# Description

The purpose of this lab is to check your understanding of the content in lecture 3a and 3b, Functions and an Introduction to Recursion.

We will focus on the material from chapter 6 of the book. The objectives are:

* To construct programs modularly from functions.
* To use common math functions available in the C++ Standard Library.
* To create functions with multiple parameters.
* The mechanisms for passing information between functions and returning results.
* How the function call/return mechanism is supported by the function call stack and activation records.
* To use random number generation to implement gameplaying applications.
* How the visibility of identifiers is limited to specific regions of programs.

# Part 1: Questions

Complete the assignment in the module *Topic 3b Lab: Functions and an Introduction to Recursion* named, "Lab 3a&b Questions", in Canvas.

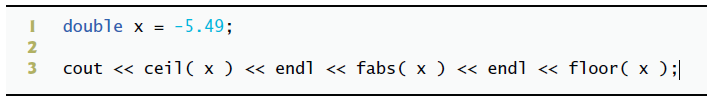
# Part 2: Activity

This part of the lab should be submitted using the Lab 3b Activity assignment in the *Topic 3b Lab: Functions and an Introduction to Recursion* module.

**You should complete the non-programming parts of this lab using this document.** **It is possible that some of the images and answer-boxes below might move or need to be resized while using them. Do your best to make your final document neat and organized.**

## Problem 1: Program Output

For each of the given program segments, read the code and write the output in the space provided below each program. [*Note:* Do not execute these programs on a computer.]

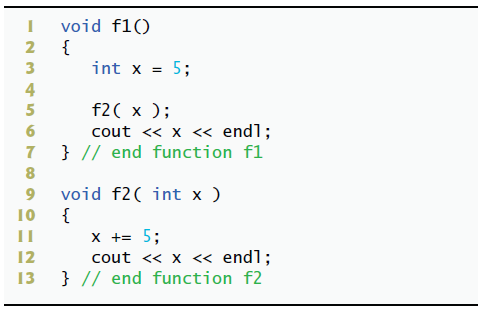
1. What is output by the given program segment? Assume that the <cmath> header file has been included.  
   

**Answer:**

**-5.0**

**5.49**

**-6.0**

1. What is output by the following program segment when function f1 is called?  
   

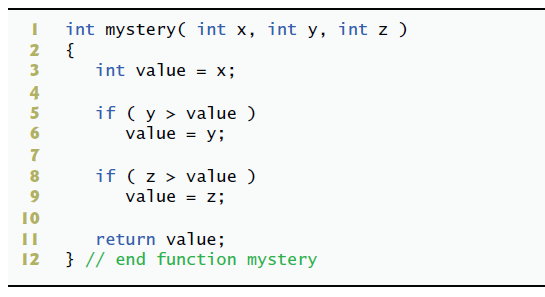
**Answer:**

**10**

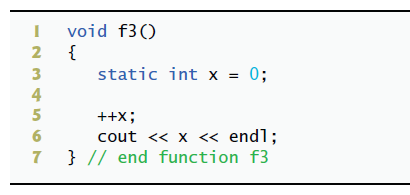
**5**

1. What is the output of

cout << mystery( 6, 2, 5 ) << endl;

assuming the following definition of mystery?  


