EXPERIMENT NO 13

Recursion in C++

Objectives:

In this lab students will learn:

- The concept of recursion.
- Basic guidelines in writing recursive functions.
- How recursion is implemented.
- Comparison between recursion and iteration.

Equipment required:

- Dev-C++/Eclipse/Visual Studio installed in PC/Windows

DISCUSSION

1. Pre-Lab

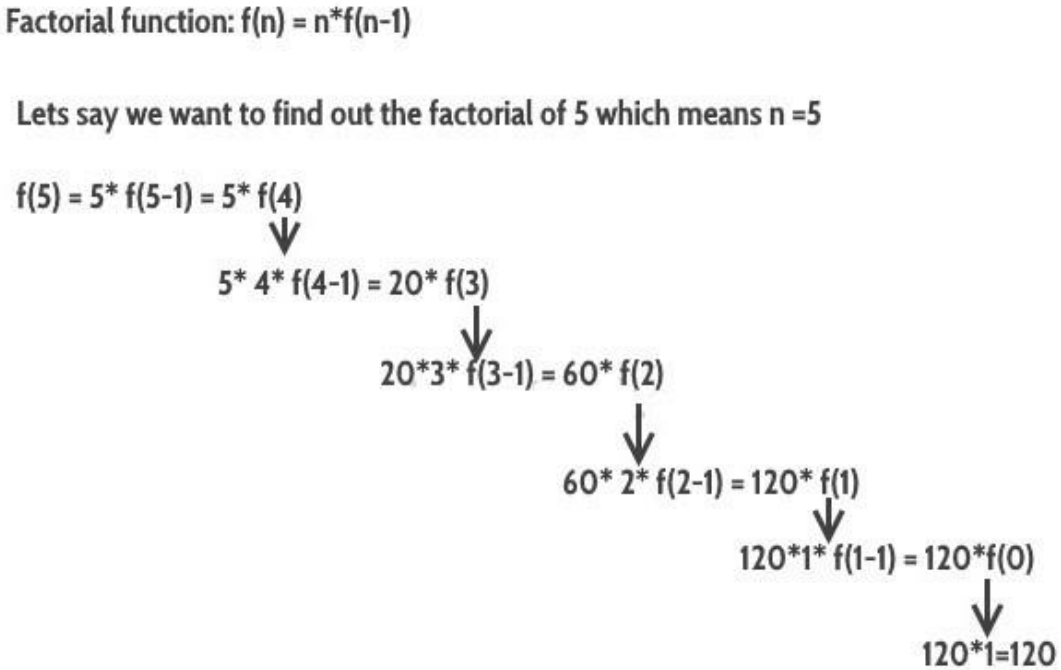
Recursion

The process in which a function calls itself is known as recursion and the corresponding function is called the recursive function. The popular example to understand the recursion is factorial function.

Factorial function is f(n) = n * f(n-1) and its base condition: if n<=1 then f(n) = 1. Base condition is used to stop the recursion.

In the following diagram, I have shown that how the factorial function is calling itself until the function reaches to the base condition.

Example



The purpose of recursion is to divide the problem into smaller problems till the base condition is reached.

Why Stack Overflow error occurs in recursion?
If the base case is not reached or not defined, then the stack overflow problem may arise. The function will keep on calling itself infinite times.

How is memory allocated to different function calls in recursion?
When any function is called from main (), the memory is allocated to it on the stack. A recursive function calls itself, the memory for a called function is allocated on top of memory allocated to calling function and different copy of local variables is created for each function call. When the base case is reached, the function returns its value to the function by whom it is called, and memory is de-allocated, and the process continues.

2. Post-Lab (Lab Tasks)

1. Write a recursive function that computes sum of all numbers from 1 to n-1, where n is given as parameter?

2. Write a C++ Program to Generate Fibonacci Sequence. The Fibonacci sequence is a series of numbers where a number is found by adding up the two numbers before it. Starting with 0 and 1, the sequence goes 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, and so forth. You should print n numbers in a Fibonacci sequence using **Recursion**

3. Write a recursive function that computes sum of all even and odd numbers from 1 to n, where n is given as parameter?

END