloccall684 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data685 = bitcast i8* %malloccall684 to [1 x i32]*
  %data_as_i8ptr686 = bitcast [1 x i32]* %data685 to i8*
  %elmptr687 = getelementptr [1 x i32], [1 x i32]* %data685, i32 0, i32 0
  store i32 1, i32* %elmptr687
  %dimsptr688 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor678, i32 0, i32 2
  store i8* %dims_as_i8ptr683, i8** %dimsptr688
  %dataptr689 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor678, i32 0, i32 3
  store i8* %data_as_i8ptr686, i8** %dataptr689
  %rc690 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor678, i32 0, i32 4
  store i8 0, i8* %rc690
  %tensor691 = bitcast %tensor_t* %raw_tensor678 to i8*
  %malloccall692 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor693 = bitcast i8* %malloccall692 to %tensor_t*
  %dtype694 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor693, i32 0, i32 0
  store i8 3, i8* %dtype694
  %ndims695 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor693, i32 0, i32 1
  store i8 1, i8* %ndims695
  %malloccall696 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims697 = bitcast i8* %malloccall696 to [1 x i64]*
  %dims_as_i8ptr698 = bitcast [1 x i64]* %dims697 to i8*
  %elmptr699 = getelementptr [1 x i64], [1 x i64]* %dims697, i64 0, i64 0
  store i64 1, i64* %elmptr699
  %malloccall700 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data701 = bitcast i8* %malloccall700 to [1 x i8*]*
  %data_as_i8ptr702 = bitcast [1 x i8*]* %data701 to i8*
  %elmptr703 = getelementptr [1 x i8*], [1 x i8*]* %data701, i64 0, i64 0
```

```
  store i8* %tensor691, i8** %elmptr703
  %dimsptr704 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor693, i32 0, i32 2
  store i8* %dims_as_i8ptr698, i8** %dimsptr704
  %dataptr705 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor693, i32 0, i32 3
  store i8* %data_as_i8ptr702, i8** %dataptr705
  %rc706 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor693, i32 0, i32 4
  store i8 0, i8* %rc706
  %tensor707 = bitcast %tensor_t* %raw_tensor693 to i8*
  %l2708 = load i8*, i8** %l2
  %access_tensor709 = call i8* @index_get(i8* %l2708, i8* %tensor707)
  %malloccall710 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor711 = bitcast i8* %malloccall710 to %tensor_t*
  %dtype712 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor711, i32 0, i32 0
  store i8 0, i8* %dtype712
  %ndims713 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor711, i32 0, i32 1
  store i8 0, i8* %ndims713
  %malloccall714 = tail call i8* @malloc(i32 0)
  %dims715 = bitcast i8* %malloccall714 to [0 x i64]*
  %dims_as_i8ptr716 = bitcast [0 x i64]* %dims715 to i8*
  %malloccall717 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data718 = bitcast i8* %malloccall717 to [1 x i32]*
  %data_as_i8ptr719 = bitcast [1 x i32]* %data718 to i8*
  %elmptr720 = getelementptr [1 x i32], [1 x i32]* %data718, i32 0, i32 0
  store i32 1, i32* %elmptr720
  %dimsptr721 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor711, i32 0, i32 2
  store i8* %dims_as_i8ptr716, i8** %dimsptr721
  %dataptr722 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor711, i32 0, i32 3
  store i8* %data_as_i8ptr719, i8** %dataptr722
  %rc723 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor711, i32 0, i32 4
  store i8 0, i8* %rc723
  %tensor724 = bitcast %tensor_t* %raw_tensor711 to i8*
  %tmpOp725 = call i8* @range(i8* %tmpOp676, i8* %access_tensor709, i8* %tensor724)
  %malloccall726 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor727 = bitcast i8* %malloccall726 to %tensor_t*
  %dtype728 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor727, i32 0, i32 0
  store i8 3, i8* %dtype728
  %ndims729 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor727, i32 0, i32 1
  store i8 1, i8* %ndims729
  %malloccall730 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims731 = bitcast i8* %malloccall730 to [1 x i64]*
```

```
  %dims_as_i8ptr732 = bitcast [1 x i64]* %dims731 to i8*
  %elmptr733 = getelementptr [1 x i64], [1 x i64]* %dims731, i64 0, i64 0
  store i64 2, i64* %elmptr733
  %malloccall734 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data735 = bitcast i8* %malloccall734 to [2 x i8*]*
  %data_as_i8ptr736 = bitcast [2 x i8*]* %data735 to i8*
  %elmptr737 = getelementptr [2 x i8*], [2 x i8*]* %data735, i64 0, i64 0
  store i8* %tmpOp627, i8** %elmptr737
  %elmptr738 = getelementptr [2 x i8*], [2 x i8*]* %data735, i64 0, i64 1
  store i8* %tmpOp725, i8** %elmptr738
  %dimsptr739 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor727, i32 0, i32 2
  store i8* %dims_as_i8ptr732, i8** %dimsptr739
  %dataptr740 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor727, i32 0, i32 3
  store i8* %data_as_i8ptr736, i8** %dataptr740
  %rc741 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor727, i32 0, i32 4
  store i8 0, i8* %rc741
  %tensor742 = bitcast %tensor_t* %raw_tensor727 to i8*
  %y743 = load i8*, i8** %y
  %access_tensor744 = call i8* @index_get(i8* %y743, i8* %tensor742)
  %y12 = alloca i8*
  store i8* null, i8** %y12
  %lhsptr745 = load i8*, i8** %y12
  call void @increase_rc(i8* %access_tensor744)
  call void @decrease_rc(i8* %lhsptr745)
  store i8* %access_tensor744, i8** %y12
  %x21746 = load i8*, i8** %x21
  %x11747 = load i8*, i8** %x11
  %tmpOp748 = call i8* @subtract(i8* %x21746, i8* %x11747)
  %y11749 = load i8*, i8** %y11
  %y12750 = load i8*, i8** %y12
  %tmpOp751 = call i8* @add(i8* %y11749, i8* %y12750)
  %tmpOp752 = call i8* @mult(i8* %tmpOp748, i8* %tmpOp751)
  ret i8* %tmpOp752
}

define i8* @StrassenMULf7(i8* %0, i8* %1) {
entry:
  %x = alloca i8*
  store i8* %0, i8** %x
  %y = alloca i8*
  store i8* %1, i8** %y
  %x1 = load i8*, i8** %x
  %shape = call i8* @shape(i8* %x1)
```

```
  %l1 = alloca i8*
  store i8* null, i8** %l1
  %lhsptr = load i8*, i8** %l1
  call void @increase_rc(i8* %shape)
  call void @decrease_rc(i8* %lhsptr)
  store i8* %shape, i8** %l1
  %y2 = load i8*, i8** %y
  %shape3 = call i8* @shape(i8* %y2)
  %l2 = alloca i8*
  store i8* null, i8** %l2
  %lhsptr4 = load i8*, i8** %l2
  call void @increase_rc(i8* %shape3)
  call void @decrease_rc(i8* %lhsptr4)
  store i8* %shape3, i8** %l2
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall5 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall5 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall6 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data = bitcast i8* %malloccall6 to [1 x i32]*
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor8 = bitcast i8* %malloccall7 to %tensor_t*
  %dtype9 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 0
  store i8 3, i8* %dtype9
  %ndims10 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 1
  store i8 1, i8* %ndims10
```

```
  %malloccall11 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims12 = bitcast i8* %malloccall11 to [1 x i64]*
  %dims_as_i8ptr13 = bitcast [1 x i64]* %dims12 to i8*
  %elmptr14 = getelementptr [1 x i64], [1 x i64]* %dims12, i64 0, i64 0
  store i64 1, i64* %elmptr14
  %malloccall15 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data16 = bitcast i8* %malloccall15 to [1 x i8*]*
  %data_as_i8ptr17 = bitcast [1 x i8*]* %data16 to i8*
  %elmptr18 = getelementptr [1 x i8*], [1 x i8*]* %data16, i64 0, i64 0
  store i8* %tensor, i8** %elmptr18
  %dimsptr19 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 2
  store i8* %dims_as_i8ptr13, i8** %dimsptr19
  %dataptr20 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 3
  store i8* %data_as_i8ptr17, i8** %dataptr20
  %rc21 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor8, i32 0, i32 4
  store i8 0, i8* %rc21
  %tensor22 = bitcast %tensor_t* %raw_tensor8 to i8*
  %l123 = load i8*, i8** %l1
  %access_tensor = call i8* @index_get(i8* %l123, i8* %tensor22)
  %malloccall24 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor25 = bitcast i8* %malloccall24 to %tensor_t*
  %dtype26 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 0
  store i8 0, i8* %dtype26
  %ndims27 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 1
  store i8 0, i8* %ndims27
  %malloccall28 = tail call i8* @malloc(i32 0)
  %dims29 = bitcast i8* %malloccall28 to [0 x i64]*
  %dims_as_i8ptr30 = bitcast [0 x i64]* %dims29 to i8*
  %malloccall31 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data32 = bitcast i8* %malloccall31 to [1 x i32]*
  %data_as_i8ptr33 = bitcast [1 x i32]* %data32 to i8*
  %elmptr34 = getelementptr [1 x i32], [1 x i32]* %data32, i32 0, i32 0
  store i32 2, i32* %elmptr34
  %dimsptr35 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 2
  store i8* %dims_as_i8ptr30, i8** %dimsptr35
  %dataptr36 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 3
  store i8* %data_as_i8ptr33, i8** %dataptr36
  %rc37 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor25, i32 0, i32 4
  store i8 0, i8* %rc37
  %tensor38 = bitcast %tensor_t* %raw_tensor25 to i8*
  %tmpOp = call i8* @floordivide(i8* %access_tensor, i8* %tensor38)
```

```
  %malloccall39 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor40 = bitcast i8* %malloccall39 to %tensor_t*
  %dtype41 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 0
  store i8 0, i8* %dtype41
  %ndims42 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 1
  store i8 0, i8* %ndims42
  %malloccall43 = tail call i8* @malloc(i32 0)
  %dims44 = bitcast i8* %malloccall43 to [0 x i64]*
  %dims_as_i8ptr45 = bitcast [0 x i64]* %dims44 to i8*
  %malloccall46 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data47 = bitcast i8* %malloccall46 to [1 x i32]*
  %data_as_i8ptr48 = bitcast [1 x i32]* %data47 to i8*
  %elmptr49 = getelementptr [1 x i32], [1 x i32]* %data47, i32 0, i32 0
  store i32 0, i32* %elmptr49
  %dimsptr50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 2
  store i8* %dims_as_i8ptr45, i8** %dimsptr50
  %dataptr51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 3
  store i8* %data_as_i8ptr48, i8** %dataptr51
  %rc52 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor40, i32 0, i32 4
  store i8 0, i8* %rc52
  %tensor53 = bitcast %tensor_t* %raw_tensor40 to i8*
  %malloccall54 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor55 = bitcast i8* %malloccall54 to %tensor_t*
  %dtype56 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 0
  store i8 3, i8* %dtype56
  %ndims57 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 1
  store i8 1, i8* %ndims57
  %malloccall58 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims59 = bitcast i8* %malloccall58 to [1 x i64]*
  %dims_as_i8ptr60 = bitcast [1 x i64]* %dims59 to i8*
  %elmptr61 = getelementptr [1 x i64], [1 x i64]* %dims59, i64 0, i64 0
  store i64 1, i64* %elmptr61
  %malloccall62 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i32))
  %data63 = bitcast i8* %malloccall62 to [1 x i8*]*
  %data_as_i8ptr64 = bitcast [1 x i8*]* %data63 to i8*
  %elmptr65 = getelementptr [1 x i8*], [1 x i8*]* %data63, i64 0, i64 0
  store i8* %tensor53, i8** %elmptr65
  %dimsptr66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 2
  store i8* %dims_as_i8ptr60, i8** %dimsptr66
  %dataptr67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 3
```

```
  store i8* %data_as_i8ptr64, i8** %dataptr67
  %rc68 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor55, i32 0, i32 4
  store i8 0, i8* %rc68
  %tensor69 = bitcast %tensor_t* %raw_tensor55 to i8*
  %l170 = load i8*, i8** %l1
  %access_tensor71 = call i8* @index_get(i8* %l170, i8* %tensor69)
  %malloccall72 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor73 = bitcast i8* %malloccall72 to %tensor_t*
  %dtype74 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 0
  store i8 0, i8* %dtype74
  %ndims75 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 1
  store i8 0, i8* %ndims75
  %malloccall76 = tail call i8* @malloc(i32 0)
  %dims77 = bitcast i8* %malloccall76 to [0 x i64]*
  %dims_as_i8ptr78 = bitcast [0 x i64]* %dims77 to i8*
  %malloccall79 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data80 = bitcast i8* %malloccall79 to [1 x i32]*
  %data_as_i8ptr81 = bitcast [1 x i32]* %data80 to i8*
  %elmptr82 = getelementptr [1 x i32], [1 x i32]* %data80, i32 0, i32 0
  store i32 1, i32* %elmptr82
  %dimsptr83 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 2
  store i8* %dims_as_i8ptr78, i8** %dimsptr83
  %dataptr84 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 3
  store i8* %data_as_i8ptr81, i8** %dataptr84
  %rc85 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor73, i32 0, i32 4
  store i8 0, i8* %rc85
  %tensor86 = bitcast %tensor_t* %raw_tensor73 to i8*
  %tmpOp87 = call i8* @range(i8* %tmpOp, i8* %access_tensor71, i8* %tensor86)
  %malloccall88 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor89 = bitcast i8* %malloccall88 to %tensor_t*
  %dtype90 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 0
  store i8 0, i8* %dtype90
  %ndims91 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 1
  store i8 0, i8* %ndims91
  %malloccall92 = tail call i8* @malloc(i32 0)
  %dims93 = bitcast i8* %malloccall92 to [0 x i64]*
  %dims_as_i8ptr94 = bitcast [0 x i64]* %dims93 to i8*
  %malloccall95 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data96 = bitcast i8* %malloccall95 to [1 x i32]*
  %data_as_i8ptr97 = bitcast [1 x i32]* %data96 to i8*
```

```
  %elmptr98 = getelementptr [1 x i32], [1 x i32]* %data96, i32 0, i32 0
  store i32 1, i32* %elmptr98
  %dimsptr99 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 2
  store i8* %dims_as_i8ptr94, i8** %dimsptr99
  %dataptr100 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 3
  store i8* %data_as_i8ptr97, i8** %dataptr100
  %rc101 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor89, i32 0, i32 4
  store i8 0, i8* %rc101
  %tensor102 = bitcast %tensor_t* %raw_tensor89 to i8*
  %malloccall103 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor104 = bitcast i8* %malloccall103 to %tensor_t*
  %dtype105 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 0
  store i8 3, i8* %dtype105
  %ndims106 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 1
  store i8 1, i8* %ndims106
  %malloccall107 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims108 = bitcast i8* %malloccall107 to [1 x i64]*
  %dims_as_i8ptr109 = bitcast [1 x i64]* %dims108 to i8*
  %elmptr110 = getelementptr [1 x i64], [1 x i64]* %dims108, i64 0, i64 0
  store i64 1, i64* %elmptr110
  %malloccall111 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data112 = bitcast i8* %malloccall111 to [1 x i8*]*
  %data_as_i8ptr113 = bitcast [1 x i8*]* %data112 to i8*
  %elmptr114 = getelementptr [1 x i8*], [1 x i8*]* %data112, i64 0, i64 0
  store i8* %tensor102, i8** %elmptr114
  %dimsptr115 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 2
  store i8* %dims_as_i8ptr109, i8** %dimsptr115
  %dataptr116 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 3
  store i8* %data_as_i8ptr113, i8** %dataptr116
  %rc117 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor104, i32 0, i32 4
  store i8 0, i8* %rc117
  %tensor118 = bitcast %tensor_t* %raw_tensor104 to i8*
  %l1119 = load i8*, i8** %l1
  %access_tensor120 = call i8* @index_get(i8* %l1119, i8* %tensor118)
  %malloccall121 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor122 = bitcast i8* %malloccall121 to %tensor_t*
  %dtype123 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 0
  store i8 0, i8* %dtype123
  %ndims124 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 1
  store i8 0, i8* %ndims124
```

```
  %malloccall125 = tail call i8* @malloc(i32 0)
  %dims126 = bitcast i8* %malloccall125 to [0 x i64]*
  %dims_as_i8ptr127 = bitcast [0 x i64]* %dims126 to i8*
  %malloccall128 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data129 = bitcast i8* %malloccall128 to [1 x i32]*
  %data_as_i8ptr130 = bitcast [1 x i32]* %data129 to i8*
  %elmptr131 = getelementptr [1 x i32], [1 x i32]* %data129, i32 0, i32 0
  store i32 2, i32* %elmptr131
  %dimsptr132 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 2
  store i8* %dims_as_i8ptr127, i8** %dimsptr132
  %dataptr133 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 3
  store i8* %data_as_i8ptr130, i8** %dataptr133
  %rc134 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor122, i32 0, i32 4
  store i8 0, i8* %rc134
  %tensor135 = bitcast %tensor_t* %raw_tensor122 to i8*
  %tmpOp136 = call i8* @floordivide(i8* %access_tensor120, i8* %tensor135)
  %malloccall137 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor138 = bitcast i8* %malloccall137 to %tensor_t*
  %dtype139 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 0
  store i8 0, i8* %dtype139
  %ndims140 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 1
  store i8 0, i8* %ndims140
  %malloccall141 = tail call i8* @malloc(i32 0)
  %dims142 = bitcast i8* %malloccall141 to [0 x i64]*
  %dims_as_i8ptr143 = bitcast [0 x i64]* %dims142 to i8*
  %malloccall144 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data145 = bitcast i8* %malloccall144 to [1 x i32]*
  %data_as_i8ptr146 = bitcast [1 x i32]* %data145 to i8*
  %elmptr147 = getelementptr [1 x i32], [1 x i32]* %data145, i32 0, i32 0
  store i32 1, i32* %elmptr147
  %dimsptr148 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 2
  store i8* %dims_as_i8ptr143, i8** %dimsptr148
  %dataptr149 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 3
  store i8* %data_as_i8ptr146, i8** %dataptr149
  %rc150 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor138, i32 0, i32 4
  store i8 0, i8* %rc150
  %tensor151 = bitcast %tensor_t* %raw_tensor138 to i8*
  %malloccall152 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor153 = bitcast i8* %malloccall152 to %tensor_t*
  %dtype154 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 0
```

```
  store i8 3, i8* %dtype154
  %ndims155 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 1
  store i8 1, i8* %ndims155
  %malloccall156 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims157 = bitcast i8* %malloccall156 to [1 x i64]*
  %dims_as_i8ptr158 = bitcast [1 x i64]* %dims157 to i8*
  %elmptr159 = getelementptr [1 x i64], [1 x i64]* %dims157, i64 0, i64 0
  store i64 1, i64* %elmptr159
  %malloccall160 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data161 = bitcast i8* %malloccall160 to [1 x i8*]*
  %data_as_i8ptr162 = bitcast [1 x i8*]* %data161 to i8*
  %elmptr163 = getelementptr [1 x i8*], [1 x i8*]* %data161, i64 0, i64 0
  store i8* %tensor151, i8** %elmptr163
  %dimsptr164 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 2
  store i8* %dims_as_i8ptr158, i8** %dimsptr164
  %dataptr165 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 3
  store i8* %data_as_i8ptr162, i8** %dataptr165
  %rc166 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor153, i32 0, i32 4
  store i8 0, i8* %rc166
  %tensor167 = bitcast %tensor_t* %raw_tensor153 to i8*
  %l1168 = load i8*, i8** %l1
  %access_tensor169 = call i8* @index_get(i8* %l1168, i8* %tensor167)
  %malloccall170 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor171 = bitcast i8* %malloccall170 to %tensor_t*
  %dtype172 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 0
  store i8 0, i8* %dtype172
  %ndims173 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 1
  store i8 0, i8* %ndims173
  %malloccall174 = tail call i8* @malloc(i32 0)
  %dims175 = bitcast i8* %malloccall174 to [0 x i64]*
  %dims_as_i8ptr176 = bitcast [0 x i64]* %dims175 to i8*
  %malloccall177 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data178 = bitcast i8* %malloccall177 to [1 x i32]*
  %data_as_i8ptr179 = bitcast [1 x i32]* %data178 to i8*
  %elmptr180 = getelementptr [1 x i32], [1 x i32]* %data178, i32 0, i32 0
  store i32 1, i32* %elmptr180
  %dimsptr181 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 2
  store i8* %dims_as_i8ptr176, i8** %dimsptr181
  %dataptr182 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 3
  store i8* %data_as_i8ptr179, i8** %dataptr182
```

```
  %rc183 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor171, i32 0, i32 4
  store i8 0, i8* %rc183
  %tensor184 = bitcast %tensor_t* %raw_tensor171 to i8*
  %tmpOp185 = call i8* @range(i8* %tmpOp136, i8* %access_tensor169, i8* %tensor184)
  %malloccall186 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor187 = bitcast i8* %malloccall186 to %tensor_t*
  %dtype188 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 0
  store i8 3, i8* %dtype188
  %ndims189 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 1
  store i8 1, i8* %ndims189
  %malloccall190 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims191 = bitcast i8* %malloccall190 to [1 x i64]*
  %dims_as_i8ptr192 = bitcast [1 x i64]* %dims191 to i8*
  %elmptr193 = getelementptr [1 x i64], [1 x i64]* %dims191, i64 0, i64 0
  store i64 2, i64* %elmptr193
  %malloccall194 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data195 = bitcast i8* %malloccall194 to [2 x i8*]*
  %data_as_i8ptr196 = bitcast [2 x i8*]* %data195 to i8*
  %elmptr197 = getelementptr [2 x i8*], [2 x i8*]* %data195, i64 0, i64 0
  store i8* %tmpOp87, i8** %elmptr197
  %elmptr198 = getelementptr [2 x i8*], [2 x i8*]* %data195, i64 0, i64 1
  store i8* %tmpOp185, i8** %elmptr198
  %dimsptr199 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 2
  store i8* %dims_as_i8ptr192, i8** %dimsptr199
  %dataptr200 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 3
  store i8* %data_as_i8ptr196, i8** %dataptr200
  %rc201 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor187, i32 0, i32 4
  store i8 0, i8* %rc201
  %tensor202 = bitcast %tensor_t* %raw_tensor187 to i8*
  %x203 = load i8*, i8** %x
  %access_tensor204 = call i8* @index_get(i8* %x203, i8* %tensor202)
  %x22 = alloca i8*
  store i8* null, i8** %x22
  %lhsptr205 = load i8*, i8** %x22
  call void @increase_rc(i8* %access_tensor204)
  call void @decrease_rc(i8* %lhsptr205)
  store i8* %access_tensor204, i8** %x22
  %malloccall206 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor207 = bitcast i8* %malloccall206 to %tensor_t*
  %dtype208 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 0
```

```
  store i8 0, i8* %dtype208
  %ndims209 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 1
  store i8 0, i8* %ndims209
  %malloccall210 = tail call i8* @malloc(i32 0)
  %dims211 = bitcast i8* %malloccall210 to [0 x i64]*
  %dims_as_i8ptr212 = bitcast [0 x i64]* %dims211 to i8*
  %malloccall213 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data214 = bitcast i8* %malloccall213 to [1 x i32]*
  %data_as_i8ptr215 = bitcast [1 x i32]* %data214 to i8*
  %elmptr216 = getelementptr [1 x i32], [1 x i32]* %data214, i32 0, i32 0
  store i32 0, i32* %elmptr216
  %dimsptr217 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 2
  store i8* %dims_as_i8ptr212, i8** %dimsptr217
  %dataptr218 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 3
  store i8* %data_as_i8ptr215, i8** %dataptr218
  %rc219 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor207, i32 0, i32 4
  store i8 0, i8* %rc219
  %tensor220 = bitcast %tensor_t* %raw_tensor207 to i8*
  %malloccall221 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor222 = bitcast i8* %malloccall221 to %tensor_t*
  %dtype223 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 0
  store i8 3, i8* %dtype223
  %ndims224 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 1
  store i8 1, i8* %ndims224
  %malloccall225 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims226 = bitcast i8* %malloccall225 to [1 x i64]*
  %dims_as_i8ptr227 = bitcast [1 x i64]* %dims226 to i8*
  %elmptr228 = getelementptr [1 x i64], [1 x i64]* %dims226, i64 0, i64 0
  store i64 1, i64* %elmptr228
  %malloccall229 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data230 = bitcast i8* %malloccall229 to [1 x i8*]*
  %data_as_i8ptr231 = bitcast [1 x i8*]* %data230 to i8*
  %elmptr232 = getelementptr [1 x i8*], [1 x i8*]* %data230, i64 0, i64 0
  store i8* %tensor220, i8** %elmptr232
  %dimsptr233 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 2
  store i8* %dims_as_i8ptr227, i8** %dimsptr233
  %dataptr234 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 3
  store i8* %data_as_i8ptr231, i8** %dataptr234
  %rc235 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor222, i32 0, i32 4
  store i8 0, i8* %rc235
```

```
  %tensor236 = bitcast %tensor_t* %raw_tensor222 to i8*
  %l2237 = load i8*, i8** %l2
  %access_tensor238 = call i8* @index_get(i8* %l2237, i8* %tensor236)
  %malloccall239 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor240 = bitcast i8* %malloccall239 to %tensor_t*
  %dtype241 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor240, i32 0, i32 0
  store i8 0, i8* %dtype241
  %ndims242 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor240, i32 0, i32 1
  store i8 0, i8* %ndims242
  %malloccall243 = tail call i8* @malloc(i32 0)
  %dims244 = bitcast i8* %malloccall243 to [0 x i64]*
  %dims_as_i8ptr245 = bitcast [0 x i64]* %dims244 to i8*
  %malloccall246 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data247 = bitcast i8* %malloccall246 to [1 x i32]*
  %data_as_i8ptr248 = bitcast [1 x i32]* %data247 to i8*
  %elmptr249 = getelementptr [1 x i32], [1 x i32]* %data247, i32 0, i32 0
  store i32 2, i32* %elmptr249
  %dimsptr250 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor240, i32 0, i32 2
  store i8* %dims_as_i8ptr245, i8** %dimsptr250
  %dataptr251 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor240, i32 0, i32 3
  store i8* %data_as_i8ptr248, i8** %dataptr251
  %rc252 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor240, i32 0, i32 4
  store i8 0, i8* %rc252
  %tensor253 = bitcast %tensor_t* %raw_tensor240 to i8*
  %tmpOp254 = call i8* @floordivide(i8* %access_tensor238, i8* %tensor253)
  %malloccall255 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor256 = bitcast i8* %malloccall255 to %tensor_t*
  %dtype257 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor256, i32 0, i32 0
  store i8 0, i8* %dtype257
  %ndims258 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor256, i32 0, i32 1
  store i8 0, i8* %ndims258
  %malloccall259 = tail call i8* @malloc(i32 0)
  %dims260 = bitcast i8* %malloccall259 to [0 x i64]*
  %dims_as_i8ptr261 = bitcast [0 x i64]* %dims260 to i8*
  %malloccall262 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data263 = bitcast i8* %malloccall262 to [1 x i32]*
  %data_as_i8ptr264 = bitcast [1 x i32]* %data263 to i8*
  %elmptr265 = getelementptr [1 x i32], [1 x i32]* %data263, i32 0, i32 0
  store i32 0, i32* %elmptr265
  %dimsptr266 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor256, i32 0, i32 2
```

