Assignment 1

Due Sep 25, 2015 by 11:59pm **Points** 100 **Submitting** a file upload
File Types zip **Available** after Sep 2, 2015 at 9pm

Implement a virtual machine interpreter for PM/0 as described in Lecture 4.

Assignment Guidelines

The program must be written in C (not C++) and must run on Eustis.

Submit a single ZIP file containing:

- Your source code
- A readme document indicating how to compile and run
- The output of a test program; you can use the example from below

Values and Bounds

Initially SP = 0; BP = 1; PC = 0; IR = 0.

MAX_STACK_HEIGHT = 2000; you can statically allocate a stack store of this size.
MAX_CODE_LENGTH = 500; you can statically allocate a code store of this size.
MAX_LEXI_LEVELS = 3.

Initialize all cells of the stack store to 0.

Input and Output

The input and output files will be named **mcode.txt** and **stacktrace.txt**. Examples are in the Files section.

- Every line of input will consist of the three numbers OP, L and M.
- The output consists first of the interpreted assembly language with line numbers, then of the execution trace of the program in the virtual machine, showing the registers and stack **after** each instruction.
  - Follow the example format as closely as you can.
  - Don’t show stack[0].
  - You need to separate each stack frame with a vertical line. The VM does not really explicitly know about stack frames, so use special case code on the CAL and OPR 0, 0 instructions to set this up.

Helpful Hints

1. Use a three-member structure for instructions.

2. Rather than actually popping twice and pushing once, you can cheat a bit on arithmetic functions and use pseudocode like the following for OPR 0, 2 (ADD):

```
{ sp <-- sp – 1; stack[sp] <-- stack[sp] + stack[sp + 1]; }
```

3. You can find base(L) with a simple iterative function like this one (the **base** parameter will be the current value of BP):

```
int base(l, base) {
    int b1 = base; // find base L levels down
    while (l > 0) {
        b1 = stack[b1 + 1];
        l--;
    }
    return b1;
}
```