

# **RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY**

## **LAB REPORT-01**

**COURSE NAME: SESSIONAL BASED ON CSE-2201  
COURSE CODE: CSE-2102**

### **SUBMITTED TO-**

**DR. MD. ALI HOSSAIN**

**ASSOCIATE PROFESSOR**

**Department of Computer Science & Engineering  
Rajshahi University of Engineering & Technology**

### **SUBMITTED BY-**

**SRABONTI DEB**

**Roll-1803163**

**Section - C**

**Department of Computer Science & Engineering  
Rajshahi University of Engineering & Technology**

**SUBMISSION DATE – 3 FEBRUARY, 2021**

**Problem Statement:** Generate a recursive algorithm and non-recursive algorithm for creating the series of Fibonacci numbers up to a given number of elements. The number of elements should be read from a file “fibonacci.txt”. Then find out the total number of steps of both algorithm and compare the number of steps (i.e., at least 10 comparisons) and include graphs if required.

**Theory:** A recursive algorithm is an algorithm which calls itself with "smaller (or simpler)" input values, and which obtains the result for the current input by applying simple operations to the returned value for the smaller (or simpler) input.

### **Recursive algorithm for Fibonacci series:**

Algorithm Fibonacci(n)

```
{    if(n<=1) then return n;
    else return Fibonacci(n-1)+Fibonacci(n-2);
}
```

Write Fibonacci(n);

### **Normal algorithm for Fibonacci series:**

Algorithm Fibonacci(n)


//Compute the nth Fibonacci number.

```
{
    if(n<=1) then
        Write(n);
    else
    {
        fnm:=0;fnm:=1;
        for i:=2 to n do
        {
            fn:=fnm1+fnm2;
            fnm2:=fnm1;fnm1=fn;
        }
        Write (fn);
    }
}
```

## Implemented code for recursive algorithm:

```
#include<bits/stdc++.h>
using namespace std;
int c=0;
int fib(int n)
{
    c++;
    if (n <= 1)
    {
        return n;
    }
    else
    {
        return fib(n-1) + fib(n-2);
    }
}
int main ()
{
    int j,n;
    ifstream file;
    file.open("fibonacci.txt");
    if(file)
    {
        while(file>>n)
        {
            cout<<"the fibonacci series of "<<n<<" numbers:\n";
            for(int i=0; i<n; i++)
            {
                cout<<fib(i)<<" ";
                c+=2;
            }
            c++;
            cout<<endl;
            cout<<"total number of counting steps:"<<c<<endl;
            c=0;
        }
    }
    else
        cout<<"file can't open"<<endl;
}
```

## Output:

 "E:\2-2\computer algorithm\Lab-1\Fibonacci(with recursion).exe"


```
the fibonacci series of 9 numbers:
0 1 1 2 3 5 8 13 21
total number of counting steps:186
the fibonacci series of 5 numbers:
0 1 1 2 3
total number of counting steps:30
the fibonacci series of 6 numbers:
0 1 1 2 3 5
total number of counting steps:47
the fibonacci series of 7 numbers:
0 1 1 2 3 5 8
total number of counting steps:74
the fibonacci series of 8 numbers:
0 1 1 2 3 5 8 13
total number of counting steps:117
the fibonacci series of 4 numbers:
0 1 1 2
total number of counting steps:19
the fibonacci series of 10 numbers:
0 1 1 2 3 5 8 13 21 34
total number of counting steps:297
the fibonacci series of 12 numbers:
0 1 1 2 3 5 8 13 21 34 55 89
total number of counting steps:765
the fibonacci series of 11 numbers:
0 1 1 2 3 5 8 13 21 34 55
total number of counting steps:476
the fibonacci series of 3 numbers:
0 1 1
total number of counting steps:12

Process returned 0 (0x0)   execution time : 0.205 s
Press any key to continue.
```

