Homework 6 – Due 9:55am (both sections) on Monday, April 9th

Primary contact for this homework: Yinggang Huang [yinggang@cs.wisc.edu]

You must do this homework in groups of **two**. This homework must be submitted **online**. No hard copies will be accepted.

Important Notes:

This homework must be submitted electronically to the Learn@UW dropbox. The files to be submitted include **binary code as binary files (*.bin)**, **pseudo-code in text files (*.txt),** and **README.txt** (see the submission guidelines below). **Do not submit files in hex or assembly! Only machine language for LC-3 is accepted for submission.**

Your programs should **always** start at address **x3000** and end with a HALT instruction (0xF025).

README file:

Download the file: README.txt.

Replace GroupMembers, UWIDs, and Section#. Replace ADDRESS with the halt address for the corresponding problem (HALTP1 = halt address for problem 1, HALTP2 = halt address for problem 2, etc.).

Submission Guidelines:

1. Please submit only one compressed or archive file (*.zip or *.tar.gz) per group to the folder **homework6**.
2. Name the archive file with the following format: Lastname1Lastname2 with .zip or .tar.gz as suffix where Lastname1 is the last name with 1st letter capitalized for one of the group members while Lastname2 is the last name for the other.
3. Your archive file should contain the following (**the files MUST be named exactly like this**):
    A. hw6_p1.txt – Pseudo-code for problem 1
    B. hw6_p1.bin – Binary code for problem 1
    C. hw6_p2.txt – Pseudo-code for problem 2
    D. hw6_p2.bin – Binary code for problem 2
    E. hw6_p3.txt – Pseudo-code for problem 3
    F. hw6_p3.bin – Binary code for problem 3
    G. hw6_p4.txt – Pseudo-code for problem 4
    H. hw6_p4.bin – Binary code for problem 4
    I. README.txt - Readme file that contains the names, student IDs, and section numbers for all members of your group and the HALT addresses for all the problems (one HALT address for each problem).
4. You can submit your code and other files as many times as you would like until the due time on the due date indicated above.

Problem 1 (3 points)

Write a short LC-3 program in PennSim that compares the 2 numbers in R1 and R2, and then puts the value 0 in R0 if R1 = R2, 1 if R1 > R2, -1 if R1 < R2. Finally, store the result to memory location 0x4000.

a. (1.5 points) Write the pseudo code for the algorithm. Please submit your pseudo code in a file **exactly** named as "hw6_p1.txt", without the double quotes, to dropbox.

b. (1.5 points) Write an LC-3 program (in machine language) based on pseudo-code from part a. Comment each line of the source code and submit the binary code to dropbox. The file name should be **exactly** "hw6_p1.bin", without the double quotes.

Problem 2 (9 points)

Write an LC-3 program that calculates factorial of n where n is a non-negative number. Use the value stored at memory location 0x4000 for input n and memory location 0x4001 to store the result of the computation. You can implement this based on the multiplication code derived in HW5. Assume the output is represented by 16 bits, the word length of LC-3.

a. (4 points) Write the pseudo code for the factorial algorithm. Please submit your pseudo code in a file **exactly** named as "hw6_p2.txt", without the double quotes, to dropbo

b. (5 points) Write an LC-3 program (in machine language) in PennSim based on pseudo-code from part a. Comment each line of the source code and submit the binary code to dropbox. The file name should be **exactly** "hw6_p2.bin", without the double quotes.

Problem 3 (10 points)

Write an LC-3 machine language program which divides the number in memory location x4000 by the number in memory location x4001 and stores the quotient at x5000 and the remainder at x5001. Assume both dividend and divisor are positive.

a. (3 points) Write the pseudo code. Please submit your pseudo code in a file **exactly** named as "hw6_p3.txt", without the double quotes, to dropbox.

b. (7 points) Write an LC-3 program (in machine language) in PennSim based on pseudo-code from part a. Comment each line of the source code and submit the binary code to dropbox. The file name should be **exactly** "hw6_p3.bin", without the double quotes.

Problem 4 (8 points)

The LC-3 has no left shift or right shift instruction.

Left shifting means that every bit in a binary string is moved to the left and the right bits are filled in with 0s. For example, the binary string 00110010 when left-shifted by one bit moves every bit one position to the left, which results in 01100100. We can shift a number to the left by one bit position by adding it to itself. For example, when the binary number 0011 is added to itself, the result is 0110 which is exactly one bit left-shift from 0011. Note that lowest bits will be filled with zeros.

But how does right-shifting sound? Let's solve it. Similar to left-shifting, right-shifting means that every bit in a binary string is moved to the right and the left bits are filled in with 0's. For example, 1010 after right-shifting by one bit results in 0101. The algorithm we want you to design should take in a 16-bit value at memory location M1 that contains the number that will have its bits shifted and a value at M2 that contains the number of bits to shift. The 16-bit result should be stored in M3. For instance, suppose M1 = 0x0010 and M2 = 0x0004. After execution of the program, M3 = 0x0100. Note that the highest bits will be filled with zeros. You can assume the MSB of the input is always 0 so that the input value to be right shifted is positive. Also assume if the value in M2 is zero or negative, no shift is performed and original 16-bit value should be stored into M3.

a. (3 points) Write the pseudo code for the right shift algorithm as described above. Please submit your pseudo code in a file **exactly** named as "hw6_p4.txt", excluding the double quotes.

b. (5 points) Write a program (in machine language) using LC-3 binary code that implements your algorithm, with the following as your memory locations:

M1: 0x5000

M2: 0x5001

M3: 0x6000

Write comments for each line of code explaining what it does. The binary code named **exactly** "hw6_p4.bin" should be submitted to the dropbox.