

CS220 - Computer System II  
Lab 1

**Due: 09/07/2016, 11:59pm**

# 1 Introduction

In this lab, you will write a function to calculate the factorial of a number.

## 2 Getting Started

Download lab1.tar.gz from blackboard and extract the contents (refer to lab0 if you don't know how to extract.) You will find prog\_lab.c, Makefile, main.o, prog\_lab.h, input1.txt. Your code will reside in prog\_lab.h and prog\_lab.c. So, these are the 2 files you will be modifying. Sample inputs for factorial will be provided in input1.txt. The file main.o contains the code required to read the input, invoke fact function and generate the output. You do not need to modify it.

## 3 Factorial of a number

Factorial of a non-negative integer  $n$ , denoted by  $n!$ , is the product of all positive integers less than or equal to  $n$ . For example,  $4! = 4 \times 3 \times 2 \times 1 = 24$ .  $0!$  is defined to be 1.

**Step 1:** Open file `prog_lab.h` in your text editor. Insert a comment that reads, "This function computes the factorial of a given number". After the comment, declare the prototype for the function as:

```
1 int fact(int n);
```

Save and close `prog_lab.h`.

**Step 2:** Next, we must provide a definition for "fact" function. Open `prog_lab.c` file in your text editor to add the definition. *Strategy:* Start by declaring the result to be 1 because factorial of a number can not be less than 1.

```
1 int result;  
  result = 1;
```

If  $n$  is 0 or 1, then we just return result, because  $0!$  is defined to be 1, and result is already set to 1.

```
if(n <= 1) return result;
```

Count from 1 up to n (or start from n down to 1,) and in each iteration, multiply result by that number and update the result. That is:

```
1 while (count > 1) {  
    /* Multiply Update the value of result to be the product of result and  
       count. */  
3    /* Decrement count */  
}
```

Finally, return the result.

**Step 3:** Build and test your code. In the terminal window, run make.

```
$ make
```

This should generate program lab1. Open input1.txt in a text editor and enter values for n, one in each line. Then, run lab1.

```
1 $ ./lab1
```

This will generate output\_input1.txt with the results. The value in the 1st line of result corresponds to the 1st input, value in the 2nd line corresponds to the second input, and so on.

## 4 Submitting the result

Run lab1 with the following values in input1.txt. Submit the corresponding output\_input1.txt file to Blackboard.

```
4  
7  
0  
9  
6
```