```
  store i8* %dims_as_i8ptr261, i8** %dimsptr266
  %dataptr267 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor256, i32 0, i32 3
  store i8* %data_as_i8ptr264, i8** %dataptr267
  %rc268 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor256, i32 0, i32 4
  store i8 0, i8* %rc268
  %tensor269 = bitcast %tensor_t* %raw_tensor256 to i8*
  %malloccall270 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor271 = bitcast i8* %malloccall270 to %tensor_t*
  %dtype272 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 0
  store i8 3, i8* %dtype272
  %ndims273 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 1
  store i8 1, i8* %ndims273
  %malloccall274 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims275 = bitcast i8* %malloccall274 to [1 x i64]*
  %dims_as_i8ptr276 = bitcast [1 x i64]* %dims275 to i8*
  %elmptr277 = getelementptr [1 x i64], [1 x i64]* %dims275, i64 0, i64 0
  store i64 1, i64* %elmptr277
  %malloccall278 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data279 = bitcast i8* %malloccall278 to [1 x i8*]*
  %data_as_i8ptr280 = bitcast [1 x i8*]* %data279 to i8*
  %elmptr281 = getelementptr [1 x i8*], [1 x i8*]* %data279, i64 0, i64 0
  store i8* %tensor269, i8** %elmptr281
  %dimsptr282 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 2
  store i8* %dims_as_i8ptr276, i8** %dimsptr282
  %dataptr283 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 3
  store i8* %data_as_i8ptr280, i8** %dataptr283
  %rc284 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor271, i32 0, i32 4
  store i8 0, i8* %rc284
  %tensor285 = bitcast %tensor_t* %raw_tensor271 to i8*
  %l2286 = load i8*, i8** %l2
  %access_tensor287 = call i8* @index_get(i8* %l2286, i8* %tensor285)
  %malloccall288 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor289 = bitcast i8* %malloccall288 to %tensor_t*
  %dtype290 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 0
  store i8 0, i8* %dtype290
  %ndims291 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 1
  store i8 0, i8* %ndims291
  %malloccall292 = tail call i8* @malloc(i32 0)
  %dims293 = bitcast i8* %malloccall292 to [0 x i64]*
  %dims_as_i8ptr294 = bitcast [0 x i64]* %dims293 to i8*
```

```
  %malloccall295 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data296 = bitcast i8* %malloccall295 to [1 x i32]*
  %data_as_i8ptr297 = bitcast [1 x i32]* %data296 to i8*
  %elmptr298 = getelementptr [1 x i32], [1 x i32]* %data296, i32 0, i32 0
  store i32 1, i32* %elmptr298
  %dimsptr299 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 2
  store i8* %dims_as_i8ptr294, i8** %dimsptr299
  %dataptr300 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 3
  store i8* %data_as_i8ptr297, i8** %dataptr300
  %rc301 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor289, i32 0, i32 4
  store i8 0, i8* %rc301
  %tensor302 = bitcast %tensor_t* %raw_tensor289 to i8*
  %tmpOp303 = call i8* @range(i8* %tmpOp254, i8* %access_tensor287, i8* %tensor302)
  %malloccall304 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor305 = bitcast i8* %malloccall304 to %tensor_t*
  %dtype306 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor305, i32 0, i32 0
  store i8 0, i8* %dtype306
  %ndims307 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor305, i32 0, i32 1
  store i8 0, i8* %ndims307
  %malloccall308 = tail call i8* @malloc(i32 0)
  %dims309 = bitcast i8* %malloccall308 to [0 x i64]*
  %dims_as_i8ptr310 = bitcast [0 x i64]* %dims309 to i8*
  %malloccall311 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data312 = bitcast i8* %malloccall311 to [1 x i32]*
  %data_as_i8ptr313 = bitcast [1 x i32]* %data312 to i8*
  %elmptr314 = getelementptr [1 x i32], [1 x i32]* %data312, i32 0, i32 0
  store i32 1, i32* %elmptr314
  %dimsptr315 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor305, i32 0, i32 2
  store i8* %dims_as_i8ptr310, i8** %dimsptr315
  %dataptr316 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor305, i32 0, i32 3
  store i8* %data_as_i8ptr313, i8** %dataptr316
  %rc317 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor305, i32 0, i32 4
  store i8 0, i8* %rc317
  %tensor318 = bitcast %tensor_t* %raw_tensor305 to i8*
  %malloccall319 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor320 = bitcast i8* %malloccall319 to %tensor_t*
  %dtype321 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 0
  store i8 3, i8* %dtype321
  %ndims322 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 1
  store i8 1, i8* %ndims322
```

```
  %malloccall323 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims324 = bitcast i8* %malloccall323 to [1 x i64]*
  %dims_as_i8ptr325 = bitcast [1 x i64]* %dims324 to i8*
  %elmptr326 = getelementptr [1 x i64], [1 x i64]* %dims324, i64 0, i64 0
  store i64 1, i64* %elmptr326
  %malloccall327 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data328 = bitcast i8* %malloccall327 to [1 x i8*]*
  %data_as_i8ptr329 = bitcast [1 x i8*]* %data328 to i8*
  %elmptr330 = getelementptr [1 x i8*], [1 x i8*]* %data328, i64 0, i64 0
  store i8* %tensor318, i8** %elmptr330
  %dimsptr331 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 2
  store i8* %dims_as_i8ptr325, i8** %dimsptr331
  %dataptr332 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 3
  store i8* %data_as_i8ptr329, i8** %dataptr332
  %rc333 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor320, i32 0, i32 4
  store i8 0, i8* %rc333
  %tensor334 = bitcast %tensor_t* %raw_tensor320 to i8*
  %l2335 = load i8*, i8** %l2
  %access_tensor336 = call i8* @index_get(i8* %l2335, i8* %tensor334)
  %malloccall337 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor338 = bitcast i8* %malloccall337 to %tensor_t*
  %dtype339 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor338, i32 0, i32 0
  store i8 0, i8* %dtype339
  %ndims340 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor338, i32 0, i32 1
  store i8 0, i8* %ndims340
  %malloccall341 = tail call i8* @malloc(i32 0)
  %dims342 = bitcast i8* %malloccall341 to [0 x i64]*
  %dims_as_i8ptr343 = bitcast [0 x i64]* %dims342 to i8*
  %malloccall344 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data345 = bitcast i8* %malloccall344 to [1 x i32]*
  %data_as_i8ptr346 = bitcast [1 x i32]* %data345 to i8*
  %elmptr347 = getelementptr [1 x i32], [1 x i32]* %data345, i32 0, i32 0
  store i32 2, i32* %elmptr347
  %dimsptr348 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor338, i32 0, i32 2
  store i8* %dims_as_i8ptr343, i8** %dimsptr348
  %dataptr349 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor338, i32 0, i32 3
  store i8* %data_as_i8ptr346, i8** %dataptr349
  %rc350 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor338, i32 0, i32 4
  store i8 0, i8* %rc350
  %tensor351 = bitcast %tensor_t* %raw_tensor338 to i8*
```

```
  %tmpOp352 = call i8* @floordivide(i8* %access_tensor336, i8* %tensor351)
  %malloccall353 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor354 = bitcast i8* %malloccall353 to %tensor_t*
  %dtype355 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor354, i32 0, i32 0
  store i8 0, i8* %dtype355
  %ndims356 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor354, i32 0, i32 1
  store i8 0, i8* %ndims356
  %malloccall357 = tail call i8* @malloc(i32 0)
  %dims358 = bitcast i8* %malloccall357 to [0 x i64]*
  %dims_as_i8ptr359 = bitcast [0 x i64]* %dims358 to i8*
  %malloccall360 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data361 = bitcast i8* %malloccall360 to [1 x i32]*
  %data_as_i8ptr362 = bitcast [1 x i32]* %data361 to i8*
  %elmptr363 = getelementptr [1 x i32], [1 x i32]* %data361, i32 0, i32 0
  store i32 1, i32* %elmptr363
  %dimsptr364 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor354, i32 0, i32 2
  store i8* %dims_as_i8ptr359, i8** %dimsptr364
  %dataptr365 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor354, i32 0, i32 3
  store i8* %data_as_i8ptr362, i8** %dataptr365
  %rc366 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor354, i32 0, i32 4
  store i8 0, i8* %rc366
  %tensor367 = bitcast %tensor_t* %raw_tensor354 to i8*
  %malloccall368 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor369 = bitcast i8* %malloccall368 to %tensor_t*
  %dtype370 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 0
  store i8 3, i8* %dtype370
  %ndims371 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 1
  store i8 1, i8* %ndims371
  %malloccall372 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims373 = bitcast i8* %malloccall372 to [1 x i64]*
  %dims_as_i8ptr374 = bitcast [1 x i64]* %dims373 to i8*
  %elmptr375 = getelementptr [1 x i64], [1 x i64]* %dims373, i64 0, i64 0
  store i64 1, i64* %elmptr375
  %malloccall376 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data377 = bitcast i8* %malloccall376 to [1 x i8*]*
  %data_as_i8ptr378 = bitcast [1 x i8*]* %data377 to i8*
  %elmptr379 = getelementptr [1 x i8*], [1 x i8*]* %data377, i64 0, i64 0
  store i8* %tensor367, i8** %elmptr379
  %dimsptr380 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 2
```

```
  store i8* %dims_as_i8ptr374, i8** %dimsptr380
  %dataptr381 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 3
  store i8* %data_as_i8ptr378, i8** %dataptr381
  %rc382 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor369, i32 0, i32 4
  store i8 0, i8* %rc382
  %tensor383 = bitcast %tensor_t* %raw_tensor369 to i8*
  %l2384 = load i8*, i8** %l2
  %access_tensor385 = call i8* @index_get(i8* %l2384, i8* %tensor383)
  %malloccall386 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor387 = bitcast i8* %malloccall386 to %tensor_t*
  %dtype388 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 0
  store i8 0, i8* %dtype388
  %ndims389 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 1
  store i8 0, i8* %ndims389
  %malloccall390 = tail call i8* @malloc(i32 0)
  %dims391 = bitcast i8* %malloccall390 to [0 x i64]*
  %dims_as_i8ptr392 = bitcast [0 x i64]* %dims391 to i8*
  %malloccall393 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data394 = bitcast i8* %malloccall393 to [1 x i32]*
  %data_as_i8ptr395 = bitcast [1 x i32]* %data394 to i8*
  %elmptr396 = getelementptr [1 x i32], [1 x i32]* %data394, i32 0, i32 0
  store i32 1, i32* %elmptr396
  %dimsptr397 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 2
  store i8* %dims_as_i8ptr392, i8** %dimsptr397
  %dataptr398 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 3
  store i8* %data_as_i8ptr395, i8** %dataptr398
  %rc399 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor387, i32 0, i32 4
  store i8 0, i8* %rc399
  %tensor400 = bitcast %tensor_t* %raw_tensor387 to i8*
  %tmpOp401 = call i8* @range(i8* %tmpOp352, i8* %access_tensor385, i8* %tensor400)
  %malloccall402 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor403 = bitcast i8* %malloccall402 to %tensor_t*
  %dtype404 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor403, i32 0, i32 0
  store i8 3, i8* %dtype404
  %ndims405 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor403, i32 0, i32 1
  store i8 1, i8* %ndims405
  %malloccall406 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims407 = bitcast i8* %malloccall406 to [1 x i64]*
  %dims_as_i8ptr408 = bitcast [1 x i64]* %dims407 to i8*
  %elmptr409 = getelementptr [1 x i64], [1 x i64]* %dims407, i64 0, i64 0
```

```
  store i64 2, i64* %elmptr409
  %malloccall410 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data411 = bitcast i8* %malloccall410 to [2 x i8*]*
  %data_as_i8ptr412 = bitcast [2 x i8*]* %data411 to i8*
  %elmptr413 = getelementptr [2 x i8*], [2 x i8*]* %data411, i64 0, i64 0
  store i8* %tmpOp303, i8** %elmptr413
  %elmptr414 = getelementptr [2 x i8*], [2 x i8*]* %data411, i64 0, i64 1
  store i8* %tmpOp401, i8** %elmptr414
  %dimsptr415 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor403, i32 0, i32 2
  store i8* %dims_as_i8ptr408, i8** %dimsptr415
  %dataptr416 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor403, i32 0, i32 3
  store i8* %data_as_i8ptr412, i8** %dataptr416
  %rc417 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor403, i32 0, i32 4
  store i8 0, i8* %rc417
  %tensor418 = bitcast %tensor_t* %raw_tensor403 to i8*
  %y419 = load i8*, i8** %y
  %access_tensor420 = call i8* @index_get(i8* %y419, i8* %tensor418)
  %y22 = alloca i8*
  store i8* null, i8** %y22
  %lhsptr421 = load i8*, i8** %y22
  call void @increase_rc(i8* %access_tensor420)
  call void @decrease_rc(i8* %lhsptr421)
  store i8* %access_tensor420, i8** %y22
  %malloccall422 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr (%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor423 = bitcast i8* %malloccall422 to %tensor_t*
  %dtype424 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor423, i32 0, i32 0
  store i8 0, i8* %dtype424
  %ndims425 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor423, i32 0, i32 1
  store i8 0, i8* %ndims425
  %malloccall426 = tail call i8* @malloc(i32 0)
  %dims427 = bitcast i8* %malloccall426 to [0 x i64]*
  %dims_as_i8ptr428 = bitcast [0 x i64]* %dims427 to i8*
  %malloccall429 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data430 = bitcast i8* %malloccall429 to [1 x i32]*
  %data_as_i8ptr431 = bitcast [1 x i32]* %data430 to i8*
  %elmptr432 = getelementptr [1 x i32], [1 x i32]* %data430, i32 0, i32 0
  store i32 0, i32* %elmptr432
  %dimsptr433 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor423, i32 0, i32 2
  store i8* %dims_as_i8ptr428, i8** %dimsptr433
  %dataptr434 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor423, i32 0, i32 3
  store i8* %data_as_i8ptr431, i8** %dataptr434
```

```
  %rc435 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor423, i32 0, i32 4
  store i8 0, i8* %rc435
  %tensor436 = bitcast %tensor_t* %raw_tensor423 to i8*
  %malloccall437 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor438 = bitcast i8* %malloccall437 to %tensor_t*
  %dtype439 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor438, i32 0, i32 0
  store i8 3, i8* %dtype439
  %ndims440 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor438, i32 0, i32 1
  store i8 1, i8* %ndims440
  %malloccall441 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims442 = bitcast i8* %malloccall441 to [1 x i64]*
  %dims_as_i8ptr443 = bitcast [1 x i64]* %dims442 to i8*
  %elmptr444 = getelementptr [1 x i64], [1 x i64]* %dims442, i64 0, i64 0
  store i64 1, i64* %elmptr444
  %malloccall445 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data446 = bitcast i8* %malloccall445 to [1 x i8*]*
  %data_as_i8ptr447 = bitcast [1 x i8*]* %data446 to i8*
  %elmptr448 = getelementptr [1 x i8*], [1 x i8*]* %data446, i64 0, i64 0
  store i8* %tensor436, i8** %elmptr448
  %dimsptr449 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor438, i32 0, i32 2
  store i8* %dims_as_i8ptr443, i8** %dimsptr449
  %dataptr450 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor438, i32 0, i32 3
  store i8* %data_as_i8ptr447, i8** %dataptr450
  %rc451 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor438, i32 0, i32 4
  store i8 0, i8* %rc451
  %tensor452 = bitcast %tensor_t* %raw_tensor438 to i8*
  %l2453 = load i8*, i8** %l2
  %access_tensor454 = call i8* @index_get(i8* %l2453, i8* %tensor452)
  %malloccall455 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor456 = bitcast i8* %malloccall455 to %tensor_t*
  %dtype457 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 0
  store i8 0, i8* %dtype457
  %ndims458 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 1
  store i8 0, i8* %ndims458
  %malloccall459 = tail call i8* @malloc(i32 0)
  %dims460 = bitcast i8* %malloccall459 to [0 x i64]*
  %dims_as_i8ptr461 = bitcast [0 x i64]* %dims460 to i8*
  %malloccall462 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data463 = bitcast i8* %malloccall462 to [1 x i32]*
```

```
  %data_as_i8ptr464 = bitcast [1 x i32]* %data463 to i8*
  %elmptr465 = getelementptr [1 x i32], [1 x i32]* %data463, i32 0, i32 0
  store i32 2, i32* %elmptr465
  %dimsptr466 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 2
  store i8* %dims_as_i8ptr461, i8** %dimsptr466
  %dataptr467 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 3
  store i8* %data_as_i8ptr464, i8** %dataptr467
  %rc468 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor456, i32 0, i32 4
  store i8 0, i8* %rc468
  %tensor469 = bitcast %tensor_t* %raw_tensor456 to i8*
  %tmpOp470 = call i8* @floordivide(i8* %access_tensor454, i8* %tensor469)
  %malloccall471 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor472 = bitcast i8* %malloccall471 to %tensor_t*
  %dtype473 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor472, i32 0, i32 0
  store i8 0, i8* %dtype473
  %ndims474 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor472, i32 0, i32 1
  store i8 0, i8* %ndims474
  %malloccall475 = tail call i8* @malloc(i32 0)
  %dims476 = bitcast i8* %malloccall475 to [0 x i64]*
  %dims_as_i8ptr477 = bitcast [0 x i64]* %dims476 to i8*
  %malloccall478 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data479 = bitcast i8* %malloccall478 to [1 x i32]*
  %data_as_i8ptr480 = bitcast [1 x i32]* %data479 to i8*
  %elmptr481 = getelementptr [1 x i32], [1 x i32]* %data479, i32 0, i32 0
  store i32 0, i32* %elmptr481
  %dimsptr482 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor472, i32 0, i32 2
  store i8* %dims_as_i8ptr477, i8** %dimsptr482
  %dataptr483 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor472, i32 0, i32 3
  store i8* %data_as_i8ptr480, i8** %dataptr483
  %rc484 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor472, i32 0, i32 4
  store i8 0, i8* %rc484
  %tensor485 = bitcast %tensor_t* %raw_tensor472 to i8*
  %malloccall486 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor487 = bitcast i8* %malloccall486 to %tensor_t*
  %dtype488 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor487, i32 0, i32 0
  store i8 3, i8* %dtype488
  %ndims489 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor487, i32 0, i32 1
  store i8 1, i8* %ndims489
  %malloccall490 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims491 = bitcast i8* %malloccall490 to [1 x i64]*
```

```
  %dims_as_i8ptr492 = bitcast [1 x i64]* %dims491 to i8*
  %elmptr493 = getelementptr [1 x i64], [1 x i64]* %dims491, i64 0, i64 0
  store i64 1, i64* %elmptr493
  %malloccall494 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data495 = bitcast i8* %malloccall494 to [1 x i8*]*
  %data_as_i8ptr496 = bitcast [1 x i8*]* %data495 to i8*
  %elmptr497 = getelementptr [1 x i8*], [1 x i8*]* %data495, i64 0, i64 0
  store i8* %tensor485, i8** %elmptr497
  %dimsptr498 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor487, i32 0, i32 2
  store i8* %dims_as_i8ptr492, i8** %dimsptr498
  %dataptr499 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor487, i32 0, i32 3
  store i8* %data_as_i8ptr496, i8** %dataptr499
  %rc500 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor487, i32 0, i32 4
  store i8 0, i8* %rc500
  %tensor501 = bitcast %tensor_t* %raw_tensor487 to i8*
  %l2502 = load i8*, i8** %l2
  %access_tensor503 = call i8* @index_get(i8* %l2502, i8* %tensor501)
  %malloccall504 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor505 = bitcast i8* %malloccall504 to %tensor_t*
  %dtype506 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor505, i32 0, i32 0
  store i8 0, i8* %dtype506
  %ndims507 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor505, i32 0, i32 1
  store i8 0, i8* %ndims507
  %malloccall508 = tail call i8* @malloc(i32 0)
  %dims509 = bitcast i8* %malloccall508 to [0 x i64]*
  %dims_as_i8ptr510 = bitcast [0 x i64]* %dims509 to i8*
  %malloccall511 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data512 = bitcast i8* %malloccall511 to [1 x i32]*
  %data_as_i8ptr513 = bitcast [1 x i32]* %data512 to i8*
  %elmptr514 = getelementptr [1 x i32], [1 x i32]* %data512, i32 0, i32 0
  store i32 1, i32* %elmptr514
  %dimsptr515 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor505, i32 0, i32 2
  store i8* %dims_as_i8ptr510, i8** %dimsptr515
  %dataptr516 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor505, i32 0, i32 3
  store i8* %data_as_i8ptr513, i8** %dataptr516
  %rc517 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor505, i32 0, i32 4
  store i8 0, i8* %rc517
  %tensor518 = bitcast %tensor_t* %raw_tensor505 to i8*
  %tmpOp519 = call i8* @range(i8* %tmpOp470, i8* %access_tensor503, i8* %tensor518)
  %malloccall520 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
```

```
  %raw_tensor521 = bitcast i8* %malloccall520 to %tensor_t*
  %dtype522 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor521, i32 0, i32 0
  store i8 0, i8* %dtype522
  %ndims523 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor521, i32 0, i32 1
  store i8 0, i8* %ndims523
  %malloccall524 = tail call i8* @malloc(i32 0)
  %dims525 = bitcast i8* %malloccall524 to [0 x i64]*
  %dims_as_i8ptr526 = bitcast [0 x i64]* %dims525 to i8*
  %malloccall527 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data528 = bitcast i8* %malloccall527 to [1 x i32]*
  %data_as_i8ptr529 = bitcast [1 x i32]* %data528 to i8*
  %elmptr530 = getelementptr [1 x i32], [1 x i32]* %data528, i32 0, i32 0
  store i32 0, i32* %elmptr530
  %dimsptr531 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor521, i32 0, i32 2
  store i8* %dims_as_i8ptr526, i8** %dimsptr531
  %dataptr532 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor521, i32 0, i32 3
  store i8* %data_as_i8ptr529, i8** %dataptr532
  %rc533 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor521, i32 0, i32 4
  store i8 0, i8* %rc533
  %tensor534 = bitcast %tensor_t* %raw_tensor521 to i8*
  %malloccall535 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor536 = bitcast i8* %malloccall535 to %tensor_t*
  %dtype537 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor536, i32 0, i32 0
  store i8 0, i8* %dtype537
  %ndims538 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor536, i32 0, i32 1
  store i8 0, i8* %ndims538
  %malloccall539 = tail call i8* @malloc(i32 0)
  %dims540 = bitcast i8* %malloccall539 to [0 x i64]*
  %dims_as_i8ptr541 = bitcast [0 x i64]* %dims540 to i8*
  %malloccall542 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data543 = bitcast i8* %malloccall542 to [1 x i32]*
  %data_as_i8ptr544 = bitcast [1 x i32]* %data543 to i8*
  %elmptr545 = getelementptr [1 x i32], [1 x i32]* %data543, i32 0, i32 0
  store i32 1, i32* %elmptr545
  %dimsptr546 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor536, i32 0, i32 2
  store i8* %dims_as_i8ptr541, i8** %dimsptr546
  %dataptr547 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor536, i32 0, i32 3
  store i8* %data_as_i8ptr544, i8** %dataptr547
  %rc548 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor536, i32 0, i32 4
  store i8 0, i8* %rc548
  %tensor549 = bitcast %tensor_t* %raw_tensor536 to i8*
```

```
  %malloccall550 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor551 = bitcast i8* %malloccall550 to %tensor_t*
  %dtype552 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor551, i32 0, i32 0
  store i8 3, i8* %dtype552
  %ndims553 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor551, i32 0, i32 1
  store i8 1, i8* %ndims553
  %malloccall554 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims555 = bitcast i8* %malloccall554 to [1 x i64]*
  %dims_as_i8ptr556 = bitcast [1 x i64]* %dims555 to i8*
  %elmptr557 = getelementptr [1 x i64], [1 x i64]* %dims555, i64 0, i64 0
  store i64 1, i64* %elmptr557
  %malloccall558 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data559 = bitcast i8* %malloccall558 to [1 x i8*]*
  %data_as_i8ptr560 = bitcast [1 x i8*]* %data559 to i8*
  %elmptr561 = getelementptr [1 x i8*], [1 x i8*]* %data559, i64 0, i64 0
  store i8* %tensor549, i8** %elmptr561
  %dimsptr562 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor551, i32 0, i32 2
  store i8* %dims_as_i8ptr556, i8** %dimsptr562
  %dataptr563 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor551, i32 0, i32 3
  store i8* %data_as_i8ptr560, i8** %dataptr563
  %rc564 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor551, i32 0, i32 4
  store i8 0, i8* %rc564
  %tensor565 = bitcast %tensor_t* %raw_tensor551 to i8*
  %l2566 = load i8*, i8** %l2
  %access_tensor567 = call i8* @index_get(i8* %l2566, i8* %tensor565)
  %malloccall568 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor569 = bitcast i8* %malloccall568 to %tensor_t*
  %dtype570 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor569, i32 0, i32 0
  store i8 0, i8* %dtype570
  %ndims571 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor569, i32 0, i32 1
  store i8 0, i8* %ndims571
  %malloccall572 = tail call i8* @malloc(i32 0)
  %dims573 = bitcast i8* %malloccall572 to [0 x i64]*
  %dims_as_i8ptr574 = bitcast [0 x i64]* %dims573 to i8*
  %malloccall575 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data576 = bitcast i8* %malloccall575 to [1 x i32]*
  %data_as_i8ptr577 = bitcast [1 x i32]* %data576 to i8*
  %elmptr578 = getelementptr [1 x i32], [1 x i32]* %data576, i32 0, i32 0
  store i32 2, i32* %elmptr578
```

```
  %dimsptr579 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor569, i32 0, i32 2
  store i8* %dims_as_i8ptr574, i8** %dimsptr579
  %dataptr580 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor569, i32 0, i32 3
  store i8* %data_as_i8ptr577, i8** %dataptr580
  %rc581 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor569, i32 0, i32 4
  store i8 0, i8* %rc581
  %tensor582 = bitcast %tensor_t* %raw_tensor569 to i8*
  %tmpOp583 = call i8* @floordivide(i8* %access_tensor567, i8* %tensor582)
  %malloccall584 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor585 = bitcast i8* %malloccall584 to %tensor_t*
  %dtype586 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor585, i32 0, i32 0
  store i8 0, i8* %dtype586
  %ndims587 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor585, i32 0, i32 1
  store i8 0, i8* %ndims587
  %malloccall588 = tail call i8* @malloc(i32 0)
  %dims589 = bitcast i8* %malloccall588 to [0 x i64]*
  %dims_as_i8ptr590 = bitcast [0 x i64]* %dims589 to i8*
  %malloccall591 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data592 = bitcast i8* %malloccall591 to [1 x i32]*
  %data_as_i8ptr593 = bitcast [1 x i32]* %data592 to i8*
  %elmptr594 = getelementptr [1 x i32], [1 x i32]* %data592, i32 0, i32 0
  store i32 1, i32* %elmptr594
  %dimsptr595 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor585, i32 0, i32 2
  store i8* %dims_as_i8ptr590, i8** %dimsptr595
  %dataptr596 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor585, i32 0, i32 3
  store i8* %data_as_i8ptr593, i8** %dataptr596
  %rc597 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor585, i32 0, i32 4
  store i8 0, i8* %rc597
  %tensor598 = bitcast %tensor_t* %raw_tensor585 to i8*
  %tmpOp599 = call i8* @range(i8* %tensor534, i8* %tmpOp583, i8* %tensor598)
  %malloccall600 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor601 = bitcast i8* %malloccall600 to %tensor_t*
  %dtype602 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor601, i32 0, i32 0
  store i8 3, i8* %dtype602
  %ndims603 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor601, i32 0, i32 1
  store i8 1, i8* %ndims603
  %malloccall604 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims605 = bitcast i8* %malloccall604 to [1 x i64]*
  %dims_as_i8ptr606 = bitcast [1 x i64]* %dims605 to i8*
  %elmptr607 = getelementptr [1 x i64], [1 x i64]* %dims605, i64 0, i64 0
```