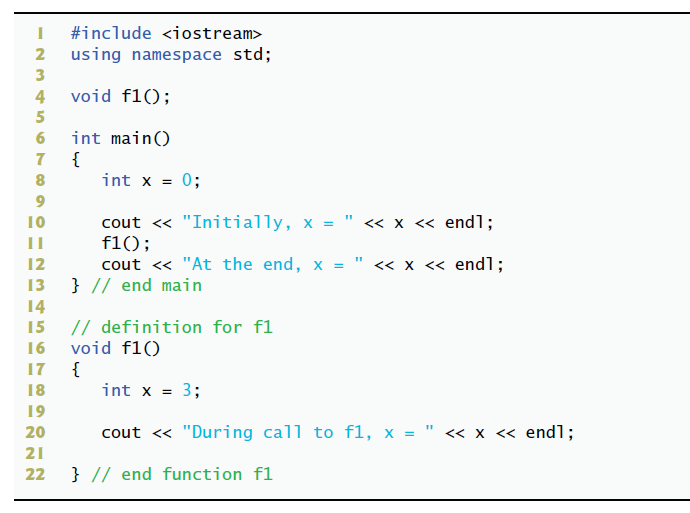
**Answer: 6**

1. What is output by the following program segment when function f3 is called twice?  
   

**Answer:**

**1**

**2**

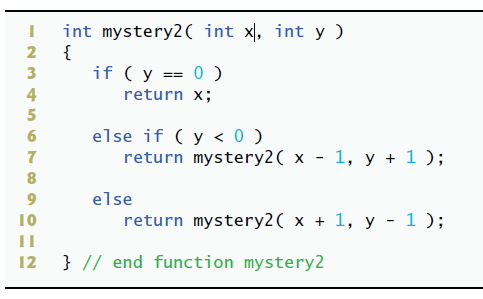
1. What is output by the following program?  
   

**Answer: Initially, x = 0**

**During call to f1, x = 3**

**At the end, x = 0**

1. What is the output of :

cout << mystery2( 5, 4 ) << endl;  
assuming the following definition of mystery2?

**Answer: 9**

1. What is the output of

cout << mystery2( 5, -4 ) << endl;

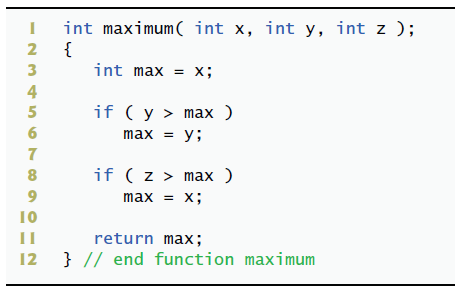
assuming the same definition of mystery2

**Answer: 1**

## Problem 2: Correct the Code

For each of the given program segments, determine if there is an error in the code. If there is an error, specify whether it is a logic or compilation error, circle the error in the program, and write the corrected code in the space provided after each problem. If the code does not contain an error, write “no error.” [Note: It is possible that a program segment may contain multiple errors.]

**Important: do not add functions or includes. Simply rewrite the code without the errors.**

1. The following program segment defines function maximum, which returns the largest of three integers:  
   



**Answer: There are two errors in this code, the first one is a compilation error because we cannot have a semicolon trailing the declaration of a function. The second is a logical error, because it appears the intended expression would have been max = z;**

**int maximum( int x, int y, int z ) {**

**int max = x;**

**if (y > max)**

**max = y;**

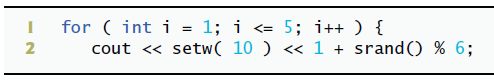
**if (z > max)**

**max = z;**

**return max;**

**}**

1. The following program segment should output five random numbers in the range from 1 to 6, inclusive:



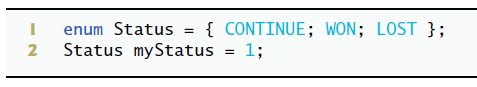


**Answer: Compilation error, srand() is what we use to seed the random generator rand().**

**for (int i = 1; i <= 5; i++ ) {**

**cout << setw(10) << 1 + rand() % 6;**

**}**

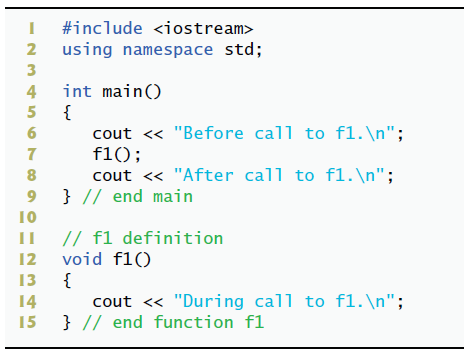
1. The following program segment creates an enumeration (Status), creates a variable of the enumeration’s type and sets it to WON:  
   

