LéPix

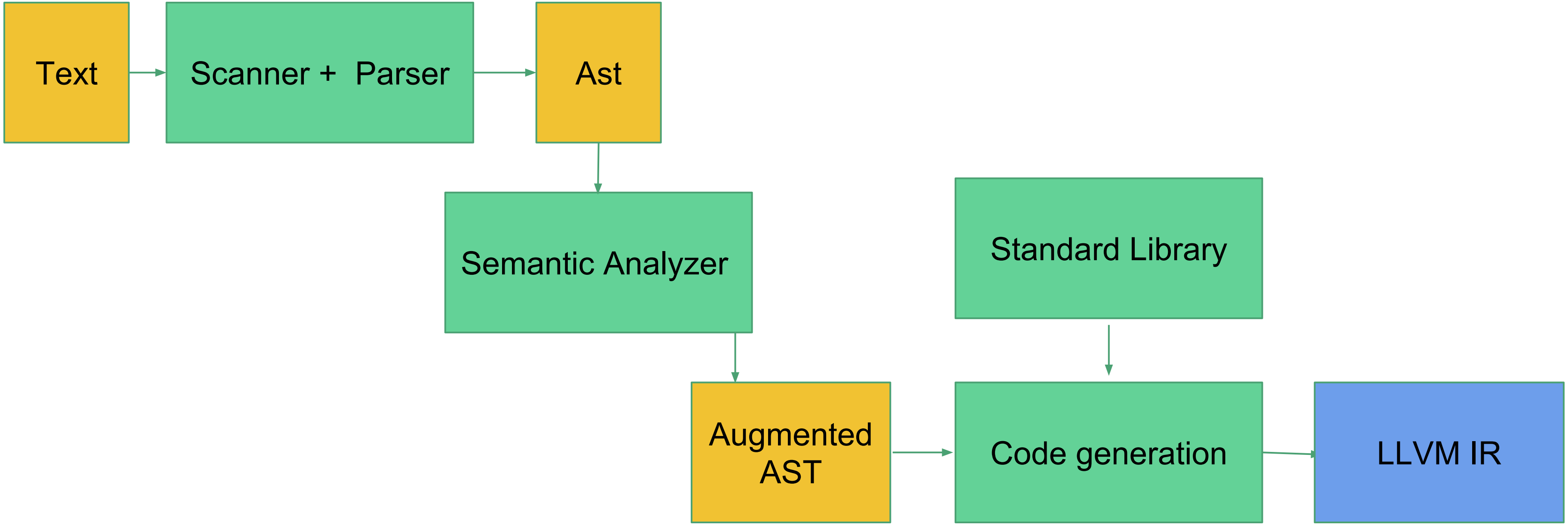
Ceci n’est pas le Photoshop

(ಠ_ಠ)>⌐■-■ (ಥ_ಥ) (•◡•) (⌐■_■)

Introduction

- Imperative programming language
- Swift-esque syntax
- Compiles to LLVM
- Floating point values and arrays to enable image manipulation

Implementation



Text

Scanner + Parser

Ast

Semantic Analyzer

Standard Library

Augmented AST

Code generation

LLVM IR

Syntax

- Syntax is reminiscent of the Swift programming language

Function declaration	fun gcd(a : int , b : int) : int { ... }
Variable declaration	var a : int = 42;
Array declaration	var arr : int[[2,2]] = [1,2,3,4];
Control Flow	C-style for , while and if-else
Array Access	arr[[1,2]]

Semantics

- Semantics are checked with a depth first traversal of the AST
- All identifiers are checked for validity
- Expressions, statements and declarations are type checked
- Static scoping and strong typing rules are used
- Implicit float to int casting on assignment

