TCSS 371 Spring 2017 HW6

Due: by midnight, Friday June 2 (or with 10% late penalty, by midnight, Saturday June 3)

In this assignment, you will add a number of features to your LC3 simulator.
The starting code for HW5 was able to load a single object file and the name of that file was supplied to the program as a command line argument. Now we allow multiple object files to be named. Each one is loaded at the address indicated in the object file. The last object file mentioned in the command line is also where execution of the program should start. We are allowing multiple object files so we can more easily initalize the trap vector table and the trap code. A typical call to the simulator will look like:

```
./MyHW6 trapvectortable.obj out.obj puts.obj halt.obj trapcalls.obj
```

You must also add the following instructions:

- `LEA` (easier than LD!)
- `JSR` (be sure to save the return address in R7)
- `RET`
- `TRAP` (save the return address, look in the trap vector table, and jump to the required trap routine)

You must implement the display output device. We will simulate a display that operates so fast that it's always ready to accept another character for output. It's not realistic, but the LC3 simulator does this too. You must

- initialize the display's ready bit, DSR[15], to 1 and
- print out the required character whenever a value is written to the DDR's memory location

There are three ways to write to a memory location (ST, STI, STR). The addresses of the DSR and DDR are as given in the book.

We no longer use opcode 13 to stop the simulator. Instead, as described on p. 277. we will use the "run latch". This is a single bit which allows the machine to run when it's on and stops the machine when it's off. It is the most significant bit of a 16 bit Machine Control Register. Like the i/o device registers, this register is memory mapped. It's memory address is 0xFFFE. So whenever your simlulator notices that bit being cleared (set to 0), the machine should stop.

In the LC3, the output routines include OUT and PUTS. They have trap vectors x21 and x22 and the trap routines are located at x0430 and x0450. If your simulator is written correctly, none of these numbers will appear in your C code. Instead your simluator must figure out what to do by examing bits of the TRAP instruction and looking in the trap vector table.

There is one set of test files posted on Canvas. It includes a minimal test of both trap calls and function calls. There are 3 trap routines but the trap vectors and trap routine locations are not the standard ones loaded into the LC3 by default. You can run the test code in the LC3, but you should load the “main” routine (trapscalls.obj) last of all. The correct output is:

```
HELLO
UWT
HELLO
UWT
----- Halting the processor -----
```

In HW6, the only output will be output produced by the runnning program. Remove all output messages previously generated as part of HW5!