```
  store i64 2, i64* %elmptr607
  %malloccall608 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data609 = bitcast i8* %malloccall608 to [2 x i8*]*
  %data_as_i8ptr610 = bitcast [2 x i8*]* %data609 to i8*
  %elmptr611 = getelementptr [2 x i8*], [2 x i8*]* %data609, i64 0, i64 0
  store i8* %tmpOp519, i8** %elmptr611
  %elmptr612 = getelementptr [2 x i8*], [2 x i8*]* %data609, i64 0, i64 1
  store i8* %tmpOp599, i8** %elmptr612
  %dimsptr613 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor601, i32 0, i32 2
  store i8* %dims_as_i8ptr606, i8** %dimsptr613
  %dataptr614 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor601, i32 0, i32 3
  store i8* %data_as_i8ptr610, i8** %dataptr614
  %rc615 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor601, i32 0, i32 4
  store i8 0, i8* %rc615
  %tensor616 = bitcast %tensor_t* %raw_tensor601 to i8*
  %y617 = load i8*, i8** %y
  %access_tensor618 = call i8* @index_get(i8* %y617, i8* %tensor616)
  %y21 = alloca i8*
  store i8* null, i8** %y21
  %lhsptr619 = load i8*, i8** %y21
  call void @increase_rc(i8* %access_tensor618)
  call void @decrease_rc(i8* %lhsptr619)
  store i8* %access_tensor618, i8** %y21
  %malloccall620 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr (%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor621 = bitcast i8* %malloccall620 to %tensor_t*
  %dtype622 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor621, i32 0, i32 0
  store i8 0, i8* %dtype622
  %ndims623 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor621, i32 0, i32 1
  store i8 0, i8* %ndims623
  %malloccall624 = tail call i8* @malloc(i32 0)
  %dims625 = bitcast i8* %malloccall624 to [0 x i64]*
  %dims_as_i8ptr626 = bitcast [0 x i64]* %dims625 to i8*
  %malloccall627 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to i32))
  %data628 = bitcast i8* %malloccall627 to [1 x i32]*
  %data_as_i8ptr629 = bitcast [1 x i32]* %data628 to i8*
  %elmptr630 = getelementptr [1 x i32], [1 x i32]* %data628, i32 0, i32 0
  store i32 0, i32* %elmptr630
  %dimsptr631 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor621, i32 0, i32 2
  store i8* %dims_as_i8ptr626, i8** %dimsptr631
  %dataptr632 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor621, i32 0, i32 3
  store i8* %data_as_i8ptr629, i8** %dataptr632
```

```
  %rc633 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor621, i32 0, i32 4
  store i8 0, i8* %rc633
  %tensor634 = bitcast %tensor_t* %raw_tensor621 to i8*
  %malloccall635 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor636 = bitcast i8* %malloccall635 to %tensor_t*
  %dtype637 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor636, i32 0, i32 0
  store i8 0, i8* %dtype637
  %ndims638 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor636, i32 0, i32 1
  store i8 0, i8* %ndims638
  %malloccall639 = tail call i8* @malloc(i32 0)
  %dims640 = bitcast i8* %malloccall639 to [0 x i64]*
  %dims_as_i8ptr641 = bitcast [0 x i64]* %dims640 to i8*
  %malloccall642 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data643 = bitcast i8* %malloccall642 to [1 x i32]*
  %data_as_i8ptr644 = bitcast [1 x i32]* %data643 to i8*
  %elmptr645 = getelementptr [1 x i32], [1 x i32]* %data643, i32 0, i32 0
  store i32 0, i32* %elmptr645
  %dimsptr646 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor636, i32 0, i32 2
  store i8* %dims_as_i8ptr641, i8** %dimsptr646
  %dataptr647 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor636, i32 0, i32 3
  store i8* %data_as_i8ptr644, i8** %dataptr647
  %rc648 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor636, i32 0, i32 4
  store i8 0, i8* %rc648
  %tensor649 = bitcast %tensor_t* %raw_tensor636 to i8*
  %malloccall650 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor651 = bitcast i8* %malloccall650 to %tensor_t*
  %dtype652 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor651, i32 0, i32 0
  store i8 3, i8* %dtype652
  %ndims653 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor651, i32 0, i32 1
  store i8 1, i8* %ndims653
  %malloccall654 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims655 = bitcast i8* %malloccall654 to [1 x i64]*
  %dims_as_i8ptr656 = bitcast [1 x i64]* %dims655 to i8*
  %elmptr657 = getelementptr [1 x i64], [1 x i64]* %dims655, i64 0, i64 0
  store i64 1, i64* %elmptr657
  %malloccall658 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data659 = bitcast i8* %malloccall658 to [1 x i8*]*
  %data_as_i8ptr660 = bitcast [1 x i8*]* %data659 to i8*
  %elmptr661 = getelementptr [1 x i8*], [1 x i8*]* %data659, i64 0, i64 0
```

```
  store i8* %tensor649, i8** %elmptr661
  %dimsptr662 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor651, i32 0, i32 2
  store i8* %dims_as_i8ptr656, i8** %dimsptr662
  %dataptr663 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor651, i32 0, i32 3
  store i8* %data_as_i8ptr660, i8** %dataptr663
  %rc664 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor651, i32 0, i32 4
  store i8 0, i8* %rc664
  %tensor665 = bitcast %tensor_t* %raw_tensor651 to i8*
  %l1666 = load i8*, i8** %l1
  %access_tensor667 = call i8* @index_get(i8* %l1666, i8* %tensor665)
  %malloccall668 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor669 = bitcast i8* %malloccall668 to %tensor_t*
  %dtype670 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor669, i32 0, i32 0
  store i8 0, i8* %dtype670
  %ndims671 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor669, i32 0, i32 1
  store i8 0, i8* %ndims671
  %malloccall672 = tail call i8* @malloc(i32 0)
  %dims673 = bitcast i8* %malloccall672 to [0 x i64]*
  %dims_as_i8ptr674 = bitcast [0 x i64]* %dims673 to i8*
  %malloccall675 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data676 = bitcast i8* %malloccall675 to [1 x i32]*
  %data_as_i8ptr677 = bitcast [1 x i32]* %data676 to i8*
  %elmptr678 = getelementptr [1 x i32], [1 x i32]* %data676, i32 0, i32 0
  store i32 2, i32* %elmptr678
  %dimsptr679 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor669, i32 0, i32 2
  store i8* %dims_as_i8ptr674, i8** %dimsptr679
  %dataptr680 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor669, i32 0, i32 3
  store i8* %data_as_i8ptr677, i8** %dataptr680
  %rc681 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor669, i32 0, i32 4
  store i8 0, i8* %rc681
  %tensor682 = bitcast %tensor_t* %raw_tensor669 to i8*
  %tmpOp683 = call i8* @floordivide(i8* %access_tensor667, i8* %tensor682)
  %malloccall684 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor685 = bitcast i8* %malloccall684 to %tensor_t*
  %dtype686 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor685, i32 0, i32 0
  store i8 0, i8* %dtype686
  %ndims687 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor685, i32 0, i32 1
  store i8 0, i8* %ndims687
  %malloccall688 = tail call i8* @malloc(i32 0)
  %dims689 = bitcast i8* %malloccall688 to [0 x i64]*
  %dims_as_i8ptr690 = bitcast [0 x i64]* %dims689 to i8*
```

```
  %malloccall691 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data692 = bitcast i8* %malloccall691 to [1 x i32]*
  %data_as_i8ptr693 = bitcast [1 x i32]* %data692 to i8*
  %elmptr694 = getelementptr [1 x i32], [1 x i32]* %data692, i32 0, i32 0
  store i32 1, i32* %elmptr694
  %dimsptr695 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor685, i32 0, i32 2
  store i8* %dims_as_i8ptr690, i8** %dimsptr695
  %dataptr696 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor685, i32 0, i32 3
  store i8* %data_as_i8ptr693, i8** %dataptr696
  %rc697 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor685, i32 0, i32 4
  store i8 0, i8* %rc697
  %tensor698 = bitcast %tensor_t* %raw_tensor685 to i8*
  %tmpOp699 = call i8* @range(i8* %tensor634, i8* %tmpOp683, i8* %tensor698)
  %malloccall700 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor701 = bitcast i8* %malloccall700 to %tensor_t*
  %dtype702 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor701, i32 0, i32 0
  store i8 0, i8* %dtype702
  %ndims703 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor701, i32 0, i32 1
  store i8 0, i8* %ndims703
  %malloccall704 = tail call i8* @malloc(i32 0)
  %dims705 = bitcast i8* %malloccall704 to [0 x i64]*
  %dims_as_i8ptr706 = bitcast [0 x i64]* %dims705 to i8*
  %malloccall707 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data708 = bitcast i8* %malloccall707 to [1 x i32]*
  %data_as_i8ptr709 = bitcast [1 x i32]* %data708 to i8*
  %elmptr710 = getelementptr [1 x i32], [1 x i32]* %data708, i32 0, i32 0
  store i32 1, i32* %elmptr710
  %dimsptr711 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor701, i32 0, i32 2
  store i8* %dims_as_i8ptr706, i8** %dimsptr711
  %dataptr712 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor701, i32 0, i32 3
  store i8* %data_as_i8ptr709, i8** %dataptr712
  %rc713 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor701, i32 0, i32 4
  store i8 0, i8* %rc713
  %tensor714 = bitcast %tensor_t* %raw_tensor701 to i8*
  %malloccall715 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor716 = bitcast i8* %malloccall715 to %tensor_t*
  %dtype717 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor716, i32 0, i32 0
  store i8 3, i8* %dtype717
  %ndims718 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor716, i32 0, i32 1
  store i8 1, i8* %ndims718
```

```
  %malloccall719 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims720 = bitcast i8* %malloccall719 to [1 x i64]*
  %dims_as_i8ptr721 = bitcast [1 x i64]* %dims720 to i8*
  %elmptr722 = getelementptr [1 x i64], [1 x i64]* %dims720, i64 0, i64 0
  store i64 1, i64* %elmptr722
  %malloccall723 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data724 = bitcast i8* %malloccall723 to [1 x i8*]*
  %data_as_i8ptr725 = bitcast [1 x i8*]* %data724 to i8*
  %elmptr726 = getelementptr [1 x i8*], [1 x i8*]* %data724, i64 0, i64 0
  store i8* %tensor714, i8** %elmptr726
  %dimsptr727 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor716, i32 0, i32 2
  store i8* %dims_as_i8ptr721, i8** %dimsptr727
  %dataptr728 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor716, i32 0, i32 3
  store i8* %data_as_i8ptr725, i8** %dataptr728
  %rc729 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor716, i32 0, i32 4
  store i8 0, i8* %rc729
  %tensor730 = bitcast %tensor_t* %raw_tensor716 to i8*
  %l1731 = load i8*, i8** %l1
  %access_tensor732 = call i8* @index_get(i8* %l1731, i8* %tensor730)
  %malloccall733 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor734 = bitcast i8* %malloccall733 to %tensor_t*
  %dtype735 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor734, i32 0, i32 0
  store i8 0, i8* %dtype735
  %ndims736 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor734, i32 0, i32 1
  store i8 0, i8* %ndims736
  %malloccall737 = tail call i8* @malloc(i32 0)
  %dims738 = bitcast i8* %malloccall737 to [0 x i64]*
  %dims_as_i8ptr739 = bitcast [0 x i64]* %dims738 to i8*
  %malloccall740 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data741 = bitcast i8* %malloccall740 to [1 x i32]*
  %data_as_i8ptr742 = bitcast [1 x i32]* %data741 to i8*
  %elmptr743 = getelementptr [1 x i32], [1 x i32]* %data741, i32 0, i32 0
  store i32 2, i32* %elmptr743
  %dimsptr744 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor734, i32 0, i32 2
  store i8* %dims_as_i8ptr739, i8** %dimsptr744
  %dataptr745 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor734, i32 0, i32 3
  store i8* %data_as_i8ptr742, i8** %dataptr745
  %rc746 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor734, i32 0, i32 4
  store i8 0, i8* %rc746
  %tensor747 = bitcast %tensor_t* %raw_tensor734 to i8*
```

```
  %tmpOp748 = call i8* @floordivide(i8* %access_tensor732, i8* %tensor747)
  %malloccall749 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor750 = bitcast i8* %malloccall749 to %tensor_t*
  %dtype751 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor750, i32 0, i32 0
  store i8 0, i8* %dtype751
  %ndims752 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor750, i32 0, i32 1
  store i8 0, i8* %ndims752
  %malloccall753 = tail call i8* @malloc(i32 0)
  %dims754 = bitcast i8* %malloccall753 to [0 x i64]*
  %dims_as_i8ptr755 = bitcast [0 x i64]* %dims754 to i8*
  %malloccall756 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data757 = bitcast i8* %malloccall756 to [1 x i32]*
  %data_as_i8ptr758 = bitcast [1 x i32]* %data757 to i8*
  %elmptr759 = getelementptr [1 x i32], [1 x i32]* %data757, i32 0, i32 0
  store i32 1, i32* %elmptr759
  %dimsptr760 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor750, i32 0, i32 2
  store i8* %dims_as_i8ptr755, i8** %dimsptr760
  %dataptr761 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor750, i32 0, i32 3
  store i8* %data_as_i8ptr758, i8** %dataptr761
  %rc762 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor750, i32 0, i32 4
  store i8 0, i8* %rc762
  %tensor763 = bitcast %tensor_t* %raw_tensor750 to i8*
  %malloccall764 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor765 = bitcast i8* %malloccall764 to %tensor_t*
  %dtype766 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor765, i32 0, i32 0
  store i8 3, i8* %dtype766
  %ndims767 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor765, i32 0, i32 1
  store i8 1, i8* %ndims767
  %malloccall768 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims769 = bitcast i8* %malloccall768 to [1 x i64]*
  %dims_as_i8ptr770 = bitcast [1 x i64]* %dims769 to i8*
  %elmptr771 = getelementptr [1 x i64], [1 x i64]* %dims769, i64 0, i64 0
  store i64 1, i64* %elmptr771
  %malloccall772 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1** null, i32 1) to
i32))
  %data773 = bitcast i8* %malloccall772 to [1 x i8*]*
  %data_as_i8ptr774 = bitcast [1 x i8*]* %data773 to i8*
  %elmptr775 = getelementptr [1 x i8*], [1 x i8*]* %data773, i64 0, i64 0
  store i8* %tensor763, i8** %elmptr775
  %dimsptr776 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor765, i32 0, i32 2
```

```
  store i8* %dims_as_i8ptr770, i8** %dimsptr776
  %dataptr777 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor765, i32 0, i32 3
  store i8* %data_as_i8ptr774, i8** %dataptr777
  %rc778 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor765, i32 0, i32 4
  store i8 0, i8* %rc778
  %tensor779 = bitcast %tensor_t* %raw_tensor765 to i8*
  %l1780 = load i8*, i8** %l1
  %access_tensor781 = call i8* @index_get(i8* %l1780, i8* %tensor779)
  %malloccall782 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor783 = bitcast i8* %malloccall782 to %tensor_t*
  %dtype784 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor783, i32 0, i32 0
  store i8 0, i8* %dtype784
  %ndims785 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor783, i32 0, i32 1
  store i8 0, i8* %ndims785
  %malloccall786 = tail call i8* @malloc(i32 0)
  %dims787 = bitcast i8* %malloccall786 to [0 x i64]*
  %dims_as_i8ptr788 = bitcast [0 x i64]* %dims787 to i8*
  %malloccall789 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data790 = bitcast i8* %malloccall789 to [1 x i32]*
  %data_as_i8ptr791 = bitcast [1 x i32]* %data790 to i8*
  %elmptr792 = getelementptr [1 x i32], [1 x i32]* %data790, i32 0, i32 0
  store i32 1, i32* %elmptr792
  %dimsptr793 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor783, i32 0, i32 2
  store i8* %dims_as_i8ptr788, i8** %dimsptr793
  %dataptr794 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor783, i32 0, i32 3
  store i8* %data_as_i8ptr791, i8** %dataptr794
  %rc795 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor783, i32 0, i32 4
  store i8 0, i8* %rc795
  %tensor796 = bitcast %tensor_t* %raw_tensor783 to i8*
  %tmpOp797 = call i8* @range(i8* %tmpOp748, i8* %access_tensor781, i8* %tensor796)
  %malloccall798 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor799 = bitcast i8* %malloccall798 to %tensor_t*
  %dtype800 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor799, i32 0, i32 0
  store i8 3, i8* %dtype800
  %ndims801 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor799, i32 0, i32 1
  store i8 1, i8* %ndims801
  %malloccall802 = tail call i8* @malloc(i32 ptrtoint (i64* getelementptr (i64, i64* null, i32 1) to
i32))
  %dims803 = bitcast i8* %malloccall802 to [1 x i64]*
  %dims_as_i8ptr804 = bitcast [1 x i64]* %dims803 to i8*
  %elmptr805 = getelementptr [1 x i64], [1 x i64]* %dims803, i64 0, i64 0
```

```
  store i64 2, i64* %elmptr805
  %malloccall806 = tail call i8* @malloc(i32 trunc (i64 mul nuw (i64 ptrtoint (i1** getelementptr
(i1*, i1** null, i32 1) to i64), i64 2) to i32))
  %data807 = bitcast i8* %malloccall806 to [2 x i8*]*
  %data_as_i8ptr808 = bitcast [2 x i8*]* %data807 to i8*
  %elmptr809 = getelementptr [2 x i8*], [2 x i8*]* %data807, i64 0, i64 0
  store i8* %tmpOp699, i8** %elmptr809
  %elmptr810 = getelementptr [2 x i8*], [2 x i8*]* %data807, i64 0, i64 1
  store i8* %tmpOp797, i8** %elmptr810
  %dimsptr811 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor799, i32 0, i32 2
  store i8* %dims_as_i8ptr804, i8** %dimsptr811
  %dataptr812 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor799, i32 0, i32 3
  store i8* %data_as_i8ptr808, i8** %dataptr812
  %rc813 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor799, i32 0, i32 4
  store i8 0, i8* %rc813
  %tensor814 = bitcast %tensor_t* %raw_tensor799 to i8*
  %x815 = load i8*, i8** %x
  %access_tensor816 = call i8* @index_get(i8* %x815, i8* %tensor814)
  %x12 = alloca i8*
  store i8* null, i8** %x12
  %lhsptr817 = load i8*, i8** %x12
  call void @increase_rc(i8* %access_tensor816)
  call void @decrease_rc(i8* %lhsptr817)
  store i8* %access_tensor816, i8** %x12
  %x12818 = load i8*, i8** %x12
  %x22819 = load i8*, i8** %x22
  %tmpOp820 = call i8* @subtract(i8* %x12818, i8* %x22819)
  %y21821 = load i8*, i8** %y21
  %y22822 = load i8*, i8** %y22
  %tmpOp823 = call i8* @add(i8* %y21821, i8* %y22822)
  %tmpOp824 = call i8* @mult(i8* %tmpOp820, i8* %tmpOp823)
  ret i8* %tmpOp824
}

define i8* @StrassenMULreduce(i8** %result) {
entry:
  %result1 = alloca i8**
  store i8** %result, i8*** %result1
  %f1 = alloca i8*
  %newa = load i8**, i8*** %result1
  %tmp = getelementptr i8*, i8** %newa, i32 0
  %newv = load i8*, i8** %tmp
  store i8* %newv, i8** %f1
  %f2 = alloca i8*
```

```
%newa2 = load i8**, i8*** %result1
%tmp3 = getelementptr i8*, i8** %newa2, i32 1
%newv4 = load i8*, i8** %tmp3
store i8* %newv4, i8** %f2
%f3 = alloca i8*
%newa5 = load i8**, i8*** %result1
%tmp6 = getelementptr i8*, i8** %newa5, i32 2
%newv7 = load i8*, i8** %tmp6
store i8* %newv7, i8** %f3
%f4 = alloca i8*
%newa8 = load i8**, i8*** %result1
%tmp9 = getelementptr i8*, i8** %newa8, i32 3
%newv10 = load i8*, i8** %tmp9
store i8* %newv10, i8** %f4
%f5 = alloca i8*
%newa11 = load i8**, i8*** %result1
%tmp12 = getelementptr i8*, i8** %newa11, i32 4
%newv13 = load i8*, i8** %tmp12
store i8* %newv13, i8** %f5
%f6 = alloca i8*
%newa14 = load i8**, i8*** %result1
%tmp15 = getelementptr i8*, i8** %newa14, i32 5
%newv16 = load i8*, i8** %tmp15
store i8* %newv16, i8** %f6
%f7 = alloca i8*
%newa17 = load i8**, i8*** %result1
%tmp18 = getelementptr i8*, i8** %newa17, i32 6
%newv19 = load i8*, i8** %tmp18
store i8* %newv19, i8** %f7
%f120 = load i8*, i8** %f1
%f421 = load i8*, i8** %f4
%tmpOp = call i8* @add(i8* %f120, i8* %f421)
%f522 = load i8*, i8** %f5
%tmpOp23 = call i8* @subtract(i8* %tmpOp, i8* %f522)
%f724 = load i8*, i8** %f7
%tmpOp25 = call i8* @add(i8* %tmpOp23, i8* %f724)
%c11 = alloca i8*
store i8* null, i8** %c11
%lhsptr = load i8*, i8** %c11
call void @increase_rc(i8* %tmpOp25)
call void @decrease_rc(i8* %lhsptr)
store i8* %tmpOp25, i8** %c11
%f326 = load i8*, i8** %f3
%f527 = load i8*, i8** %f5
```

```
  %tmpOp28 = call i8* @add(i8* %f326, i8* %f527)
  %c12 = alloca i8*
  store i8* null, i8** %c12
  %lhsptr29 = load i8*, i8** %c12
  call void @increase_rc(i8* %tmpOp28)
  call void @decrease_rc(i8* %lhsptr29)
  store i8* %tmpOp28, i8** %c12
  %f230 = load i8*, i8** %f2
  %f431 = load i8*, i8** %f4
  %tmpOp32 = call i8* @add(i8* %f230, i8* %f431)
  %c21 = alloca i8*
  store i8* null, i8** %c21
  %lhsptr33 = load i8*, i8** %c21
  call void @increase_rc(i8* %tmpOp32)
  call void @decrease_rc(i8* %lhsptr33)
  store i8* %tmpOp32, i8** %c21
  %f134 = load i8*, i8** %f1
  %f235 = load i8*, i8** %f2
  %tmpOp36 = call i8* @subtract(i8* %f134, i8* %f235)
  %f337 = load i8*, i8** %f3
  %tmpOp38 = call i8* @add(i8* %tmpOp36, i8* %f337)
  %f639 = load i8*, i8** %f6
  %tmpOp40 = call i8* @add(i8* %tmpOp38, i8* %f639)
  %c22 = alloca i8*
  store i8* null, i8** %c22
  %lhsptr41 = load i8*, i8** %c22
  call void @increase_rc(i8* %tmpOp40)
  call void @decrease_rc(i8* %lhsptr41)
  store i8* %tmpOp40, i8** %c22
  %c1142 = load i8*, i8** %c11
  %c1243 = load i8*, i8** %c12
  %malloccall = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor = bitcast i8* %malloccall to %tensor_t*
  %dtype = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 0
  store i8 0, i8* %dtype
  %ndims = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 1
  store i8 0, i8* %ndims
  %malloccall44 = tail call i8* @malloc(i32 0)
  %dims = bitcast i8* %malloccall44 to [0 x i64]*
  %dims_as_i8ptr = bitcast [0 x i64]* %dims to i8*
  %malloccall45 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data = bitcast i8* %malloccall45 to [1 x i32]*
```

```
  %data_as_i8ptr = bitcast [1 x i32]* %data to i8*
  %elmptr = getelementptr [1 x i32], [1 x i32]* %data, i32 0, i32 0
  store i32 0, i32* %elmptr
  %dimsptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 2
  store i8* %dims_as_i8ptr, i8** %dimsptr
  %dataptr = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 3
  store i8* %data_as_i8ptr, i8** %dataptr
  %rc = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor, i32 0, i32 4
  store i8 0, i8* %rc
  %tensor = bitcast %tensor_t* %raw_tensor to i8*
  %cat = call i8* @cat(i8* %c1142, i8* %c1243, i8* %tensor)
  %c2146 = load i8*, i8** %c21
  %c2247 = load i8*, i8** %c22
  %malloccall48 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor49 = bitcast i8* %malloccall48 to %tensor_t*
  %dtype50 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor49, i32 0, i32 0
  store i8 0, i8* %dtype50
  %ndims51 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor49, i32 0, i32 1
  store i8 0, i8* %ndims51
  %malloccall52 = tail call i8* @malloc(i32 0)
  %dims53 = bitcast i8* %malloccall52 to [0 x i64]*
  %dims_as_i8ptr54 = bitcast [0 x i64]* %dims53 to i8*
  %malloccall55 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data56 = bitcast i8* %malloccall55 to [1 x i32]*
  %data_as_i8ptr57 = bitcast [1 x i32]* %data56 to i8*
  %elmptr58 = getelementptr [1 x i32], [1 x i32]* %data56, i32 0, i32 0
  store i32 0, i32* %elmptr58
  %dimsptr59 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor49, i32 0, i32 2
  store i8* %dims_as_i8ptr54, i8** %dimsptr59
  %dataptr60 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor49, i32 0, i32 3
  store i8* %data_as_i8ptr57, i8** %dataptr60
  %rc61 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor49, i32 0, i32 4
  store i8 0, i8* %rc61
  %tensor62 = bitcast %tensor_t* %raw_tensor49 to i8*
  %cat63 = call i8* @cat(i8* %c2146, i8* %c2247, i8* %tensor62)
  %malloccall64 = tail call i8* @malloc(i32 ptrtoint (%tensor_t* getelementptr
(%tensor_t, %tensor_t* null, i32 1) to i32))
  %raw_tensor65 = bitcast i8* %malloccall64 to %tensor_t*
  %dtype66 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor65, i32 0, i32 0
  store i8 0, i8* %dtype66
  %ndims67 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor65, i32 0, i32 1
  store i8 0, i8* %ndims67
```

```
  %malloccall68 = tail call i8* @malloc(i32 0)
  %dims69 = bitcast i8* %malloccall68 to [0 x i64]*
  %dims_as_i8ptr70 = bitcast [0 x i64]* %dims69 to i8*
  %malloccall71 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32* null, i32 1) to
i32))
  %data72 = bitcast i8* %malloccall71 to [1 x i32]*
  %data_as_i8ptr73 = bitcast [1 x i32]* %data72 to i8*
  %elmptr74 = getelementptr [1 x i32], [1 x i32]* %data72, i32 0, i32 0
  store i32 1, i32* %elmptr74
  %dimsptr75 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor65, i32 0, i32 2
  store i8* %dims_as_i8ptr70, i8** %dimsptr75
  %dataptr76 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor65, i32 0, i32 3
  store i8* %data_as_i8ptr73, i8** %dataptr76
  %rc77 = getelementptr inbounds %tensor_t, %tensor_t* %raw_tensor65, i32 0, i32 4
  store i8 0, i8* %rc77
  %tensor78 = bitcast %tensor_t* %raw_tensor65 to i8*
  %cat79 = call i8* @cat(i8* %cat, i8* %cat63, i8* %tensor78)
  ret i8* %cat79
}
```

7. Lessons Learned

7.1. Xiangrong Xu

Functional programming language is very fun. It's my first time trying to write in a functional programming language. At the beginning I thought it was impossible for me just to understand the codes on slides, but gradually I found it very interesting. Throughout our study and implementation of our own language compilers I gradually realize and appreciate that functional programming languages have really powerful features like pattern matching and are perfect for compiler designs.

And for our project, the designing stages of development took a lot of time. Actually, for several times we changed our ideas and thought about whether we could finish this work, but in the end we did and did very well. If we were given more time, maybe we could give a more complete version of TENLab.

In the end, I really appreciate my teammates! They helped me a lot during the development of this project. It's so nice to have them as my teammates!

7.2. Xincheng Xie

The first lesson I learned is how to use automatic machines, that is, DFA/NFA and Context free grammar to analyse a language. I also learned how a compiler generally works, which gives me a big picture of how to go deep into different stages.

Additionally, it's really interesting to learn OCaml. It is a brand-new language that I have never learned. I also learned LLVM IR and how the front end and back end work together.

For this project, in order to achieve a somewhat dynamic language, I read a lot of CPython codes and materials, and found out how to implement our own tensor. This process is really challenging and I learn a lot about how a language works.

7.3. Songqing Ye

LLVM IR is fun (maybe). In the course, I learned the syntax of the very powerful .mly and .mll file. Also, I have a chance to see what LLVM code looks like. This is really new and interesting to me.

For the project, my teammates are really reliable. I can implement my function on top of the others' code, without knowing to every detail how it works. That is like working in a top company. I think most important for a team is fluent communication and trust in each other.

7.4. Senhong Liu

Through this project, I learned two important things. The first one is that before you start working, you need to think it through. Before I actually translated the sast into the llvm ir, I didn't work out the whole architecture, and therefore, I usually changed my idea about the parser, about the semantic check. We didn't start the final code work on Codegen until much later. Therefore, we don’t have time to implement all those interesting ideas in our minds. Another thing is that it’s my first time writing programs using OCaml and functional programming as a whole. Similarly, it’s also my first time learning the whole workflow of how to design and implement a new language and translate it into LLvm IR, which interests me a lot.

Appendix

Ast.ml

