**Experiment No. 3b**

**Title :** Implementation of Fibonacci series using recursion

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

**Algorithm :**

**S1 :** Start

**S2 :** Declare an integer variable and a function fibonacci ()

**S3 :** Call the function and send the value whose Fibonacci number has to be found

**S4 :** in 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.

**S5 :** Stop

**Code :**

#include<iostream>

using namespace std;

int fib(int n)

{

if(n<=1)

{

return(1);

}

else{

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

}

}

int main()

{

int n,x;

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

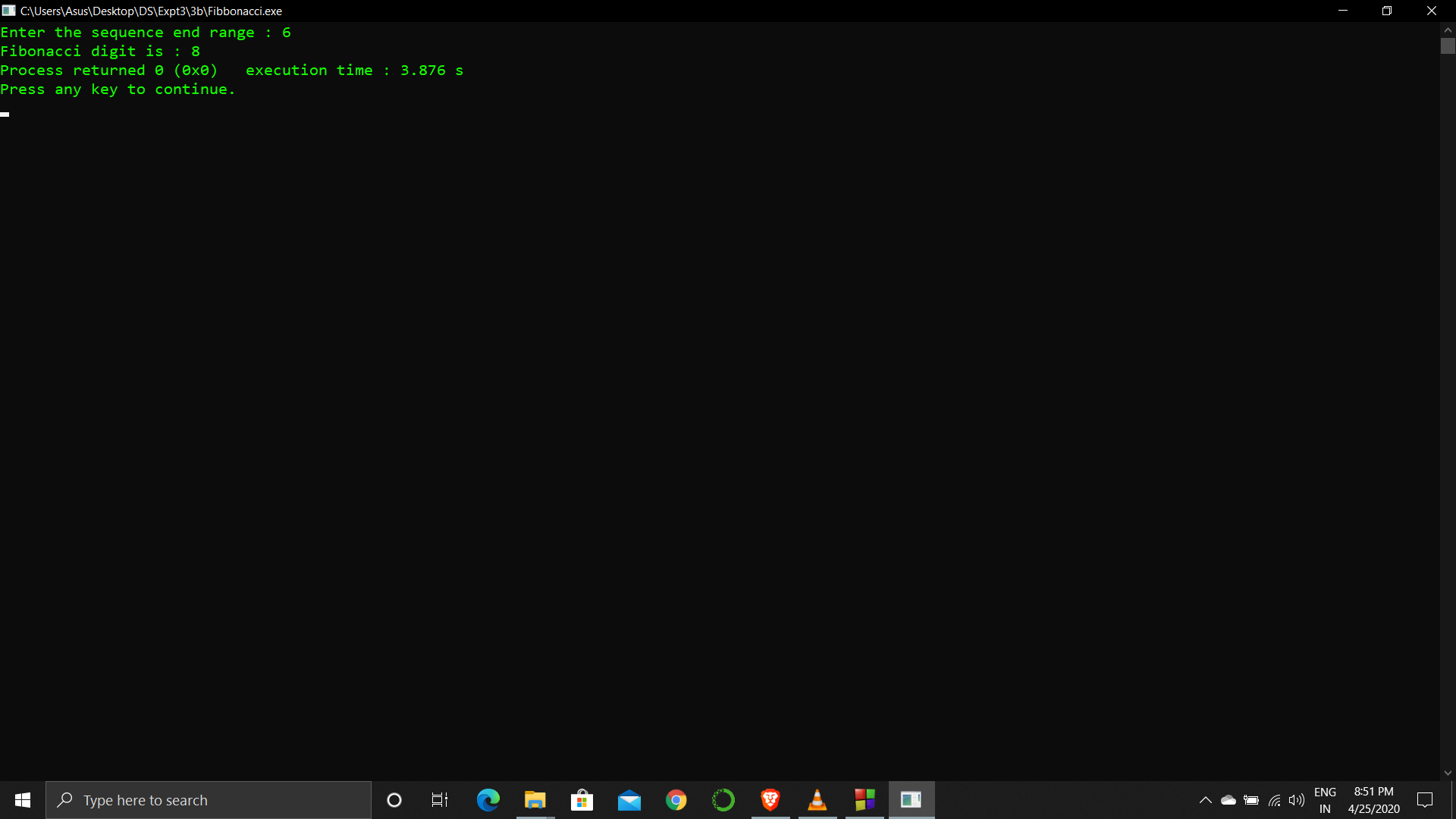
cin>>n;

