**QUESTION:**

Write two programs for finding factorials of a given non-negative integers  
a) using a FOR loop  
b) using a recursive function  
  
Then test both the programs with different input starting from 10. Observe which one is performing better, and state why.

1. **Using FOR loop:**

#include <stdio.h>

int main()

{

int n, i;

unsigned long long fact = 1;

printf("Enter an integer: ");

scanf("%d", &n);

// shows error if the user enters a negative integer

if (n < 0)

printf("Error! Factorial of a negative number doesn't exist.");

else {

for (i = 1; i <= n; ++i)

{

fact \*= i;

}

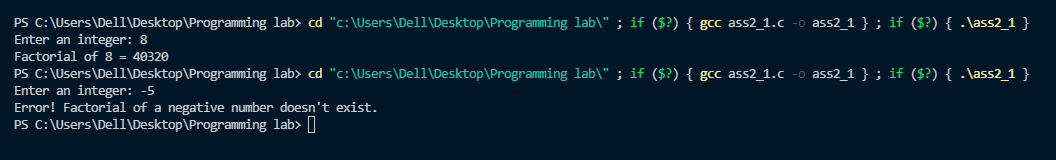
printf("Factorial of %d = %llu", n, fact);

}

return 0;

}

**Output:**



1. **using a recursive function:**

#include <stdio.h>

double fac(int n)

{

if(n<=1)

return 1;

return n\*fac(n-1);

}

void main()

{

int num;

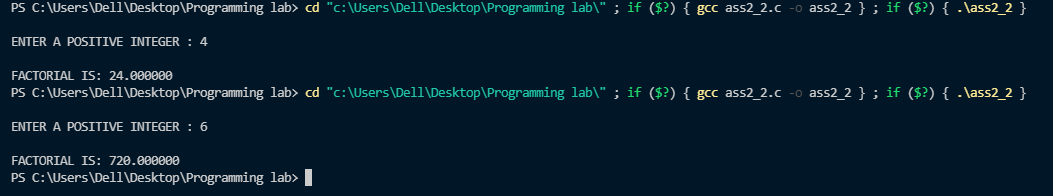
printf("\nENTER A POSITIVE INTEGER : ");

scanf("%d",&num);

printf("\nFACTORIAL IS: %f\n",fac(num));

}

**Output:**



1. **Performance of both the programs:**

A program is called recursive when an entity calls itself. A program is call iterative when there is a loop (or repetition).

Theoretically Loop should be faster than Recursion based methods. Recursion uses more memory than a loop. Recursion can be faster when the variable mutation or condition controlling the loop requires heavy computation.

Although Loop method involved extra step of loop variable condition checking and updation. Recursive approach will not involve such repeated steps due to its Stacked CALL structure.

So we can conclude that, in case of small values iteration that is the loops are much more effectives, Whereas in case of large values recursive is a good option.