

DSA REPORT

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

Assignment Set: 1

Problem No: 1

Problem Statement:

Write a program to compute the factorial of an integer n iteratively and recursively. Check when there is overflow in the result and change the data types for accommodating higher values of inputs.

Solution Approach:

For recursive calculation of factorial of a number, we use the recurrence $\text{fact}(n) = n * \text{fact}(n-1)$, where $\text{fact}(0) = 1$.

For iterative approach we can use the same principle, but instead of finding all the factorials, we initialize a variable with 1 ($\text{fact}=1$), and multiply every integer less than or equal to the given number, using a loop. The point is we store the factorial of 1 and use it to calculate the desired factorial.

For finding the overflow point, we can just run a loop calculating $\text{fact}(i)$, and break the loop when factorial becomes negative, showing that overflow has occurred.

For a better maximum limit, we can use some higher range data types like long long int, whose overflow point can be calculated using the same method.

Structured Pseudocode:

```
FUNCTION FACT_RECUR ( INT N ):
    IF N == 1:
        RETURN 1
    ELSE
        RETURN N * FACT_RECUR( N-1 )
```

```
FUNCTION FACT_ITER ( INT N ):
    F = 1
    FOR i=2 to n DO
```

DSA REPORT

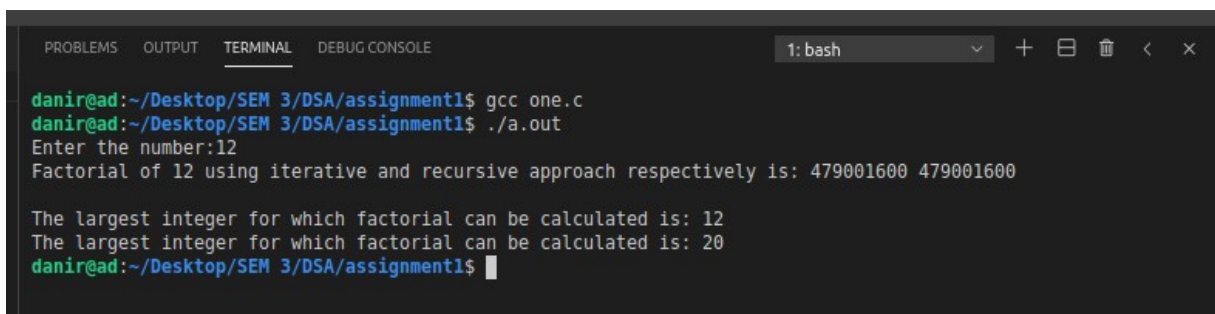
Name: Anirban Das **Roll: 001910501077** **Class: BCSE -II** **Sem: First** **Session: 2020-21**

```
F = F * i
RETURN F
```

```
FUNCTION FIND_MAX_INT ( ):
    RES=1
    F=1 (int)
    WHILE true DO:
        IF R < 0:
            BREAK
        R = R + 1
        F = F * R
    RETURN R
```

```
FUNCTION FIND_MAX_LONG ( ):
    RES=1
    F=1 (long long int)
    WHILE true DO:
        IF R < 0:
            BREAK
        R = R + 1
        F = F * R
    RETURN R
```

Results:



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ gcc one.c
danir@ad:~/Desktop/SEM 3/DSA/assignment1$ ./a.out
Enter the number:12
Factorial of 12 using iterative and recursive approach respectively is: 479001600 479001600

The largest integer for which factorial can be calculated is: 12
The largest integer for which factorial can be calculated is: 20
danir@ad:~/Desktop/SEM 3/DSA/assignment1$
```

There is a overflow of value of factorial at the integer number 13 (in case of int data type). The same overflow occurs at the integer number 21 (in case of long long data type).

DSA REPORT

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

Discussions:

The Time Complexiy in both recursive and iterative approach is $O(n)$. The space complexity, however, is $O(1)$ in case of iterative and $O(n)$ in case of recursive.

From the results above, we can conclude that the largest int and long long number whose factorial can be calculated are 12 and 20 respectively.

The space complexity is $O(1)$.

Source Code:

FILE NAME: “one.c”

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