Recursive Functions

Goals

- Create and use recursive functions

In this lab, we will write a program that uses three recursive functions.

Requirements:

Important: You must use the array for this lab, no vectors allowed.

1. **First Recursive Function**

Write a function that recursively **prints a string in reverse**. The function has **ONLY** one parameter of type string. It **prints** the reversed character to the screen followed by a newline character.

Example: Input of “Hello, world” should output “dlrow ,olleH(newline)”.

Note: Your recursive function just needs to **print** the reverse of the string rather than actually reversing the string itself.

2. **Second Recursive Function**

Write a function that recursively **calculates the sum of an array of integers**. The function has 2 parameters:

- A pointer to the integer array
- An integer for the number of elements in the array.

The function must use a recursive call to sum the value of all integers.

3. **Third Recursive Function**

Write a function that recursively **calculates the triangular number** of an **integer N**. You can set an up-limit for N. You don’t need to handle extra large integers.

The function has one parameter that take in integer N.

Example: If the integer N is 3, the function should output the triangular number 6, since 1 + 2 + 3 = 6.

For more information on triangular number:
https://en.wikipedia.org/wiki/Triangular_number

(Links to an external site.)

Menu

Your program needs to demonstrate all three functions by providing a menu.

The menu should provide user choices to **select one of the three functions to call**, after prompting user input for function call and the function outputs results, the menu should **go back to the first menu** to let the user continue choosing functions to call.

If user choose **function #1**, the menu should prompt the user to enter a **string** and your program reversely prints the string

Note: Use getline() in standard library so the input takes space characters.

If user chooses **function #2**, the menu should first prompt the user an **integer input for the number of integers** in the array, then **a series of integers to fill the array**. Afterwards, the program prints sum of the array of integers.

If the user chooses **function #3**, the menu should prompt the user **an integer**, then the program prints the triangular number of that number.

In addition to the 3 function options inside the first menu, the menu must also provide the **option to quit the program**. You can add input validation functions into these the menu to make it robust.

File organization

You can put all your recursive functions in a single .cpp file or separate them into different .cpp files. However, you must separate the implementation and declaration in .cpp and .hpp files. **Note:** You do not need to make a class for this lab.

What you need to submit

- All the program files including header and source files (.cpp/.hpp)
- Makefile

Important: Put all the files in a single .zip file and submit it on Canvas.

Grading

- Programming style – 10%
- Implement the function to reversely print string – 20%
- Implement the function to calculate the sum of the array – 25%
- Implement the function to calculate the triangular number – 25%
- Implement a program with a menu to call each function or to exit – 20%