Lab 9

Comp 11 - Summer Session — Recurse, Recurse, Recurse

9.1 Description

In this lab we will implement factorial with and without recursion.

Our objectives are the following:

- Write a function with a function definition: *int factorial(int n)*. You will implement this solution iteratively.
- Write a function with a function definition: *int recFactorial(int n)*. You will implement this solution recursively.

9.2 Files

You may use the following code to help get you started.

```
#include <iostream> // library for std::cout

// Your function definitions are here for you.
int factorial(int n){
    //
    // Your code here for the iterative solution
    //
    // What is the return type?
}

int recFactorial(int n){
    //
    // Your code here for the recursive solution
    //
    // What is the return type?
}

// Entry point to the program.
```

```cpp
int main(){

    // print out our results
    // Note both results should match!
    std::cout << factorial(4) << "\n";
    std::cout << recFactorial(4) << "\n\n";

    std::cout << factorial(5) << "\n";
    std::cout << recFactorial(5) << "\n\n";

    std::cout << factorial(6) << "\n";
    std::cout << recFactorial(6) << "\n\n";

    std::cout << factorial(7) << "\n";
    std::cout << recFactorial(7) << "\n";

  return 0;
}
```

Listing 9.1: Factorial Starter Code

9.3 Output

9.4 Refresher

Remember that all recursive programs have a base case. Start with the base case (i.e. the smallest solvable problem). After that, the second part of factorial will then be thinking about how to make the problem smaller.

9.5 Submission

```
provide comp11 lab9 your_file.cpp README
```

Listing 9.2: Submit Assignment

9.6 Going Further

Did you enjoy this lab? Want to try out some additional commands to go further?

- Try writing factorial as a lambda.