RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Lab Report 1

Course Code: *CSE 2202*

Course Title: *Sessional Based on CSE 2201.*

Submitted By:

Name : Shanjid Hasan Nishat

Roll No : 1803172

Section : ‘C’

Date of Submission: 06/02/ 2021

Submitted To:

Dr. Md. Ali Hossain

Associate Professor,

Dept. of Computer Science and Engineering.

Rajshahi University of Engineering & Technology.

Problem Statement: Comparison of recursive and non-recursive method for generating Fibonacci series.

Description and Algorithm:

Fibonacci series is a sequence of numbers where each number is the sum of the two preceding numbers starting from 0 and 1. We can generate Fibonacci series by iterative method and recursive method. In iterative method the time complexity of generating one Fibonacci number is **O(n)**. On the other hand, in recursive method the time complexity of generating one Fibonacci number is **O(2^n)** or **exponential**.

- Algorithm for **recursive** method:

```
Fibonacci(n):
  IF n <= 1 THEN
     return n
  return Fibonacci(n-1) + Fibonacci(n-2)
```

- Algorithm for **non-recursive** method:

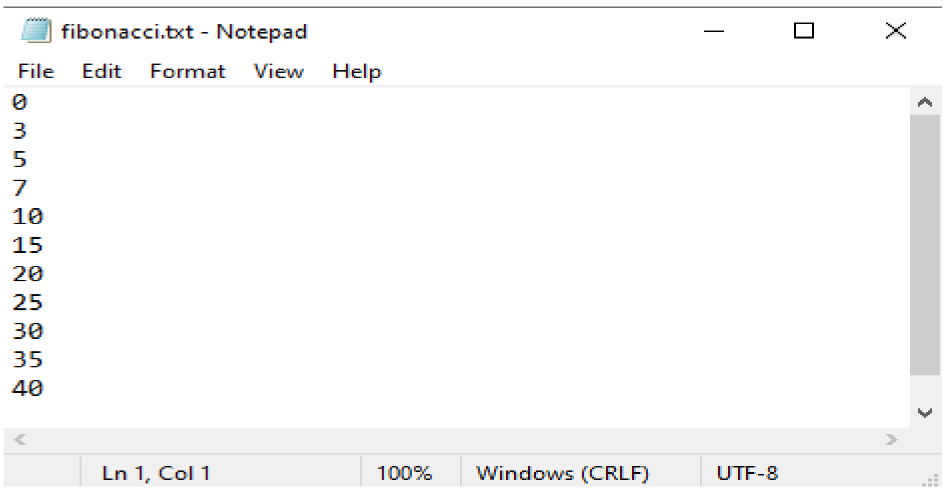
```
Fibonacci(n):
   a:=1
   b:=0
   x:=0
   IF n<=1 THEN
       return n;
   FOR  i = 2 to n:
       x:=a+b
      b:=a
       a:=x
   return x
```

Code:

```
#include <bits/stdc++.h>
using namespace std;
long long arr[100010],len;
long long rec_cont = 0, non_rec_cnt = 0;
long long rec_fib(long long n)
{rec_cont++;
    rec_cont+=2;
    if((n==1)|| n==0)
    {rec_cont++;
        return n;
    }
    else
    {
        rec_cont+=2;
        return(rec_fib(n-1)+rec_fib(n-2));
    }
}
int fib(int n)
{
    int b=0;    non_rec_cnt++;
    int a=1;    non_rec_cnt++;
    int x=0;    non_rec_cnt++;
    int i;  non_rec_cnt++;
    non_rec_cnt+=2;
    if (n==0 || n==1)
    {
        non_rec_cnt++;
        return n;
    }
    non_rec_cnt++;
    for(i=2; i<=n; i++)
    {non_rec_cnt+=2;
        x=a+b;  non_rec_cnt+=2;
        b=a;    non_rec_cnt++;
        a=x;    non_rec_cnt++;
    }
    non_rec_cnt++;
    return x;
}
void readFile(string fname)
{
    long long x,i=0;
    ifstream inFile;
    inFile.open(fname);
    if (!inFile)
    {
        cout << "Cannot open file.\n";
        exit(1);
    }
    while (inFile >> x)
    {
        arr[i++] = x;
    }
    inFile.close();
    len = i;
}
```

