# **WHY IS IT A BAD IDEA TO USE RECURSION METHOD TO FIND THE FIBONACCI OF A NUMBER?**

What is the Fibonacci series;

The Fibonacci series is a sequence of numbers i.e. 0 1 1 2 3 5 8 13 21… Where the next element in the sequence is a sum of the current element and the element before it.

The C code snippet below shows how the Fibonacci sequence can be derived using recursion;

#include<stdio.h>

int fib(int n)

{

   if (n <= 1)

      return n;

   return fib(n-1) + fib(n-2);

}

Why it is not advisable to use recursion to derive the Fibonacci series;

This is due to the nature of a recursive function, a recursive function in programming is a function that calls on itself until it reaches the simplest form of itself, in the case of the code above ”fib(1)” and “fib(0)” can be regarded as the simplest versions of the recursion function. Afterwards the function generates its solution by building back up from its simplest form to the required level i.e. fib(n).

This process yet relatively simple to understand has a time complexity (speed of an algorithm) of O(2n), which is an exponential time complexity, meaning the time required to generate the required result increases exponentially with an increase in the value of “n”. But when compared to the time complexity of a Fibonacci series derived using dynamic programming (loops), which is O(n) (linear), It is easy to understand why the recursive method is considerably slower.

The recursive method also has a space complexity of O(n) while that of dynamic programming has a space complexity of O(1), meaning the recursive method also takes up extra space.

Therefore, if speed and space optimization is what you require which in turn is always better in the long run, the recursive method for deriving the value of an element in the Fibonacci series is not advisable to use.