```
(************ Abstract Syntax Tree types for TENLab ************)

type bop =
(* Arithmetic operators*)
Add | Sub | Mul | DotMul | Div | Pow | DotPow | Mod | FlrDiv

(* Relational operators *)
| Eq | Geq | Gt | Leq | Lt | Neq

(* Logical operators *)
| And | Or

type tensortype = INT_Tensor | FLOAT_Tensor

type dimtype = TensorTup of tensortype * int * int list | VoidTup

(* Unary operators *)
type uop = Not | Neg | Transpose

(* Literal *)
type literal =
 IntLit of int
| FloatLit of float

type operator_name =
 AddSymbol
| MinusSymbol
| MulSymbol

type expr =
| FId of string
| Tensor of tensor
| VarTs of expr list
| ASexpr of asexpr
| EmptyTensor
| StringLit of string
```

```
| Binop of expr * bop * expr
| Unop of uop * expr
| Range of expr * expr * expr
(* Built-in functions *)
| Print of expr
| Shape of expr
| Cat of expr * expr * expr
| Any of expr
| All of expr
| Sum of expr
| Ones of expr
| Zeros of expr
| Rand of expr
| Len of expr
| Int_Of of expr
| Float_Of of expr
| Floor of expr
| Ceil of expr
| Round of expr
| Abs of expr
| Log of expr
| Inverse of expr
(* | Solve of expr * expr *)
(* | Svd of expr
| Eig of expr
| Eigv of expr *)

| FuncCall of expr * expr list

and asexpr =
 Identifier of string
| IdentifierInd of string * expr list

(* Tensor *)
and tensor =
 LRTensor of tensor
| NPTensor of tensor
| LRTensors of tensor * tensor
| NPTensors of tensor * tensor
| Tensor0 of literal

(* Statements *)
```

```
type stmt =
 EmptyStmt
| Expr of expr
| Assign of asexpr * expr
(* | FuncSign of string * string list *)
| FuncDecl of string * string list * stmt list
| IfStmt of expr * stmt list * stmt list
| ForStmt of string * expr * stmt list
| WhileStmt of expr * stmt list
| PEInvoke of string
| PEEnd of string
(* Keyword statement *)
| Return of expr
(* | Break
| Continue
| Exit of expr *)

type pofunc = {
 operator : string;
 params : string list;
 headstmt : stmt list;
 mapfuncs : (string * stmt list) list;
 reducefunc : stmt list;
}

type program = (string * pofunc list) list * stmt list



let string_of_program (_, _) = "TODO"
```

Parser.mly

```
%{ open Ast %}

%token SEP EOF
// arithmetic operators
%token PLUS SUBTRACT MULTIPLICATION DOT_MULTIPLICATION DIVIDE POWER DOT_POWER
TRANSPOSE MOD FLOOR_DIVIDE
// relational operators
```

```
%token IS_EQUAL IS_GEQ IS_GT IS_LEQ IS_LT IS_NOT_EQUAL
// logical operators
%token AND OR NOT
// parentheses and brackets
%token LEFT_PARENTHESIS RIGHT_PARENTHESIS LEFT_CURLY_BRACKET RIGHT_CURLY_BRACKET
LEFT_SQUARE_BRACKET RIGHT_SQUARE_BRACKET
// delimiters
%token COMMA COLON
// assignment
%token ASSIGNMENT
// keywords
// TODO: support special keywords, e.g., read, print, shape, cat
// build-in functions
%token ANY ALL SUM ONES ZEROS RAND LEN INT_OF FLOAT_OF FLOOR CEIL ROUND ABS LOG
INVERSE PRINT SHAPE CAT
/* %token SOLVE SVD EIG EIGV  */

// TODO: string or char?
%token IF ELIF ELSE FOR WHILE IN CONTINUE BREAK RETURN EXIT DEFINE INT FLOAT STRING
VAR PARALLEL_DEFINE OVERLOAD MAP REDUCE USING RETURN END NIL
%token <string> OPERATOR_INDICATOR
%token <int> INT_LITERAL
%token <string> STRING_LITERAL
%token <float> FLOAT_LITERAL
%token <string> IDENTIFIER

%left SEP
%nonassoc RIGHT_SQUARE_BRACKET RIGHT_CURLY_BRACKET
%nonassoc LEFT_SQUARE_BRACKET LEFT_CURLY_BRACKET
%nonassoc COLON
%nonassoc COMMA
%nonassoc NOELSE
%left RETURN
%right ASSIGNMENT
%left OR
%left AND
%left IS_EQUAL IS_NOT_EQUAL
%left IS_GEQ IS_GT IS_LEQ IS_LT
%left PLUS SUBTRACT
%left MULTIPLICATION DOT_MULTIPLICATION DIVIDE MOD FLOOR_DIVIDE
%right POWER DOT_POWER
%left TRANSPOSE
```

```
%right NOT

%nonassoc RIGHT_PARENTHESIS
%nonassoc LEFT_PARENTHESIS

%start main
%type <Ast.program> main

%%

main:
 pes normal_stmts EOF { ($1, $2) }



/*****************************************************************************
**
                              Statements
*****************************************************************************
*/

normal_stmts:
| { [] }
| normal_stmt normal_stmts { $1::$2 }

func_stmts:
| { [] }
| func_stmt func_stmts { $1::$2 }

loop_stmts:
| { [] }
| loop_stmt loop_stmts { $1::$2 }

func_loop_stmts:
| { [] }
| func_loop_stmt func_loop_stmts { $1::$2 }

/*
* A TENLab file should consist of a bunch of statements.
* Here we define all the possible statements:
* (i)    a function declaration or function call
* (ii)   an exprssion inside/outside the body of function
* (iii)  tensor declaration or assignment
```

```
* (iv)   a return/break/continue/exit statement
* (v)    if statement/if-else statement
* (vi)   TODO: if-elif statement
* (vii)  for statement
* (viii) while statement
* (ix)   TODO: more statments, e.g., built-in function?
*/
stmt:
| expr SEP { Expr($1) }
// TODO: support a, b = 1, 2?
| asexpr ASSIGNMENT expr SEP { Assign($1, $3) }
| USING IDENTIFIER SEP { PEInvoke($2) }
| END IDENTIFIER SEP { PEEnd($2) }
/* | EXIT LEFT_PARENTHESIS expr RIGHT_PARENTHESIS SEP { Exit($3) } */

normal_stmt:
| stmt { $1 }
| DEFINE IDENTIFIER LEFT_PARENTHESIS params RIGHT_PARENTHESIS func_stmt_body
{ FuncDecl($2, $4, $6) }
| IF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS stmt_body %prec NOELSE { IfStmt($3, $5,
[EmptyStmt]) }
| IF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS stmt_body ELSE stmt_body { IfStmt($3, $5,
$7) }
| FOR LEFT_PARENTHESIS IDENTIFIER IN expr RIGHT_PARENTHESIS loop_stmt_body
{ ForStmt($3, $5, $7) }
| WHILE LEFT_PARENTHESIS expr RIGHT_PARENTHESIS loop_stmt_body { WhileStmt($3, $5) }
| RETURN expr SEP { raise(Failure ("Return outside functions")) }
/* | BREAK SEP { raise(Failure ("Break outside loops"))  }
| CONTINUE SEP { raise(Failure ("Continue outside loops"))  } */

func_stmt:
| stmt { $1 }
| IF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS func_stmt_body %prec NOELSE { IfStmt($3,
$5, [EmptyStmt]) }
| IF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS func_stmt_body ELSE func_stmt_body
{ IfStmt($3, $5, $7) }
| FOR LEFT_PARENTHESIS IDENTIFIER IN expr RIGHT_PARENTHESIS func_loop_stmt_body
{ ForStmt($3, $5, $7) }
| WHILE LEFT_PARENTHESIS expr RIGHT_PARENTHESIS func_loop_stmt_body { WhileStmt($3,
$5) }
| RETURN expr SEP { Return($2) }
/* | BREAK SEP { raise(Failure ("Break outside loops"))  }
```

```
| CONTINUE SEP { raise(Failure ("Continue outside loops"))  } */

loop_stmt:
| stmt { $1 }
| IF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS loop_stmt_body %prec NOELSE { IfStmt($3,
$5, [EmptyStmt]) }
| IF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS loop_stmt_body ELSE loop_stmt_body
{ IfStmt($3, $5, $7) }
| FOR LEFT_PARENTHESIS IDENTIFIER IN expr RIGHT_PARENTHESIS loop_stmt_body
{ ForStmt($3, $5, $7) }
| WHILE LEFT_PARENTHESIS expr RIGHT_PARENTHESIS loop_stmt_body { WhileStmt($3, $5) }
/* | BREAK SEP { Break }
| CONTINUE SEP { Continue } */
| RETURN expr SEP { raise(Failure ("Return outside functions")) }

func_loop_stmt:
| stmt { $1 }
| IF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS func_loop_stmt_body %prec NOELSE
{ IfStmt($3, $5, [EmptyStmt]) }
| IF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS func_loop_stmt_body ELSE
func_loop_stmt_body { IfStmt($3, $5, $7) }
| FOR LEFT_PARENTHESIS IDENTIFIER IN expr RIGHT_PARENTHESIS func_loop_stmt_body
{ ForStmt($3, $5, $7) }
| WHILE LEFT_PARENTHESIS expr RIGHT_PARENTHESIS func_loop_stmt_body { WhileStmt($3,
$5) }
/* | BREAK SEP { Break }
| CONTINUE SEP { Continue } */
| RETURN expr SEP { Return($2) }

stmt_body: LEFT_CURLY_BRACKET normal_stmts RIGHT_CURLY_BRACKET { $2 }

func_stmt_body: LEFT_CURLY_BRACKET func_stmts RIGHT_CURLY_BRACKET { $2 }

loop_stmt_body: LEFT_CURLY_BRACKET loop_stmts RIGHT_CURLY_BRACKET { $2 }

func_loop_stmt_body: LEFT_CURLY_BRACKET func_loop_stmts RIGHT_CURLY_BRACKET { $2 }

/*************************************************************************
**
                        Function Call
*************************************************************************
*/
```

```
/* func_signature: IDENTIFIER LEFT_PARENTHESIS params RIGHT_PARENTHESIS { FuncSign($1,
$3) } */

/* We support the following form of function call:
*      (i)  call the function directly
*      (ii) call the function and assign the return value to variable(s)
*/
// TODO: support a, b = foo(), foo()? *)
func_call: IDENTIFIER LEFT_PARENTHESIS exprs RIGHT_PARENTHESIS { FuncCall(FId($1),
$3) }

exprs:
| { [] }
| expr_list { $1 }

expr_list:
| expr { [$1] }
| expr COMMA expr_list { $1 :: $3 }

params:
| { [] }
| param_list { $1 }

param_list:
| IDENTIFIER { [$1] }
| IDENTIFIER COMMA param_list { $1 :: $3 }

/*************************************************************************
**
                  Parallel Environment
*************************************************************************
*/

pes:
| { [] }
| pe pes { $1::$2 }

pe: PARALLEL_DEFINE IDENTIFIER LEFT_CURLY_BRACKET po_list RIGHT_CURLY_BRACKET { ($2,
$4) }

po_list:
```

```
| po { [$1] }
| po po_list { $1 :: $2 }

po: OVERLOAD OPERATOR_INDICATOR LEFT_PARENTHESIS params RIGHT_PARENTHESIS
LEFT_CURLY_BRACKET normal_stmts map_funcs reduce_func RIGHT_CURLY_BRACKET
{ {
 operator = $2;
 params = $4;
 headstmt = $7;
 mapfuncs = $8;
 reducefunc = $9;
} }

map_funcs:
| MAP IDENTIFIER func_stmt_body { [($2, $3)] }
| MAP IDENTIFIER func_stmt_body map_funcs { ($2, $3) :: $4 }

reduce_func: REDUCE func_stmt_body { $2 }

/*****************************************************************************
**
       All possible expressions, including binary expression and unary expression
*****************************************************************************
*/
expr:
// multi-dim data type
| VAR vtensor { VarTs($2) }
| tensor { Tensor($1) }
| NIL { EmptyTensor }
| asexpr { ASexpr($1) }
| STRING_LITERAL { StringLit($1) }
// Expression within parenthesis
| LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { $2 }
// Binary expression
| expr PLUS expr { Binop($1, Add, $3) }
| expr SUBTRACT expr { Binop($1, Sub, $3) }
| expr MULTIPLICATION expr { Binop($1, Mul, $3) }
| expr DOT_MULTIPLICATION expr { Binop($1, DotMul, $3) }
| expr DIVIDE expr { Binop($1, Div, $3) }
| expr POWER expr { Binop($1, Pow, $3) }
| expr DOT_POWER expr { Binop($1, DotPow, $3) }
| expr MOD expr { Binop($1, Mod, $3) }
```

```
| expr FLOOR_DIVIDE expr { Binop($1, FlrDiv, $3) }
| expr IS_EQUAL expr { Binop($1, Eq, $3) }
| expr IS_GEQ expr { Binop($1, Geq, $3) }
| expr IS_GT expr { Binop($1, Gt, $3) }
| expr IS_LEQ expr { Binop($1, Leq, $3) }
| expr IS_LT expr { Binop($1, Lt, $3) }
| expr IS_NOT_EQUAL expr { Binop($1, Neq, $3) }
| expr AND expr { Binop($1, And, $3) }
| expr OR expr { Binop($1, Or, $3) }
// Unary expression
| NOT expr { Unop(Not, $2) }
| SUBTRACT expr { Unop(Neg, $2) }
| expr TRANSPOSE { Unop(Transpose, $1) }
// A special expression, numerical range. *)
| expr COLON expr COLON expr { Range($1, $3, $5) }
// built-in functions
// TODO: necessary to do the syntax check?
| PRINT LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Print($3) }
| SHAPE LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Shape($3) }
| CAT LEFT_PARENTHESIS expr COMMA expr COMMA expr RIGHT_PARENTHESIS { Cat($3, $5,
$7) }
| ANY LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Any($3) }
| ALL LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { All($3) }
| SUM LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Sum($3) }
| ONES LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Ones($3) }
| ZEROS LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Zeros($3) }
| RAND LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Rand($3) }
| LEN LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Len($3) }
| INT_OF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Int_Of($3) }
| FLOAT_OF LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Float_Of($3) }
| FLOOR LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Floor($3) }
| CEIL LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Ceil($3) }
| ROUND LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Round($3) }
| ABS LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Abs($3) }
| LOG LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Log($3) }
| INVERSE LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Inverse($3) }
/* | SOLVE LEFT_PARENTHESIS expr COMMA expr RIGHT_PARENTHESIS { Solve($3, $5) } */
/* | SVD LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Svd($3) }
| EIG LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Eig($3) }
| EIGV LEFT_PARENTHESIS expr RIGHT_PARENTHESIS { Eigv($3) } */
// function call
| func_call { $1 }
```

```
asexpr:
   IDENTIFIER { Identifier($1) }
 | IDENTIFIER vtensor { IdentifierInd($1, $2) }

// tensor
tensor:
   LEFT_SQUARE_BRACKET tensor COMMA n_tensor RIGHT_SQUARE_BRACKET { LRTensors($2,
$4) }
 | LEFT_SQUARE_BRACKET tensor RIGHT_SQUARE_BRACKET { LRTensor($2) }
 | SUBTRACT INT_LITERAL { Tensor0(IntLit(-$2)) }
 | SUBTRACT FLOAT_LITERAL { Tensor0(FloatLit(-.$2)) }
 | INT_LITERAL { Tensor0(IntLit($1)) }
 | FLOAT_LITERAL { Tensor0(FloatLit($1)) }

n_tensor:
   tensor COMMA n_tensor { NPTensors($1, $3) }
 | tensor { NPTensor($1) }

vtensor:
   LEFT_SQUARE_BRACKET expr COMMA n_vtensor RIGHT_SQUARE_BRACKET { $2 :: $4 }
 | LEFT_SQUARE_BRACKET expr RIGHT_SQUARE_BRACKET { [$2] }

n_vtensor:
   expr { [$1] }
 | expr COMMA n_vtensor { $1 :: $3 }
```

Scanner.mll

```
{
       open Parser
}

let digit = ['0' - '9']
let digits = digit+
let exponent = ['e' 'E'] ['+' '-']? digits

rule tokenize = parse
 [' ' '\t' '\r' '\n'] { tokenize lexbuf }
```

```
| [';'] { SEP }

(* comments *)
| '#' { comment_line lexbuf }
| "'''" { comment_block lexbuf }

(* arithmetic operators *)
| '+' { PLUS }
| '-' { SUBTRACT }
| '*' { MULTIPLICATION }
| ".*" { DOT_MULTIPLICATION }
| '/' { DIVIDE }
| '^' { POWER }
| ".^" { DOT_POWER }
| "'" { TRANSPOSE }
| "%" { MOD }
| "//" { FLOOR_DIVIDE }

(* relational operators *)
| "==" { IS_EQUAL }
| ">=" { IS_GEQ }
| '>' { IS_GT }
| "<=" { IS_LEQ }
| '<' { IS_LT }
| "!=" { IS_NOT_EQUAL }

(* logical operators *)
| "&&" { AND }
| "||" { OR }
| '!' { NOT }

(* parentheses and brackets *)
| '(' { LEFT_PARENTHESIS }
| ')' { RIGHT_PARENTHESIS }
| '{' { LEFT_CURLY_BRACKET }
| '}' { RIGHT_CURLY_BRACKET }
| '[' { LEFT_SQUARE_BRACKET }
| ']' { RIGHT_SQUARE_BRACKET }

(* delimiters *)
| ',' { COMMA }
| ':' { COLON }
```

```
(* assignment *)
| "=" { ASSIGNMENT }

(* keywords *)
| "if" { IF }
| "elif" { ELIF }
| "else" { ELSE }
| "for" { FOR }
| "while" { WHILE }
| "in" { IN }
(* | "continue" { CONTINUE }
| "break" { BREAK } *)
| "return" { RETURN }
(* | "exit" { EXIT } *)
| "def" { DEFINE }
| "int" { INT }
| "float" { FLOAT }
| "var" { VAR }
| "nil" { NIL }

(* build-in function*)
| "any" { ANY }
| "all" { ALL }
| "sum" { SUM }
| "ones" { ONES }
| "zeros" { ZEROS }
| "rand" { RAND }
| "int_of" { INT_OF }
| "float_of" { FLOAT_OF }
| "floor" { FLOOR }
| "ceil" { CEIL }
| "round" { ROUND }
| "abs" { ABS }
| "log" { LOG }
| "inverse" { INVERSE }
(* | "solve" { SOLVE }
| "svd" { SVD }
| "eig" { EIG }
| "eign" { EIGV } *)
| "print" { PRINT }
| "shape" { SHAPE }
```

```
| "cat" { CAT }
(* Parallel Environment keywords *)
| "parallel_define" { PARALLEL_DEFINE }
| "overload" { OVERLOAD }
| "map" { MAP }
| "reduce" { REDUCE }
| "using" { USING }
| "end" { END }

(* Operator Names *)
| "__"['*''+''-']"__" as oname { OPERATOR_INDICATOR(oname) }

(* identifiers and literals *)
| digits as lit { INT_LITERAL(int_of_string lit) }
| ['a'-'z''A'-'Z']['a'-'z''A'-'Z''0'-'9''_']* as id { IDENTIFIER(id) }
| ('"'[^'"''\\']*('\\'_[^'"''\\']*)*'"') as str { STRING_LITERAL(String.sub str 1
(String.length str - 2)) }
| (digits '.' digit* exponent? | digits exponent | '.' digits exponent?) as lit
{ FLOAT_LITERAL(float_of_string lit) }

| eof { EOF }

(* handle illegal character*)
| _ as ch {raise (Failure("illegal character " ^ Char.escaped ch))}


and comment_line = parse
'\n' { tokenize lexbuf }
| _ { comment_line lexbuf }

and comment_block = parse
"''''" { tokenize lexbuf }
| _ { comment_block lexbuf }
```

Sast.ml

```
(* Semantically-checked Abstract Syntax Tree and functions for printing it *)

(* a hello world version of sast *)
open Ast
```

```
type sdimtype = STensorTup of tensortype * int * int array | SVoidTup

(*type index = int * int array * literal array*)

type sexpr = sdimtype * sx
and sx =
 SVoidExpr (* for semantic check only *)
| SFId of string
| STensor of literal array
| SVtensor of sexpr list
| SASexpr of sasexpr
| SStringLit of string
| SEmptyTensor
| SBinop of sexpr * bop * sexpr
| SUnop of uop * sexpr
| SRange of sexpr * sexpr * sexpr
(* Built-in functions *)
| SPrint of sexpr
| SShape of sexpr
| SCat of sexpr * sexpr * sexpr
| SAny of sexpr
| SAll of sexpr
| SSum of sexpr
| SOnes of sexpr
| SRand of sexpr
| SZeros of sexpr
| SLen of sexpr
| SInt_Of of sexpr
| SFloat_Of of sexpr
| SFloor of sexpr
| SCeil of sexpr
| SRound of sexpr
| SAbs of sexpr
| SLog of sexpr
| SInverse of sexpr
(* | SSolve of sexpr * sexpr *)
(* | SSvd of sexpr
| SEig of sexpr
| SEigv of sexpr *)
| SFuncCall of string * sexpr list
```

```
and sasexpr =
 Identifier of string
| IdentifierInd of string * sexpr list

type sstmt =
 SEmptyStmt
| SExpr of sexpr
| SAssign of sasexpr * sexpr
(* | SFuncSign of string * string list *)
| SFuncDecl of string * string list * sstmt list
| SIfStmt of sexpr * sstmt list * sstmt list
| SWhileStmt of sexpr * sstmt list
| SForStmt of string * sexpr * sstmt list
(* Keyword statement *)
| SPEInvoke of string
| SPEEnd of string
| SReturn of sexpr
(* | SBreak
| SContinue
| SExit of sexpr *)

type spofunc = {
 soperator : string;
 sparams : string list;
 smapfuncs : (string * sstmt list) list;
 sreducefunc : sstmt list;
}

type accspofunc = SDEF | SPO of spofunc

type spe = {
 sadd : accspofunc;
 sminus : accspofunc;
 smulti : accspofunc;
}

type program = (string * spe) list * sstmt list

(* let rec string_of_sexpr = function
 SId(str1) -> str1
| STensor(st1) -> string
```

```
| SBinop(se1, bop, se2) -> string_of_sexpr se1 ^ " " ^ string_of_bop bop ^ " " ^
string_of_sexpr se2
| SUnop(uop, se1) -> string_of_uop uop ^ " " ^ string_of_sexpr se1
| SRange(se1, se2, se3) -> string_of_sexpr se1 ^ ":" ^ string_of_sexpr se2 ^ ":" ^
string_of_sexpr se3

(* built-in functions *)
| SPrint(se1) -> "print(" ^ string_of_sexpr se1 ^ ")"
(* | Shape(e1) -> "shape(" ^ string_of_expr e1 ^ ")"
| Cat(e1, e2, e3) -> "cat(" ^ string_of_expr e1 ^ "," ^ string_of_expr e2 ^
string_of_expr e3 ^ ")"
| Any(e1) -> "any (" ^ string_of_expr e1 ^ ")"
| All(e1) -> "all (" ^ string_of_expr e1 ^ ")"
| Sum(e1) -> "sum (" ^ string_of_expr e1 ^ ")"
| Ones(e1) -> "ones (" ^ string_of_expr e1 ^ ")"
| Zeros(e1) -> "zeros (" ^ string_of_expr e1 ^ ")"
| Len(e1) -> "len (" ^ string_of_expr e1 ^ ")"
| Int_Of(e1) -> "int_of (" ^ string_of_expr e1 ^ ")"
| Float_Of(e1) -> "float_of (" ^ string_of_expr e1 ^ ")"
| Floor(e1) -> "floor (" ^ string_of_expr e1 ^ ")"
| Ceil(e1) -> "ceil (" ^ string_of_expr e1 ^ ")"
| Round(e1) -> "round (" ^ string_of_expr e1 ^ ")"
| Abs(e1) -> "abs(" ^ string_of_expr e1 ^ ")"
| Log(e1) -> "log (" ^ string_of_expr e1 ^ ")"
| Inverse(e1) -> "inverse (" ^ string_of_expr e1 ^ ")"
| Solve(e1, e2) -> "solve (" ^ string_of_expr e1 ^ ", " ^ string_of_expr e2 ^ ")"
| Svd(e1) -> "svd (" ^ string_of_expr e1 ^ ")"
| Eig(e1) -> "eig (" ^ string_of_expr e1 ^ ")"
| Eigv(e1) -> "Eigv (" ^ string_of_expr e1 ^ ")"
| FuncCall(e1, e2) -> (string_of_expr e1 ^ "(" ^ (String.concat "," (List.map
string_of_expr e2))) ^ ")" *)

let rec string_of_sstmt = function
 SExpr(se1) -> string_of_sexpr se1 ^ ";"
| SReturn(se1) -> "return " ^ string_of_sexpr se1 ^ "\n"
| SBreak -> "break\n"
| SContinue -> "continue\n"
| SExit(se1) -> "exit" ^ string_of_sexpr se1 ^ "\n"

and string_of_sprogram l = String.concat "" (List.map string_of_sstmt l) ^ "\n" *)
let string_of_sprogram _ = "TODO!"
```

Semant.ml

```
(* Semantic checking for the TENLab compiler *)
open Ast
open Sast

module StringHash = Hashtbl.Make(struct
 type t = string
 let equal x y = x = y
 let hash = Hashtbl.hash
end)

let symbol_table = StringHash.create 10
let function_table = StringHash.create 10
let pe_table = StringHash.create 10

let rec equal_dim d1 d2=
 match d1, d2 with
   d10::d1_, d20::d2_ -> if d10 <> d20 then false else equal_dim d1_ d2_
 | [], [] -> true
 | _, [] -> false
 | [], _ -> false

let rec check_tensor = function
 Tensor0(x) -> (match x with
   IntLit(_) -> (TensorTup(INT_Tensor, 0, [-1]), [|x|])
 | FloatLit(_) -> (TensorTup(FLOAT_Tensor, 0, [-1]), [|x|]))
| LRTensor(x) ->
   (match check_tensor(x) with
     (TensorTup(t, nd, d0::d_), y) -> (TensorTup(t, nd+1, -1::-d0::d_), y)
   | _ -> raise (Failure("ought not occur")))
| NPTensor(x) -> check_tensor(x)
| LRTensors(x1, x2) ->
   let tdy1 = check_tensor(x1) in
   let tdy2 = check_tensor(x2) in
     (match tdy1, tdy2 with
       (TensorTup(t1, nd1, d10::d1_), y1), (TensorTup(t2, _, d20::d2_), y2) ->
         if t1 = t2 && equal_dim d1_ d2_ then
           (TensorTup(t1, nd1+1, -1::-(d10 + d20)::d1_), Array.append y1 y2)
         else if t1 <> t2 then raise (Failure("invalid type"))
         else raise (Failure("invalid dim"))
     | _, _ -> raise (Failure("ought not occur")))
```

```
| NPTensors(x1, x2) ->
   let tdy1 = check_tensor(x1) in
   let tdy2 = check_tensor(x2) in
     (match tdy1, tdy2 with
       (TensorTup(t1, nd1, d10::d1_), y1), (TensorTup(t2, _, d20::d2_), y2) ->
         if t1 = t2 && equal_dim d1_ d2_ then
           (TensorTup(t1, nd1, (d10 + d20)::d1_), Array.append y1 y2)
         else if t1 <> t2 then raise (Failure("invalid type"))
         else raise (Failure("invalid dim"))
     | _, _ -> raise (Failure("ought not occur")))

