Title: Recursive functions in C

Objective:

The main objectives of this lab are to

- Print Fibonacci numbers till nth term.
- Factorial of a number.

Theory:

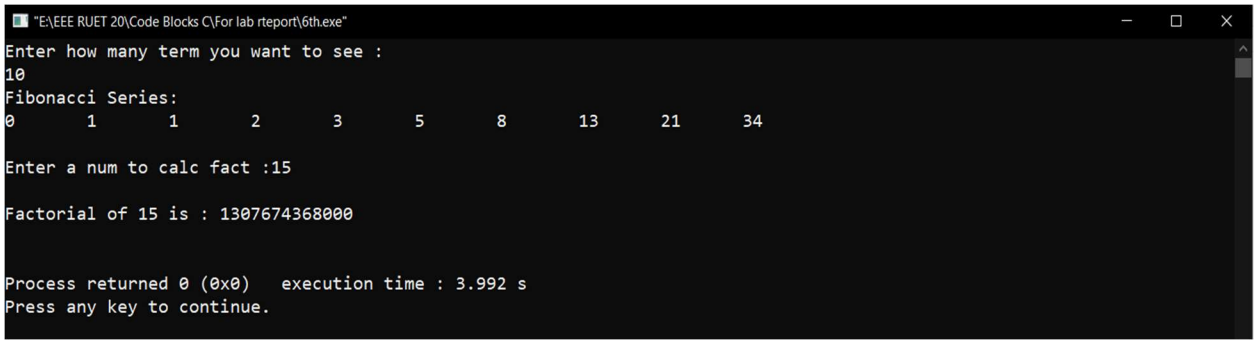
Recursion is the process which comes into existence when a function calls a copy of itself to work on a smaller problem. Any function which calls itself is called recursive function, and such function calls are called recursive calls. Recursion involves several numbers of recursive calls. However, it is important to impose a termination condition of recursion. Recursion code is shorter than iterative code however it is difficult to understand.
Recursion cannot be applied to all the problem, but it is more useful for the tasks that can be defined in terms of similar subtasks. For Example, recursion may be applied to sorting, searching, and traversal problems.
Generally, iterative solutions are more efficient than recursion since function call is always overhead. Any problem that can be solved recursively, can also be solved iteratively. However, some problems are best suited to be solved by the recursion, for example, tower of Hanoi, Fibonacci series, factorial finding, etc.

Source Code:

```
1. /// C Program to print fibonacci series using recursion
2.
3. #include<stdio.h>
4. // User Define function to display fibonacci series.
5. int fibonacci(int n)
6. {
7.     if (n==0)
8.         return 0;
9.     else if (n==1)
10.             return 1;
11.         else
12.             return (fibonacci(n-1)+fibonacci(n-2));
13.     }
14.     // User define function to display factorial of a number.
15.     long long int fact (int n)
16.     {
17.         if (n==1 || n==0)
18.             return 1;
19.         else
20.             return n*fact(n-1);
21.     }
22.
23.     int main()
24.     {
25.
26.         int i,c=0,n;
27.         printf("Enter how many term you want to see : \n");
28.         scanf("%d",&n);
29.         printf("Fibonacci Series: \n");
30.         for (i=1; i<=n; i++)
31.         {
32.             printf("%d\t",fibonacci(c)); // Recursion of [ int
    fibonacci (int n) ]
33.             c++;
34.         }
35.
36.         printf("\n\n");
37.
```

```
38.      /// C Program to print Factorial of a number using
   recursion
39.
40.
41.          {
42.              long long int n;
43.              printf("Enter a num to calc fact :");
44.              scanf("%lld",&n);
45.              long long int factorial= fact(n); // Recursion of [
   long long int fact (int n)) ]
46.              printf("\nFactorial of %d is : %lld",n,factorial);
47.          }
48.          printf("\n\n");
49.          return 0;
50.      }
```

Output:



Discussion and Conclusion:

In this program, I work on a special function that call recursive function. It is such type of function that calls itself in its function. In the first section I made a user define function named Fibonacci() then I made another user define function named fact(). The first program takes input from the user that how many term user want to see of Fibonacci series and in the next program it is also takes a input from them user that at which number user want to see the answer of that number factorial.