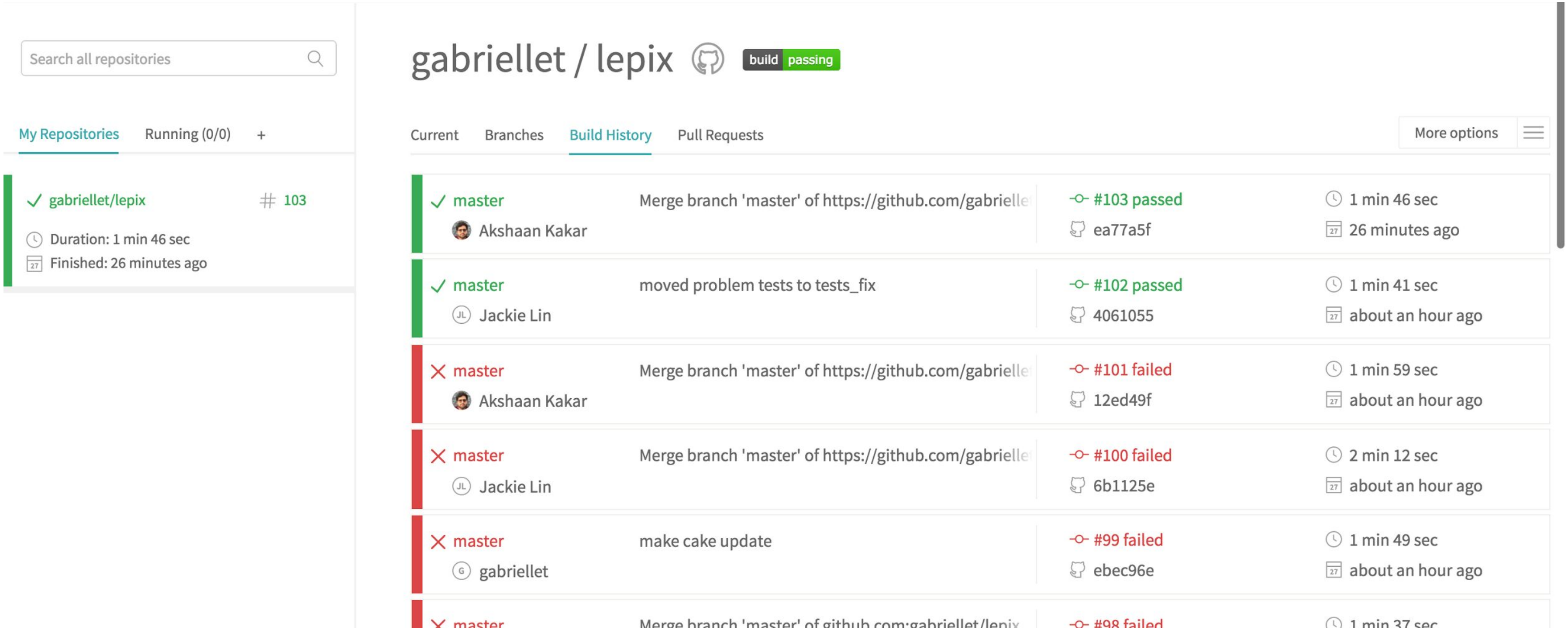
Codegen

- Codegen is performed on a semantically checked, augmented AST
- Symbol table holds pointers to global and local variables (including arrays)
- Functions definitions are held in a function table
- 2-D matrices are allocated as 1-D arrays and indexed accordingly
- Standard library is automatically included at compile time
- Stdlib:
    - Math : Modulo, Square Root, Prime Sieve, GCD etc.
    - I/O : Printing to console for different types (Int, Bool, Float), Printing an RGB image in PPM format

Testing: Building Regression Test Suite

- Began with simple examples based on how we wanted our language to look/behave (e.g. hello world)
- Based first set of tests on MicroC
- Revised test suite whenever language syntax/behavior changed
- Expanded test suite whenever new functionality was added to language (e.g. 1D array and 2D array)
- Coverage: tested for both expected behavior and potential user error when applicable