**Answer: Compilation error, formatting incorrect**

**enum Status { CONTINUE, WON, LOST };**

**Status myStatus = WON;**

1. The following program should display three lines of text:



**Answer: Compilation error, there should be a declaration of the prototype function f1 before the main function.**

**#include <iostream>**

**using namespace std;**

**void f1();**

**int main() {**

**cout << “Before call to f1.\n”;**

**f1();**

**cout << “After call to f1.\n”;**

**}**

**void f1() {**

**cout << “During call to f1.\n”;**

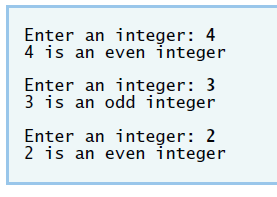
**}**

## Problem 3: Programming Exercises

Write these programs using C++. Your output must be consistent with the examples provided.

1. Write a program called, **"even.cpp"**, that takes as user-input a series of integers and passes them one at a time to function even, which uses a bit mask to determine whether an integer is even. The function should take an integer argument and return true if the integer is even and false otherwise. The mask will be the value 1 and should be bit-wise ANDed with the value entered to mask all but the least significant digit.

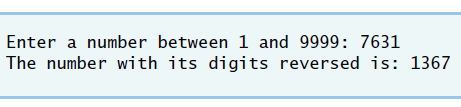
Sample output:



1. Write a program called, "reverse\_digits.cpp", with a function called, "printReverseDigits ". The function takes an integer value and prints the number with its digits reversed. For example, given the number 7631, the function should print 1367. Your program should take user input and print its reverse until the user enters a sentinel value (like 0).

\*\*\* Important, **you must use RECURSION for this.** Also, printReverseDigits should take its arguments pass-by-reference but use safety.

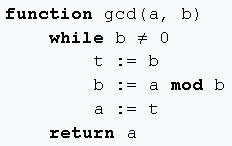
Sample output:



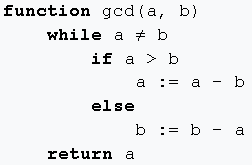
1. The greatest common divisor (GCD) of two integers is the largest integer that evenly divides into each of the two integers. Write a program called, "gcd.cpp", that tests a function called gcd which uses the following Euclidean algorithm and returns the greatest common divisor of two integers.

**\*\*\*Important, these algorithms show a loop, but you must write it using RECURSION.**

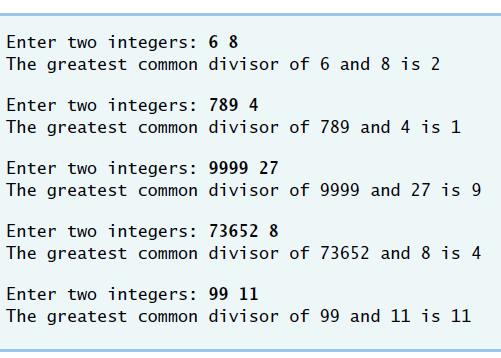
**This program should t**ake in a sentinel value (like 0) to stop the program. This behavior is not shown in the sample output. Note, this pseudocode requires a > b.



**OR**



Sample output:



# What to Submit for Lab 3b Activity

* This document completed.
* The source code file called, "even.cpp", from programming exercise 1 .
* The source code file called, "reverse\_digits.cpp ", programming exercise 2.
* The source code file called, "gcd.cpp ", programming exercise 3.
* A screenshot of the output for *each program* (three separate screenshots).

Note: your screenshots must include the entire Visual Studio Code window. Do not include your desktop or anything else in the image. Do not take a picture with a camera or phone, use your computer to create a screenshot.