**Experiment No: 3b.**

**Title :**Fibonacci series using recursion in C++.

**Problem Statement :** Implementing Fibonacci series of any number using recursion.

**Algorithm :**

1. Start
2. Declare an integer variable and a function fibonacci ()
3. Make a function call to fibonacci by sending the value whose Fibonacci number is to be found
4. Within Fibonacci function if the integer is equal to 1 return 1 else call the function by sending the value (n-1) and (n-2) and add both.
5. Stop

**Code :**

#include<iostream>

using namespace std;

int fibo(int n)

{

if(n<=1)

{

return(1);

}

else{

return((fibo(n-1)+fibo(n-2)));

}

}

int main()

{

int n,x;

cout<<"Enter the sequence end range : ";

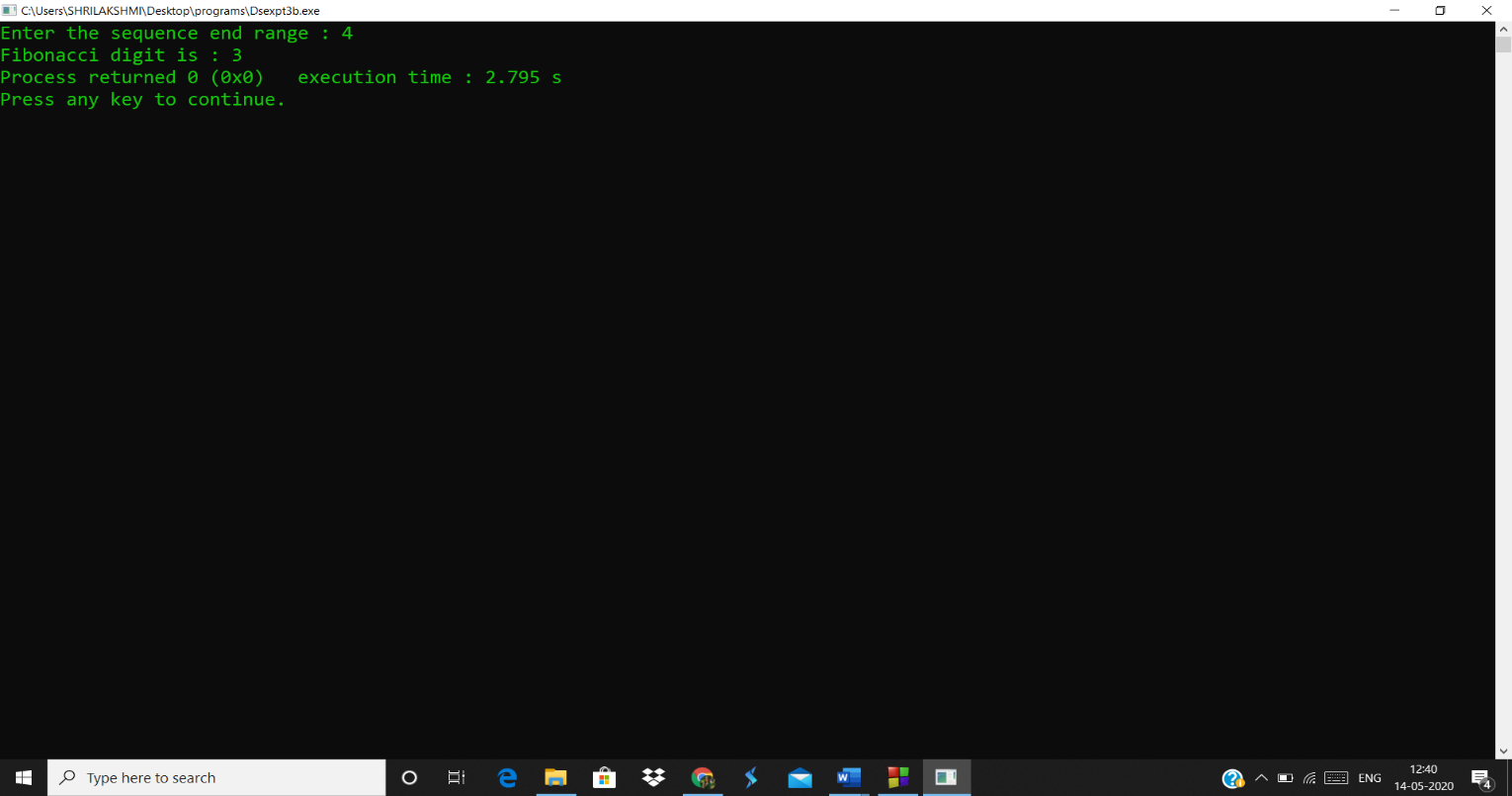
cin>>n;

