Lab 13: Recursive Functions

Recursive functions are functions that call themselves. (You can also have a series of functions that form a recursive loop - function A calls function B, function B calls function C, ... , until the last function calls function A again.) To avoid an infinite series of function calls recursive functions must have a ‘stopping condition’ in addition to a ‘recursive call’. The stopping condition makes sure that eventually the series of function calls halts. Without a stopping condition the program will run out of memory, typically resulting in a segmentation fault.

Exercise 13.1

The factorial of a natural number N (written as N!) is

\[ N! = N \times (N - 1) \times (N - 2) \times ... \times 1 \]

it can also be defined recursively as:

\[
0! = 1 \\
1! = 1 \\
N! = N \times (N - 1)!
\]

Write two functions to calculate factorials. The first function should be recursive and should not use a loop; the second function should use a loop and should not be recursive.

Use the time command (time ./a.out) to determine which function is faster. You may need to calculate the factorial of a fairly large number to detect a difference. To determine which function is faster you will need to ‘hard code’ N, not allow the user to input it. If the program waits for user input, it will throw off the time measurement.

Turn in your code and output showing that both functions correctly calculate factorials. In addition, report which of the functions is faster and which of the functions is shorter.

Exercise 13.2

The greatest common divisor \( GCD(a, b) \) of two positive integers \( a \) and \( b \), is the largest divisor common to both \( a \) and \( b \). For example, \( GCD(3, 5) = 1 \), \( GCD(15, 90) = 15 \), and \( GCD(126, 350) = 14 \).

An iterative function for calculating the GCD is:
int GCD( int a, int b )
{
    while( a != b )
    {
        if( a > b)
            a = a - b;
        else
            b = b - a;
    }
    return a;
}

A recursive algorithm for GCD

\[
\begin{align*}
GCD(a, b) &= a \quad \text{if } a = b, \\
GCD(a, b) &= GCD(a - b, b) \quad \text{if } a > b, \\
GCD(a, b) &= GCD(a, b - a) \quad \text{if } a < b.
\end{align*}
\]

Use the time command (\texttt{time ./a.out}) to determine which function is faster. You may need to calculate the GCD of two fairly large number to detect a difference. To determine which function is faster you will need to ‘hard code’ \(a\) and \(b\), not allow the user to input them.

Turn in your code and output showing that both functions correctly calculate the GCD. In addition, report which of the functions is faster and which of the functions is shorter.