(*let check_type = function
 INT_Tensor -> 0
| FLOAT_Tensor -> 1

let rec check_intv = function
 LRTensor(x) -> check_intv(NPTensor(x))
| NPTensor(x) -> (match check_tensor(x) with
 | (TensorTup(t, n, d::d_), y) -> if check_type(t) = 0 then
   (1, [(n, Array.of_list d_, y)]) else raise (Failure("Index must be integer
tensor"))
 | (_, _) -> raise (Failure("ought not occur")))
| LRTensors(x1, x2) -> check_intv(NPTensors(x1, x2))
| NPTensors(x1, x2) -> let y1 = check_intv(NPTensor(x1)) and y2 = check_intv(x2) in
   (match y1, y2 with
     (n1, ind1), (n2, ind2) -> (n1+n2, ind1@ind2))
| _ -> raise (Failure("wrong format of index"))*)

(* expr -> sexpr *)
let rec check_expr symbol_table function_table = function
| FId(id) -> if StringHash.mem symbol_table id then StringHash.remove symbol_table id;
            if StringHash.mem function_table id then (SVoidTup, SFId(id)) else raise
(Failure("function " ^ id ^ " not defined"))
| Binop(x1, bop, x2) -> (SVoidTup, SBinop(check_expr symbol_table function_table x1,
bop, check_expr symbol_table function_table x2))
| Unop(uop, x) -> (SVoidTup, SUnop(uop, check_expr symbol_table function_table x))
| Range(x1, x2, x3) ->
 let x1_ = check_expr symbol_table function_table x1 in
 let x2_ = check_expr symbol_table function_table x2 in
 let x3_ = check_expr symbol_table function_table x3 in
   (SVoidTup, SRange(x1_, x2_, x3_))
| Tensor(x) -> (match check_tensor(x) with
```

```
 | (TensorTup(t, n, _::d_), y) -> (STensorTup(t, n, Array.of_list d_), STensor(y))
 | (_, _) -> raise (Failure( "ought not occur")))
| VarTs(x) ->
    let x_ = List.map (check_expr symbol_table function_table) x in
            (SVoidTup, SVtensor(x_))
| StringLit(s) -> (SVoidTup, SStringLit(s))
| EmptyTensor -> (SVoidTup, SEmptyTensor)
| ASexpr(x) -> (match x with
 | Identifier(id) -> if StringHash.mem function_table id then StringHash.remove
function_table id;
            if StringHash.mem symbol_table id then (SVoidTup,
SASexpr(Identifier(id)))
            else raise (Failure( "variable " ^ id ^ " not defined"))
 | IdentifierInd(id, x) ->
    let x_ = List.map (check_expr symbol_table function_table) x in
                if StringHash.mem function_table id then StringHash.remove
function_table id;
                if StringHash.mem symbol_table id then (SVoidTup,
SASexpr(IdentifierInd(id, x_)))
                else raise (Failure( "variable " ^ id ^ " not defined")))
| Print(e) -> (SVoidTup, SPrint(check_expr symbol_table function_table e))
| Shape(e) -> (SVoidTup, SShape(check_expr symbol_table function_table e))
| Zeros(e) -> (SVoidTup, SZeros(check_expr symbol_table function_table e))
| Ones(e) -> (SVoidTup, SOnes(check_expr symbol_table function_table e))
| Rand(e) -> (SVoidTup, SRand(check_expr symbol_table function_table e))
| Sum(e) -> (SVoidTup, SSum(check_expr symbol_table function_table e))
| Any(e) -> (SVoidTup, SAny(check_expr symbol_table function_table e))
| All(e) -> (SVoidTup, SAll(check_expr symbol_table function_table e))
| Floor(e) -> (SVoidTup, SFloor(check_expr symbol_table function_table e))
| Ceil(e) -> (SVoidTup, SCeil(check_expr symbol_table function_table e))
| Round(e) -> (SVoidTup, SRound(check_expr symbol_table function_table e))
| Int_Of(e) -> (SVoidTup, SInt_Of(check_expr symbol_table function_table e))
| Float_Of(e) -> (SVoidTup, SFloat_Of(check_expr symbol_table function_table e))
| Abs(e) -> (SVoidTup, SAbs(check_expr symbol_table function_table e))
| Log(e) -> (SVoidTup, SLog(check_expr symbol_table function_table e))
| Len(e) -> (SVoidTup, SLen(check_expr symbol_table function_table e))
| Inverse(e) -> (SVoidTup, SInverse(check_expr symbol_table function_table e))
| Cat(e1, e2, e3) -> (SVoidTup, SCat(check_expr symbol_table function_table e1,
check_expr symbol_table function_table e2, check_expr symbol_table function_table e3))
| FuncCall(e1, e2) -> let _ = check_expr symbol_table function_table e1 in
                    let e2_ = List.map (check_expr symbol_table function_table) e2 in
                    let id = match e1 with
```

```
                        FId(id) -> id
                      | _ -> raise (Failure("invalid function id")) in
                    let argc = StringHash.find function_table id in
                    if argc <> List.length(e2) then raise (Failure("the number of
arguments mismatch"))
                    else (SVoidTup, SFuncCall(id, e2_))

(* stmt -> sstmt *)
let rec check_stmt symbol_table function_table = function
 EmptyStmt -> SEmptyStmt
| Expr(e) -> SExpr(check_expr symbol_table function_table e)
| Assign(s, e2) ->
                    let sexpr = check_expr symbol_table function_table e2 in
     (match s with
           Identifier(str1) ->
                    ignore(StringHash.add symbol_table str1 sexpr);
SAssign(Identifier(str1), sexpr)
         | IdentifierInd(id, x) ->
               let x_ = List.map (check_expr symbol_table function_table) x in
                    if StringHash.mem function_table id then StringHash.remove
function_table id;
                    if StringHash.mem symbol_table id then SAssign(IdentifierInd(id,
x_), sexpr)
                    else raise (Failure( "variable " ^ id ^ " not defined"))
     )
| IfStmt(e1, s1, s2) -> let local_symbol_table = StringHash.copy symbol_table in
                      let local_function_table = StringHash.copy function_table in
                      let e1_ = check_expr local_symbol_table local_function_table e1
in (* TODO: check if boolean expression *)
                      let s1_ = List.map (check_stmt local_symbol_table
local_function_table) s1 in
                      let s2_ = List.map (check_stmt local_symbol_table
local_function_table) s2 in
                      SIfStmt(e1_, s1_, s2_)
| WhileStmt(e1, s1) -> let local_symbol_table = StringHash.copy symbol_table in
                    let local_function_table = StringHash.copy function_table in
                    let e1_ = check_expr local_symbol_table local_function_table e1
in
                    let s1_ = List.map (check_stmt local_symbol_table
local_function_table) s1 in
                    SWhileStmt(e1_, s1_)
| ForStmt(str1, e1, s1) -> let local_symbol_table = StringHash.copy symbol_table in
```

```
                        ignore(StringHash.add local_symbol_table str1 (SVoidTup,
SVoidExpr));
                        let local_function_table = StringHash.copy function_table in
                        let e1_ = check_expr local_symbol_table local_function_table
e1 in
                        let s1_ = List.map (check_stmt local_symbol_table
local_function_table) s1 in
                        SForStmt(str1, e1_, s1_)
| FuncDecl(str1, str2, s1) -> let local_symbol_table = StringHash.copy symbol_table in
                            let argc = List.length(str2) in
                            List.iter (fun s -> StringHash.add local_symbol_table s
(SVoidTup, SVoidExpr)) str2;
                            ignore(StringHash.add function_table str1 argc);
                            let local_function_table = StringHash.copy function_table
in
                            let s1_ = List.map(check_stmt local_symbol_table
local_function_table) s1 in
                            SFuncDecl(str1, str2, s1_)
| Return(e1) -> let e1_ = check_expr symbol_table function_table e1 in
              SReturn(e1_)
(* | Break -> SBreak
| Continue -> SContinue
| Exit(e1) -> let e1_ = check_expr symbol_table function_table e1 in
            SExit(e1_) *)
| PEInvoke(s1) -> let z = StringHash.find pe_table s1 in
if (z=1) then SPEInvoke(s1) else raise (Failure("PE does not exist"))
| PEEnd(s1) -> let z = StringHash.find pe_table s1 in
if (z=1) then  SPEEnd(s1)  else raise (Failure("PE does not exist"))

let check_mapf s f (name,statements) =
let ps = StringHash.copy s in
let fs = StringHash.copy f in
(name, List.map (check_stmt ps fs) statements)

let check_po po =
let psymbol_table = StringHash.create 10 in
let pfunction_table = StringHash.create 10 in
ignore(List.iter (fun s -> StringHash.add psymbol_table s (SVoidTup, SVoidExpr))
po.params);
let app x =
ignore(List.iter (fun (name,_) -> StringHash.add psymbol_table name (SVoidTup,
SVoidExpr)) po.mapfuncs); x
```

```
in
let oprewrite = function
 | "__+__" -> "ADD"
 | "__-__" -> "SUB"
 | "__*__" -> "MUL"
 | _ -> raise (Failure("invalid operator"))
in
{
 soperator = oprewrite po.operator;
 sparams = po.params;
 smapfuncs = List.map (check_mapf psymbol_table pfunction_table) po.mapfuncs;
 sreducefunc = List.map (check_stmt psymbol_table pfunction_table) (app po.reducefunc)
}

let fill_pe pe spo =
match spo.soperator with
| "ADD" -> {sadd = SPO(spo);
            sminus = pe.sminus;
            smulti = pe.smulti;}
| "SUB" -> {sadd = pe.sadd;
            sminus = SPO(spo);
            smulti = pe.smulti;}
| "MUL" -> {sadd = pe.sadd;
            sminus = pe.sminus;
            smulti = SPO(spo);}
| _ -> raise (Failure("invalid operator"))

let check_pe (name, pos) =
ignore(StringHash.add pe_table name 1);
(name, List.fold_left fill_pe {sadd = SDEF; sminus = SDEF; smulti = SDEF;} (List.map
check_po pos))

let check (pes,stmts) =
let z1 = List.map check_pe pes in
let z2 = List.map (check_stmt symbol_table function_table) stmts in
(z1, z2)
```

codegen.ml

```
(* Code generation: translate takes a semantically checked AST and
produces LLVM IR
```

```
(*
LLVM tutorial: Make sure to read the OCaml version of the tutorial

http://llvm.org/docs/tutorial/index.html

Detailed documentation on the OCaml LLVM library:

http://llvm.moe/
http://llvm.moe/ocaml/

*)
module L = Llvm
open Ast
open Sast

module StringHash = Hashtbl.Make(struct
 type t = string
 let equal x y = x = y
 let hash = Hashtbl.hash
end)

type namespace = {
   symbol_table: L.llvalue StringHash.t;
   function_table: (L.llvalue * L.llbuilder) StringHash.t;
   func: L.llvalue;
   builder: L.llbuilder;
   global: bool;
   env : string;
}

type po = {
 funcnum : int;
 mapfuncs : L.llvalue;
 refunc : L.llvalue;
}

type actualpo = DEF | PO of po


type pe = {
 add : actualpo;
 minus : actualpo;
```

```
 multi : actualpo;
}

exception E of string (* debug information *)

let translate (spes,sstmts) =
 let global_symbol_table = StringHash.create 10 in
 let global_function_table = StringHash.create 10 in
 let global_pe_table = StringHash.create 10 in

 let context = L.global_context() in

 let the_module = L.create_module context "TENLab" in

 let int_t     = L.i32_type      context
 and i8_t      = L.i8_type       context
 and bool_t    = L.i1_type       context
 and float_t   = L.double_type   context
 and i64_t     = L.i64_type      context
 and void_t    = L.void_type     context  in
 let i8ptr_t   = L.pointer_type  i8_t
 and tensor_t = L.named_struct_type context "tensor_t" in
 let i8ptr_ptr_t = L.pointer_type i8ptr_t in
 let map_func_t = L.function_type i8ptr_t [|i8ptr_t; i8ptr_t|]
 and reduce_func_t = L.function_type i8ptr_t [|i8ptr_ptr_t|] in
 let map_func_ptr_t = L.pointer_type map_func_t in
 let map_func_ptr_ptr_t = L.pointer_type map_func_ptr_t
 and reduce_func_ptr_t = L.pointer_type reduce_func_t in
       L.struct_set_body tensor_t [| i8_t; i8_t; i8ptr_t; i8ptr_t; i8_t |] false;

 (* init *)
 let function_type = L.function_type i8_t [||] in
 let main_function = L.define_function "main" function_type the_module in

 let gen_array ltype arr =
     let n = Array.length arr in
     let newarray = if ltype = int_t then Array.make n (L.const_int int_t 0)
                    else Array.make n (L.const_int float_t 0) in
     for i = 0 to (Array.length arr)-1 do
       newarray.(i) <-
         (match arr.(i) with
           IntLit(x) -> L.const_int ltype x
```

```
        | FloatLit(x) -> L.const_float ltype x)
    done; newarray in

let gen_char ltype arr =
  let n = Array.length arr in
  let newarray = Array.make (n+1) (L.const_int ltype 0) in
  for i = 0 to (Array.length arr)-1 do
      newarray.(i) <-  L.const_int ltype (int_of_char arr.(i))
  done; newarray in

let gen_dim ltype arr =
  let n = Array.length arr in
  let newarray = Array.make n (L.const_int ltype 0) in
  for i = 0 to (Array.length arr)-1 do
      newarray.(i) <-  L.const_int ltype arr.(i)
  done; newarray in

let gen_value ltype value = L.const_int ltype value in

(* set local pointer *)
let set_localptr local_builder symbol_table args_name args_val =
  ignore(StringHash.remove symbol_table args_name);
  let alloca = L.build_alloca i8ptr_t args_name local_builder in
      ignore(L.build_store args_val alloca local_builder);
      StringHash.add symbol_table args_name alloca in


(* create a new tensor *)
let build_tensor the_namespace ltype itype dtype ndims dims data =
  let tensor = L.build_malloc tensor_t "raw_tensor" the_namespace.builder in
  (* store dtype *)
  let dtypeptr = L.build_struct_gep tensor 0 "dtype" the_namespace.builder in
  ignore(L.build_store dtype dtypeptr the_namespace.builder);

  (* store ndims *)
  let ndimsptr = L.build_struct_gep tensor 1 "ndims" the_namespace.builder in
  ignore(L.build_store ndims ndimsptr the_namespace.builder);

  (* store dims *)
  let size = Array.length dims in
  let dimsptr = L.build_malloc (L.array_type i64_t size) "dims" the_namespace.builder
in
```

```
   let dimsptr_as_i8ptr = L.build_bitcast dimsptr i8ptr_t "dims_as_i8ptr"
the_namespace.builder in
   let store_dims dim idx =
     let gep_addr = L.build_gep dimsptr [|L.const_int i64_t 0; L.const_int i64_t idx|]
"elmptr" the_namespace.builder in
     ignore(L.build_store dim gep_addr the_namespace.builder);
   in
   ignore(List.iter2 store_dims (Array.to_list dims) (List.init size (fun i -> i)));

   (* store data *)
   let size = Array.length data in
   let dataptr = L.build_malloc (L.array_type ltype size) "data" the_namespace.builder
in
   let dataptr_as_i8ptr = L.build_bitcast dataptr i8ptr_t "data_as_i8ptr"
the_namespace.builder in

   let store_data ten idx =
     let gep_addr = L.build_gep dataptr [|L.const_int itype 0; L.const_int itype idx|]
"elmptr" the_namespace.builder in
     ignore(L.build_store ten gep_addr the_namespace.builder);
   in
   ignore(List.iter2 store_data (Array.to_list data) (List.init size (fun i -> i)));

   let dimsptr_of_tensor = L.build_struct_gep tensor 2 "dimsptr" the_namespace.builder
in
   ignore(L.build_store dimsptr_as_i8ptr dimsptr_of_tensor the_namespace.builder);

   let dataptr_of_tensor = L.build_struct_gep tensor 3 "dataptr" the_namespace.builder
in
   ignore(L.build_store dataptr_as_i8ptr dataptr_of_tensor the_namespace.builder);

   (* init reference count *)
   let rcptr = L.build_struct_gep tensor 4 "rc" the_namespace.builder in
   ignore(L.build_store (L.const_int i8_t 0) rcptr the_namespace.builder);

   L.build_bitcast tensor i8ptr_t "tensor" the_namespace.builder
 in

 (* add terminal for function definition *)
 let add_terminal builder instr =
     match L.block_terminator (L.insertion_block builder) with
     Some _ -> ()    (* do nothing if terminator is there *)
```

```
    | None -> ignore (instr builder) in

(* built-in function *)
let add_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let add_func : L.llvalue =
  L.declare_function "add" add_t the_module in
let subtract_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let subtract_func : L.llvalue =
  L.declare_function "subtract" subtract_t the_module in
let negative_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let negative_func : L.llvalue =
  L.declare_function "negative" negative_t the_module in
let mult_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let mult_func : L.llvalue =
  L.declare_function "mult" mult_t the_module in
let dotmul_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let dotmul_func : L.llvalue =
  L.declare_function "dotmul" dotmul_t the_module in
let divide_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let divide_func : L.llvalue =
  L.declare_function "divide" divide_t the_module in
let floordivide_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let floordivide_func : L.llvalue =
  L.declare_function "floordivide" floordivide_t the_module in
let matpow_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let matpow_func : L.llvalue =
  L.declare_function "matpow" matpow_t the_module in
let dotpow_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let dotpow_func : L.llvalue =
  L.declare_function "dotpow" dotpow_t the_module in
let mod_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let mod_func : L.llvalue =
```

```
  L.declare_function "mod" mod_t the_module in
let transpose_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let transpose_func : L.llvalue =
  L.declare_function "transpose" transpose_t the_module in
let equal_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let equal_func : L.llvalue =
  L.declare_function "equal" equal_t the_module in
let notequal_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let notequal_func : L.llvalue =
  L.declare_function "notequal" notequal_t the_module in
let greater_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let greater_func : L.llvalue =
  L.declare_function "greater" greater_t the_module in
let greaterequal_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let greaterequal_func : L.llvalue =
  L.declare_function "greaterequal" greaterequal_t the_module in
let less_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let less_func : L.llvalue =
  L.declare_function "less" less_t the_module in
let lessequal_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let lessequal_func : L.llvalue =
  L.declare_function "lessequal" lessequal_t the_module in
let range_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t; i8ptr_t |] in
let range_func : L.llvalue =
  L.declare_function "range" range_t the_module in
let print_t : L.lltype =
  L.function_type void_t [| i8ptr_t |] in
let print_func : L.llvalue =
  L.declare_function "print" print_t the_module in
let logicaland_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let logicaland_func : L.llvalue =
  L.declare_function "logicaland" logicaland_t the_module in
let logicalor_t : L.lltype =
```

```
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t |] in
let logicalor_func : L.llvalue =
  L.declare_function "logicalor" logicalor_t the_module in
let logicalnot_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let logicalnot_func : L.llvalue =
  L.declare_function "logicalnot" logicalnot_t the_module in
let len_t : L.lltype =
  L.function_type int_t [| i8ptr_t |] in
let len : L.llvalue =
  L.declare_function "len" len_t the_module in

let zeros_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let zeros_func : L.llvalue =
  L.declare_function "zeros" zeros_t the_module in
let cat_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t; i8ptr_t; i8ptr_t |] in
let cat_func : L.llvalue =
  L.declare_function "cat" cat_t the_module in
let shape_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let shape_func : L.llvalue =
  L.declare_function "shape" shape_t the_module in
let ones_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let ones_func : L.llvalue =
  L.declare_function "ones" ones_t the_module in
let tensor_rand_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let tensor_rand_func : L.llvalue =
  L.declare_function "tensor_rand" tensor_rand_t the_module in
let sum_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let sum_func : L.llvalue =
  L.declare_function "sum" sum_t the_module in
let any_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let any_func : L.llvalue =
  L.declare_function "any" any_t the_module in
let all_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
```

```
let all_func : L.llvalue =
  L.declare_function "all" all_t the_module in
let tensor_abs_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let tensor_abs_func : L.llvalue =
  L.declare_function "tensor_abs" tensor_abs_t the_module in
let tensor_log_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let tensor_log_func : L.llvalue =
  L.declare_function "tensor_log" tensor_log_t the_module in
let tensor_floor_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let tensor_floor_func : L.llvalue =
  L.declare_function "tensor_floor" tensor_floor_t the_module in
let tensor_ceil_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let tensor_ceil_func : L.llvalue =
  L.declare_function "tensor_ceil" tensor_ceil_t the_module in
let tensor_round_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let tensor_round_func : L.llvalue =
  L.declare_function "tensor_round" tensor_round_t the_module in
let int_of_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let int_of_func : L.llvalue =
  L.declare_function "int_of" int_of_t the_module in
let float_of_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let float_of_func : L.llvalue =
  L.declare_function "float_of" float_of_t the_module in
let inverse_t : L.lltype =
  L.function_type i8ptr_t [| i8ptr_t |] in
let inverse_func : L.llvalue =
    L.declare_function "inverse" inverse_t the_module in

let bool_of_zero_t : L.lltype =
  L.function_type bool_t [| i8ptr_t |] in
let bool_of_zero : L.llvalue =
  L.declare_function "bool_of_zero" bool_of_zero_t the_module in

let pe_calc_t : L.lltype =
```

```
 L.function_type i8ptr_t [|map_func_ptr_ptr_t; int_t;  reduce_func_ptr_t; i8ptr_t;
i8ptr_t|] in
 let pe_calc : L.llvalue =
 L.declare_function "pe_calc" pe_calc_t the_module in
 let index_get_t : L.lltype =
 L.function_type i8ptr_t [|i8ptr_t; i8ptr_t|] in
 let index_get : L.llvalue =
 L.declare_function "index_get" index_get_t the_module in
 let index_get_int_t : L.lltype =
 L.function_type i8ptr_t [|i8ptr_t; int_t|] in
 let index_get_int : L.llvalue =
 L.declare_function "index_get_int" index_get_int_t the_module in
 let index_put_t : L.lltype =
 L.function_type void_t [|i8ptr_t; i8ptr_t; i8ptr_t|] in
 let index_put : L.llvalue =
 L.declare_function "index_put" index_put_t the_module in
 let increase_rc_t : L.lltype =
 L.function_type void_t [| i8ptr_t |] in
 let increase_rc: L.llvalue =
 L.declare_function "increase_rc" increase_rc_t the_module in
 let decrease_rc_t : L.lltype =
 L.function_type void_t [| i8ptr_t |] in
 let decrease_rc: L.llvalue =
 L.declare_function "decrease_rc" decrease_rc_t the_module in

 let lookup id the_namespace = if StringHash.mem the_namespace.symbol_table id then
StringHash.find the_namespace.symbol_table id
                              else if the_namespace.global then let init =
L.define_global id (L.const_pointer_null i8ptr_t) the_module in
                                                                  ignore(StringHash.add
the_namespace.symbol_table id init);
                                                                  init
                              else let init = L.build_alloca i8ptr_t id
the_namespace.builder in
                                   ignore(L.build_store (L.const_pointer_null
i8ptr_t) init the_namespace.builder);
                                   ignore(StringHash.add the_namespace.symbol_table
id init);
                                   init in

 let pe_add the_namespace expr1 expr2 =
   let pe = the_namespace.env in
```

```
   let builder = the_namespace.builder in
   let pef = StringHash.find global_pe_table pe in
     match pef.add with
     | DEF -> L.build_call add_func [| expr1 ; expr2 |]
     | PO(po) ->
       let newpr = L.build_load po.refunc "newpr" builder in
       let newmp = L.build_struct_gep po.mapfuncs 0 "newmp" builder in
       L.build_call pe_calc [| newmp; (L.const_int int_t po.funcnum); newpr ;expr1 ;
expr2 |]
 in
 let pe_sub the_namespace expr1 expr2 =
   let pe = the_namespace.env in
   let builder = the_namespace.builder in
   let pef = StringHash.find global_pe_table pe in
     match pef.minus with
     | DEF -> L.build_call subtract_func [| expr1 ; expr2 |]
     | PO(po) ->
       let newpr = L.build_load po.refunc "newpr" builder in
       let newmp = L.build_struct_gep po.mapfuncs 0 "newmp" builder in
       L.build_call pe_calc [| newmp; (L.const_int int_t po.funcnum); newpr ;expr1 ;
expr2 |]
 in
 let pe_mul the_namespace expr1 expr2 =
   let pe = the_namespace.env in
   let builder = the_namespace.builder in
   let pef = StringHash.find global_pe_table pe in
     match pef.multi with
     | DEF -> L.build_call mult_func [| expr1 ; expr2 |]
     | PO(po) ->
       let newpr = L.build_load po.refunc "newpr" builder in
       let newmp = L.build_struct_gep po.mapfuncs 0 "newmp" builder in
       L.build_call pe_calc [| newmp; (L.const_int int_t po.funcnum); newpr ;expr1 ;
expr2 |]
 in
 (* expression translation *)
 let rec genExpr the_namespace se = match se with
   | (_, SBinop(se1, op, se2)) ->
       let se1_ = genExpr the_namespace se1
       and se2_ = genExpr the_namespace se2 in
       (match op with
         Add -> (match the_namespace.env with
         | "default" -> L.build_call add_func [| se1_ ; se2_ |]
```

```
        | _ -> pe_add the_namespace se1_ se2_)
      | Sub -> (match the_namespace.env with
        | "default" -> L.build_call subtract_func [| se1_ ; se2_ |]
        | _ -> pe_sub the_namespace se1_ se2_)
      | Mul -> (match the_namespace.env with
        | "default" -> L.build_call mult_func [| se1_ ; se2_ |]
        | _ -> pe_mul the_namespace se1_ se2_)
      | DotMul -> L.build_call dotmul_func [| se1_ ; se2_ |]
      | Div -> L.build_call divide_func [| se1_ ; se2_ |]
      | FlrDiv -> L.build_call floordivide_func [| se1_ ; se2_ |]
      | Pow -> L.build_call matpow_func [| se1_ ; se2_ |]
      | DotPow -> L.build_call dotpow_func [| se1_ ; se2_ |]
      | Mod -> L.build_call mod_func [| se1_ ; se2_ |]
      | Eq -> L.build_call equal_func [| se1_ ; se2_ |]
      | Neq -> L.build_call notequal_func [| se1_ ; se2_ |]
      | Geq -> L.build_call greaterequal_func [| se1_ ; se2_ |]
      | Gt -> L.build_call greater_func [| se1_ ; se2_ |]
      | Leq -> L.build_call lessequal_func [| se1_ ; se2_ |]
      | Lt -> L.build_call less_func [| se1_ ; se2_ |]
      | And -> L.build_call logicaland_func [| se1_ ; se2_ |]
      | Or -> L.build_call logicalor_func [| se1_ ; se2_ |]
      ) "tmpOp" the_namespace.builder
      (* cast_voidpt_to_tensor the_namespace "tmpOp" tmpOp *)
  | (_, SUnop(uop, se1)) ->
    let se1_ = genExpr the_namespace se1 in
    (match uop with
      Transpose -> L.build_call transpose_func
    | Not -> L.build_call logicalnot_func
    | Neg -> L.build_call negative_func
    ) [| se1_ |] "tmpOp" the_namespace.builder
  | (STensorTup(t, n, d), STensor(y)) ->
      (match t with
        INT_Tensor -> build_tensor the_namespace int_t int_t (gen_value i8_t 0)
(gen_value i8_t n) (gen_dim i64_t d) (gen_array int_t y)
      | FLOAT_Tensor -> build_tensor the_namespace float_t i64_t (gen_value i8_t 1)
(gen_value i8_t n) (gen_dim i64_t d) (gen_array float_t y)
      )
  | (_, SStringLit(s)) -> build_tensor the_namespace i8_t i8_t (gen_value i8_t 2)
(gen_value i8_t 1) (gen_dim i64_t [| String.length s |]) (gen_char i8_t
(Array.of_seq(String.to_seq s)))
  | (_, SEmptyTensor) -> build_tensor the_namespace i8_t i8_t (gen_value i8_t 21)
(gen_value i8_t 0) (gen_dim i64_t [||]) (gen_dim i8_t [||])
```