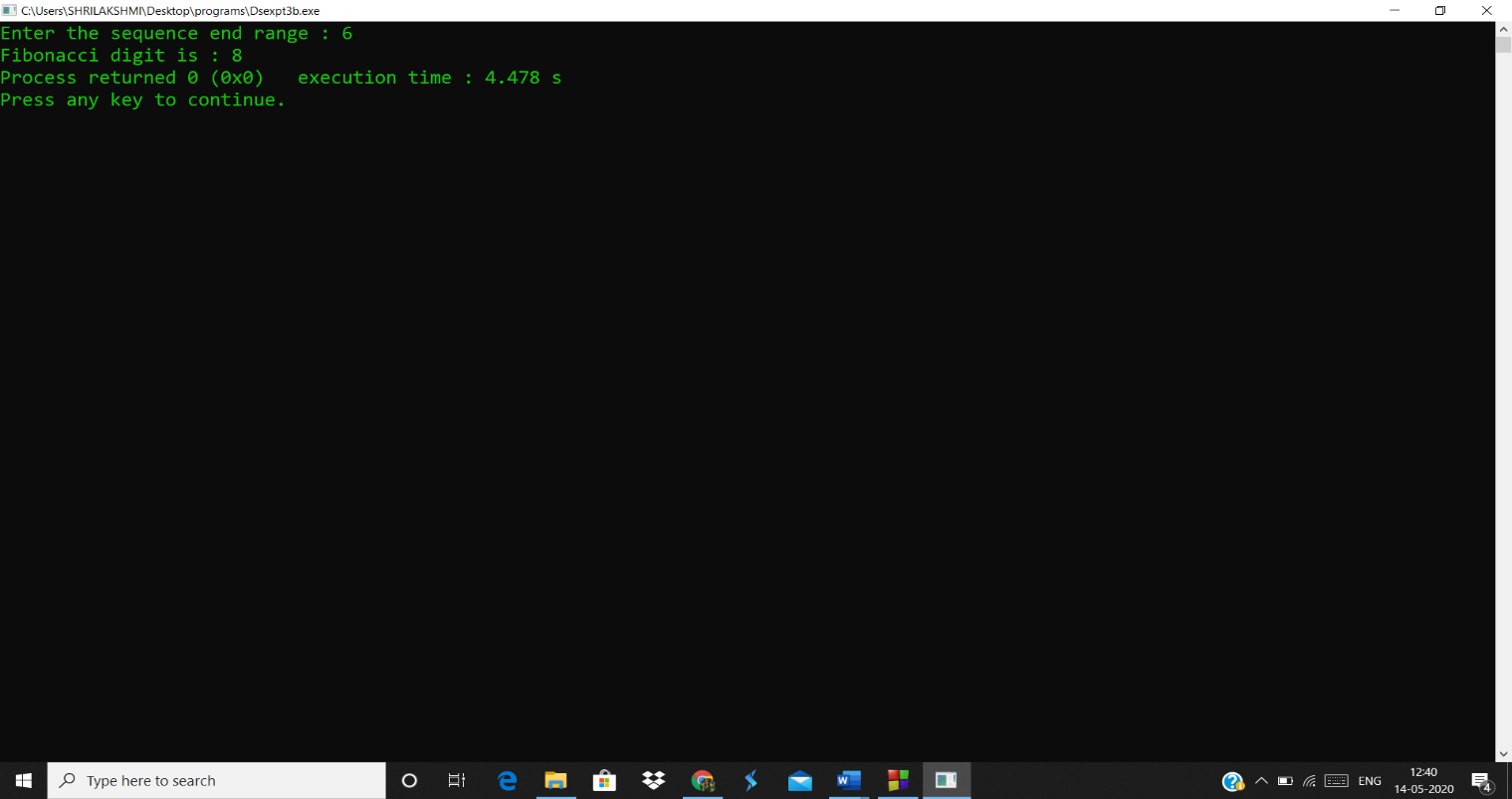
x = fibo((n-1));

cout<<"Fibonacci digit is : "<<x;

}

**Output :**

****

****

**Analysis :**

* Recursion is a very costly process when compared to looping because we can perform the operations using both.
* The program returns the number in Fibonacci series according to the position sent but does not give the entire series until that position.