ECE 362 HW 6. (60 points)

In this homework, you learn how to write a subroutine, how to pass arguments to the subroutine, and how to call a subroutine in the main program.

Write the programs for the following problems. Try your best to **make them correct and concise**. The only way to make sure they work is to run them in the lab computer. However, it is not required.

1. (5 pts) (Passing argument using registers and using a for-loop) Write a subroutine to compare two equal sized character strings. The number of characters in the string is in register B. The starting address of one string is in register X. The starting address of the other string is in register Y. If the strings are equal, the subroutine returns a $11 in register B. Otherwise, the subroutine returns a $22 in register B.

2. (5 pts) (Passing argument using registers and using a while-loop) Write a subroutine to compare two character strings. The last character of each string is indicated by 0. The starting address of one string is in register X. The starting address of the other string is in register Y. If the strings are equal, the subroutine returns a $33 in register B. Otherwise, the subroutine returns a $44 in register B.

3. (5 pts) Write the main program that calls the subroutine developed in problem 2.

4. (10 pts) (Bit checking and manipulation) Write a subroutine that finds the average of an N element array of 16-bit signed integers. The starting address of the array is in register X and N is in register Y. If the result is even, turn on LED connected to port S bit 0 and off LED connected to port S bit 1; otherwise turn off LED connected to port S bit 0 and on LED connected to port S bit 1. If the result is positive, turn off LED connected to port S bit 7; otherwise turn on LED connected to port S bit 7. Assume no overflow occurs.

5. (5 pts) Write a main program to call subroutine defined in problem 4 500 times.

6. (10 pts) (Passing argument using stack) Write a subroutine to compare two character strings. The last character of each string is indicated by 0. The starting address of one string is pushed first in the stack. The starting address of the other string is pushed next in the stack. If the strings are equal, the subroutine returns a $77 in register B. Otherwise, the subroutine returns a $88 in register B.

7. (5 pts) Write a main program that calls the subroutine developed in problem 6.

8. (10 pts) <u>Write a subroutine</u> that performs the following tasks (1) receive two 8-bit input arguments, (2) create a byte-size local variable, (3) clear bits 4 and 1 of the first argument, (4) set bits 7 and 3 of the second argument, (5) and the arguments together, (6) store the result on the created local variable, and (7) return the result back. All arguments and the result between the calling program and the subroutine are **passed using stack**. Do not use any global variable.

9. (5 pts) Write a main program that calls the subroutine developed in problem 8, and use the result.