

# Computer Science 210 –Computer Organization

## Homework Exercise 7

Due on Github by 11:59 PM, Monday 21 March

Your turn in will consist of one assembly language source file named **factorial.asm**. You must work with a partner that you have not previously worked with.

Unlike your earlier programs, this program should make maximum use of subroutines. Several of these, such as **MUL**, **DIV**, and **FACTORIAL**, are available in the **factorial.asm** file in **project7.zip**. You may assume that none of the numeric operands will be negative.

Be sure to include comments that describe what each routine does and its input and/or output parameters. Check the existing routines and the style guidelines for models of documentation if you're unsure about it.

Your subroutines should communicate with their callers via registers only. Any data variables used in a subroutine should not be used in another subroutine or in the main program. Likewise, no subroutine should use the main program's data variable. Registers should be backed up and restored when appropriate. You should use the stack resource for register backup.

The program includes a stack management facility to support subroutine calls. Read the code and note its organization. Note how the stack is used to back up and restore registers, resulting in a minimal data area for each routine.

Assemble and run the program. Run it first with the value 5 for **NUMBER** and observe the changes to **NUMBER**, **QUOTIENT**, and **REMAINDER**. Do the same with the value 4 for **NUMBER**.

1. You will now make this program truly interactive, by allowing the user to input a number and display the resulting factorial of that number in the LC3 console. Add routines to the program to input a string from the keyboard, convert a string of digits to an integer, and convert an integer to a string of digits. The new routines are named **GETS**, **INT**, and **TOSTRING**. All of these functions were discussed in lecture. You wrote most of the code for the input function in the last assignment. Be sure to thoroughly document each routine with a register dictionary, pseudocode, and appropriate comments. Remember to use the stack resource to back up registers when needed.
2. Modify the main program code so that it prompts the user for a number and computes and prints the factorial of that number. Here is a pseudocode for this main program process:

```
buffer = input("Enter a number: ")
number = int(buffer)
result = factorial(number)
print("The factorial of", str(number), "is", str(result))
```