Homework 7 // Due at lecture Friday April 25th
Primary contact for this homework: Shengnan Wang [shengnan at cs dot wisc dot edu]
You must do this homework alone. Please staple multiple pages together. Make sure you list your name and section number on the front page.

The following homework requires you to use the LC-3 simulator. Please follow the instructions from HW#5 problem 9 for opening the LC-3 simulator and instructions from HW#6 for submitting screen shots of your code.

Problem 1(10 points)

Given following LC-3 program :

```
        .ORIG   x3005
        LEA     R2, DATA
        LDR     R4, R2, #0
LOOP    ADD     R4, R4, # -3
        BRzp    LOOP
        TRAP    x25
DATA    .FILL   x000B
        .END
```

a. Run the given program in the LC-3 simulator and describe what the program does?
b. How many times will the instruction at the memory address labeled LOOP execute?

Problem 2 (15 points)

a. What does the following program do?

```
        .ORIG x3000
        LD R2, ZERO
        LD R0, M0
        LD R1, M1
LOOP    BRz   DONE
        ADD R2, R2, R0
        ADD R1, R1, #-1
        BR  LOOP
DONE    ST  R2, RESULT
        HALT
RESULT .FILL x0000
ZERO    .FILL x0000
M0      .FILL x0004
```

```
M1      .FILL x0803
        .END
```

b. What value will be contained in RESULT after the program runs to completion?
c. Create and show the symbol table entries generated by the LC-3 assembler.

Problem 3 (20 points)

We want the following program fragment to shift R3 to the left by four bits, but it has an error in it.

```
        .ORIG  x3000
        AND R2, R2, #0
        ADD R2, R2, #4
LOOP    BRz     DONE
        ADD  R2, R2, #-1
        ADD  R3, R3, R3
        BR   LOOP
DONE    HALT
        .END
```

a. Identify the error.
b. Explain how to fix it.

PS: Definition of shift: For binary numbers, it is a bitwise operation that shifts all of the bits of its operand; every bit in the operand is simply moved to the right or to the left a given number of bit positions, and the vacant bit-positions are filled in. In this problem, we use *arithmetic shift*. In an *arithmetic shift*, the bits that are shifted out of either end are discarded. In a left arithmetic shift, zeros are shifted in on the right; in a right arithmetic shift, the sign bit is shifted in on the left, thus preserving the sign of the operand. The following examples use an 16-bit register:

0010 0100 0001 0111 Left-Shift by one bit = 0100 1000 0010 1110;

If you do the left shifting twice, then you shift the register to the left by two bits:

0010 0100 0001 0111 Left-Shift by two bits = 1001 0000 0101 1100;

Problem 4 (25 points)

a. The following LC-3 program compares two character strings of the same length. The source strings are in the .STRINGZ form. The first string starts at memory location x4000, and the second string starts at memory location x4100. If the strings are the same, the program terminates with the value 1 in R5.
(1)Create a flowchart (like Figure 6.3 in ItCS) for this comparison problem;

(2)Insert instructions at (a), (b), and (c) that will complete the program.

```
          .ORIG   x3000
          LD      R1, FIRST
          LD      R2, SECOND
          AND   R0, R0, #0
LOOP    ---------------------------  (a)
          LDR   R4, R2, #0
          BRz    NEXT
          ADD   R1, R1, #1
          ADD   R2, R2, #1
          -------------------------  (b)
          -------------------------  (c)
          ADD  R3, R3, R4
          BRz   LOOP
          AND  R5, R5, #0
          BRnzp  DONE
NEXT    AND R5, R5, #0
          ADD R5, R5, #1
DONE    TRAP   x25
FIRST   .FILL    x4000
SECOND .FILL  x4100
          .END
```

b. If we remove the assumption that the two strings are of the same length, that is, we don't know beforehand the lengths of the two strings we are going to compare, then how will you change the program? Draw a flowchart and revise the program from a. (you should write down a program as well as draw a flowchart).

Problem 5 (30 points)

Write an LC-3 assembly language program that divides 2 positive integer numbers (and discards the remainder). Your code should use a looping construct. Store the result in R3. R3 = X/Y.

For this problem, use X = 0x003D and Y = 0x0005.The value of X is stored at 0x3200 and the value of Y is stored at 0x3201. Your program must start at 0x3000. Submit a screen shot of your code. Also submit a printout of your commented code in assembly language (.asm file)