Testing: Continuous Integration with Travis



✓ gabriellet/lepix # 103

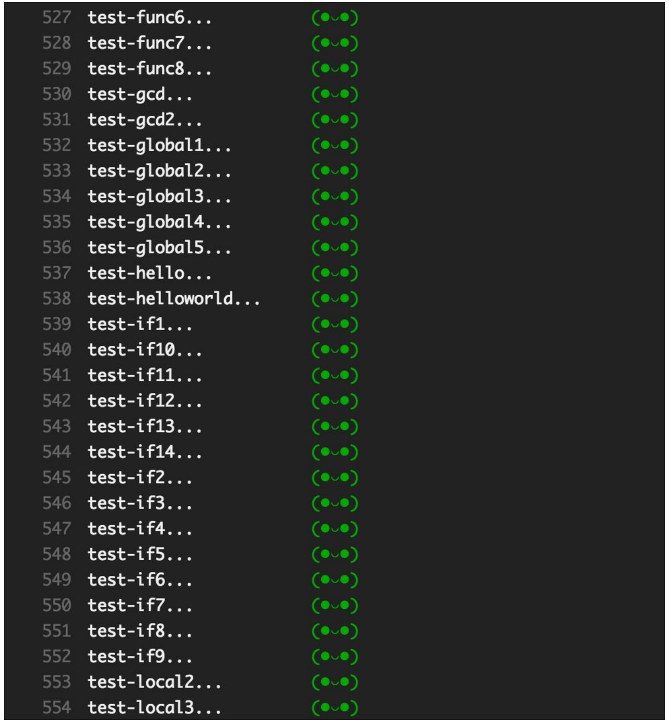
Duration: 1 min 46 sec

Finished: 26 minutes ago

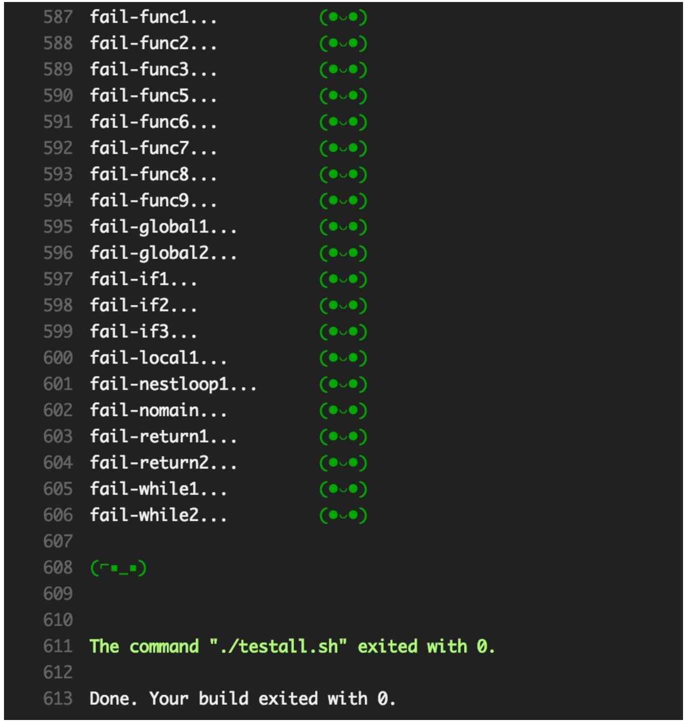
gabriellet / lepix

✓ master Akshaan Kakar	Merge branch 'master' of https://github.com/gabrielle	#103 passed ea77a5f	1 min 46 sec 26 minutes ago
✓ master Jackie Lin	moved problem tests to tests_fix	#102 passed 4061055	1 min 41 sec about an hour ago
✗ master Akshaan Kakar	Merge branch 'master' of https://github.com/gabrielle	#101 failed 12ed49f	1 min 59 sec about an hour ago
✗ master Jackie Lin	Merge branch 'master' of https://github.com/gabrielle	#100 failed 6b1125e	2 min 12 sec about an hour ago
✗ master gabriellet	make cake update	#99 failed ebec96e	1 min 49 sec about an hour ago
✗ master	Merge branch 'master' of github.com:gabriellet/lepix	#98 failed	1 min 37 sec

Testing: When Everything Works (•◡•) (⌐■_■)