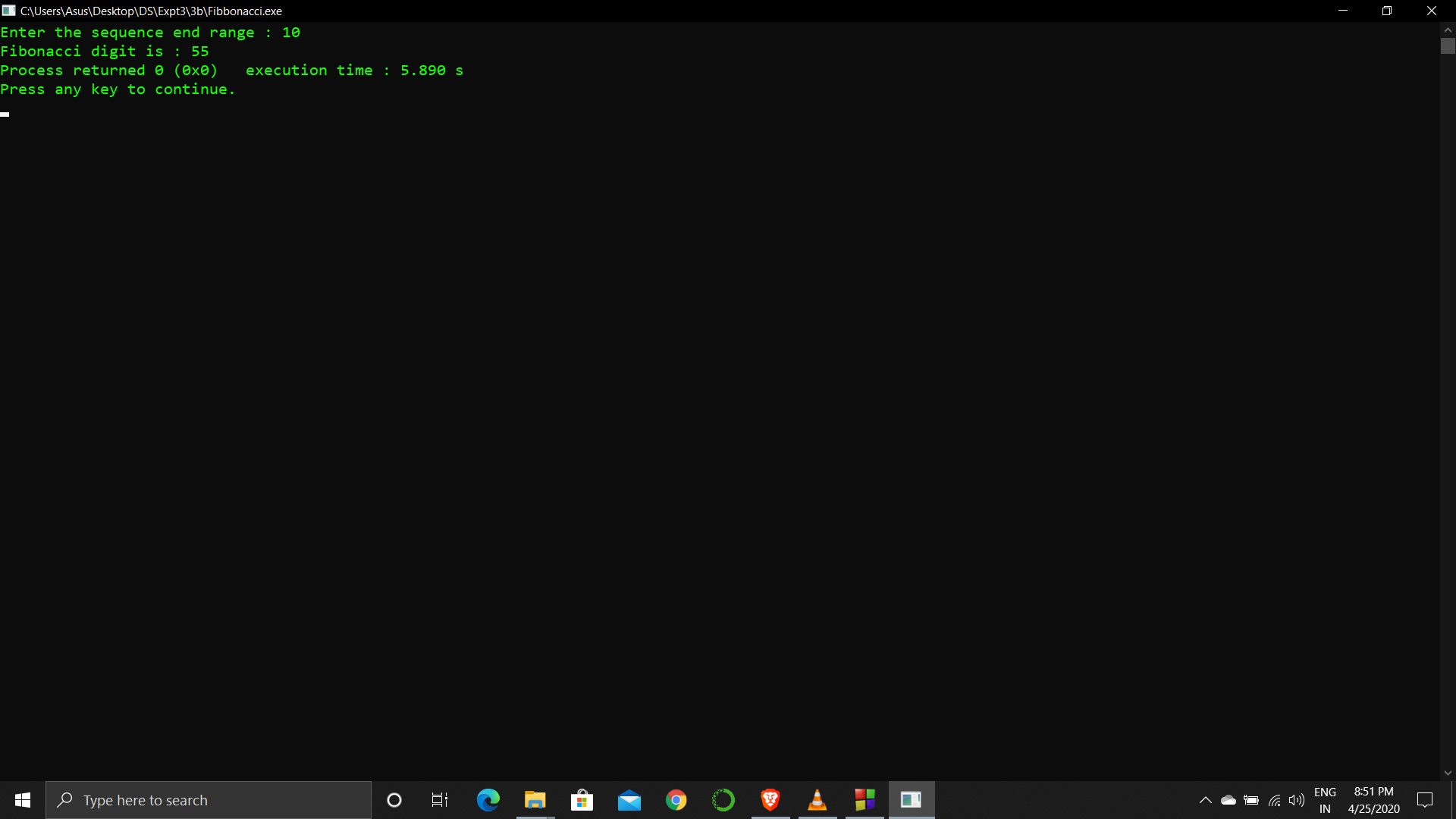
x = fib((n-1));

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.