## Implemented code for normal algorithm:

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int i,n1,n2,n3,n,c=0;
    ifstream file;
    file.open("fibonacci.txt");
    if(file)
    {
        while(file>>n)
        {
            if(n<=1)
            {
                cout<<"the fibonacci series of "<<n<<" numbers:\n";
                cout<<0;
            }
            else
            {
                n1=0;
                n2=1;
                cout<<"the fibonacci series of "<<n<<" numbers:\n";
                cout<<n1<<" "<<n2<<" ";
                for(i=2; i<n; i++)
                {
                    n3=n1+n2;
                    n1=n2;
                    n2=n3;
                    c=c+5;
                    cout<<n3<<" ";
                }
                c++;
                cout<<"\n"<<"total number of counting steps:"<<c<<"\n";
                c=0;
                n1=0,n2=1;
            }
        }
    }
    else
        cout<<"file can't open"<<endl;
    return 0;
}
```

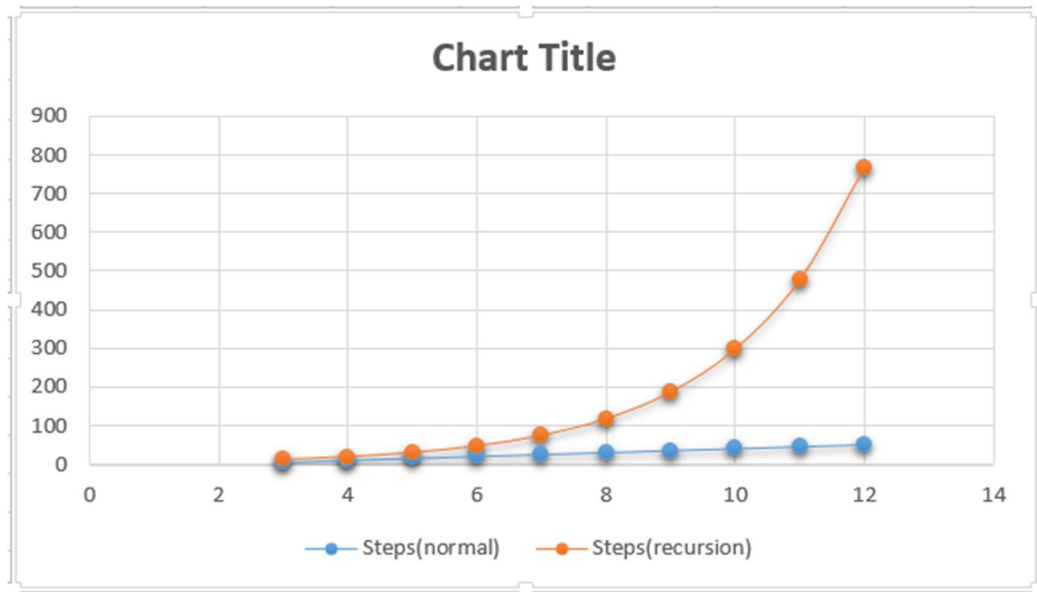
## Output:

 "E:\2-2\computer algorithm\Lab-1\Fibonacci(without recursion).exe"

```
the fibonacci series of 9 numbers:
0 1 1 2 3 5 8 13 21
total number of counting steps:36
the fibonacci series of 5 numbers:
0 1 1 2 3
total number of counting steps:16
the fibonacci series of 6 numbers:
0 1 1 2 3 5
total number of counting steps:21
the fibonacci series of 7 numbers:
0 1 1 2 3 5 8
total number of counting steps:26
the fibonacci series of 8 numbers:
0 1 1 2 3 5 8 13
total number of counting steps:31
the fibonacci series of 4 numbers:
0 1 1 2
total number of counting steps:11
the fibonacci series of 10 numbers:
0 1 1 2 3 5 8 13 21 34
total number of counting steps:41
the fibonacci series of 12 numbers:
0 1 1 2 3 5 8 13 21 34 55 89
total number of counting steps:51
the fibonacci series of 11 numbers:
0 1 1 2 3 5 8 13 21 34 55
total number of counting steps:46
the fibonacci series of 3 numbers:
0 1 1
total number of counting steps:6
```

```
Process returned 0 (0x0)   execution time : 2.066 s
Press any key to continue.
```

### Graph:



### Discussion & Conclusion:

Here we see that the number of counting steps for recursive algorithm is more than those of normal algorithm because recursive algorithm calls itself and for this the iteration process continues sometimes unnecessarily. On the other hand, normal algorithm doesn't conduct unnecessary iteration. So, if we give large number of input, normal algorithm will work faster than recursive algorithm.