# Data Structures – Lesson 4: Recursion

B.Sc. 3rd Semester, Paper C5

Paulami Basu Ray
Assistant Professor
Department of Computer Science & Applications
Prabhat Kumar College, Contai

## What is Recursion?

Any function which calls itself is called recursive.

A recursive method solves a problem by calling a copy of itself to work on a smaller problem.

This is called the recursion step.

The recursion step can result in many more such recursive calls.

## Why Recursion?

- Recursion is a useful technique borrowed from Mathematics.
- Recursive code is generally shorter and easier to write than iterative code.
- Generally loops are turned into recursive functions when they are compiled or interpreted.
- Recursion is most useful for tasks that can be defined in terms of similar subtasks.
- For example, sort, search and traversal problems often have recursive solutions.

## Format of a Recursive Function

```
if( test for the base case )
    return some base case value

else if ( test for another base case value )
    return some other base case value

//the recursive case
else
    return ( some work and then a recursive call )
```

## Example

n! = 1,

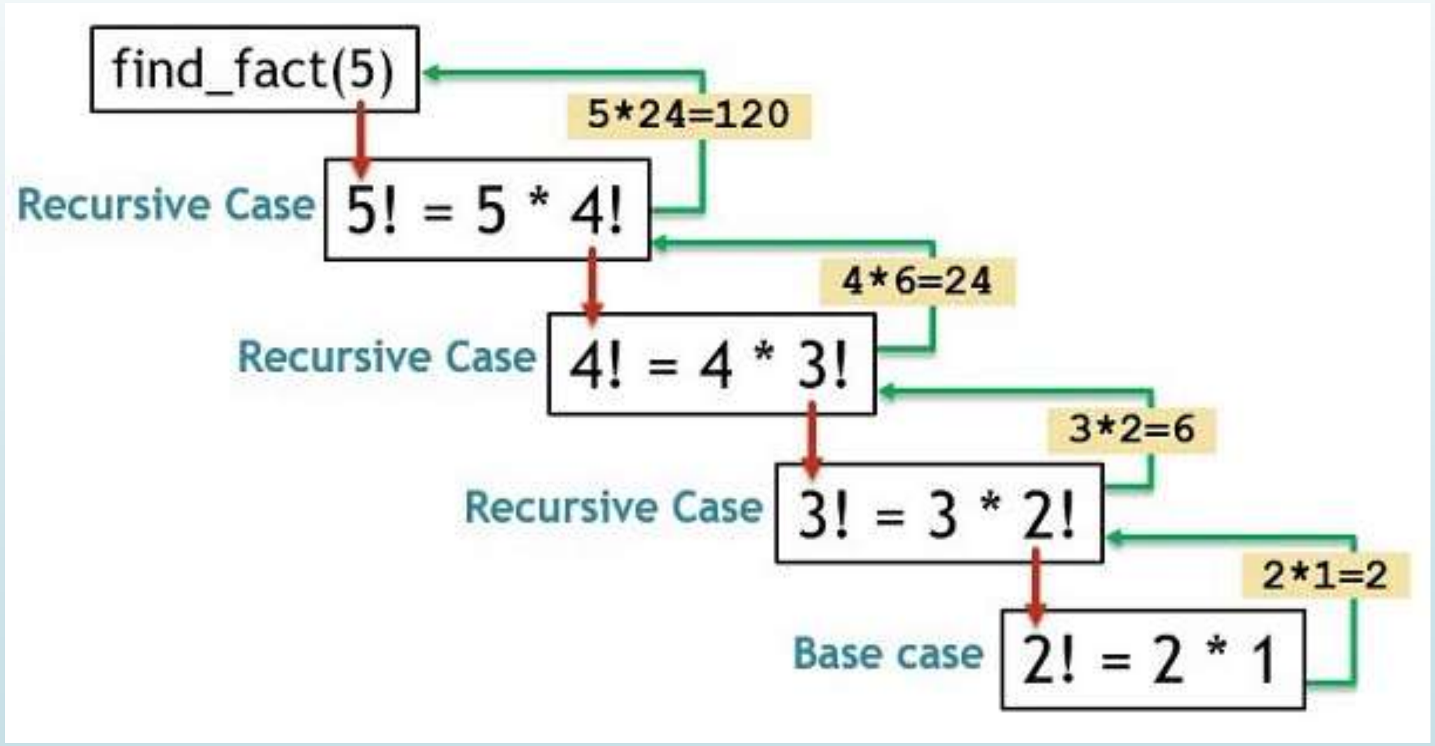
if n = 0

n! = n * (n-1)!

If n > 0

```
int factorial(int n)
{
        if ( n == 1 )
                return 1;
        else
                return ( n * factorial(n-1));
}
```

## Pictorial Representation



# Recursive Program in C to find the sum of n Natural Numbers

```
#include <stdio.h>
int addNumbers(int n);

 void main()
 {
               int num;
               printf("Enter a positive integer: ");
               scanf("%d", &num);
               printf("Sum = %d",
               addNumbers(num));
               r
}
int addNumbers(int n)
 {
               if (n != 0)
                              return n + addNumbers(n - 1);
               else
                              return n;
 }
```