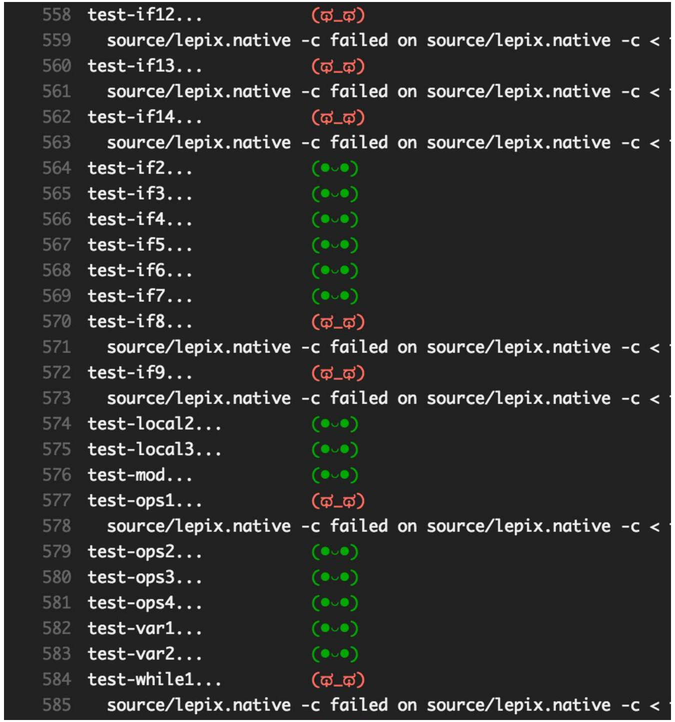


```
test-func6...          (•◡•)
test-func7...          (•◡•)
test-func8...          (•◡•)
test-gcd...            (•◡•)
test-gcd2...           (•◡•)
test-global1...        (•◡•)
test-global2...        (•◡•)
test-global3...        (•◡•)
test-global4...        (•◡•)
test-global5...        (•◡•)
test-hello...          (•◡•)
test-helloworld...     (•◡•)
test-if1...            (•◡•)
test-if10...           (•◡•)
test-if11...           (•◡•)
test-if12...           (•◡•)
test-if13...           (•◡•)
test-if14...           (•◡•)
test-if2...            (•◡•)
test-if3...            (•◡•)
test-if4...            (•◡•)
test-if5...            (•◡•)
test-if6...            (•◡•)
test-if7...            (•◡•)
test-if8...            (•◡•)
test-if9...            (•◡•)
test-local2...         (•◡•)
test-local3...         (•◡•)
```

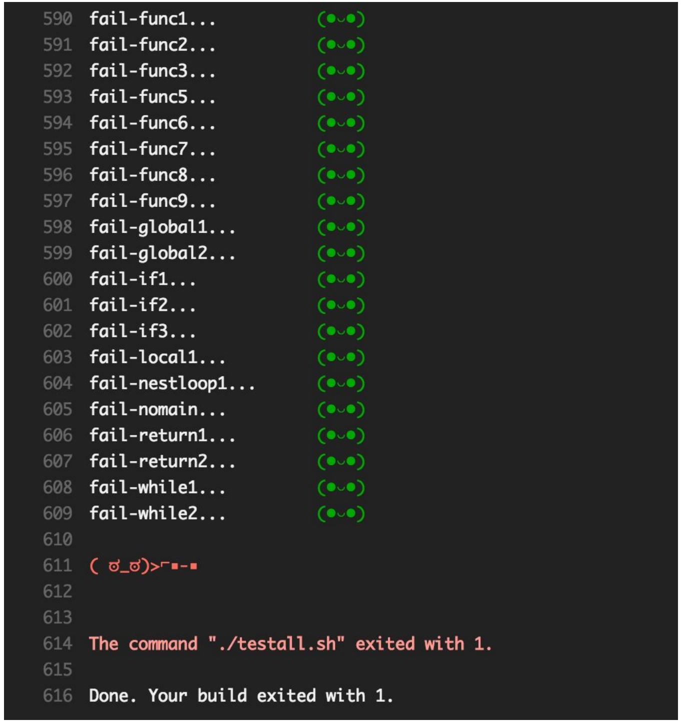


```
fail-func1...           (•◡•)
fail-func2...           (•◡•)
fail-func3...           (•◡•)
fail-func5...           (•◡•)
fail-func6...           (•◡•)
fail-func7...           (•◡•)
fail-func8...           (•◡•)
fail-func9...           (•◡•)
fail-global1...         (•◡•)
fail-global2...         (•◡•)
fail-if1...             (•◡•)
fail-if2...             (•◡•)
fail-if3...             (•◡•)
fail-local1...          (•◡•)
fail-nestloop1...       (•◡•)
fail-nomain...          (•◡•)
fail-return1...         (•◡•)
fail-return2...         (•◡•)
fail-while1...          (•◡•)
fail-while2...          (•◡•)

(⌐■_■)


The command "./testall.sh" exited with 0.

Done. Your build exited with 0.
```

