Important Notes:

- Answers to questions 1 and 2 should be handed to your instructor in hard copy on the due date during your class time.
- Questions 3 and 4 must be submitted electronically to the Learn@UW dropbox before 11.59 AM on the due date. Submit only one archive file, named exactly like: Lastname_Firstname[.zip] to the dropbox folder homework7. You have to submit the .txt files (which contains the assembly code) for these problems and the files should be named hw7_p#.txt, i.e, the file for problem 3 should be named hw7_p3.txt. Since your code is tested automatically, it is important to stick to this naming convention, otherwise you will lose credit, even if your code is working correctly.
  - To give you an example, if Sujith was to submit his code, he will put hw7_p3.txt (assembly code for problem 3) and hw7_p4.txt (assembly code for problem 4) into a folder called surendran_sujith (since surendran is his last name and sujith is his first name). He will then zip this folder, and submit it into homework7 dropbox (before the deadline!).

- The programs which you write should always start at address x3000 and end with a HALT instruction (HALT).

- You can submit your code for problem 3 and 4 as many times as you want, until 1:59 PM on Friday, April 24, 2014. We will consider your latest submission for grading.
Problem 1 (4 points)

Consider the following assembly program:

```
.ORIG x3000
LD R2, INPUT_NUM
LD R3, BITS
AND R0, R0, #0
ADD R1, R0, #1
NEXT AND R4, R2, R1
BRnp SKIP
ADD R0, R0, #1
SKIP ADD R1, R1, R1
ADD R3, R3, #-1
BRzp NEXT
ST R0, RESULT
HALT

INPUT_NUM .FILL x3210
BITS .FILL xF
RESULT .BLKW #1

.END
```

a) (2 Points) Run the program on PennSim and give a brief explanation of what the program does. (i.e., Specify how the value at RESULT relates to value at the INPUT_NUM when the program reaches HALT.)
This will give the number of zeros in the INPUT_NUM and put this value in RESULT.

b) (1 Point) How many times does the instruction at label NEXT execute?
   16

c) (1 Point) What value will be contained in R0 after the execution of the program?
   R0 will contain the number of 0s in the INPUT_NUM
Problem 2 (6 Points)

Consider the following assembly code:

```
.ORIG x4000
LEA R3, INPUT
LD R1, SIZE
ADD R3, R3, R1
LOOP
  LDR R0, R3, 0
  ADD R3, R3, -1
  ADD R1, R1, -1
  BRp LOOP
HALT
INPUT .STRINGZ "Problem_2"
STRING .BLKW 5
SIZE .FILL x12
.END
```

a) (4 Points) In the first pass, the assembler creates the symbol table. Fill in the symbol table created by the assembler for this program (You may extend this table to any number of rows)

<table>
<thead>
<tr>
<th>Symbol Name / Label</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOOP</td>
<td>x4003</td>
</tr>
<tr>
<td>INPUT</td>
<td>x4008</td>
</tr>
<tr>
<td>STRING</td>
<td>x4012</td>
</tr>
<tr>
<td>SIZE</td>
<td>x4017</td>
</tr>
</tbody>
</table>

b) (2 Points) In the second pass, the assembler creates a binary (.obj) version of the program, using the entries from the symbol table. Write the binary code generated for the first two instructions (LEA and LD)

`LEA R3, INPUT = 1110 011 000000111`
`LD R1, SIZE = 0010 001 000010101`
Problem 3 (12 Points)

Assume that you are appointed as a TA for this CS/ECE 252 course in Spring 2016, and assume that 25 students have enrolled in this course. After grading all their exams, you finally decided to use LC-3 for calculating the performance of the class. For doing so, assume you have stored the total marks of each of the students in consecutive memory locations starting at 0x4000.

Write a program in LC-3, which starts at memory location 0x3000, to do the following:

a) (2 Points) Calculate the maximum marks obtained by student and store this value in the memory location corresponding to label “MAX_MARKS” (see template hw7_p4.txt).

b) (2 Points) Calculate the minimum marks obtained by a student and store this value in memory location corresponding to label “MIN_MARKS”.

c) (5 Points) Calculate the average marks of the class and store this value in memory location corresponding to label “AVG_MARKS”. If this turns out to be a fraction, round it off to the next higher integer. For example, if the average turns out to be 30.02, the average should be rounded off to 31.

d) (3 Points) Calculate the number of students who have scored below class average (which has been rounded off) and store this value in memory location corresponding to label “BELOW_AVG”.

