**Experiment Number: 3a**

**Title:** Factorial Using Recursion Using C++.

**Problem Statement**: Write a C++ program to calculate the factorial of a given number using recursion.

**Algorithm:**

1. Read the number and declare the variable fact to store the factorial of the number.
2. We know that the factorial of a number n is fact(n)=n\*fact(n-1)

Where, n is the number

fact means factorial.

1. The above equation acts as base condition.
2. First check if the number is equal to zero if zero return the value 1

If not zero then call the function recursively according to the equation n\*fact(n-1).

1. **Return the variable fact.**

**Code:**

#include<iostream>

using namespace std;

int factorial(int n);

int main()

{

int num;

cout<<"\nEnter the number:";

cin>>num;

cout<<"\nFactorial of entered number:"<<factorial(num);

return 0;

}

int factorial(int n)

{

int fact;

if(n==0)

return 1;

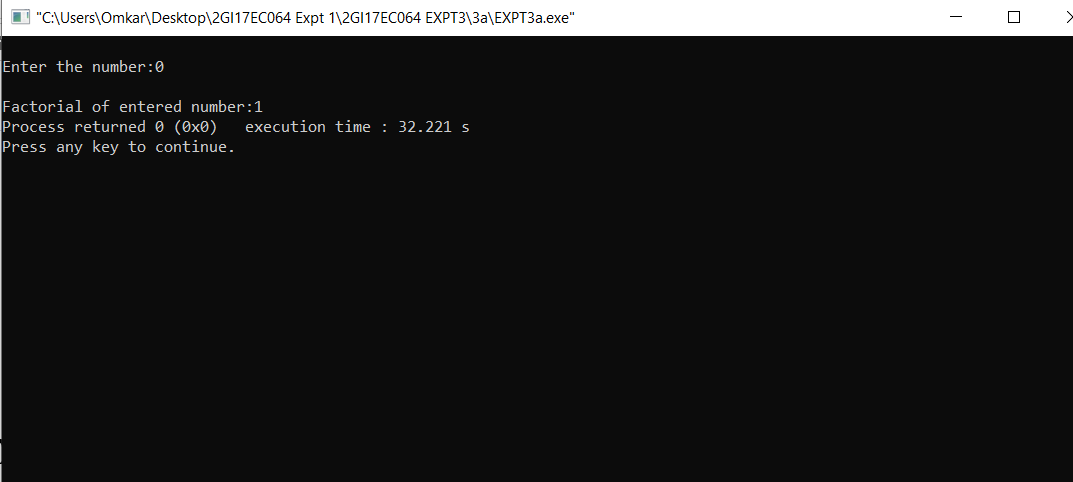
else

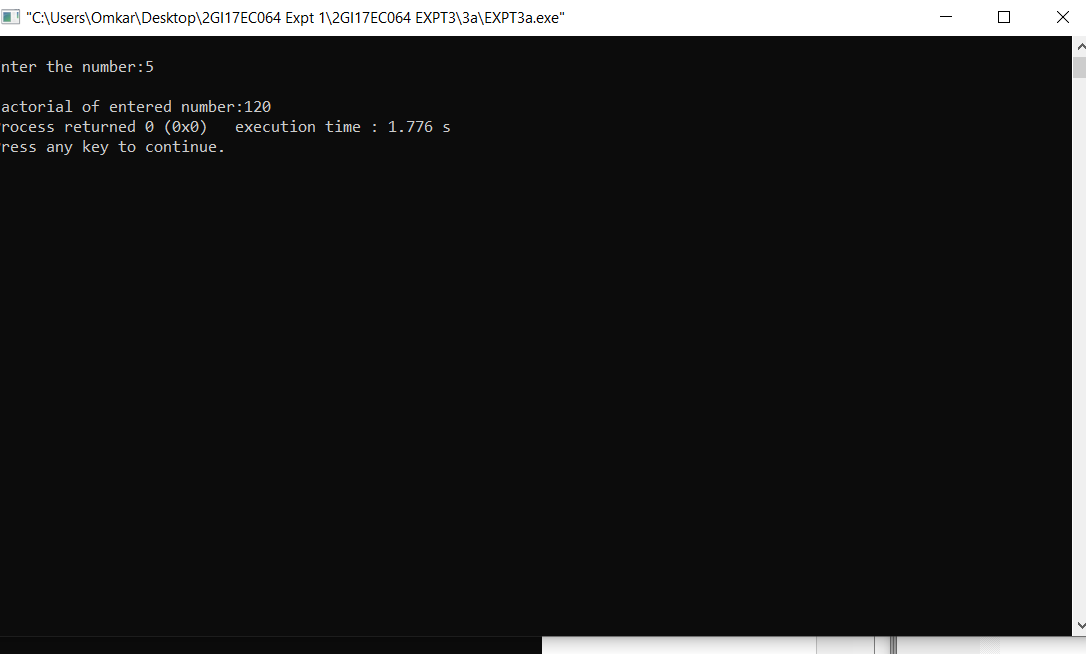
fact=n\*factorial(n-1);

return fact;

}

**Output:**

****



**5. Analysis/Limitations:**

There are following limitations of above implementation.

Disadvantages of recursion:

* Fairly slower than its iterative solution.
* For each step we make a **recursive** call to a function.
* May cause stack-overflow if the **recursion** goes too deep to solve the problem.
* Difficult to debug and trace the values **with** each step of **recursion**.