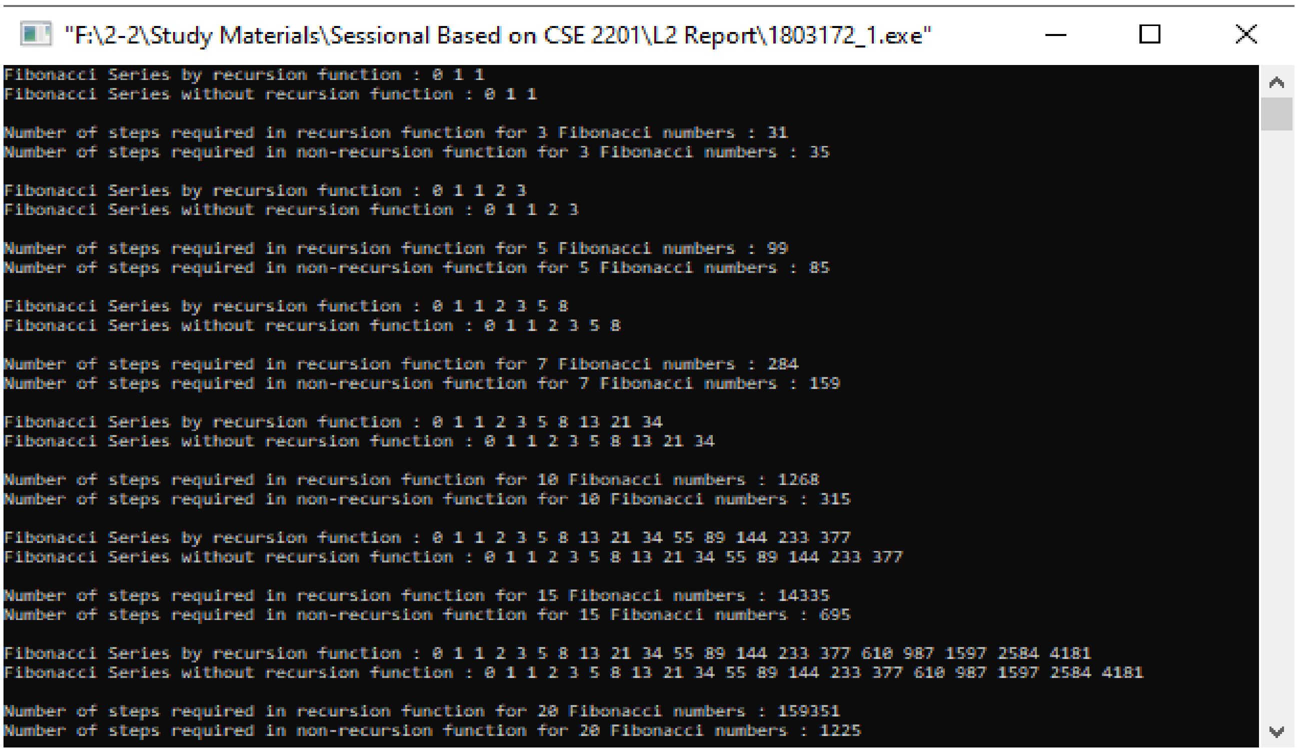
```
int main()
{
    readFile("fibonacci.txt");
    for(long long i = 0 ; i < len ; i++)
    {
        rec_cont = 0;
        non_rec_cnt = 0;
        cout << "Fibonacci Series by recursion function : ";
        rec_cont++;
        for(long long j = 0 ; j < arr[i] ; j++)
        {rec_cont+=2;
            rec_cont++;
            cout<<rec_fib(j)<<" ";
        }
        cout<<endl;

        cout << "Fibonacci Series without recursion function : ";
        non_rec_cnt++;
        for(long long j = 0 ; j < arr[i] ; j++)
        {non_rec_cnt+=2;
            cout<<fib(j)<<" ";
        }
        cout<<endl<<endl;
        cout<<"Number of steps required in recursion function for
"<<arr[i]<<" Fibonacci numbers : "<<rec_cont;
        cout<<"\nNumber of steps required in non-recursion function for
"<<arr[i]<<" Fibonacci numbers : "<<non_rec_cnt;
        cout<<endl<<endl;
    }
    return 0;
}
```

Input:



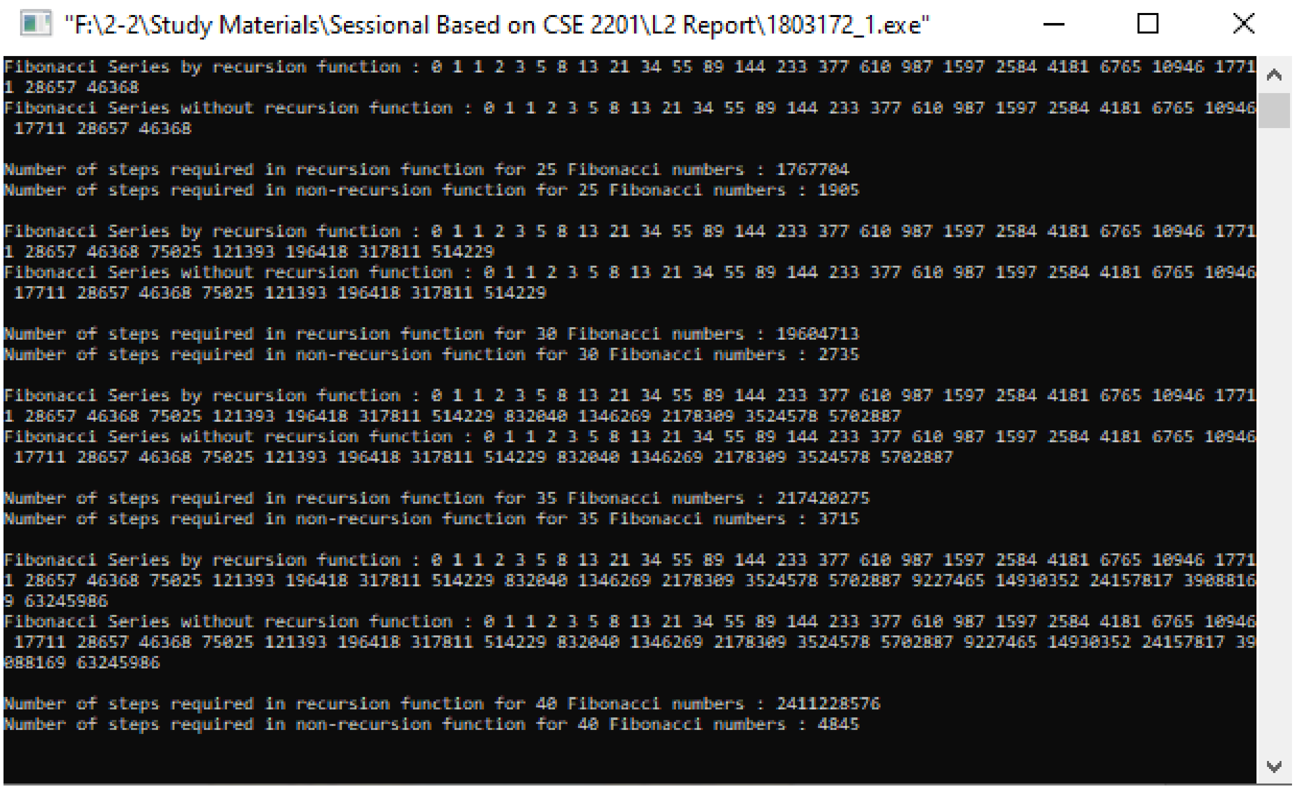
```
0
3
5
7
10
15
20
25
30
35
40
```

Output:



"F:\2-2\Study Materials\Sessional Based on CSE 2201\L2 Report\1803172_1.exe"