Testing: When Everything Breaks (ಥ_ಥ) (ಠ_ಠ)>⌐■-■

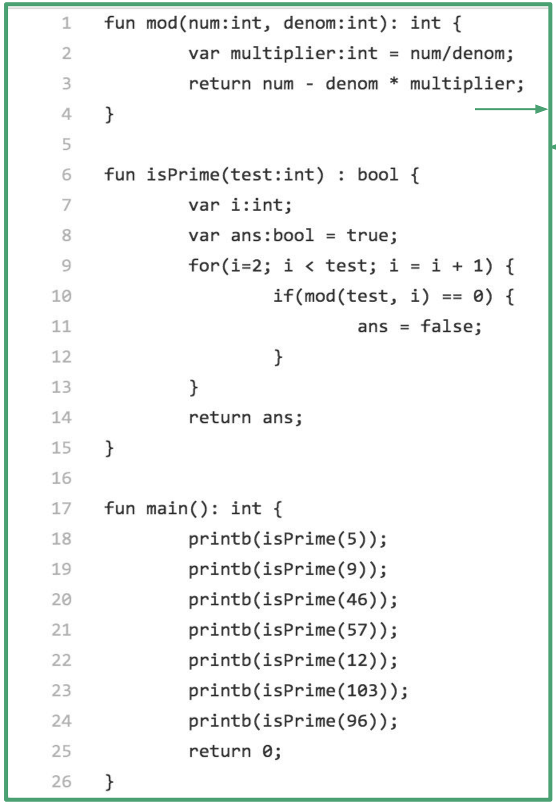


```
test-if12...            (ಥ_ಥ)
  source/lepix.native -c failed on source/lepix.native -c <
test-if13...            (ಥ_ಥ)
  source/lepix.native -c failed on source/lepix.native -c <
test-if14...            (ಥ_ಥ)
  source/lepix.native -c failed on source/lepix.native -c <
test-if2...             (•◡•)
test-if3...             (•◡•)
test-if4...             (•◡•)
test-if5...             (•◡•)
test-if6...             (•◡•)
test-if7...             (•◡•)
test-if8...             (ಥ_ಥ)
  source/lepix.native -c failed on source/lepix.native -c <
test-if9...             (ಥ_ಥ)
  source/lepix.native -c failed on source/lepix.native -c <
test-local2...          (•◡•)
test-local3...          (•◡•)
test-mod...             (•◡•)
test-ops1...            (ಥ_ಥ)
  source/lepix.native -c failed on source/lepix.native -c <
test-ops2...            (•◡•)
test-ops3...            (•◡•)
test-ops4...            (•◡•)
test-var1...            (•◡•)
test-var2...            (•◡•)
test-while1...          (ಥ_ಥ)
  source/lepix.native -c failed on source/lepix.native -c <
```



```
fail-func1...           (•◡•)
fail-func2...           (•◡•)
fail-func3...           (•◡•)
fail-func5...           (•◡•)
fail-func6...           (•◡•)
fail-func7...           (•◡•)
fail-func8...           (•◡•)
fail-func9...           (•◡•)
fail-global1...         (•◡•)
fail-global2...         (•◡•)
fail-if1...             (•◡•)
fail-if2...             (•◡•)
fail-if3...             (•◡•)
fail-local1...          (•◡•)
fail-nestloop1...       (•◡•)
fail-nomain...          (•◡•)
fail-return1...         (•◡•)
fail-return2...         (•◡•)
fail-while1...          (•◡•)
fail-while2...          (•◡•)

( ಠ_ಠ)>⌐■-■


The command "./testall.sh" exited with 1.

Done. Your build exited with 1.
```