```
  | (_, SVtensor(x)) ->
    let x_ = Array.of_list(List.map (genExpr the_namespace) x) in
          let dims = gen_dim i64_t [|Array.length(x_)|] in
          let data = x_ in
            build_tensor the_namespace i8ptr_t i64_t (gen_value i8_t 3) (gen_value
i8_t 1) dims data
    (*let rec gen_vartensor = function
      Tensor0(x) -> (match x with
        IntLit(_) -> [|build_tensor the_namespace int_t int_t (gen_value i8_t 0)
(gen_value i8_t 0) (gen_dim i8_t [||]) (gen_array int_t [|x|])|]
      | FloatLit(_) -> [|build_tensor the_namespace float_t i64_t (gen_value i8_t 1)
(gen_value i8_t 0) (gen_dim i8_t [||]) (gen_array float_t [|x|])|])
    | LRTensor(x) -> let y = gen_vartensor(x) in
                     let dims = gen_dim i8_t [|Array.length(y)|] in
                     let data =  y in
                     let ptrx = build_tensor the_namespace i8ptr_t i64_t (gen_value
i8_t 3) (gen_value i8_t 1) dims data in
                     [|ptrx|]
    | LRTensors(x1, x2) -> let y1 = gen_vartensor(x1) and y2 = gen_vartensor(x2) in
                           let dims = gen_dim i8_t
[|Array.length(y1)+Array.length(y2)|] in
                           let data = Array.append y1 y2 in
                           let ptrx = build_tensor the_namespace i8ptr_t i64_t
(gen_value i8_t 3) (gen_value i8_t 1) dims data in
                           [|ptrx|]
    | NPTensor(x) -> gen_vartensor(x)
    | NPTensors(x1, x2) -> let y1 = gen_vartensor(x1) and y2 = gen_vartensor(x2) in
                           Array.append y1 y2
    in let y = gen_vartensor(x) in y.(0)*)
  | (_, SASexpr(x)) -> (match x with
      Identifier(id) -> L.build_load (lookup id the_namespace) id
the_namespace.builder
    | IdentifierInd(s, x) ->
      let x_ = Array.of_list(List.map (genExpr the_namespace) x) in
          let dims = gen_dim i64_t [|Array.length(x_)|] in
          let data = x_ in
          let xptr = build_tensor the_namespace i8ptr_t i64_t (gen_value i8_t 3)
(gen_value i8_t 1) dims data in
          let sptr = L.build_load (lookup s the_namespace) s the_namespace.builder in
          L.build_call index_get [|sptr; xptr|] "access_tensor" the_namespace.builder
      (*(match x with
        (nlist, indlist) -> let rec gen_indlist = function
```

```
            [] -> [||]
          | (n, d, y)::indlist_ -> let i0 = genExpr the_namespace
(STensorTup(INT_Tensor, n, d), STensor(y)) in
            let y1 = [|i0|] and y2 = gen_indlist(indlist_) in Array.append y1 y2 in
            let dims = gen_dim i8_t [|nlist|] in
            let data = gen_indlist(indlist) in
            let xptr = build_tensor the_namespace i8ptr_t i64_t (gen_value i8_t 3)
(gen_value i8_t 1) dims data in
            let sptr = L.build_load (lookup s the_namespace) s the_namespace.builder in
            L.build_call index_get [|sptr; xptr|] "access_tensor" the_namespace.builder
          )*)
      )
    | (_, SRange(se1, se2, se3)) ->
              let se1_ = genExpr the_namespace se1 in
              let se2_ = genExpr the_namespace se2 in
              let se3_ = genExpr the_namespace se3 in
              L.build_call range_func [| se1_; se2_; se3_ |] "tmpOp"
the_namespace.builder
    | (_, SPrint(se1)) -> let se1_ = genExpr the_namespace se1 in
                          L.build_call print_func [| se1_ |] "" the_namespace.builder
    | (_, SZeros(se1)) -> let se1_ = genExpr the_namespace se1 in
                          L.build_call zeros_func [| se1_ |] "zeros"
the_namespace.builder
    | (_, SShape(se1)) -> let se1_ = genExpr the_namespace se1 in
                          L.build_call shape_func [| se1_ |] "shape"
the_namespace.builder
    | (_, SOnes(se1)) -> let se1_ = genExpr the_namespace se1 in
                          L.build_call ones_func [| se1_ |] "ones"
the_namespace.builder
    | (_, SRand(se1)) -> let se1_ = genExpr the_namespace se1 in
                          L.build_call tensor_rand_func [| se1_ |] "rand"
the_namespace.builder
    | (_, SSum(se1)) -> let se1_ = genExpr the_namespace se1 in
                          L.build_call sum_func [| se1_ |] "sum" the_namespace.builder
    | (_, SAny(se1)) -> let se1_ = genExpr the_namespace se1 in
                          L.build_call any_func [| se1_ |] "any" the_namespace.builder
    | (_, SAll(se1)) -> let se1_ = genExpr the_namespace se1 in
                          L.build_call all_func [| se1_ |] "all" the_namespace.builder
    | (_, SAbs(se1)) -> let se1_ = genExpr the_namespace se1 in
                          L.build_call tensor_abs_func [| se1_ |] "abs"
the_namespace.builder
    | (_, SLog(se1)) -> let se1_ = genExpr the_namespace se1 in
```

```
                        L.build_call tensor_log_func [| se1_ |] "log"
the_namespace.builder
   | (_, SInt_Of(se1)) -> let se1_ = genExpr the_namespace se1 in
                        L.build_call int_of_func [| se1_ |] "int_of"
the_namespace.builder
   | (_, SFloat_Of(se1)) -> let se1_ = genExpr the_namespace se1 in
                        L.build_call float_of_func [| se1_ |] "float_of"
the_namespace.builder
   | (_, SFloor(se1)) -> let se1_ = genExpr the_namespace se1 in
                        L.build_call tensor_floor_func [| se1_ |] "abs"
the_namespace.builder
   | (_, SCeil(se1)) -> let se1_ = genExpr the_namespace se1 in
                        L.build_call tensor_ceil_func [| se1_ |] "ceil"
the_namespace.builder
   | (_, SRound(se1)) -> let se1_ = genExpr the_namespace se1 in
                        L.build_call tensor_round_func [| se1_ |] "round"
the_namespace.builder
   | (_, SInverse(se1)) -> let se1_ = genExpr the_namespace se1 in
                        L.build_call inverse_func [| se1_ |] "inverse"
the_namespace.builder
   | (_, SCat(se1, se2, se3)) -> let se1_ = genExpr the_namespace se1
                                 and se2_ = genExpr the_namespace se2
                                 and se3_ = genExpr the_namespace se3 in
                        L.build_call cat_func [| se1_ ; se2_ ; se3_ |] "cat"
the_namespace.builder
   | (_, SFuncCall(str1, se1)) -> let (the_function, _) = StringHash.find
the_namespace.function_table str1 in
                                  let argv = List.map (genExpr the_namespace) se1 in
                                  L.build_call the_function (Array.of_list argv) "ret"
the_namespace.builder
   | (_, _) -> gen_value i8ptr_t 0 in

 let build_fn fname argc =
   let ftype : L.lltype =
     L.function_type i8ptr_t (Array.make argc i8ptr_t) in
   let the_function = L.define_function fname ftype the_module in
   let builder = L.builder_at_end context (L.entry_block the_function) in
   (the_function, builder)
 in

 let rec stmt the_namespace = function
     SEmptyStmt -> the_namespace
```

```
   | SExpr(se) -> ignore(genExpr the_namespace se); the_namespace
   | SAssign(s, se1) ->
           (match s with
             Identifier(id) ->
                         let rhs = genExpr the_namespace se1 in
                         let lhs = lookup id the_namespace in
                         let lhsptr = L.build_load lhs "lhsptr" the_namespace.builder
in
                         ignore(L.build_call increase_rc [| rhs |] ""
the_namespace.builder);
                         ignore(L.build_call decrease_rc [| lhsptr |] ""
the_namespace.builder);
                         ignore(L.build_store rhs lhs the_namespace.builder);
                         the_namespace
           | IdentifierInd(s, x) ->
                         let rhs = genExpr the_namespace se1 in
                         let x_ = Array.of_list(List.map (genExpr the_namespace) x) in
                           let dims = gen_dim i64_t [|Array.length(x_)|] in
                           let data = x_ in
                           let xptr = build_tensor the_namespace i8ptr_t i64_t
(gen_value i8_t 3) (gen_value i8_t 1) dims data in
                           let sptr = L.build_load (lookup s the_namespace) s
the_namespace.builder in
                           ignore(L.build_call index_put [|sptr; xptr; rhs|] ""
the_namespace.builder);
                           the_namespace
                 (*(match x with
                         (nlist, indlist) -> let rec gen_indlist = function
                           [] -> [||]
                         | (n, d, y)::indlist_ -> let i0 = genExpr the_namespace
(STensorTup(INT_Tensor, n, d), STensor(y)) in
                           let y1 = [|i0|] and y2 = gen_indlist(indlist_) in
Array.append y1 y2 in
                           let dims = gen_dim i8_t [|nlist|] in
                           let data = gen_indlist(indlist) in
                           let xptr = build_tensor the_namespace i8ptr_t i64_t
(gen_value i8_t 3) (gen_value i8_t 1) dims data in
                           let sptr = L.build_load (lookup s the_namespace) s
the_namespace.builder in
                           L.build_call index_put [|sptr; xptr; rhs|] ""
the_namespace.builder
                 ); the_namespace*)
```

```
          )
   | SFuncDecl(str1, str2, ss1) -> let argc = List.length(str2) in
                                  let (the_function, the_builder) = build_fn str1 argc in
                                  ignore(StringHash.add the_namespace.function_table str1
(the_function, the_builder));
                                  let local_symbol_table = StringHash.copy
the_namespace.symbol_table in
                                  List.iter (fun id -> StringHash.remove local_symbol_table
id) str2;
                                  let local_function_table = StringHash.copy
the_namespace.function_table in
                                  let argv = Array.to_list (L.params the_function) in
                                  List.iter2 (set_localptr the_builder local_symbol_table)
str2 argv;
                                  let build_return b = L.build_ret (L.const_pointer_null
i8ptr_t) b in
                                  let local_namespace = {symbol_table = local_symbol_table;
function_table = local_function_table; func = the_function; builder = the_builder;
global = false; env = the_namespace.env} in
                                  ignore(List.fold_left stmt local_namespace ss1);
                                  ignore(add_terminal the_builder build_return);
                                  the_namespace (* return the main namespace *)
   | SIfStmt(se1, ss1, ss2) -> let se1_ = genExpr the_namespace se1 in
                               let bool_val = L.build_call bool_of_zero [| se1_ |]
"bool" the_namespace.builder in
                               let merge_bb = L.append_block context "merge"
the_namespace.func in
                               let build_br_merge = L.build_br merge_bb in
                               let then_bb = L.append_block context "then"
the_namespace.func in
                               let local_builder = L.builder_at_end context then_bb in
                               let local_namespace = {symbol_table =
the_namespace.symbol_table;
                                                      function_table =
the_namespace.function_table;
                                                      func = the_namespace.func;
                                                      builder = local_builder;
                                                      global = the_namespace.global;
                                                      env = the_namespace.env} in
                               let local_namespace = List.fold_left stmt
local_namespace ss1 in
```

```
                                ignore(add_terminal local_namespace.builder
build_br_merge);
                                let else_bb = L.append_block context "else"
the_namespace.func in
                                let local_builder = L.builder_at_end context else_bb in
                                let local_namespace = {symbol_table =
the_namespace.symbol_table;
                                                       function_table =
the_namespace.function_table;
                                                       func = the_namespace.func;
                                                       builder = local_builder;
                                                       global = the_namespace.global;
                                                       env = the_namespace.env} in
                                let local_namespace = List.fold_left stmt
local_namespace ss2 in
                                ignore(add_terminal local_namespace.builder
build_br_merge);
                                ignore(L.build_cond_br bool_val then_bb else_bb
the_namespace.builder);
                                let builder = L.builder_at_end context merge_bb in
                                {symbol_table = the_namespace.symbol_table;
function_table = the_namespace.function_table; func = the_namespace.func; builder =
builder; global = the_namespace.global; env = the_namespace.env}
   | SWhileStmt(se1, ss1) -> let pred_bb = L.append_block context "while"
the_namespace.func in
                            ignore(L.build_br pred_bb the_namespace.builder);
                            let body_bb = L.append_block context "while_body"
the_namespace.func in
                            let local_builder = L.builder_at_end context body_bb in
                            let local_namespace = {symbol_table =
the_namespace.symbol_table;
                                                   function_table =
the_namespace.function_table;
                                                   func = the_namespace.func;
                                                   builder = local_builder;
                                                   global = the_namespace.global;
                                                   env = the_namespace.env} in
                            let local_namespace = List.fold_left stmt local_namespace
ss1 in
                            ignore(add_terminal local_namespace.builder (L.build_br
pred_bb));
                            let pred_builder = L.builder_at_end context pred_bb in
```

```
                              let local_namespace = {symbol_table =
the_namespace.symbol_table;
                                                     function_table =
the_namespace.function_table;
                                                     func = the_namespace.func;
                                                     builder = pred_builder;
                                                     global = the_namespace.global;
                                                     env = the_namespace.env} in
                              let se1_ = genExpr local_namespace se1 in
                              let bool_val = L.build_call bool_of_zero [| se1_ |]
"bool" local_namespace.builder in
                              let merge_bb = L.append_block context "merge"
the_namespace.func in
                              ignore(L.build_cond_br bool_val body_bb merge_bb
pred_builder);
                              let builder = L.builder_at_end context merge_bb in
                              {symbol_table = the_namespace.symbol_table;
function_table = the_namespace.function_table; func = the_namespace.func; builder =
builder; global = the_namespace.global;env = the_namespace.env}
    | SForStmt(str1, se1, ss1) -> let idxptr = L.build_alloca int_t "idxptr"
the_namespace.builder in
                                  ignore(L.build_store (L.const_int int_t (0)) idxptr
the_namespace.builder);
                                  let tensor = genExpr the_namespace se1 in
                                  let size = L.build_call len [| tensor |] "length"
the_namespace.builder in
                                  let pred_bb = L.append_block context "for"
the_namespace.func in
                                  ignore(L.build_br pred_bb the_namespace.builder);
                                  let pred_builder = L.builder_at_end context pred_bb
in
                                  let indicator = L.build_load idxptr "idx"
pred_builder in
                                  let new_indicator = L.build_add indicator
(L.const_int int_t 1) "new_idx"  pred_builder in
                                  ignore(L.build_store new_indicator idxptr
pred_builder);
                                  let cond = (L.build_icmp L.Icmp.Sgt) new_indicator
size "condition" pred_builder in

                                  let body_bb = L.append_block context "for_body"
the_namespace.func in
```

```
                                let body_builder = L.builder_at_end context body_bb
in
                                let local_namespace = {symbol_table =
the_namespace.symbol_table;
                                                       function_table =
the_namespace.function_table;
                                                       func = the_namespace.func;
                                                       builder = body_builder;
                                                       global = the_namespace.global;
                                                       env = the_namespace.env} in
                                (* let indicator_as_tensor = build_tensor
local_namespace int_t int_t (gen_value i8_t 0) (gen_value i8_t 1) [| gen_value i8_t 1
|] [| indicator |] in *)
                                let str1ptr = lookup str1 local_namespace in
                                let indicator_as_tensor = L.build_call index_get_int
[|tensor; indicator|] "indicator_as_tensor" body_builder in
                                ignore(L.build_store indicator_as_tensor str1ptr
body_builder);
                                let local_namespace = List.fold_left stmt
local_namespace ss1 in
                                ignore(add_terminal local_namespace.builder
(L.build_br pred_bb));

                                let merge_bb = L.append_block context "merge"
the_namespace.func in
                                ignore(L.build_cond_br cond merge_bb body_bb
pred_builder);
                                let builder = L.builder_at_end context merge_bb in
                                {symbol_table = the_namespace.symbol_table;
function_table = the_namespace.function_table; func = the_namespace.func; builder =
builder; global = the_namespace.global; env = the_namespace.env}
   | SReturn(se1) -> ignore(L.build_ret (genExpr the_namespace se1)
the_namespace.builder); the_namespace
   | SPEInvoke(str1) -> {symbol_table = the_namespace.symbol_table; function_table =
the_namespace.function_table; func = the_namespace.func; builder =
the_namespace.builder; global = the_namespace.global; env = str1}
   | SPEEnd(_) -> {symbol_table = the_namespace.symbol_table; function_table =
the_namespace.function_table; func = the_namespace.func; builder =
the_namespace.builder; global = the_namespace.global; env = "default"}
  and pedecl main_namespace (pename, pebody) =
     let map_value_helper bigname po_map_array idx (name, _) =
       let fname = bigname ^ name in
```

```
      let tmp = L.build_struct_gep po_map_array idx "tmp" main_namespace.builder in
      let ftmp = match (L.lookup_function fname the_module) with
                   Some z-> z
                 | _ -> raise (Failure("Bug"))in
      ignore(L.build_store ftmp tmp main_namespace.builder);
      idx+1
    in

    let pofp_creator pename pofunc =
      let poname = pename ^ pofunc.soperator in
      let funcn = List.length(pofunc.smapfuncs) in
      let po_map_id = poname ^ "maps" in
      let po_map_array = L.define_global po_map_id (L.const_null (L.array_type
map_func_ptr_t funcn)) the_module in
      (* ignore(StringHash.add main_namespace.symbol_table po_map_id po_map_array);
*)
      ignore(List.fold_left (map_value_helper poname po_map_array) 0
pofunc.smapfuncs);
      let po_reduce_id = poname ^ "reduce" in
      let po_reduce = L.define_global po_reduce_id (L.const_pointer_null
reduce_func_ptr_t) the_module in
      (* ignore(StringHash.add main_namespace.symbol_table po_reduce_id po_reduce);
*)
      let ftmp = match (L.lookup_function (poname ^ "reduce") the_module) with
                   Some z-> z
                 | _ -> raise (Failure("Bug")) in
      ignore(L.build_store ftmp po_reduce main_namespace.builder);
      PO({
        funcnum = funcn;
        mapfuncs = po_map_array;
        refunc = po_reduce;
      })
    in
    let build_pofunc bigname paras (name, stmts) =
      let fname = bigname ^ name in
      let (the_function, the_builder) = build_fn fname 2 in
      let local_symbol_table = StringHash.create 10 in
      let local_function_table = StringHash.create 10 in
      let argv = Array.to_list (L.params the_function) in
      List.iter2 (set_localptr the_builder local_symbol_table) paras argv;
      let build_return b = L.build_ret (L.const_pointer_null i8ptr_t) b in
```

```
      let local_namespace = {symbol_table = local_symbol_table; function_table =
local_function_table; func = the_function; builder = the_builder; global = false; env
= "default"} in
      ignore(List.fold_left stmt local_namespace stmts);
      ignore(add_terminal the_builder build_return);
    in
    let reduce_value_helper local_symbol_table the_builder reduce_array idx name =
      let alloca = L.build_alloca i8ptr_t name the_builder in
      let newa = L.build_load reduce_array "newa" the_builder in
      let tmp = L.build_gep newa [|(L.const_int int_t idx)|] "tmp" the_builder in
      let newv = L.build_load tmp "newv" the_builder in
      ignore(L.build_store newv alloca the_builder);
      ignore(StringHash.add local_symbol_table name alloca);
      idx+1
    in

    let build_reducefunc name vars stmts =
      let the_function = L.define_function name reduce_func_t the_module in
      let the_builder = L.builder_at_end context (L.entry_block the_function) in
      let local_symbol_table = StringHash.create 10 in
      let local_function_table = StringHash.create 10 in

      let argval = List.hd(Array.to_list (L.params the_function)) in
      ignore(L.set_value_name "result" argval);
      let alloca = L.build_alloca i8ptr_ptr_t "result" the_builder in
      ignore(L.build_store argval alloca the_builder);

      ignore(List.fold_left (reduce_value_helper local_symbol_table the_builder
alloca) 0 vars);

      let build_return b = L.build_ret (L.const_pointer_null i8ptr_t) b in
      let local_namespace = {symbol_table = local_symbol_table; function_table =
local_function_table; func = the_function; builder = the_builder; global = false; env
= "default"} in
      ignore(List.fold_left stmt local_namespace stmts);
      ignore(add_terminal the_builder build_return);
    in
    let pogenerate pename pofunc =
      let poname = pename ^ pofunc.soperator in
      ignore(List.iter (build_pofunc poname pofunc.sparams) pofunc.smapfuncs);
      ignore(build_reducefunc (poname ^ "reduce") (List.map (fun (name, _) -> name)
pofunc.smapfuncs) pofunc.sreducefunc);
```

```
      pofp_creator pename pofunc
    in
    let podecl pename pof =
    match pof with
      SDEF -> DEF
    | SPO(p) -> pogenerate pename p
    in
    ignore(StringHash.add global_pe_table pename
    {
      add = podecl pename pebody.sadd;
      minus = podecl pename pebody.sminus;
      multi = podecl pename pebody.smulti;
    });
    main_namespace

in

let main_builder = L.builder_at_end context (L.entry_block main_function) in
let main_namespace = {symbol_table = global_symbol_table; function_table =
global_function_table; func = main_function; builder = main_builder; global = true;
env = "default"} in
let main_namespace = List.fold_left pedecl main_namespace spes in
let main_namespace = List.fold_left stmt main_namespace sstmts in
ignore(L.build_ret (L.const_int i8_t 0) main_namespace.builder);
the_module;
```

lib/builtin.cc

```
#include "tensor.h"
#include <array>

torch::Tensor add_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   torch::Tensor z_t = x_t + y_t;
   return z_t;
}

extern "C" void *add(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;
```

```
    check(x->type == y->type, "Not consistent type");
    if (y->ndim != 0) {
        check(x->ndim == y->ndim, "Not consistent dimension\n");
        bool flag = true;
        for (int i = 0; i < x->ndim; i++) {
            if (x->dims[i] != y->dims[i]) {
                flag = false;
                break;
            }
        }
        check(flag == true, "Not consistent dimension\n");
    }


    return (void *)fromTensor(add_t(toTensor(x), toTensor(y)));
}

torch::Tensor all_t(const torch::Tensor &x_t)
{
    torch::Tensor a_t = torch::all(x_t);
    return a_t.toType(torch::kInt32);
}

extern "C" void *all(void *a)
{
    tensor *x = (tensor *)a;

    return (void *)fromTensor(all_t(toTensor(x)));
}

torch::Tensor any_t(const torch::Tensor &x_t)
{
    torch::Tensor a_t = torch::any(x_t);
    return a_t.toType(torch::kInt32);
}

extern "C" void *any(void *a)
{
    tensor *x = (tensor *)a;

    return (void *)fromTensor(any_t(toTensor(x)));
```

```
}

extern "C" bool bool_of_zero(void *a)
{
   tensor *x = (tensor *)a;
   if (*(int *)(x -> data) == 0) {
       return false;
   } else {
       return true;
   }
}

torch::Tensor cat_t(const torch::Tensor &x_t, const torch::Tensor &y_t, const
torch::Tensor &dim_t)
{
   torch::Tensor a_t = torch::cat({x_t, y_t}, dim_t.item().to<int64_t>());
   return a_t;
}

extern "C" void *cat(void *a, void *b, void *c)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;
   tensor *z = (tensor *)c;

   check(x->type == y->type, "Not consistent type");
   check(z->ndim == 0, "Second tensor should be 0-dim tensor");

   return (void *)fromTensor(cat_t(toTensor(x), toTensor(y), toTensor(z)));
}

torch::Tensor divide_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   torch::Tensor z_t = x_t.div(y_t.item());
   return z_t;
}

extern "C" void *divide(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;
```

```
    check(x->type == y->type, "Not consistent type");
    check(y->ndim == 0, "Second tensor should be 0-dim tensor");

    return (void *)fromTensor(divide_t(toTensor(x), toTensor(y)));
}

torch::Tensor dotmul_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
    torch::Tensor z_t = x_t * y_t;
    return z_t;
}

extern "C" void *dotmul(void *a, void *b)
{
    tensor *x = (tensor *)a;
    tensor *y = (tensor *)b;

    check(x->type == y->type, "Not consistent type\n");
    if (y->ndim != 0) {
        check(x->ndim == y->ndim, "Not consistent dimension\n");
        bool flag = true;
        for (int i = 0; i < x->ndim; i++) {
            if (x->dims[i] != y->dims[i]) {
                flag = false;
                break;
            }
        }
        check(flag == true, "Not consistent dimension\n");
    }

    return (void *)fromTensor(dotmul_t(toTensor(x), toTensor(y)));
}


torch::Tensor dotpow_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
    int64_t exp = y_t.item().to<int64_t>();
    torch::Tensor z_t = x_t;
    for (int i = 1; i < exp; i++)
        z_t = z_t * x_t;
    return z_t;
}
```

```
extern "C" void *dotpow(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   check(y->type == 0, "Second tensor should have int type\n");
   check(y->ndim == 0, "Second tensor should be 0-dim\n");

   return (void *)fromTensor(dotpow_t(toTensor(x), toTensor(y)));
}

torch::Tensor equal_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   torch::Tensor z_t = x_t == y_t;
   return z_t.toType(torch::kInt32);
}

extern "C" void *equal(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   //check(x->type == y->type, "Not consistent type");
   if (x->type != y->type) {
       tensor *ret = (tensor *)malloc(sizeof(tensor));
       ret->type = 0;
       ret->ndim = 0;
       ret->dims = NULL;
       ret->data = malloc(sizeof(int));
       *(int *)(ret->data) = 0;
       return (void *)ret;
   }
   if (x->type == 21) {
       tensor *ret = (tensor *)malloc(sizeof(tensor));
       ret->type = 0;
       ret->ndim = 0;
       ret->dims = NULL;
       ret->data = malloc(sizeof(int));
       *(int *)(ret->data) = 1;
       return (void *)ret;
   }
```

```
   return (void *)fromTensor(equal_t(toTensor(x), toTensor(y)));
}


torch::Tensor floordivide_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   std::string rounding_mode = "floor";
   torch::Tensor z_t = torch::div(x_t, y_t.item(), rounding_mode);
   return z_t;
}

extern "C" void *floordivide(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   check(x->type == y->type, "Not consistent type\n");
   check(y->ndim == 0, "Second tensor should be 0-dim tensor\n");

   return (void *)fromTensor(floordivide_t(toTensor(x), toTensor(y)));
}

torch::Tensor greater_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   torch::Tensor z_t = x_t > y_t;
   return z_t.toType(torch::kInt32);
}

extern "C" void *greater(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   check(x->type == y->type, "Not consistent type\n");
   check(x->ndim == y->ndim, "Not consistent dimension\n");
   bool flag = true;
   for (int i = 0; i < x->ndim; i++) {
       if (x->dims[i] != y->dims[i]) {
           flag = false;
           break;
       }
```

```
    }
    check(flag == true, "Not consistent dimension\n");

    return (void *)fromTensor(greater_t(toTensor(x), toTensor(y)));
}

torch::Tensor greaterequal_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
    torch::Tensor z_t = x_t >= y_t;
    return z_t.toType(torch::kInt32);
}

extern "C" void *greaterequal(void *a, void *b)
{
    tensor *x = (tensor *)a;
    tensor *y = (tensor *)b;

    check(x->type == y->type, "Not consistent type\n");
    check(x->ndim == y->ndim, "Not consistent dimension\n");
    bool flag = true;
    for (int i = 0; i < x->ndim; i++) {
        if (x->dims[i] != y->dims[i]) {
            flag = false;
            break;
        }
    }
    check(flag == true, "Not consistent dimension\n");

    return (void *)fromTensor(greaterequal_t(toTensor(x), toTensor(y)));
}

torch::Tensor less_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
    torch::Tensor z_t = x_t < y_t;
    return z_t.toType(torch::kInt32);
}

extern "C" void *less(void *a, void *b)
{
    tensor *x = (tensor *)a;
    tensor *y = (tensor *)b;
```

