

Name: Mohiuddin Mondal  
Roll no. 001910501043  
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:** Take a variable 'sum' initialised to 0, 'prev1' = 1 and 'prev2' = 1; then run a loop where prev1 and prev2 are added with sum and prev1, prev2 are updated accordingly to store previous two fibonacci numbers, thus we get required nth fibonacci number.

**Structured Pseudocode:**

```
Read n
sum=0, prev1=1, prev2=1;
if n<=0:
    Invalid Number
else if n<=2:
    print 1
    end program
else
    for i=3 to n do:
        sum = prev1 + prev2
        prev1 = prev2
        prev2 = sum
    end for
end if
print sum
end program
```

**Results:**

output1:

```
zulfiqar@zulqarnain: ~/assignmentGit/secondYear/DSAAssignment/assignment1
zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$ ./a.out
Enter a number: 34
Iterative computation of 34 = 5702887
Recursive computation of 34 = 5702887

zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$ ./a.out
Enter a number: 91
Iterative computation of 91 = 4660046610375530309
Number is too big to compute recursively. :)

zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$ ./a.out
Enter a number: 92
Iterative computation of 92 = 7540113804746346429
Number is too big to compute recursively. :)

zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$ ./a.out
Enter a number: 93
Iterative computation of 93 = -6246583658587674878
Number is too big to compute recursively. :)

zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$
```

output2:

```
zulfiqar@zulqarnain: ~/assignmentGit/secondYear/DSAAssignment/assignment1
zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$ gcc assignment2-b.c
zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$ ./a.out
Enter a number: 92
92th Fibonacci Numer: 7540113804746346429

zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$ ./a.out
Enter a number: 93
93th Fibonacci Numer: 12200160415121876738

zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$ ./a.out
Enter a number: 111
111th Fibonacci Numer: 70492524767089125814114

zulfiqar@zulqarnain:~/assignmentGit/secondYear/DSAAssignment/assignment1$
```

## Discussion

- In the results section , output1: We see 93th number is negative or overflow has occurred for built in *long long* datatype.
- output2: here, an array of integer is used to represent an integer and corresponding function are defined to add such numbers. Thus we see here no overflow occurs and higher values are shown.

**Separate files containing commented source code:** Two source code are attached,:

assignment2-a.c :- It computes factorial using built in data type *long long* (64bit)

assignment2-b.c :- It computes factorial using an array of digit to represent the number.