

## Lab 8: Recursion

This is a brief lab to help you practice recursion. Recursion is the idea of making a function call itself from inside itself. This has the effect of making the function go into a series of digressions into itself over and over again.

In this lab you will write a function that calculates the Fibonacci sequence. The Fibonacci sequence looks like this:

0, 1, 1, 2, 3, 5, 8, 13, 21

The sum of the previous two values determines the next value in the sequence. Notice how the first and second numbers in the sequence (0 and 1) are summed together to reveal the 3<sup>rd</sup> value which is 1. Notice then that the 2<sup>nd</sup> and 3<sup>rd</sup> values are summed together to get 2 and the 3<sup>rd</sup> and 4<sup>th</sup> value are summed together to get the next value which is 3.

Your function should take an integer as input representing how many numbers in the Fibonacci sequence to display and then calculate and print all of the numbers.

You should write two versions of this function, one named `fib()` and another named `fibR()`. The function `fib()` should be written without recursion using loops. `fibR()` should be written using recursion but should produce the same exact output as `fib()`. The return type of both functions should be void and they should take an integer as input.

Your main function should ask the user how many numbers in the Fibonacci sequence they would like to see. Next it should ask the user if they would like to use the ordinary `fib()` function or the `fibR()` function. Based on their response, one or the other function should be called with either passing the integer value representing how many numbers in the sequence to display as input.

Your program ask both questions over and over again until the user types a 'q' when they are being asked which function they would like to use.

Before the program ends, it should say goodbye to the user.