```
    check(x->type == y->type, "Not consistent type\n");
    check(x->ndim == y->ndim, "Not consistent dimension\n");
    bool flag = true;
    for (int i = 0; i < x->ndim; i++) {
        if (x->dims[i] != y->dims[i]) {
            flag = false;
            break;
        }
    }
    check(flag == true, "Not consistent dimension\n");

    return (void *)fromTensor(less_t(toTensor(x), toTensor(y)));
}

torch::Tensor lessequal_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
    torch::Tensor z_t = x_t <= y_t;
    return z_t.toType(torch::kInt32);
}

extern "C" void *lessequal(void *a, void *b)
{
    tensor *x = (tensor *)a;
    tensor *y = (tensor *)b;

    check(x->type == y->type, "Not consistent type\n");
    check(x->ndim == y->ndim, "Not consistent dimension\n");
    bool flag = true;
    for (int i = 0; i < x->ndim; i++) {
        if (x->dims[i] != y->dims[i]) {
            flag = false;
            break;
        }
    }
    check(flag == true, "Not consistent dimension\n");

    return (void *)fromTensor(lessequal_t(toTensor(x), toTensor(y)));
}

torch::Tensor logicaland_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
    torch::Tensor z_t = torch::logical_and(x_t,y_t);
```

```
    return z_t.toType(torch::kInt32);
}

extern "C" void *logicaland(void *a, void *b)
{
    tensor *x = (tensor *)a;
    tensor *y = (tensor *)b;

    check(x->type == y->type, "Not consistent type\n");
    check(x->ndim == y->ndim, "Not consistent dimension\n");
    bool flag = true;
    for (int i = 0; i < x->ndim; i++) {
        if (x->dims[i] != y->dims[i]) {
            flag = false;
            break;
        }
    }
    check(flag == true, "Not consistent dimension\n");

    return (void *)fromTensor(logicaland_t(toTensor(x), toTensor(y)));
}

torch::Tensor logicalnot_t(const torch::Tensor &x_t)
{
    torch::Tensor z_t = torch::logical_not(x_t);
    return z_t.toType(torch::kInt32);
}

extern "C" void *logicalnot(void *a)
{
    tensor *x = (tensor *)a;

    return (void *)fromTensor(logicalnot_t(toTensor(x)));
}

torch::Tensor logicalor_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
    torch::Tensor z_t = torch::logical_or(x_t,y_t);
    return z_t.toType(torch::kInt32);
}

extern "C" void *logicalor(void *a, void *b)
```

```
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   check(x->type == y->type, "Not consistent type\n");
   check(x->ndim == y->ndim, "Not consistent dimension\n");
   bool flag = true;
   for (int i = 0; i < x->ndim; i++) {
       if (x->dims[i] != y->dims[i]) {
           flag = false;
           break;
       }
   }
   check(flag == true, "Not consistent dimension\n");

   return (void *)fromTensor(logicalor_t(toTensor(x), toTensor(y)));
}

torch::Tensor matpow_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   torch::Tensor z_t = torch::linalg::matrix_power(x_t, y_t.item().to<int64_t>());
   return z_t;
}

extern "C" void *matpow(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   check(y->type == 0, "Second tensor should have int type\n");
   check(y->ndim == 0, "Second tensor should be 0-dim tensor\n");

   return (void *)fromTensor(matpow_t(toTensor(x), toTensor(y)));
}

torch::Tensor mod_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   torch::Tensor z_t = torch::fmod(x_t, y_t.item());
   return z_t;
}

extern "C" void *mod(void *a, void *b)
```

```
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   check(x->type == y->type, "Second tensor should have int type\n");
   check(y->ndim == 0, "Second tensor should be 0-dim tensor\n");

   return (void *)fromTensor(mod_t(toTensor(x), toTensor(y)));
}

torch::Tensor mult_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   torch::Tensor z_t = torch::matmul(x_t, y_t);
   return z_t;
}

extern "C" void *mult(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   check(x->type == y->type, "Not consistent type\n");
   check(x->ndim <= 2, "Matrix multiplication for dim >= 3 not allowed\n");
   check(y->ndim <= 2, "Matrix multiplication for dim >= 3 not allowedn\n");
   check(x->dims[x->ndim-1] == y->dims[0], "Not consistent dimension for matrix
multiplication\n");

   return (void *)fromTensor(mult_t(toTensor(x), toTensor(y)));
}

torch::Tensor notequal_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   torch::Tensor z_t = x_t != y_t;
   return z_t.toType(torch::kInt32);
}

extern "C" void *notequal(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   //check(x->type == y->type, "Not consistent type\n");
```

```
    if (x->type != y->type) {
        tensor *ret = (tensor *)malloc(sizeof(tensor));
        ret->type = 0;
        ret->ndim = 0;
        ret->dims = NULL;
        ret->data = malloc(sizeof(int));
        *(int *)(ret->data) = 1;
        return (void *)ret;
    }
    if (x->type == 21) {
        tensor *ret = (tensor *)malloc(sizeof(tensor));
        ret->type = 0;
        ret->ndim = 0;
        ret->dims = NULL;
        ret->data = malloc(sizeof(int));
        *(int *)(ret->data) = 0;
        return (void *)ret;
    }
    check(x->ndim == y->ndim, "Not consistent dimension\n");
    bool flag = true;
    for (int i = 0; i < x->ndim; i++) {
        if (x->dims[i] != y->dims[i]) {
            flag = false;
            break;
        }
    }
    check(flag == true, "Not consistent dimension\n");

    return (void *)fromTensor(notequal_t(toTensor(x), toTensor(y)));
}

void print_var(tensor *x)
{
    if (x->type != 3) {
        if (x->type == 21)
            printf("Nil\n");
        else if (x->type == 2)
            printf("%s\n", (char *)x->data);
        else
            std::cout << toTensor(x) << std::endl;
        return;
    }
```

```
    int8_t dim = x->dims[0], i;
    tensor **data = (tensor **)x->data;
    printf("[");
    for (i = 0; i < dim; i++) {
        print_var(data[i]);
        if (i != dim-1)
            printf(",");
    }
    printf("]");
}

extern "C" void print(void *a)
{
    tensor *x = (tensor *)a;
    if (x->type == 3) {
        print_var(x);
        printf("\n");
    } else if (x->type == 2)
        printf("%s\n", (char *)x->data);
    else if (x->type == 21)
        printf("Nil\n");
    else
        std::cout << toTensor(x) << std::endl;
}

extern "C" void print_int(int a)
{
    printf("%d\n", a);
}

extern "C" void *range(void *a, void *b, void *c)
{
    tensor *x = (tensor *)a;
    tensor *y = (tensor *)b;
    tensor *z = (tensor *)c;

    check(x->type == 0 && y->type == 0 && z->type == 0, "Slice must be integer");
    check(x->ndim == 0 && y->ndim == 0 && z->ndim == 0, "Slice must be integer");

    int indx = *(int *)x->data;
    int indy = *(int *)y->data;
    int indz = *(int *)z->data;
```

```
    tensor *n = fromTensor(torch::arange(indx, indy, indz).to(torch::kInt32));
    int len = n->dims[0];
    free(n->dims);
    n->dims = (int64_t *)malloc(sizeof(int64_t) * (n->ndim + 3));
    n->dims[0] = len;
    n->dims[1] = indx;
    n->dims[2] = indy;
    n->dims[3] = indz;
    n->type = 4;
    return (void *)n;
}

torch::Tensor shape_t(const torch::Tensor &x_t)
{
    c10::IntArrayRef a = x_t.sizes();
    int64_t size = a.size();
    int64_t dim_tmp[size];
    for (int i = 0; i < size; i++){
        dim_tmp[i] = (int64_t)(a[i]);
    }
    c10::IntArrayRef a_dim (size);
    auto options = torch::TensorOptions().dtype(torch::kInt64);
    torch::Tensor z_t = torch::from_blob(dim_tmp, a_dim, options);
    return z_t.toType(torch::kInt32).clone();
}

extern "C" void *shape(void *a)
{
    tensor *x = (tensor *)a;
    if (x->type == 3) {
        tensor *ret = (tensor *)malloc(sizeof(tensor));
        ret->type = 0;
        ret->ndim = 0;
        ret->dims = NULL;
        ret->data = malloc(sizeof(int));
        *(int *)(ret->data) = (int)(x->dims[0]);
        return (void *)ret;
    }

    return (void *)fromTensor(shape_t(toTensor(x)));
}
```

```
torch::Tensor subtract_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   torch::Tensor z_t = x_t - y_t;
   return z_t;
}

torch::Tensor negative_t(const torch::Tensor &x_t)
{
   return torch::neg(x_t);
}

extern "C" void *subtract(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   check(x->type == y->type, "Not consistent type\n");
   if (y->ndim != 0) {
       check(x->ndim == y->ndim, "Not consistent dimension\n");
       bool flag = true;
       for (int i = 0; i < x->ndim; i++) {
           if (x->dims[i] != y->dims[i]) {
               flag = false;
               break;
           }
       }
       check(flag == true, "Not consistent dimension\n");
   }


   return (void *)fromTensor(subtract_t(toTensor(x), toTensor(y)));
}

extern "C" void *negative(void *a)
{
   tensor *x = (tensor *)a;

   return (void *)fromTensor(negative_t(toTensor(x)));
}
```

```
torch::Tensor transpose_t(const torch::Tensor &x_t)
{
   torch::Tensor z_t = x_t.transpose(-1, 0).contiguous();
   return z_t;
}

extern "C" void *transpose(void *a)
{
   tensor *x = (tensor *)a;

   return (void *)fromTensor(transpose_t(toTensor(x)));
}

torch::Tensor zeros_t(const torch::Tensor &x_t)
{
   int64_t size = x_t.sizes()[0];
   std::vector<int64_t> dims;
   for (int i = 0; i < size; i++)
       dims.push_back(x_t[i].item().to<int64_t>());
   at::IntArrayRef ndims (dims);
   torch::Tensor z_t = torch::zeros(ndims);
   return z_t.toType(torch::kInt32);
}

extern "C" void *zeros(void *a)
{
   tensor *x = (tensor *)a;
   check(x->type == 0, "Not consistent type");
   check(x->ndim == 1 , "Dimension should be 1");
   return (void *)fromTensor(zeros_t(toTensor(x)));
}

torch::Tensor sum_t(const torch::Tensor &x_t)
{
   torch::Tensor z_t = torch::sum(x_t);
   return z_t;
}

extern "C" void *sum(void *a)
{
   tensor *x = (tensor *)a;
```

```
    if (x->type == 0)
        return (void *)fromTensor(sum_t(toTensor(x)).toType(torch::kInt32));
    else
        return (void *)fromTensor(sum_t(toTensor(x)).toType(torch::kFloat64));
}

torch::Tensor ones_t(const torch::Tensor &x_t)
{
    int64_t size = x_t.sizes()[0];
    std::vector<int64_t> dims;
    for (int i = 0; i < size; i++)
        dims.push_back(x_t[i].item().to<int64_t>());
    at::IntArrayRef ndims (dims);
    torch::Tensor z_t = torch::ones(ndims);
    return z_t.toType(torch::kInt32);
}

extern "C" void *ones(void *a)
{
    tensor *x = (tensor *)a;
    check(x->type == 0, "Not consistent type");
    check(x->ndim == 1 , "Dimension should be 1");
    return (void *)fromTensor(ones_t(toTensor(x)));
}

torch::Tensor tensor_floor_t(const torch::Tensor &x_t)
{
    return torch::floor(x_t.toType(torch::kFloat64));
}


extern "C" void *tensor_floor(void *a)
{
    tensor *x = (tensor *)a;

    return (void *)fromTensor(tensor_floor_t(toTensor(x)));
}

torch::Tensor tensor_ceil_t(const torch::Tensor &x_t)
{
    return torch::ceil(x_t.toType(torch::kFloat64));
}
```

```
extern "C" void *tensor_ceil(void *a)
{
   tensor *x = (tensor *)a;

   return (void *)fromTensor(tensor_ceil_t(toTensor(x)));
}

torch::Tensor tensor_round_t(const torch::Tensor &x_t)
{
   return torch::round(x_t.toType(torch::kFloat64));
}

extern "C" void *tensor_round(void *a)
{
   tensor *x = (tensor *)a;
   if (x->type == 0)
       return a;
   return (void *)fromTensor(tensor_round_t(toTensor(x)));
}

torch::Tensor tensor_abs_t(const torch::Tensor &x_t)
{
   return torch::abs(x_t);
}

extern "C" void *tensor_abs(void *a)
{
   tensor *x = (tensor *)a;

   return (void *)fromTensor(tensor_abs_t(toTensor(x)));
}

torch::Tensor tensor_log_t(const torch::Tensor &x_t)
{
   return torch::log(x_t.toType(torch::kFloat64));
}

extern "C" void *tensor_log(void *a)
{
   tensor *x = (tensor *)a;
```

```
   return (void *)fromTensor(tensor_log_t(toTensor(x)));
}

torch::Tensor inverse_t(const torch::Tensor &x_t)
{
   return torch::inverse(x_t.toType(torch::kFloat64));
}

extern "C" void *inverse(void *a)
{
   tensor *x = (tensor *)a;
   check(x->ndim == 2, "Only 2-d square matrix allowed for inverse operation\n");
   check(x->dims[0] == x->dims[1], "Only 2-d square matrix allowed for inverse
operation\n");

   return (void *)fromTensor(inverse_t(toTensor(x)));
}

torch::Tensor int_of_t(const torch::Tensor &x_t)
{
   return x_t.toType(torch::kInt32);
}

extern "C" void *int_of(void *a)
{
   tensor *x = (tensor *)a;

   return (void *)fromTensor(int_of_t(toTensor(x)));
}

torch::Tensor float_of_t(const torch::Tensor &x_t)
{
   return x_t.toType(torch::kFloat64);
}

extern "C" void *float_of(void *a)
{
   tensor *x = (tensor *)a;

   return (void *)fromTensor(float_of_t(toTensor(x)));
}
```

```
torch::Tensor solve_t(const torch::Tensor &x_t, const torch::Tensor &y_t)
{
   auto z_t = torch::solve(x_t,y_t);
   return std::get<0>(z_t);  // tbm
}

extern "C" void *solve(void *a, void *b)
{
   tensor *x = (tensor *)a;
   tensor *y = (tensor *)b;

   check(x->type == y->type, "Not consistent type\n");

   return (void *)fromTensor(subtract_t(toTensor(x), toTensor(y)));
}

torch::Tensor tensor_rand_t(const torch::Tensor &x_t)
{
   int64_t size = x_t.sizes()[0];
   std::vector<int64_t> dims;
   for (int i = 0; i < size; i++)
       dims.push_back(x_t[i].item().to<int64_t>());
   at::IntArrayRef ndims (dims);
   torch::Tensor z_t = torch::rand(ndims);
   return z_t.toType(torch::kFloat64);
}

extern "C" void *tensor_rand(void *a)
{
   tensor *x = (tensor *)a;
   check(x->type == 0, "Not consistent type");
   check(x->ndim == 1 , "Dimension should be 1");
   return (void *)fromTensor(tensor_rand_t(toTensor(x)));
}
```

lib/gc.cc

```
#include "tensor.h"
#include <stdlib.h>
```

```
void free_tensor(tensor *a)
{
   tensor *x = a;
   if (x -> rc < 0) {
       printf("bug comming :)!\n");
   } else if (x -> rc > 0) {
       return;
   }

   free(x);
}

extern "C" void increase_rc(void *a)
{
   tensor *x = (tensor *)a;
   if (x -> type != 3) {
       x -> rc += 1;
       return;
   }

   x -> rc += 1;
   int8_t dim = x->dims[0], i;
   tensor **data = (tensor **)x->data;
   for (i = 0; i < dim; i++) {
       increase_rc(data[i]);
   }
}

extern "C" void decrease_rc(void *a)
{
   if (a == NULL) return;
   tensor *x = (tensor *)a;
   if (x -> type != 3) {
       x -> rc -= 1;
       if (x -> rc == 0) {
           free_tensor(x);
       }
       return;
   }

   x -> rc -= 1;
   int8_t dim = x->dims[0], i;
```

```
    tensor **data = (tensor **)x->data;
    for (i = 0; i < dim; i++) {
        decrease_rc(data[i]);
    }

    if (x -> rc == 0) {
        free_tensor(x);
    }
}
```

lib/index.cc

```
#include "tensor.h"

torch::Tensor index_get_t(const torch::Tensor &x_t,
    const c10::ArrayRef<torch::indexing::TensorIndex> &y_t)
{
    return x_t.index(y_t);
}

tensor *index_get_v(tensor *x, tensor *y)
{
    int8_t dim = y->dims[0];

    check(dim == 1, "Invalid index of vartensor");

    tensor *index = ((tensor **)y->data)[0];
    tensor **data = (tensor **)x->data;
    check(index->type == 0 || index->type == 4, "Index must be integer");
    int32_t *ind_data = (int32_t *)index->data;

    dim = index->ndim;
    check(dim == 1 || dim == 0, "Invalid index of vartensor");

    if (dim == 0) {
        check(ind_data[0] < x->dims[0], "Out of range");
        return data[ind_data[0]];
    }

    int8_t numInd = index->dims[0];
    tensor **ret_data = (tensor **)malloc(sizeof(tensor *) * numInd);
```

```
    dim = x->dims[0];
    for (int i = 0; i < numInd; i++) {
        check(ind_data[i] < dim, "Out of range");
        ret_data[i] = data[ind_data[i]];
    }

    tensor *ret_ten = (tensor *)malloc(sizeof(tensor));
    ret_ten->type = 3;
    ret_ten->ndim = 1;
    ret_ten->dims = (int64_t *)malloc(sizeof(int64_t));
    ret_ten->dims[0] = numInd;
    ret_ten->data = (void *)ret_data;
    return ret_ten;
}

extern "C" void *index_get(void *tena, void *inda)
{
    tensor *tenx = (tensor *)tena;
    tensor *indx = (tensor *)inda;
    int8_t dim = indx->dims[0];

    if (tenx->type == 3)
        return (void *)index_get_v(tenx, indx);

    std::vector<torch::indexing::TensorIndex> tmplist;

    tensor **indices = (tensor **)indx->data;
    for (int i = 0; i < dim; i++) {
        if (indices[i]->type == 4) {
            tmplist.push_back(torch::indexing::Slice
                (indices[i]->dims[1], indices[i]->dims[2], indices[i]->dims[3]));
            continue;
        }
        check(indices[i]->type == 0, "Index must be integer");
        tmplist.push_back(toTensor(indices[i]).to(torch::kInt64));
    }
    c10::ArrayRef<torch::indexing::TensorIndex> indlist(tmplist);

    return (void *)fromTensor(index_get_t(toTensor(tenx), indlist));
}

extern "C" void *index_get_int(void *tena, int inda)
```

```
{
   tensor *indx = (tensor *)malloc(sizeof(tensor));
   indx->type = 0;
   indx->ndim = 0;
   indx->dims = NULL;
   indx->data = (void *) malloc(sizeof(int));
   memcpy(indx->data, &inda, sizeof(int));

   tensor *indx2 = (tensor *)malloc(sizeof(tensor));
   tensor **tmp = (tensor **)malloc(sizeof(tensor *));
   tmp[0] = indx;
   indx2->type = 3;
   indx2->ndim = 1;
   indx2->dims = (int64_t *)malloc(sizeof(int64_t));
   indx2->dims[0] = 1;

   indx2->data = (void *)tmp;

   return index_get(tena, (void *)indx2);
}

void index_put_t(torch::Tensor x_t,
   const c10::ArrayRef<torch::indexing::TensorIndex> &y_t,
   torch::Tensor z_t)
{
   x_t.index_put_(y_t, z_t);
}

void index_put_v(tensor *x, tensor *y, tensor *z)
{
   int8_t dim = y->dims[0];

   check(dim == 1, "Invalid index of vartensor");

   tensor *index = ((tensor **)y->data)[0];
   tensor **data = (tensor **)x->data;
   check(index->type == 0, "Index must be integer");
   int32_t *ind_data = (int32_t *)index->data;

   dim = index->ndim;
   check(dim == 1 || dim == 0, "Invalid index of vartensor");
```

```
    if (dim == 0) {
        check(ind_data[0] < x->dims[0], "Out of range");
        data[ind_data[0]] = z;
        return;
    }

    check(z->type == 3, "Invalid input");
    int8_t numInd = index->dims[0];
    check(numInd == z->dims[0], "Inconsistent input size");
    tensor **newdata = (tensor **)z->data;

    dim = x->dims[0];
    for (int i = 0; i < numInd; i++) {
        check(ind_data[i] < dim, "Out of range");
        data[ind_data[i]] = newdata[i];
    }
}

extern "C" void index_put(void *tena, void *inda, void *ntena)
{
    tensor *tenx = (tensor *)tena;
    tensor *indx = (tensor *)inda;
    tensor *ntenx = (tensor *)ntena;
    tensor *teny;
    int8_t dim = indx->dims[0];

    if (tenx->type == 3)
        return index_put_v(tenx, indx, ntenx);

    std::vector<torch::indexing::TensorIndex> tmplist;

    tensor **indices = (tensor **)indx->data;
    for (int i = 0; i < dim; i++) {
        if (indices[i]->type == 4) {
            tmplist.push_back(torch::indexing::Slice
                (indices[i]->dims[1], indices[i]->dims[2], indices[i]->dims[3]));
            continue;
        }
        check(indices[i]->type == 0, "Index must be integer");
        tmplist.push_back(toTensor(indices[i]).to(torch::kInt64));
    }
    c10::ArrayRef<torch::indexing::TensorIndex> indlist(tmplist);
```

```
    index_put_t(toTensor(tenx), indlist, toTensor(ntenx));
}
```

lib/pe.cc

```
#include "tensor.h"
#include <thread>
#include <vector>
#include <future>
#include <iostream>

using namespace std;

typedef void* (*pf)(void*, void*);
typedef void* (*rd)(void**);

extern "C" void *pe_calc(pf* mapfunctions, int num, rd reduce, void* a, void* b)
{
    int i;
    future<void*> pres[num];
    void *res[num];

    for (i = 0;i < num;i++) {
        pres[i] = async(((*mapfunctions[i])), a, b);
    }

    for (i = 0;i < num;i++) {
        res[i] = pres[i].get();
    }

    return (*reduce)(res);
}
```

lib/tensor.cc

```
#include "tensor.h"

torch::Dtype toType(const tensor * const a)
{
```

```
    switch (a->type) {
        case 0: return torch::kInt32;
        case 1: return torch::kFloat64;
        case 4: return torch::kInt32;

        default: check(0, "Invalid type");
    }
}

int8_t fromType(const torch::Dtype &a_type)
{
    switch (a_type) {
        case torch::kInt32: return 0;
        case torch::kFloat64: return 1;

        default: check(0, "Invalid type");
    }
}

int8_t typeSize(int8_t type)
{
    switch (type) {
        case 0: return sizeof(int);
        case 1: return sizeof(double);
        case 4: return sizeof(int);

        default: check(0, "Invalid type");
    }
}

torch::Tensor toTensor(const tensor * const a)
{
    int64_t *dim_tmp;

    dim_tmp = new int64_t[a->ndim];
    for (int i = 0; i < a->ndim; i++)
        dim_tmp[i] = a->dims[i];
    c10::IntArrayRef a_dim (dim_tmp, dim_tmp+a->ndim);

    torch::Tensor a_t = torch::from_blob(a->data, a_dim, toType(a));

    delete [] dim_tmp;
```

```
    return a_t;
}

tensor *fromTensor(const torch::Tensor &a_t)
{
    tensor *a = (tensor *)malloc(sizeof(tensor));

    a->type = fromType(a_t.scalar_type());
    a->ndim = (int8_t)a_t.dim();

    a->dims = (int64_t *)malloc(sizeof(int64_t)*a->ndim);
    for (int i = 0; i < a->ndim; i++)
        a->dims[i] = a_t.size(i);

    unsigned int eleBytes = typeSize(a->type) * torch::numel(a_t);
    a->data = malloc(eleBytes);
    memcpy(a->data, a_t.data_ptr(), eleBytes);
    a->rc = 1; // todo: 0 or 1?

    return a;
}

extern "C" int len(void *a)
{
    tensor *x = (tensor *)a;
    if (x->type != 3)
        return toTensor(x).size(0);
    else
        return x->dims[0];
}
```

lib/tensor.h

```
#ifndef __TENSOR_H
#define __TENSOR_H

#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
```

```
#include <errno.h>

#include <iostream>
#include <torch/torch.h>

#define check(cond, ...) \
do {\
   if (cond)\
       break;\
   fprintf(stderr, __VA_ARGS__);\
   if (errno)\
       perror(" ");\
   exit(1);\
} while(0)

typedef struct tensor
{
   int8_t type;
   int8_t ndim;
   int64_t *dims;
   void *data;
   int8_t rc;
} tensor;

torch::Dtype toType(const tensor * const a);
int8_t fromType(const torch::Dtype &a_type);
torch::Tensor toTensor(const tensor * const a);
tensor *fromTensor(const torch::Tensor &a_t);

#endif /* __TENSOR_H */
```

Makefile

```
.PHONY: test
test: all generate.sh test.sh
   ./generate.sh ./tests/*-test/test-*.tl
   ./test.sh ./tests/*-test/test-*.tl
   ./generate.sh ./tests/fails/fail-*.tl

.PHONY: all
all: tenlab.native
```

```
tenlab.native:
   opam config exec -- \
   ocamlbuild -I src -use-ocamlfind tenlab.native

.PHONY: clean
clean: cleandir
   rm -rf *.ll *.out *.s *.diff *.exe *.err
   rm -rf tenlab.native
   rm -rf _build
   rm -rf pe.o

cleandir :
   @if [ -d build ]; then make -C build clean; \
   else echo "build not exist"; fi
```

Generate.sh

```
# Path to the LLVM interpreter
LLI="lli"

# Path to the LLVM compiler
LLC="llc"

# Path to the C compiler
CC="cc"

# Path to the tenlab compiler.
TENLAB="./tenlab.native"

globallog=testall.log
rm -f $globallog

# Run <args>
# Report the command, run it, and report any errors
Run() {
   echo $* 1>&2
   eval $* || {
   SignalError "$1 failed on $*"
   return 1
```

```
    }
}

Compare() {
    generatedfiles="$generatedfiles $3"
    echo diff -b $1 $2 ">" $3 1>&2
    diff -b "$1" "$2" > "$3" 2>&1 || {
    SignalError "$1 differs"
    echo "FAILED $1 differs from $2" 1>&2
    }
}

Generate() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
                              s/.tl//'`
    reffile=`echo $1 | sed 's/.tl$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Generating $basename.s" 1>&2

    Run "$TENLAB" "./$1" ">" "${basename}.ll" &&
    Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">" "${basename}.s"
}

CheckFail() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
                              s/.tl//'`
    reffile=`echo $1 | sed 's/.tl$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""
```

```
    generatedfiles="$generatedfiles ${basename}.err ${basename}.diff" &&
    RunFail "$TENLAB" "<" $1 "2>" "${basename}.err" ">>" $globallog &&
    Compare ${basename}.err ${reffile}.err ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
        rm -f $generatedfiles
    echo "OK"
    echo "###### SUCCESS" 1>&2
    else
    echo "###### FAILED" 1>&2
    globalerror=$error
    fi
}

RunFail() {
    echo $* 1>&2
    eval $* && {
    SignalError "failed: $* did not report an error"
    return 1
    }
    return 0
}

if [ $# -ge 1 ]
then
    files=$@
else
    files="tests/test-*.tl tests/fail-*.tl"
fi

for file in $files
do
    case $file in
    *test-*)
        Generate $file 2
        ;;
    *fail-*)
        CheckFail $file 2 >> $globallog
        ;;
    *)
```

