COMP 2280 Winter 2022 - Lab 1 – Due Feb. 11th

The goal of this lab is to make you comfortable with using the LC-3 editor and simulator. You should read the **Guide to Using LC3Tools** and watch the LC3 Walkthrough Video (found on UMLearn) before attempting this lab. Other information about the LC3 can be found at the textbook's website, or on lc3tutor.org.

Write the answers in the highlighted spaces (Adobe Acrobat Reader or other programs should be able to do it) and submit a PDF copy of this file to the appropriate Dropbox, along with your .asm files (**NOT** .obj files)

Question 1

1. Type in the program following these instructions into the LC-3 editor and save it as **HelloWorld.asm** in some working directory.
2. **Assemble** the program and ensure there are no mistakes. If there are mistakes, fix them and re-assemble the program.
3. Open **Windows Explorer (or Finder on a Mac)** and ensure that you have the following files in your working directory: **HelloWorld.asm and HelloWorld.obj.** There may be other files but just ignore them.
4. Start/Switch to the **LC-3 Simulator** and open the file HelloWorld.obj if it is not already.
5. Choose **Run**, to run the program. Ensure that the LC3 Console window has the correct output.
6. Now, in the **Jump to Location** field, write x3000 and click the **blue arrow** button to set the program counter to x3000. This will reset the program to start executing at the start of the program.
7. **Add breakpoint**s at instructions located at x3001 (TRAP PUTS) and at x3003 (HALT). There should be a red stop sign at the start of the lines.
8. Clear the LC3 console window by choosing the top-right button near the console panel.
    a. At this point, the value of PC register (in hexadecimal) is: x3000.
9. Run the program. The execution of your program should stop at line 0x3001 due to the breakpoint. The message is not yet displayed in the LC3 console window.
    a. What is the value of R0 (in hexadecimal) at this point: x3004?
    b. What does this value correspond to? **Hint: Take a look at the memory.** corresponds to "H"
10. Run the program from this point. Your program will pause at the next breakpoint (at x3003).
    a. What is the value of R0 (in hexadecimal) at this point: x0000?
11. Remove all breakpoints and run the program from this point. Did it work correctly?

    Yes the program worked correctly

```
1  ; Example of Hello world for LC-3
2  ; Prints Hello World.
3  ; Save as HelloWorld.asm
4
5      .orig   x3000   ; the following instructions
6              ; start at addr 0x3000
7              lea r0,mesg ; load addr of mesg into register r0
8          trap    x22 ; print string pointed to by r0
9      and r0,r0,x0
10         halt        ; halt the program
11
12 mesg     .stringz "Hello world.\n"
13             .end
14
```

Question 2

1. Type in the program following these instructions into the LC-3 editor and save it as **sum.asm**, assemble it and load **sum.obj** into the LC3 simulator.
2. Add in a breakpoint to the line lines **and `r2,r2,x0`** and **`add r3,r1,r2`**.
3. Run the program. When the program stops at the first breakpoint :
   a. R1 = x0003 (in hex)
   b. R2 = x0000 (in hex),
   c. R3 = x0000 (in hex).
4. Continue the program. When the program stops at the second breakpoint:
   a. R1 = x0003 (in hex)
   b. R2 = x000A (in hex) ,
   c. R3 = x0000 (in hex) .
5. Continue the program. When the program stops by itself:
   a. R1 = x7FFF (in hex)
   b. R2 = x000A (in hex) ,
   c. R3 = x000D (in hex) .
   d. Note the value of R1, why do you think it has a different value?
   The value of r1 was overwritten with MASK_HI .FILL x7FFF when halt is executed
6. Reset the program counter to x3000 and remove all the breakpoints.
7. Single step through the program line by line, using **step over** button.
   a. The value of register R1, just before the **halt** instruction is executed, is:
   x0003 (in hex).

```
1  ; This example adds the contents of two registers and stores it
2  ; into a third
3
4       .orig   x3000
5
6       and r3,r3,x0    ; clear r3
7       and r1,r1,x0    ; clear r1
8   add     r1,r1,x3    ; r1 is set to 3
9       and r2,r2,x0    ; clear r2
10      add r2,r2,xA   ; r2 is set to 10 (base 10)
11  add r3,r1,r2   ; r3 <- r1 + r2
12
13      halt
14      .end
15
```