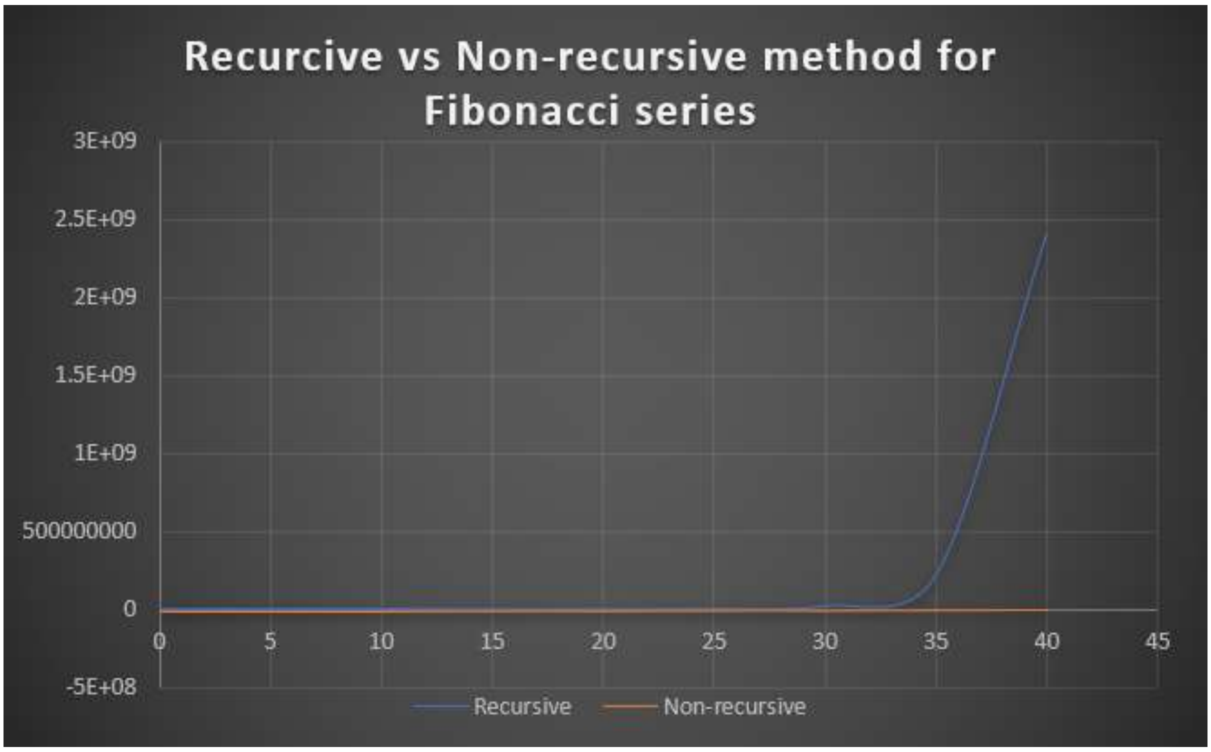
```
Fibonacci Series by recursion function : 0 1 1
Fibonacci Series without recursion function : 0 1 1

Number of steps required in recursion function for 3 Fibonacci numbers : 31
Number of steps required in non-recursion function for 3 Fibonacci numbers : 35

Fibonacci Series by recursion function : 0 1 1 2 3
Fibonacci Series without recursion function : 0 1 1 2 3

Number of steps required in recursion function for 5 Fibonacci numbers : 99
Number of steps required in non-recursion function for 5 Fibonacci numbers : 85

Fibonacci Series by recursion function : 0 1 1 2 3 5 8
Fibonacci Series without recursion function : 0 1 1 2 3 5 8

Number of steps required in recursion function for 7 Fibonacci numbers : 284
Number of steps required in non-recursion function for 7 Fibonacci numbers : 159

Fibonacci Series by recursion function : 0 1 1 2 3 5 8 13 21 34
Fibonacci Series without recursion function : 0 1 1 2 3 5 8 13 21 34

Number of steps required in recursion function for 10 Fibonacci numbers : 1268
Number of steps required in non-recursion function for 10 Fibonacci numbers : 315

Fibonacci Series by recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
Fibonacci Series without recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

Number of steps required in recursion function for 15 Fibonacci numbers : 14335
Number of steps required in non-recursion function for 15 Fibonacci numbers : 695

Fibonacci Series by recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
Fibonacci Series without recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181

Number of steps required in recursion function for 20 Fibonacci numbers : 159351
Number of steps required in non-recursion function for 20 Fibonacci numbers : 1225
```



"F:\2-2\Study Materials\Sessional Based on CSE 2201\L2 Report\1803172_1.exe"

```
Fibonacci Series by recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 1771
1 28657 46368
Fibonacci Series without recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946
 17711 28657 46368

Number of steps required in recursion function for 25 Fibonacci numbers : 1767704
Number of steps required in non-recursion function for 25 Fibonacci numbers : 1905

Fibonacci Series by recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 1771
1 28657 46368 75025 121393 196418 317811 514229
Fibonacci Series without recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946
 17711 28657 46368 75025 121393 196418 317811 514229

Number of steps required in recursion function for 30 Fibonacci numbers : 19604713
Number of steps required in non-recursion function for 30 Fibonacci numbers : 2735

Fibonacci Series by recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 1771
1 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887
Fibonacci Series without recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946
 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887

Number of steps required in recursion function for 35 Fibonacci numbers : 217420275
Number of steps required in non-recursion function for 35 Fibonacci numbers : 3715

Fibonacci Series by recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 1771
1 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 3908816
9 63245986
Fibonacci Series without recursion function : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946
 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 39
088169 63245986

Number of steps required in recursion function for 40 Fibonacci numbers : 2411228576
Number of steps required in non-recursion function for 40 Fibonacci numbers : 4845
```

Graph:



Discussion and conclusion:

In this problem we have seen the comparison of recursive and non-recursive method for generating Fibonacci series. Here from the graph er can see that the recursive method takes a lot more steps to get the sequence than the non-recursive or iterative method. So, we can conclude that **non-recursive** method is **more efficient than recursive** method **for generating Fibonacci series**.