CS270 Recitation 7
"LC-3 Programming Introduction"

Goals

1. To learn how to write a basic LC-3 program with functions, conditionals, and a loop.
2. To learn how to use the LC-3 assembler and simulator to debug assembly code.

The Assignment

Make a subdirectory called R7 for the recitation, all files should reside in this subdirectory. Copy the file from the link to the R7 directory, a listing of the code with some comments removed is shown below.

http://www.cs.colostate.edu/~cs270/CurrentSemester/recitations/R7/R7.asm

```
                .ORIG x3000
                BR Main

; A jump table defined as an array of addresses
Functions       .FILL IntAdd          ; address of add      (option 0)
                .FILL IntSub          ; address of subtract (option 1)
                .FILL IntMul          ; address of multiply (option 2)

Main            LEA R0,Functions      ; get base of jump table
                LD  R1,Option         ; get option to use, no error checking
                ADD R0,R0,R1          ; add index of array
                LDR R0,R0,#0          ; get address of function
                JSRR R0               ; call selected function
                HALT

; Parameters and return values for all functions
Option          .BLKW 1               ; which function to call
Param1          .BLKW 1               ; space to specify first parameter
Param2          .BLKW 1               ; space to specify second parameter
Result          .BLKW 1               ; space to store result

; End reserved section: do not change ANYTHING in reserved section!
;-------------------------------------------------------------------------
IntAdd                                ; Your code goes here
                                      ; Solution has ~4 instructions
                 RET
;-------------------------------------------------------------------------
IntSub                                ; Your code goes here
                                      ; Solution has ~6 instructions
                 RET
;-------------------------------------------------------------------------
IntMul                                ; Your code goes here
                                      ; Solution has ~9 instructions
                 RET
                .END
```

1) Use the LC-3 assembler to transform your assembly code into object code that can run on the LC-3 simulator:

$ ~cs270/lc3tools/lc3as R7.asm

2) Load the LC-3 simulator and the TA will help you step through an invocation of one of the LC-3 subroutines:

$ ~cs270/lc3tools/lc3sim-tk &

3) Implement the IntAdd subroutine, using the following algorithm:

- Load the Param1 parameter into a register
- Load the Param2 parameter into a register
- Add the registers storing Param1 and Param2 into another register
- Store the result into the Result memory location and return

4) Test the IntAdd subroutine in the simulator using Option = 0, Param1 = 0x1234, and Param2 = 0x3456. The answer in Result should be 0x468A. Try a negative value as well.

5) Implement the IntSub subroutine, which is a clone of IntAdd, however you must negate the second operand before the addition. Use the 2's complement to do this, as follows:

- Negate the register storing Param2
- Increment Param2 using an immediate add

6) Test the IntSub subroutine in the simulator using Option = 1, Param1 = 0x8765, and Param2 = 0x3456. The answer in Result should be 0x530F.

7) Implement the IntMul subroutine, using the following algorithm:

- Initialize a register for the result to zero
- Load the Param1 parameter into a register
- If zero, go to the exit code
- Load the Param2 parameter into a register
- If zero, go to the exit code
- In a loop, add the Param1 to the result, and decrement Param2
- Continue in the loop while Param2 is positive
- In the exit code store the Result value and return

7) Test the IntMul subroutine in the simulator, using Option = 2, Param1 = 0x1234, and Param2 = 0x0003. The answer in Result should be 0x369C. Try a negative value as well.

8) Submit to the drop box in RamCT for Recitation 7 and show your code to the TA.