

Name: \_\_\_\_\_

Date: \_\_\_\_\_

## CS270 Recitation 9 “LC-3 Stack Exercise”

### Goals

- Further practice with LC-3 assembly code
- Learn about stack protocols
- Prepare for Programming Assignment 7

### The Assignment

Make a subdirectory called R9 for the recitation assignment, in which all files should reside. Now download the source file from the following link to the R9 directory and open it in the text editor:

<http://www.cs.colostate.edu/~cs270/.Spring16/recitations/R9/r9.asm>

You will see a fully implemented COMPUTE function, and partial code with comments for the MAIN function. The goal of this assignment is to fill in the code in MAIN that calls the function, as shown below.

### ;; Main program

```
MAIN    LD    R6, STACK        ; Initialize stack pointer
                                     ; 1) Load parameters
                                     ; 2) Push parameters
                                     ; 3) Call function
                                     ; 4) Pop result
                                     ; 5) Clean up the parameters
                                     ; 6) Store result

        HALT                    ; Program complete
```

Each comment may correspond to one or more assembly instructions. Note that the stack pointer is initialized already. You must **1)** load each of the parameters to COMPUTE from memory, **2)** push these parameters on the stack in reverse order, then **3)** call COMPUTE. On return, you must **4)** pop the result into a register, **5)** clean up (or pop) the stack to remove the three parameters, and **6)** store the result in memory. The COMPUTE function is completely defined, and has a comment that shows how it would be implemented in C. The parameter values for the call and a memory location for the result are already allocated. When the code is complete, assemble it and run in the simulator:

```
$ ~cs270/lc3tools/lc3as r9.asm
$ ~cs270/lc3tools/lc3sim-tk &
```

Now step the code in the simulator, drawing a picture of the stack using the table below. Even though the stack addresses are already filled in, do not assume that all of the locations shown are used.

<i>Address</i>	<i>Stack Description</i>	<i>Real Value</i>
x3ff9		
x3ffa		
x3ffb		
x3ffc		
x3ffd		
x3ffe		
x3fff		

What is the final value of the stack pointer in R6 when HALT is executed? \_\_\_\_\_

Has the program restored the stack pointer after the function call? \_\_\_\_\_

What is the value stored in RESULT when HALT is executed? \_\_\_\_\_

Is RESULT correct based on the implementation of Compute defined in the comment? \_\_\_\_\_

**When finished, please show this answer sheet to the teaching assistant, and turn in r9.asm to the dropbox to receive credit for the recitation.**