**Problem Statement:**

Generate a recursive algorithm for creating the series of Fibonacci numbers up to a givennumber of elements. The number of elements should be read from a file fibonacci.txt. Your task is to implement the code of your algorithm and then find out the total number of steps. Finally compare the number of steps (i.e., at least 10 comparisons) with an algorithm which do not use recursive method forgenerating the Fibonacci numbers (include graphs if required).

**Details Description and Algorithm:**

Fibonacci series is just like : 0,1,1,2,3,5,8,13…….

Here I implement and compare two algorithm which create the series and count the number of step needed.

**First Algorithm:**

Fibo\_Function(A,n) // A is an arry of size n

{

If(n<=2) then

{

If(n=1) then a[1]:=0;

else a[1]:=0;a[2]:=0;

}

else

{

for i:=3 to n do

a[i]:=a[i-1]+a[i-2];

}

}

**Second Algorithm:**

Fibo\_Function(n) // n is the nth Fibonacci number of the series

{

If(n<=1) then return n;

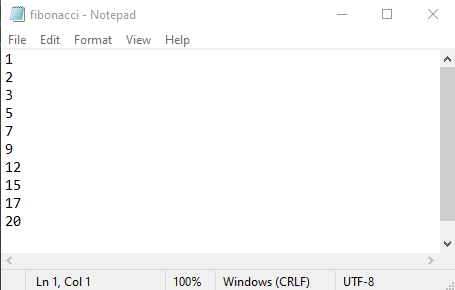
return Fibo\_Function(n-1)+ Fibo\_Function(n-2);

}

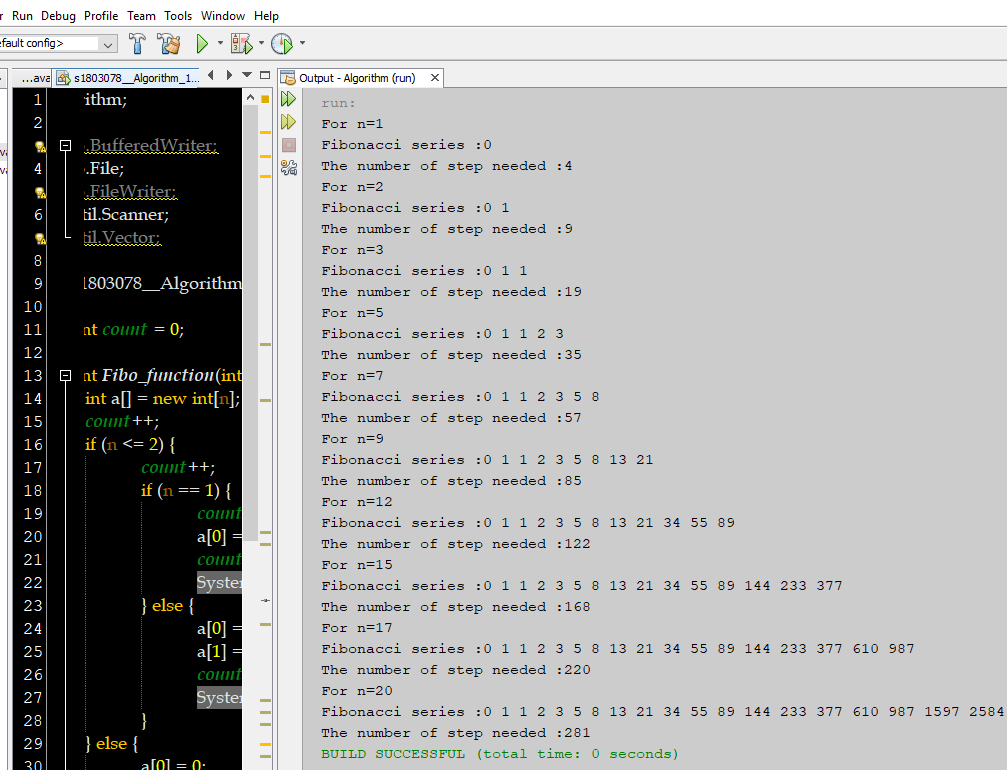
**Implemented Code(by Algorithm-1):**

|  |
| --- |
| **package algorithm;**  **import java.io.BufferedWriter;**  **import java.io.File;**  **import java.io.FileWriter;**  **import java.util.Scanner;**  **import java.util.Vector;**  **public class s1803078\_\_Algorithm\_1 {**  **static int count = 0;**  **static int Fibo\_function(int n) {**  **int a[] = new int[n];**  **count++;**  **if (n <= 2) {**  **count++;**  **if (n == 1) {**  **count++;**  **a[0] = 0;**  **count++;**  **System.out.print("Fibonacci series :"+a[0]);**  **} else {**  **a[0] = 0;**  **a[1] = 1;**  **count += 3;**  **System.out.print("Fibonacci series :"+a[0]+" "+a[1]);**  **}**  **} else {**  **a[0] = 0;**  **a[1] = 1;**  **count += 3;**  **for (int i = 2; i < n; i++) {**  **count++;**  **a[i]=a[i-1]+a[i-2];**  **count++;**  **}**  **count++;**  **System.out.print("Fibonacci series :");**  **for(int i=0;i<n;i++){System.out.print(a[i]+" ");}**  **count+=n;**  **}**  **return count;**  **}**  **public static void main(String[] args) {**  **Scanner ob = new Scanner(System.in);**  **try {**  **File f = new File("fibonacci.txt");**  **Scanner ob1 = new Scanner(f);**  **while (ob1.hasNext()) {**  **int n = Integer.valueOf(ob1.next());**  **System.out.println("For n=" + n);**  **System.out.println("\nThe number of step needed :" + Fibo\_function(n));**  **//System.out.println();**  **}**  **} catch (Exception e) {**  **System.out.println("Exception found.");**  **}**  **}**  **}** |