Note:
- You MUST use this template for writing your code: hw7_p4.txt.
- You can assume that addition of marks of all the students will not generate overflow.
- You may use the script file (hw7p4_script.txt) for testing your code. The script has some default values for marks for each student. However, please note that we will be finally testing your code with other values for marks (not with the default ones).
  - Before you use the script, make sure that your code is getting compiled successfully.
  - Then you can open Pennsim (if it is closed) and type “script hw7p4_script.txt” in the command line of Pennsim. This script will reset the system, load your code and OS and then tests your code with some default values of marks. If any of the tests fail, you will get a FALSE output on the screen. If all tests are successful, you will see the following 4 true statements (for the default values):
    - TRUE (check MAX_MARKS 100)
    - TRUE (check MIN_MARKS 70)
    - TRUE (check AVG_MARKS 84)
    - TRUE (check BELOW_AVG 10)
.ORIG x3000

CALCULATE_MAX
    LD R1, NUM_STUDENTS ; R1 holds the total number of students in class
    LD R2, MARKS_ADDRESS ; R2 holds the starting address of marks (ie,x4000)
    AND R3, R3, #0
    LOOP1
      LDR R4, R2, #0
      NOT R5, R4
      ADD R5, R5, #1
      ADD R5, R5, R3
      BRp NEXT1
      ADD R3, R4, #0
    NEXT1
      ADD R2, R2, #1
      ADD R1, R1, #1
      BRp LOOP1
      LEA R0, MAX_MARKS
      STR R3, R0, #0

CALCULATE_MIN
    LD R1, NUM_STUDENTS ; R1 holds the total number of students in class
    LD R2, MARKS_ADDRESS ; R2 holds the starting address of marks (ie,x4000)
    LD R3, MAX_MARKS; R3 gets initialized to the maximum marks value
    LOOP2
      LDR R4, R2, #0
      NOT R5, R4
      ADD R5, R5, #1
      ADD R5, R5, R3
      BRn NEXT2
      ADD R3, R4, #0
    NEXT2
      ADD R2, R2, #1
      ADD R1, R1, #1
      BRp LOOP2
      LEA R0, MIN_MARKS
      STR R3, R0, #0

CALCULATE_AVERAGE
    LD R1, NUM_STUDENTS ; R1 holds the total number of students in class
    LD R2, MARKS_ADDRESS ; R2 holds the starting address of marks (ie,x4000)
    AND R3, R3, #0
    AND R6, R6, #0
    GEN_SUM LDR R4, R2, #0
ADD R3, R3, R4 ; Sum will be stored in R3
ADD R2, R2, #1
ADD R1, R1, #-1
BRp GEN_SUM

LD R1, NUM_STUDENTS
NOT R5, R1
ADD R5, R5, 1 ; R5 has value -20
ADD R4, R3, #0
LOOP3  ADD R4, R4, #0
        BRnz DONE_AVG
ADD R4, R4, R5 ; Subtract 20 from total
ADD R6, R6, #1 ; increment everytime
BRnzp LOOP3

DONE_AVG LEA R0, AVG_MARKS
        STR R6, R0, #0

BELOW_AVERAGE

LD R1, NUM_STUDENTS ; R1 holds the total number of students in class
LD R2, MARKS_ADDRESS ; R2 holds the starting address of marks (i.e., x4000)
LD R3, AVG_MARKS ; R3 has the average marks obtained by a student
AND R6, R6, #0 ; Where the total number of students below average is stored

LOOP4  LDR R4, R2, #0
        NOT R5, R4
ADD R5, R5, #1
ADD R5, R5, R3
BRnz NEXT4
ADD R6, R6, #1

NEXT4  ADD R2, R2, #1
        ADD R1, R1, #-1
        BRp LOOP4
        LEA R0, BELOW_AVG
        STR R6, R0, #0

LAST    HALT

; DATA REGION
; >>>>> INSERT ANY DATA YOU NEED HERE >>>>>>>
Problem 4 (8 Points)

Write a program which reads a string of lowercase alphabets starting at label "INPUT". It then checks if the string is a palindrome or not. (A palindrome is a word or phrase that reads the same forward or reversed. For example, “kayak” and “racecar” are palindromes.) If the string is a palindrome, the program should store a value 1 in the memory location corresponding to label “PALINDROME”. Else, it should store a value 0 into this memory location.

Note:

- The string is always terminated by a NULL character (ASCII value 0x0). If the string is empty (the character at label INPUT is the NULL character), the value at PALINDROME should be 0.

- Use this template for writing your code <PROVIDE LINK>:

```
.ORIG x3000

;>>>>>>>>>>>>>>INSERT YOUR CODE BELOW>>>>>>>>>>>>>>>>>>

LEA R0, INPUT ;Grab address of first char in string
LEA R1, PALINDROME ;Grab address of PALINDROME label
ADD R1, R1, -2 ;Subtract 2 to get last char in string

LOOP LDR R2, R0, 0 ;Grab first char in string
LDR R3, R1, 0 ;Grab last char in string
NOT R4, R3
ADD R4, R4, 1
ADD R4, R4, R2
BR np LAST ;If first char and last char don't match, end program
ADD R0, R0, 1 ;Otherwise, grab address of next char from front
ADD R1, R1, -1 ;Grab address of next char from end
NOT R5, R1
ADD R5, R5, 1
ADD R5, R5, R0
BRzp YES ;Test if end of palindrome has been found
BR LOOP

YES AND R6, R6, 0 ;Store 1 if palindrome was found
ADD R6, R6, 1
ST R6, PALINDROME

LAST HALT

;DATA REGION

;>>>>>>>>INSERT ANY DATA YOU NEED HERE>>>>>>>>>
```
INPUT .STRINGZ "abcdcba"
PALINDROME .BLKW 1
.END