THE JOY OF ENGINEERING

COMPUTER SCIENCE/COMPUTER ENGINEERING

Michael Yan, Eric Leung, Binna Han

Calc-E
int keyboard_key()
{
  // initialize keyboard, reset every column to high
  keyboard_init();
  int c; // goes through columns
  int r; // goes through rows
  for (c = 0; c < columns; c++)
  {
    keyboard_column_low(c);

    for (r = 0; r < rows; r++)
    {
      if (!keyboard_row_read(r))
      {
        return keysPressed[c][r];
      }
    }
  }
  return -1; // no keys are pressed
}
void keyboard_get_entry(struct entry *result)
{
    int ent = keyboard_key();
    // if something is pressed
    if (ent != -1)
    {
        ... 
    }
}
int keyboard_key()
{
    // initialize keyboard, reset every column to high
    keyboard_init();
    int c; // goes through columns
    int r; // goes through rows
    for (c = 0 ; c < columns; c++)
    {
        keyboard_column_low(c);

        for (r = 0 ; r < rows ; r++)
        {
            if (!keyboard_row_read(r))
            {
                return keysPressed[c][r];
            }
        }
    }
    return -1; // no keys are pressed
}
void keyboard_get_entry(struct entry *result)
{
    int ent = keyboard_key();
    // if something is pressed
    if (ent != -1)
    {
        ...}
}
char printScreen[] = "           ";
int pos = 1; // save a space for negative sign
int pressed = 1; //Variable that determines if any key is being pressed.
int lengthOfScreen = 11;

for (;;)
{
    keyboard_get_entry(&entry);  //====== built on the first part of lab3

    if (entry.number == NO_KEY_BEING_PRESSED)
    {
    pressed = 1;
    }
    //Wait for a key to be pressed
    if (entry.number != NO_KEY_BEING_PRESSED && & pressed == 1)
    {
    }
for (;;)
{
    int num1;
    int num2;
    int result;

    keyboard_get_entry(&entry); // Built on lab 3

    if (entry.operation == '\r') // If the user enters a number add it to the stack
    {
        stack[stackPointer++] = entry.number; // Advance the stack pointer
        lcd_init();
    }

    if (entry.number != INT_MAX) // If a number is entered (INT_MAX is returned if operation)
    {
        stack[stackPointer] = entry.number;
    }
LAB 4 MAIN.C (2/3)

if (entry.operation == '+' || entry.operation == '-' || entry.operation == '*' || entry.operation == '/') {
  if (entry.number == INT_MAX) // Only an operation is entered
    {
        stackPointer--;
    }  
    num1 = stack[stackPointer--]; // Pop the first number and decrement the stack pointer
    num2 = stack[stackPointer]; // Pop the next number
switch(entry.operation)
{
    case '+':
        result = num1 + num2;
        stack[stackPointer] = result; //Push the result onto the stack
        lcd_print_int(result);
        break;
    case '-':
        result = num2 - num1;
        stack[stackPointer] = result;
        lcd_print_int(result);
        break;
    case '*':
        result = num1 * num2;
        stack[stackPointer] = result;
        lcd_print_int(result);
        break;
    case '/': //Division does not work due to faults we cannot control
        result = num1 / num2;
        stack[stackPointer] = result;
        lcd_print_int(result);
        break;
}
stackPointer++;
return 0;
char printScreen[] = "  ";
int pos = 1; // save a space for negative sign
int pressed = 1; //Variable that determines if any key is being pressed.
int lengthOfScreen = 11;

for (;;)
{
    keyboard_get_entry(&entry); // built on the first part of lab3

    if (entry.number == NO_KEY_BEING_PRESSED)
    {
        pressed = 1;
    }
    //Wait for a key to be pressed
    if (entry.number != NO_KEY_BEING_PRESSED && &pressed == 1)
    {
        ...
        pressed = 0; //So that it doesn't print the same number a bajillion times
    }
}
for (;;) {
    int num1;
    int num2;
    int result;

    keyboard_get_entry(&entry); // Built on lab 3

    if (entry.operation == '\r') // If the user enters a number add it to the stack
    {
        stack[stackPointer++] = entry.number; // Advance the stack pointer
        lcd_init();
    }

    if (entry.number != INT_MAX) // If a number is entered (INT_MAX is returned if operation)
    {
        stack[stackPointer] = entry.number;
    }
if (entry.operation == '+' || entry.operation == '-' || entry.operation == '*' || entry.operation == '/')
{
    if (entry.number == INT_MAX) // Only an operation is entered
    {
        stackPointer--; // Pop the first number and decrement the stack pointer
    }
    stackPointer--; // Pop the next number
    num1 = stack[stackPointer--]; // Pop the first number and decrement the stack pointer
    num2 = stack[stackPointer]; // Pop the next number
switch(entry.operation) {
    case '+':
        result = num1 + num2;
        stack[stackPointer] = result; //Push the result onto the stack
        lcd_print_int(result);
        break;
    case '-':
        result = num2 - num1;
        stack[stackPointer] = result;
        lcd_print_int(result);
        break;
    case '*':
        result = num1 * num2;
        stack[stackPointer] = result;
        lcd_print_int(result);
        break;
    case '/': //Division does not work due to faults we cannot control
        result = num1 / num2;
        stack[stackPointer] = result;
        lcd_print_int(result);
        break;
}
stackPointer++;
}
return 0;
HOW WE FEEL

 лично

 Enjoyable Experience
GOOD LUCK WITH FINAL EXAMS!

AND

Have a RELAXING Christmas Vacation!

Bye bye Calc-E! We’ll miss you~