50 pts

Name 1: ____________________________________

Name 2: ____________________________________

Class Day / Time: ____________________________________

Due Date: ____________________________________

Lab #11: Intro to Recursion - GCD

For this lab you need to write a program that will read in two values from a user and output the greatest common divisor (using a recursive implementation of the Euclidean algorithm) to a file. In Lab #3, you implemented this program using an iterative method.

Greatest Common Divisor
In mathematics, the greatest common divisor (GCD) of two or more integers (when at least one of them is not zero) is the largest positive integer that divides the numbers without a remainder. If one of them is zero then the larger value is the GCD.

Euclidean Algorithm
The Euclidean algorithm works by performing successive long divisions of two numbers that are unequal. Based on the result of the division of the larger number divided by the smaller number, the algorithm will either terminate and return the GCD, or will perform another long division using the smallest of the two original values as the numerator, and the remainder of the calculated long division as the divisor. The algorithm continues dividing and updating the numerator and divisor until the remainder of the division is 0.

Recursion
The Euclidean algorithm can be solved using iteration (as in Lab#3: Intro to Functions – GCD) or by recursion. Recursion is defined as a function repeatedly calling itself. Recursion is similar to looping in that the parameters serve as loop control variables, the parameters are initialized outside of the function by the initial function call, the same body of code is executed repeatedly, each call/iteration advances the solution toward completion and changes the parameter(s) for each subsequent call. However, the control flow between looping and recursion differ. When looping, the control flow of the loop structure does not leave the procedure/method it is currently executing in; whereas with recursion, looping is performed by successive (recursive) function calls where each recursive call invokes a new instance of the procedure on the call stack, and hence loops through the same body of code. A recursive function is usually defined as having one or multiple base-case(s), which terminates the recursion, and one or more inductive or recursive case(s). As a general rule, it is **important** to test base cases before making recursive calls to avoid infinite recursion.

A summary of a recursive Euclidian algorithm & example follow:

$$\text{For } m \geq n > 0,\ \text{gcd}(m,n) = \begin{cases} \text{Base case: If n divides m with no remainder, return n} \\ \text{Recursive Case: Otherwise, return gcd(n, remainder of m/n)} \end{cases}$$

From the calling procedure: 1) Initialize: Make the initial call	**Ex:** `gcd(30, 12);`
Once inside the function: 2) Check the Base Case: if n divides m with no remainder, return n 3) Change: Update Parameters & Make Recursive Call	**Ex:** `gcd(12, remainder of 30/12);`

For example:
Let's say we want to find the GCD of 74 & 32.
We would first divide 74 and 32.
74 / 32 = 2 r 10

Next we take the smaller number (32) and divide it by the remainder (10).
32 / 10 = 3 r 2

Again we take the smaller number (10) and divide it by the new remainder (2). We repeat this process until the remainder is 0.
10 / 2 = 5 r 0

Once the remainder is 0 we stop and our GCD is the last non-zero remainder, which in this case is 2.

For the GCD write a function to read in the two values, a function to calculate the GCD, and a function to output the results. Have the code run 4 times.

Test with the following inputs:

72, 32
99, 30
48, 18
12, 0

Screen INPUT/OUTPUT - should be formatted as follows –

```
Enter the first integer:   72
Enter the second integer: 32

Enter the first integer:   99
Enter the second integer: 30

…

Thank you for using my GCD calculator!
```

OUTPUT File format -

```
The GCD for 72 & 32 = 8

The GCD for 99 & 30 = 3
…
```

Turn in as a single PDF file (IN THIS ORDER)

1. Screen I/O - cut and pasted to a txt file within eclipse and output
2. Output File
3. Header File
4. Main - documented according to the requirements & printed from eclipse
5. Functions (in order in which they are called) - documented according to the requirements & printed from eclipse

```
****************************************************
*   PROGRAMMED BY : Carlos Aguilera
*   CLASS         : CS1B
*   SECTION       : MW: 7:30p - 9:50p
*   LAB #11       : Recursion GCD
****************************************************

Enter the first integer:  72
Enter the second integer: 32

Enter the first integer:  99
Enter the second integer: 30

Enter the first integer:  48
Enter the second integer: 18

Enter the first integer:  12
Enter the second integer: 0

Thank you for using my GCD calculator!
```

```
The GCD for 72 & 32 = 8
The GCD for 99 & 30 = 9
The GCD for 48 & 18 = 12
The GCD for 12 & 0 = 0
```

```cpp
#ifndef HEADER_H_
#define HEADER_H_

#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>

/**************************************************************************
 * heading
 --------------------------------------------------------------------------
 * FUNCTION: this function prints out heading
 * @param
 * RETURNS - void
 *************************************************************************/
void heading();

/**************************************************************************
 * gcd
 --------------------------------------------------------------------------
 * FUNCTION: This is a recursive function that takes in two values and finds the
 * greatest common divisor for them
 * @param m value being divided
 * @param n value to divide by
 * RETURNS - int
 *************************************************************************/
int gcd(int m, int n);

#endif // HEADER_H_

/**************************************************************************
 * AUTHOR      : Carlos Aguilera
 * STUDENT ID  : 1152562
 * LAB #11     : Recursion GCD
 * CLASS       : CS1B
 * SECTION     : M-W
 * DUE DATE    : 04.25.22
 *************************************************************************/

/*************************************************************************
 * GCD Recursion
 * -----------------------------------------------------------------------
 * PROGRAM: This program is a GCD calculator using recursion its fairly simple
 * so ill let you go through the code
 * -----------------------------------------------------------------------
 *************************************************************************/

#include "../include/header.h"

int main() {
  std::fstream oFile;
  int m, n;
  oFile.open("OutputFile.txt", std::ios::out);

  heading();
  for (int i = 0; i < 4; i++) {
    std::cout << std::left;
    std::cout << std::setw(26) << "Enter the first integer:";
    std::cin >> m;
    std::cout << std::setw(26) << "Enter the second integer:";
    std::cin >> n;
    std::cout << "\n";
    std::cout << std::right;
    oFile << "The GCD for " << m << " & " << n << " = " << gcd(m, n) << "\n";
  }
  std::cout << "Thank you for using my GCD calculator!\n";
  oFile.close();
  return 0;
}

#include "../include/header.h"

void heading() {

  /*************************************************************************
   * CONSTANTS
   * -----------------------------------------------------------------------
   * OUTPUT - USED FOR CLASS HEADING
   * -----------------------------------------------------------------------
   * PROGRAMMER : Programmer's Name
   * CLASS      : Student's Course
   * SECTION    : Class Days and Times
   * LAB_NUM    : Lab Number (specific to this lab)
   * LAB_NAME   : Title of the Lab
   ************************************************************************/
  const char PROGRAMMER[] = "Carlos Aguilera";
  const char CLASS[]      = "CS1B";
  const char SECTION[]    = "MW: 7:30p - 9:50p";
  const int LAB_NUM       = 11;
  const char LAB_NAME[]   = "Recursion GCD";

  std::cout << std::left;
  std::cout << "****************************************************\n";
  std::cout << "*   PROGRAMMED BY : " << PROGRAMMER << std::endl;
  std::cout << "*   " << std::setw(14) <<"CLASS" << ": " << CLASS << std::endl;
  std::cout << "*   " << std::setw(14) <<"SECTION" << ": " << SECTION << std::endl;
  std::cout << "*   LAB #" << std::setw(9) << LAB_NUM << ": " << LAB_NAME << std::endl;
  std::cout << "****************************************************\n\n";
  std::cout << std::right;
}
```