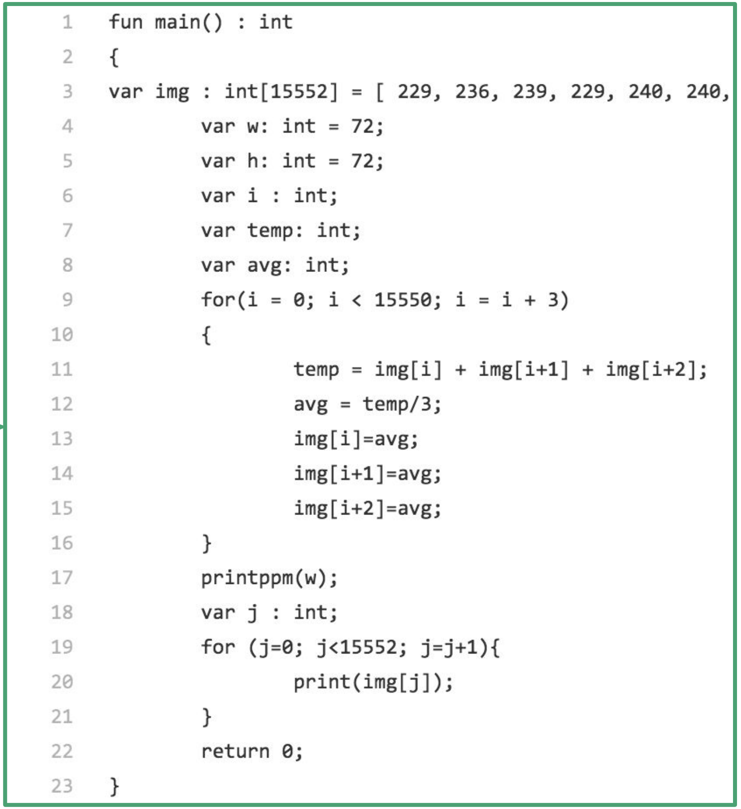
Example LePix programs



```
fun mod(num:int, denom:int): int {
        var multiplier:int = num/denom;
        return num - denom * multiplier;
}

fun isPrime(test:int) : bool {
        var i:int;
        var ans:bool = true;
        for(i=2; i < test; i = i + 1) {
                if(mod(test, i) == 0) {
                        ans = false;
                }
        }
        return ans;
}

fun main(): int {
        printb(isPrime(5));
        printb(isPrime(9));
        printb(isPrime(46));
        printb(isPrime(57));
        printb(isPrime(12));
        printb(isPrime(103));
        printb(isPrime(96));
        return 0;
}
```

Prime Sieve

RGB Neg Filter



```
fun main() : int
{
var img : int[15552] = [ 229, 236, 239, 229, 240, 240,
        var w: int = 72;
        var h: int = 72;
        var i : int;
        var temp: int;
        var avg: int;
        for(i = 0; i < 15550; i = i + 3)
        {
                temp = img[i] + img[i+1] + img[i+2];
                avg = temp/3;
                img[i]=avg;
                img[i+1]=avg;
                img[i+2]=avg;
        }
        printppm(w);
        var j : int;
        for (j=0; j<15552; j=j+1){
                print(img[j]);
        }
        return 0;
}
```

Images

(⌐■_■)

```
fun demo(): void {}
```