**Sample Input (fibonacci.txt):**

****

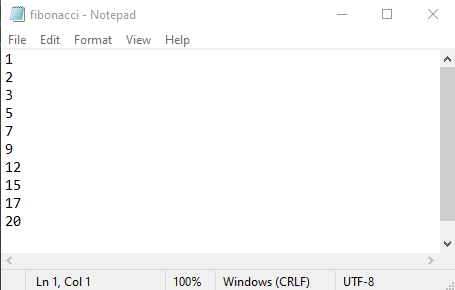
**Output:**

****

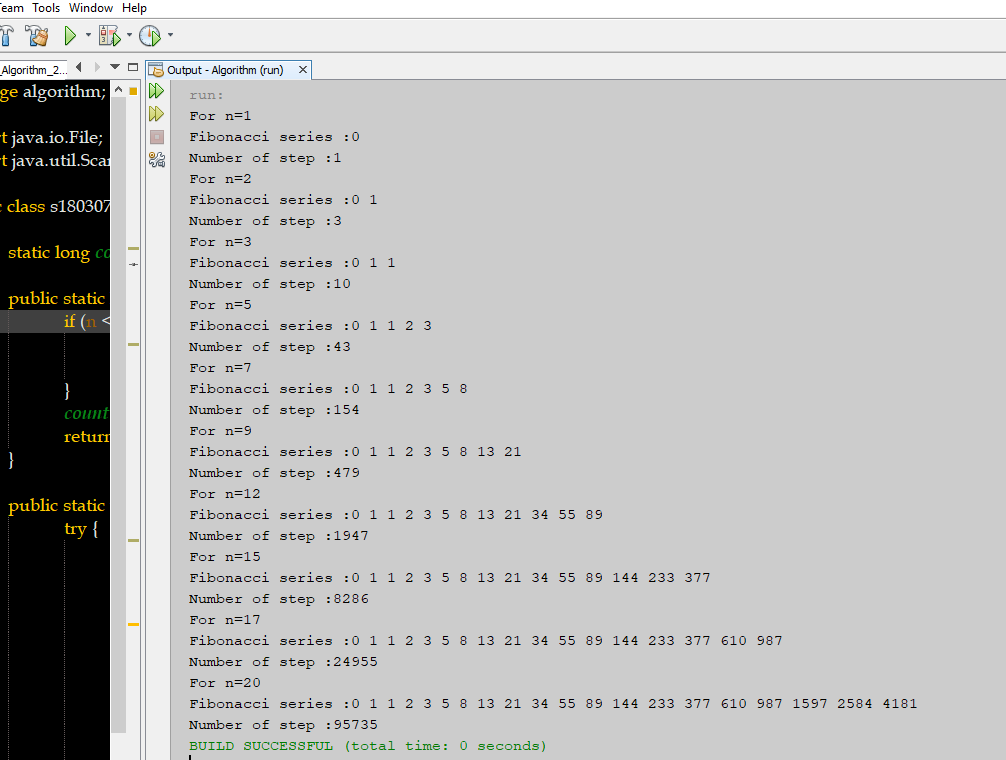
**Implemented Code(by Algorithm-2):**

|  |
| --- |
| **package algorithm;**  **import java.io.File;**  **import java.util.Scanner;**  **public class s1803078\_\_Algorithm\_2 {**  **static long count = 0;**  **public static long fibo\_function(long n) {**  **if (n <= 1) {**  **count++;**  **return n;**  **}**  **count+=3;**  **return fibo\_function(n - 2) + fibo\_function(n - 1);**  **}**  **public static void main(String args[]) {**  **try {**  **File f = new File("fibonacci.txt");**  **Scanner ob = new Scanner(f);**  **while (ob.hasNext()) {**  **String s = ob.nextLine();**  **long n = Long.valueOf(s);**  **System.out.print("For n=" + n + "\nFibonacci series :");**  **for (int i = 0; i < n; i++) {**  **System.out.print( fibo\_function(i) + " ");**  **}**  **//System.out.println();**  **System.out.println("\nNumber of step :"+count);**  **}**  **} catch (Exception e) {**  **System.out.println("exception found.");**  **}**  **}**  **}** |

**Sample Input (fibonacci.txt):**

****

**Output:**

****

**Comparing Input Output by Graph :**

**Discussion and Conclusion:**

By comparing these two Algorithm, we see Algorithm-2 takes more step than Algorithm-1. That’s why ,Algorithm-1 is better than Algorithm-2.