```
        echo "unknown file type $file"
        globalerror=1
        ;;
    esac
done
```

Test.sh

```
#!/bin/sh

# Regression testing script for TENLab
# Step through a list of files
#  Compile, run, and check the output of each expected-to-work test
#  Compile and check the error of each expected-to-fail test

# Path to the LLVM interpreter
LLI="lli"

# Path to the LLVM compiler
LLC="llc"

# Path to the C compiler
CC="cc"

# Path to the tenlab compiler.
TENLAB="./tenlab.native"

# Set time limit for all operations
ulimit -t 30

globallog=testall.log
rm -f $globallog
error=0
globalerror=0

keep=0

Usage() {
    echo "Usage: testall.sh [options] [.tl files]"
    echo "-k    Keep intermediate files"
```

```
    echo "-h    Print this help"
    exit 1
}

SignalError() {
    if [ $error -eq 0 ] ; then
    echo "FAILED"
    error=1
    fi
    echo "  $1"
}

# Compare <outfile> <reffile> <difffile>
# Compares the outfile with reffile.  Differences, if any, written to difffile
Compare() {
    generatedfiles="$generatedfiles $3"
    echo diff -b $1 $2 ">" $3 1>&2
    diff -b "$1" "$2" > "$3" 2>&1 || {
    SignalError "$1 differs"
    echo "FAILED $1 differs from $2" 1>&2
    }
}

# Run <args>
# Report the command, run it, and report any errors
Run() {
    echo $* 1>&2
    eval $* || {
    SignalError "$1 failed on $*"
    return 1
    }
}

# RunFail <args>
# Report the command, run it, and expect an error
RunFail() {
    echo $* 1>&2
    eval $* && {
    SignalError "failed: $* did not report an error"
    return 1
    }
    return 0
```

```
}

Check() {
   error=0
   basename=`echo $1 | sed 's/.*\\///
                           s/.tl//'`
   reffile=`echo $1 | sed 's/.tl$//'`
   basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

   echo -n "$basename..."

   echo 1>&2
   echo "###### Testing $basename" 1>&2

   generatedfiles=""

   if [ -d build ]; then echo "build exist"; else mkdir build; fi

   generatedfiles="$generatedfiles ${basename}.ll ${basename}.s ${basename}.exe
${basename}.out" &&
   # Run "$TENLAB" "$1" ">" "${basename}.ll" &&
   # Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">" "${basename}.s" &&
   # Run "$CC" "-o" "${basename}.exe" "${basename}.s" "./lib/add.o" "./lib/print.o"
"./lib/mult.o" &&
   Run "cd build" &&
   Run "cmake .." "-DSOURCE_FILE:FILEPATH=${basename}" &&
   Run "make" &&
   Run "cd .." &&
   Run "./${basename}.exe" > "${basename}.out" &&
   Compare ${basename}.out ${reffile}.out ${basename}.diff

   # Report the status and clean up the generated files

   if [ $error -eq 0 ] ; then
   if [ $keep -eq 0 ] ; then
       rm -f $generatedfiles
   fi
   echo "OK"
   echo "###### SUCCESS" 1>&2
   else
   echo "###### FAILED" 1>&2
   globalerror=$error
```

```
    fi
}

CheckFail() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
                              s/.tl//'`
    reffile=`echo $1 | sed 's/.tl$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.err ${basename}.diff" &&
    RunFail "$TENLAB" "<" $1 "2>" "${basename}.err" ">>" $globallog &&
    Compare ${basename}.err ${reffile}.err ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
    if [ $keep -eq 0 ] ; then
        rm -f $generatedfiles
    fi
    echo "OK"
    echo "###### SUCCESS" 1>&2
    else
    echo "###### FAILED" 1>&2
    globalerror=$error
    fi
}

while getopts kdpsh c; do
    case $c in
    k) # Keep intermediate files
        keep=1
        ;;
    h) # Help
        Usage
```

```
       ;;
   esac
done

shift `expr $OPTIND - 1`

LLIFail() {
 echo "Could not find the LLVM interpreter \"$LLI\"."
 echo "Check your LLVM installation and/or modify the LLI variable in testall.sh"
 exit 1
}

which "$LLI" >> $globallog || LLIFail

if [ $# -ge 1 ]
then
   files=$@
else
   files="tests/test-*.tl tests/fail-*.tl"
fi

for file in $files
do
   case $file in
   *test-*)
       Check $file 2>> $globallog
       ;;
   *fail-*)
       CheckFail $file 2>> $globallog
       ;;
   *)
       echo "unknown file type $file"
       globalerror=1
       ;;
   esac
done

exit $globalerror
```

Rundemo.sh

```
# Path to the LLVM interpreter
LLI="lli"

# Path to the LLVM compiler
LLC="llc"

# Path to the C compiler
CC="cc"

# Path to the tenlab compiler.
TENLAB="./tenlab.native"

Run() {
   echo $* 1>&2
   eval $* || {
   SignalError "$1 failed on $*"
   return 1
   }
}

Generate() {
   basename=`echo $1 | sed 's/.*\\///
                           s/.tl//'`

   echo -n "$basename..."

   echo 1>&2
   echo "###### Generating $basename.s" 1>&2

   Run "$TENLAB" "./$1" ">" "${basename}.ll" &&
   Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">" "${basename}.s"
}

Test() {
   error=0
   basename=`echo $1 | sed 's/.*\\///
                           s/.tl//'`

   echo -n "$basename..."
   echo "###### Testing $basename" 1>&2
```

```
   generatedfiles=""

   if [ -d build ]; then echo "build exist"; else mkdir build; fi

   generatedfiles="$generatedfiles ${basename}.ll ${basename}.s ${basename}.exe
${basename}.out" &&
   Run "cd build" &&
   Run "cmake .." "-DSOURCE_FILE:FILEPATH=${basename}" &&
   Run "make" &&
   Run "cd .." &&
   Run "./${basename}.exe"

   # Report the status and clean up the generated files

   if [ $error -eq 0 ] ; then
   rm -f $generatedfiles
   echo "OK"
   else
   echo "###### FAILED" 1>&2
   fi
}

TimeTest() {
   error=0
   basename=`echo $1 | sed 's/.*\\///
                           s/.tl//'`

   echo -n "$basename..."
   echo "###### Testing $basename" 1>&2

   generatedfiles=""

   if [ -d build ]; then echo "build exist"; else mkdir build; fi

   generatedfiles="$generatedfiles ${basename}.ll ${basename}.s ${basename}.exe
${basename}.out" &&
   eval "cd build" &&
   eval "cmake .." "-DSOURCE_FILE:FILEPATH=${basename}" &&
   eval "make" &&
   eval "cd .." &&
   Run "time ./${basename}.exe"
```

```
   # Report the status and clean up the generated files

   if [ $error -eq 0 ] ; then
   rm -f $generatedfiles
   echo "OK"
   else
   echo "###### FAILED" 1>&2
   fi
}


if [ $# -eq 1 ]; then
   Generate $1
   Test $1
else
   Generate $1
   TimeTest $1
fi
```

Tests:

```
<test-print-literal.tl>
a = [[1, 1], [1, 1]];
b = [[1, 2], [1, 2]];
c = (a + b) * a;
print(c);
<test-print-literal.out>
 5  5
 5  5
[ CPUIntType{2,2} ]
--------------------
<test-neg.tl>
a = -10;
print(a);
<test-neg.out>
-10
[ CPUIntType{} ]
```

```
-------------------
<test-mult.tl>
a = [[1, 2, 3], [4, 5, 6]];
b = [[1], [1], [1]];
c = a * b;
print(c);
<test-mult.out>
  6
 15
[ CPUIntType{2,1} ]

-------------------
<test-gl.tl>
a = 1;
b = 1;
c = 2;
def foo(a) {
    a = 2;
    print(b);
    b = c;
    print(b);
}

foo(3);
print(a);
print(b);
<test-gl.out>
1
[ CPUIntType{} ]
2
[ CPUIntType{} ]
1
[ CPUIntType{} ]
2
[ CPUIntType{} ]

-------------------
<test-assign.tl>
a = [[1,4,2,5,7,8],[3,2,1,5,7,9]];
a[0:2:1,1:3:1] = [[10,12],[13,14]];
print(a);
<test-assign.out>
```

```
  1  10  12   5   7   8
  3  13  14   5   7   9
[ CPUIntType{2,6} ]

-------------------
<test-using.tl>
parallel_define et {
    overload __+__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x+y;
                  return z;
            }
            reduce {
                  return f1+f2;
            }
      }
}

using et;
print(1+2);

<test-using.out>
6
[ CPUIntType{} ]

-------------------
<test-pesub.tl>
parallel_define et {
    overload __-__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x-y;
                  return z;
            }
            reduce {
                  return f1+f2;
```

```
            }
      }
}

using et;
print(2-1);


<test-pesub.out>
4
[ CPUIntType{} ]
-------------------
<test-pereal.tl>
parallel_define fast_add {
    overload __+__ (x,y) {
            map f1 {
                  l = shape(x);
                  z = x[0:l[0]//2:1]+y[0:l[0]//2:1];
                  return z;
            }
            map f2 {
                  l = shape(x);
                  z = x[l[0]//2:l[0]:1]+y[l[0]//2:l[0]:1];
                  return z;
            }
            reduce {
                  return cat(f1,f2,0);
            }
      }
}


a = zeros([20]);
b = ones([20]);
using fast_add;
c = a+b;
print(c);
```

```
<test-pereal.out>
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
 1
[ CPUIntType{20} ]

--------------------
<test-pemul.tl>
parallel_define et {
    overload __*__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x-y;
                  return z;
            }
            reduce {
                  return f1+f2;
            }
      }
}
```

```
using et;
print(2*1);
<test-pemul.out>
4
[ CPUIntType{} ]
-------------------
<test-peall.tl>
parallel_define et {
    overload __+__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x+y;
                  return z;
            }
            reduce {
                  return f1+f2;
            }
      }
      overload __*__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x-y;
                  return z;
            }
            reduce {
                  return f1+f2;
            }
      }
      overload __-__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x-y;
                  return z;
            }
```

```
            reduce {
                  return f1+f2;
            }
      }
}

using et;
print(1+2);
print(2-1);
print(3*1);
end et;
print(1+2);

<test-peall.out>
6
[ CPUIntType{} ]
4
[ CPUIntType{} ]
6
[ CPUIntType{} ]
3
[ CPUIntType{} ]

--------------------
<test-peadd.tl>
parallel_define et {
    overload __+__ (x,y) {
            map f1 {
                  z = x[0]+y[0];
                  return z;
            }
            map f2 {
                  z = x[1]+y[1];
                  return z;
            }
            reduce {
                  return f1+f2;
            }
      }
}

using et;
print([1,1]+[2,2]);
```

```
<test-peadd.out>
6
[ CPUIntType{} ]
-------------------
<test-pe.tl>
parallel_define et {
    overload __+__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x+y;
                  return z;
            }
            reduce {
                  return f1+f2;
            }
      }
}

using et;
print(1+2);
end et;
print(1+2);

<test-pe.out>
6
[ CPUIntType{} ]
3
[ CPUIntType{} ]
-------------------
<test-end.tl>
parallel_define et {
    overload __+__ (x,y) {
            map f1 {
                  z = x+y;
                  return z;
            }
            map f2 {
                  z = x+y;
                  return z;
            }
```

```
            reduce {
                  return f1+f2;
            }
      }
}

using et;
print(1+2);
end et;
print("--------");
print(1+2);
<test-end.out>
6
[ CPUIntType{} ]
--------
3
[ CPUIntType{} ]

--------------------
<test-built-in-zeros.tl>
a = [2,2,3];
b = zeros(a);
print(b);

<test-built-in-zeros.out>
(1,.,.) =
  0  0  0
  0  0  0

(2,.,.) =
  0  0  0
  0  0  0
[ CPUIntType{2,2,3} ]
--------------------
<test-built-in-sum.tl>
a = [2,2,3];
print(sum(a));
b = [2.0,2.2,3.5];
print(sum(b));
<test-built-in-sum.out>
7
[ CPUIntType{} ]
7.7
```

```
[ CPUDoubleType{} ]
--------------------
<test-built-in-shape.tl>
a = [[2,3,4,5],[5,1,4,3],[2,3,1,3]];
b = shape(a);
print(b);

<test-built-in-shape.out>
 3
 4
[ CPUIntType{2} ]

--------------------
<test-built-in-round.tl>
a = [1,2,4];
print(round(a));
b = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0];
print(round(b));

<test-built-in-round.out>
 1
 2
 4
[ CPUIntType{3} ]
 0
 0
 0
 0
 0
 1
 1
 1
 1
 1
[ CPUDoubleType{10} ]
--------------------
<test-built-in-rand.tl>
a = [2,2,3];
b = rand(a);
print("rand test");
<test-built-in-rand.out>
rand test
--------------------
```

```
<test-built-in-ones.tl>
a = [2,2,3];
b = ones(a);
print(b);

<test-built-in-ones.out>
(1,.,.) =
  1  1  1
  1  1  1

(2,.,.) =
  1  1  1
  1  1  1
[ CPUIntType{2,2,3} ]
-------------------
<test-built-in-log.tl>
a = [[10,11,12],[14,15,16]];
print(log(a));
b = [1.0,3.2,5.6];
print(log(b));
<test-built-in-log.out>
 2.3026  2.3979  2.4849
 2.6391  2.7081  2.7726
[ CPUDoubleType{2,3} ]
 0.0000
 1.1632
 1.7228
[ CPUDoubleType{3} ]

-------------------
<test-built-in-inverse.tl>
a = [[1,2,3],[4,5,6],[7,-8,9]];
print(inverse(a));
<test-built-in-inverse.out>
-0.9687 -0.0625  0.6979
 0.4375  0.1250 -0.2292
 0.0313 -0.0625  0.0312
[ CPUDoubleType{3,3} ]

-------------------
<test-built-in-int-of.tl>
a = [1.0,2.2,3.9];
print(int_of(a));
```

```
<test-built-in-int-of.out>
 1
 2
 3
[ CPUIntType{3} ]
-------------------
<test-built-in-floor.tl>
a = [[10,11,12],[14,15,16]];
print(floor(a));
b = [1.0,3.2,5.6];
print(floor(b));
<test-built-in-floor.out>
 10  11  12
 14  15  16
[ CPUDoubleType{2,3} ]
 1
 3
 5
[ CPUDoubleType{3} ]

-------------------
<test-built-in-float-of.tl>
a = [1,2,3];
c = float_of(a);
print(c);

b = [2.1,3.4,6.7];
print(c * b);
<test-built-in-float-of.out>
 1
 2
 3
[ CPUDoubleType{3} ]
29
[ CPUDoubleType{} ]

-------------------
<test-built-in-ceil.tl>
a = [[10,11,12],[14,15,16]];
print(ceil(a));
b = [1.0,3.2,5.6];
print(ceil(b));
<test-built-in-ceil.out>
```

```
 10  11  12
 14  15  16
[ CPUDoubleType{2,3} ]
 1
 4
 6
[ CPUDoubleType{3} ]

-------------------
<test-built-in-cat.tl>
a = [[1,2,3],[5,6,7]];
b = [[10,11,12],[14,15,16]];
c = cat(a,b,0);
print(c);
c = cat(a,b,1);
print(c);
<test-built-in-cat.out>
  1   2   3
  5   6   7
 10  11  12
 14  15  16
[ CPUIntType{4,3} ]
  1   2   3  10  11  12
  5   6   7  14  15  16
[ CPUIntType{2,6} ]

-------------------
<test-built-in-any.tl>
a = [0,1,2];
if (any(a)){
    print("any-built-in functions works!");
}
b = [0,0,0];
if (any(b)){
    print("any-built-in functions doesn't work");
}
<test-built-in-any.out>
any-built-in functions works!
-------------------
<test-built-in-all.tl>
a = [0,1,2];
if (all(a)){
    print("all-built-in functions doesn't work");
```

```
}
b = [2,1,2];
if (all(b)){
    print("all-built-in functions works!");
}
<test-built-in-all.out>
all-built-in functions works!
-------------------
<test-built-in-abs.tl>
a = [-1,2];
print(abs(a));
b = [-1.0,0.0,-2.4];
print(abs(b));
<test-built-in-abs.out>
 1
 2
[ CPUIntType{2} ]
 1.0000
 0.0000
 2.4000
[ CPUDoubleType{3} ]
-------------------
<test-transpose.tl>
a = [[1,2,3],[4,5,6]];
print(a);
print(a');
<test-transpose.out>
 1  2  3
 4  5  6
[ CPUIntType{2,3} ]
 1  4
 2  5
 3  6
[ CPUIntType{3,2} ]

-------------------
<test-subtract.tl>
a = [[1.1,2.0],[3.5,4.2]];
b = [[0.1,0.3],[2.2,6.2]];
print(a-b);
print(a-0.2);
<test-subtract.out>
 1.0000  1.7000
```

```
 1.3000 -2.0000
[ CPUDoubleType{2,2} ]
 0.9000  1.8000
 3.3000  4.0000
[ CPUDoubleType{2,2} ]

-------------------
<test-print-literal.tl>
a = [[1, 1], [1, 1]];
b = [[1, 2], [1, 2]];
c = (a + b) * a;
print(c);
<test-print-literal.out>
 5  5
 5  5
[ CPUIntType{2,2} ]
-------------------
<test-notequal.tl>
a = [[1,2,3],[4,5,6]];
b = [[1,3,5],[3,5,6]];
print(a != b);
<test-notequal.out>
 0  1  1
 1  0  0
[ CPUIntType{2,3} ]

-------------------
<test-neg.tl>
a = -10;
print(a);
b = -[-1.2, 2.3, -0.5];
print(b);
<test-neg.out>
-10
[ CPUIntType{} ]
 1.2000
-2.3000
 0.5000
[ CPUDoubleType{3} ]

-------------------
<test-mult.tl>
a = [[1, 2, 3], [4, 5, 6]];
```

```
b = [[1], [1], [1]];
c = a * b;
print(c);
<test-mult.out>
  6
 15
[ CPUIntType{2,1} ]

-------------------
<test-mod.tl>
a = [[1.1,2.0],[3.5,4.2]];
print(a % 1.6);
<test-mod.out>
 1.1000  0.4000
 0.3000  1.0000
[ CPUDoubleType{2,2} ]

-------------------
<test-minus.tl>
a = [[1, 2, 3], [4, 5, 6]];
b = [[-1, -2, -3], [-4, -5, -6]];
c = a - b;
print(c);
<test-minus.out>
  2   4   6
  8  10  12
[ CPUIntType{2,3} ]

-------------------
<test-matpow.tl>
a = [[1.1,2.0],[3.5,4.2]];
print(a^0);
print(a^1);
print(a^2);
<test-matpow.out>
 1  0
 0  1
[ CPUDoubleType{2,2} ]
 1.1000  2.0000
 3.5000  4.2000
[ CPUDoubleType{2,2} ]
  8.2100  10.6000
 18.5500  24.6400
```

```
[ CPUDoubleType{2,2} ]

--------------------
<test-logicalor.tl>
a = [[1,0,3],[4,0,-1]];
b = [[1,0,5],[-1,5,-1]];
print(a || b);
<test-logicalor.out>
 1  0  1
 1  1  1
[ CPUIntType{2,3} ]

--------------------
<test-logicalnot.tl>
a = [[1,0,3],[4,0,-1]];
print(!a);
<test-logicalnot.out>
 0  1  0
 0  1  0
[ CPUIntType{2,3} ]

--------------------
<test-logicaland.tl>
a = [[1,0,3],[4,0,-1]];
b = [[1,0,5],[-1,5,-1]];
print(a && b);
<test-logicaland.out>
 1  0  1
 1  0  1
[ CPUIntType{2,3} ]

--------------------
<test-lessequal.tl>
a = [[1,2,3],[4,5,6]];
b = [[1,3,5],[3,5,6]];
print(a <= b);
<test-lessequal.out>
 1  1  1
 0  1  1
[ CPUIntType{2,3} ]

--------------------
<test-less.tl>
```

```
a = [[1,2,3],[4,5,6]];
b = [[1,3,5],[3,5,6]];
print(a < b);
<test-less.out>
 0  1  1
 0  0  0
[ CPUIntType{2,3} ]

-------------------
<test-greaterequal.tl>
a = [[1,2,3],[4,5,6]];
b = [[1,3,5],[3,5,6]];
print(a >= b);
<test-greaterequal.out>
 1  0  0
 1  1  1
[ CPUIntType{2,3} ]

-------------------
<test-greater.tl>
a = [[1,2,3],[4,5,6]];
b = [[1,3,5],[3,5,6]];
print(a > b);
<test-greater.out>
 0  0  0
 1  0  0
[ CPUIntType{2,3} ]
-------------------
<test-floordivide.tl>
a = [[1.1,2.0],[3.5,4.2],[3.9, 0.6]] // 2.0;
print(a);
<test-floordivide.out>
 0  1
 1  2
 1  0
[ CPUDoubleType{3,2} ]

-------------------
<test-equal.tl>
a = [[1,2,3],[4,5,6]];
b = [[1,3,5],[3,5,6]];
print(a == b);
<test-equal.out>
```

```
 1  0  0
 0  1  1
[ CPUIntType{2,3} ]

-------------------
<test-dotpow.tl>
a = [[1.1,2.0],[3.5,4.2]] .^ 2;
print(a);
<test-dotpow.out>
  1.2100   4.0000
 12.2500  17.6400
[ CPUDoubleType{2,2} ]

-------------------
<test-dotmul.tl>
a = [[1, 2, 3], [4, 5, 6]];
b = [[-1, -2, -3], [-4, -5, -6]];
c = a .* b;
print(c);
d = a .* 2;
print(d);
e = float_of(a) .* 0.2;
print(e);
<test-dotmul.out>
-1 -4 -9
-16 -25 -36
[ CPUIntType{2,3} ]
  2   4   6
  8  10  12
[ CPUIntType{2,3} ]
 0.2000  0.4000  0.6000
 0.8000  1.0000  1.2000
[ CPUDoubleType{2,3} ]

-------------------
<test-divide.tl>
a = [[1.1,2.0],[3.5,4.2]] / 0.5;
print(a);
<test-divide.out>
 2.2000  4.0000
 7.0000  8.4000
[ CPUDoubleType{2,2} ]
```

```
------------------
<test-assign.tl>
a = [[1,4,2,5,7,8],[3,2,1,5,7,9]];
a[0:2:1,1:3:1] = [[10,12],[13,14]];
print(a);
<test-assign.out>
  1  10  12   5   7   8
  3  13  14   5   7   9
[ CPUIntType{2,6} ]

------------------
<test-add.tl>
a = [1,2,4];
b = [2,3,4];
print(a+b);
print(a+-b);
print(a+1);

<test-add.out>
 3
 5
 8
[ CPUIntType{3} ]
-1
-1
 0
[ CPUIntType{3} ]
 2
 3
 5
[ CPUIntType{3} ]
------------------
<test-while.tl>
a = 1;
b = 2;

while (a) {
    if (b == 10) {
        a = 0;
    }
    print(b);
    b = b + 1;
}
```

```
<test-while.out>
2
[ CPUIntType{} ]
3
[ CPUIntType{} ]
4
[ CPUIntType{} ]
5
[ CPUIntType{} ]
6
[ CPUIntType{} ]
7
[ CPUIntType{} ]
8
[ CPUIntType{} ]
9
[ CPUIntType{} ]
10
[ CPUIntType{} ]

--------------------
<test-noret.tl>
def foo(a, b) {
    print(a);
    print(b);
}

foo(1, 2);
<test-noret.out>
1
[ CPUIntType{} ]
2
[ CPUIntType{} ]
--------------------
<test-if4.tl>
a = [1,0];
if (!a) {
      if (a) {
            print("wrong");
      } else {
            print("yes");
      }
} else {
```

```
      print("ok");
}
<test-if4.out>
ok

-------------------
<test-if3.tl>
a = [1,0];
if (!a) {
      print("yes");
} else {
      print("no");
}
<test-if3.out>
no

-------------------
<test-if2.tl>
a = [0,0];
if (a) {
      print("yes");
} else {
      print("no");
}
<test-if2.out>
no

-------------------
<test-if1.tl>
a = [1,0];
if (a) {
      print("yes");
} else {
      print("no");
}
<test-if1.out>
yes

-------------------
<test-func6.tl>
e = 100;
def foo(a, b) {
    d = a+e;
```

```
      if (e < 5) {
            return 10;
      } else {
            return 88;
      }
    return e;
}
e = 0;

f = foo(1, 2);
print(f);
<test-func6.out>
10
[ CPUIntType{} ]

-------------------
<test-func5.tl>
e = 100;
def foo(a, b) {
    d = a+e;
      if (e < 5) {
            return 10;
      } else {
            return 88;
      }
    return e;
}

f = foo(1, 2);
print(f);
<test-func5.out>
88
[ CPUIntType{} ]

-------------------
<test-func4.tl>
e = 100;
def foo(a, b) {
    d = a+e;
    return e;
}

f = foo(1, 2);
```

```
print(f);
<test-func4.out>
100
[ CPUIntType{} ]

--------------------
<test-func3.tl>

e = 100;
def foo(a, b) {
    e = a+b;
    return e;
}

f = foo(1, 2);
print(f);
<test-func3.out>
3
[ CPUIntType{} ]

--------------------
<test-func2.tl>
def foo(a, b) {
    c = a;
    d = b;
    e = c + d;
    return e;
}

f = foo(1, 2);
print(f);
<test-func2.out>
3
[ CPUIntType{} ]
--------------------
<test-func1.tl>
def foo(a) {
    return a;
}

f = foo(1);
print(f);
<test-func1.out>
```

```
1
[ CPUIntType{} ]

------------------
<test-forvar.tl>
a = var [[0, 1], 2];
for (i in a)
{
    print(i);
}
<test-forvar.out>
[ 0
 1
[ CPUIntType{2} ]
]
[2
[ CPUIntType{} ]
]

------------------
<test-for.tl>
a = [1, 2, 3];
b = 1;
for (i in a) {
    print(i);
}
<test-for.out>
 1
[ CPUIntType{1} ]
 2
[ CPUIntType{1} ]
 3
[ CPUIntType{1} ]

------------------
<fail-undefined3.tl>
def foo() {
      s = 1;
}
a = s;

<fail-undefined3.err>
Fatal error: exception Failure("variable s not defined")
```

```
<fail-undefined2.tl>
a = foo();
<fail-undefined2.err>
Fatal error: exception Failure("function foo not defined")
<fail-undefined1.tl>
a = b;
<fail-undefined1.err>
Fatal error: exception Failure("variable b not defined")
<fail-return.tl>
def foo(a, b) {
    c = a;
    d = b;
    e = c + d;
    return e;
}

f = foo(1, 2);
return 2;
print(f);
<fail-return.err>
Fatal error: exception Failure("Return outside functions")
<fail-badtype.tl>
a = [1,1.0];
<fail-badtype.err>
Fatal error: exception Failure("invalid type")
<fail-badfuncparam.tl>
def foo(a,b,c) {

}
a = foo(1);
<fail-badfuncparam.err>
Fatal error: exception Failure("the number of arguments mismatch")
<fail-baddim.tl>
a = [1,[2,3]];
<fail-baddim.err>
Fatal error: exception Failure("invalid dim")
<test-tree.tl>
tree = var [nil, 0, nil]; # [left, value, right]
tree[0] = var [var [nil, 1, nil], 2, var [nil, 4, var [nil, 5, nil]]];
tree[2] = var [nil, 6, nil];

def preorder(t) {
    if (t != nil) {
```

```
        print(t[1]);
        preorder(t[0]);
        preorder(t[2]);
    }
    return 1;
}

print("Pre-Order");
preorder(tree);
<test-tree.out>
Pre-Order
0
[ CPUIntType{} ]
2
[ CPUIntType{} ]
1
[ CPUIntType{} ]
4
[ CPUIntType{} ]
5
[ CPUIntType{} ]
6
[ CPUIntType{} ]

-------------------
<test-gc.tl>
a = var [[0, 1], 2];
a = var [[0, 3], 3];
<test-gc.out>

-------------------
```