DSA REPORT

Name: Anirban Das Roll: 001910501077 Class: BCSE -II Sem: First Session: 2020-21

Assignment Set: 1

Problem No: 2

Problem Statement:

Write a program to generate the nth Fibonacci number iteratively and recursively. Check when there is overflow in the result and change the data types for accommodating higher values of inputs. Plot the fibonacci number vs n graph.

Solution Approach:

For recursive approach to get fibonacci, we can use the recurrence fib(n)= fib(n-1)+fib(n-2), where fib(1) = 0 and fib(2) = 1. We can use these as the breaking conditions for the recursion.

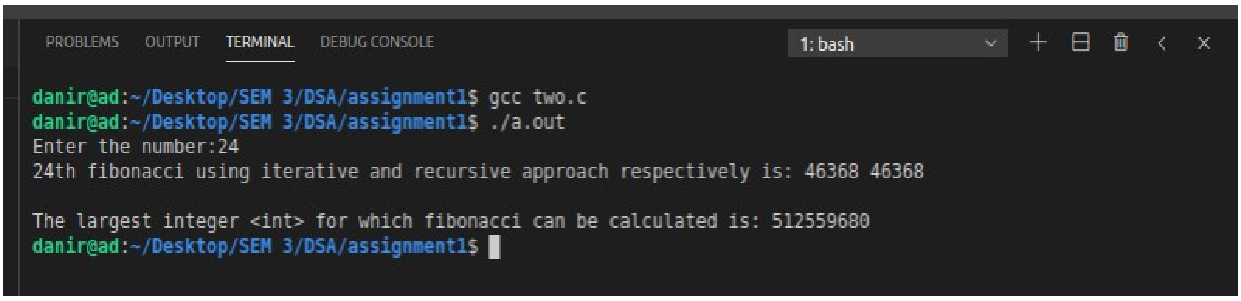
For iterative calculation, using the same recurence, we can see that if we store the last 2 fibonacci values, we can get the desired fibonacci value. This can be done using a single for loop trom 3 to n, updating prev1 and prev2(storing the last 2 fibos) in each iteration.

For findng the overflow point of int, we can just run a loop calculating fib(i), fib(i-1) and fib(i-2) and check if the condition fib(i-1) + fib(i-2) == fib(i) is satisfied, if the don’t, the value of fib(i) will get overflowed. The overflow point can be increased by taking data types such as long long int.

Structured Pseudocode:

```
FUNCTION   FIBO_RECUR ( INT N ):
        IF N <= 1:
                RETURN  N
        ELSE
                RETURN  FIBO_RECUR( N-1 ) + FIBO_RECUR( N-2 )


FUNCTION   FIBO_ITER ( INT N ):
        IF  N<=1:
                RETURN  N
        DECLARE  ARRAY[N+1]
        ARRAY[0]=0  ARRAY[1]=1
        FOR i=2 TO N+1 DO:
                ARRAY[i]=ARRAY[i-1]+ARRAY[i-2]
        RETURN ARRAY[N]

FUNCTION   FIND_MAX_INT ( ):
        TEMP=0
        RES=1
        FIB=1 (int)
        WHILE FIB+LAST>FIB DO:
                TEMP=FIB
                FIB = FIB + LAST
                LAST = TEMP
        RETURN FIB
```

Results:



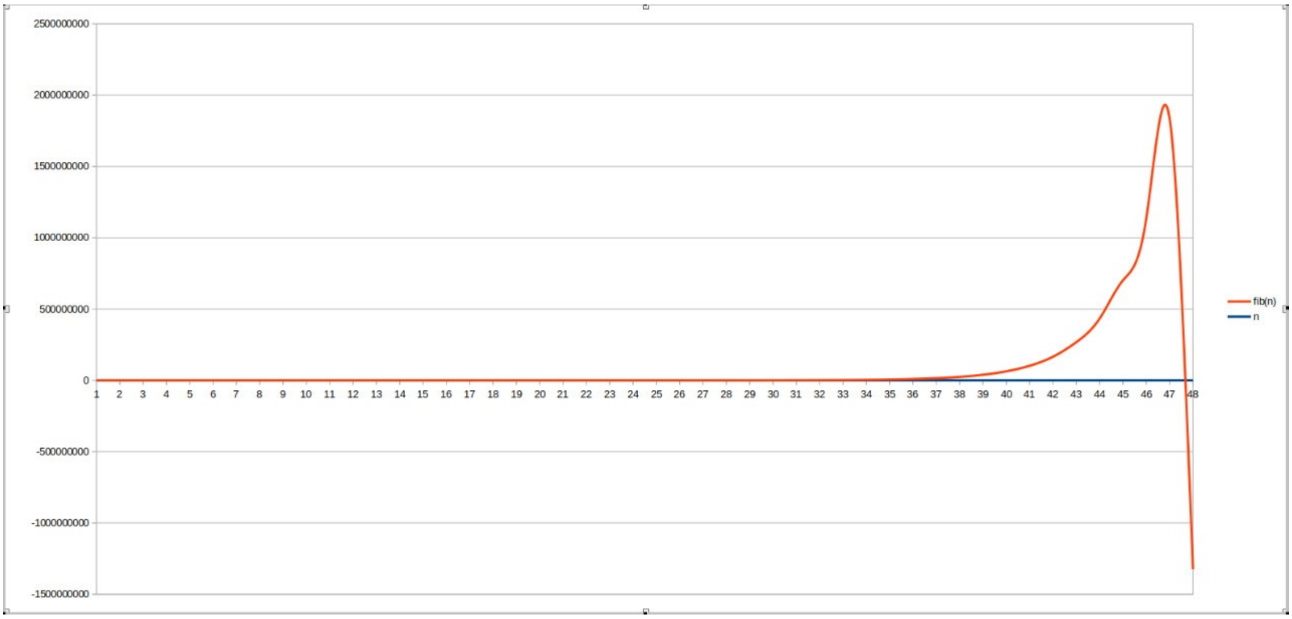
```
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ gcc two.c
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ ./a.out
Enter the number:24
24th fibonacci using iterative and recursive approach respectively is: 46368 46368

The largest integer <int> for which fibonacci can be calculated is: 512559680
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ 
```

There is a overflow in value of int for the fibbonacci number of 512559681 (in case of int data type).

DSA REPORT

Name: Anirban Das Roll: 001910501077   Class: BCSE -II   Sem: First   Session: 2020-21

Discussions:



The above graph shows fibonacci(red) vs integer number(blue). The fall at the end of the graph depicts the overflow in the value of fibonacci for int data type.

The Time Complexiy in recursive version is O(2^n). The space complexity, however, is O(1) in case of iterative and O(n) in case of recursive.

From the results we can conclude the largest fibonacci of type int that can be calculated is 512559680.

Source Code:

FILE NAME: “two.c”

(can be found in the following link: https://drive.google.com/drive/folders/1-nNb6aRleNLE1mcE58i85096fDmDUCvd?usp=sharing)