Pre requisites :

- Loops
- Basic java syntax

List of concepts involved :

- Introduction to recursion
- Need of recursion
- Base case and recursive calls
- Given an integer, find its factorial
- Given an integer n , find nth fibonacci number
- Given two integers a and b. Find the value of a^b using recursion.
- Efficient way of power calculation
- Staircase problem

Introduction to recursion

A recursive function solves a particular problem by calling a copy of itself and solving smaller subproblems of the original problems. Many more recursive calls can be generated as and when required. It is essential to know that we should provide a certain case in order to terminate this recursion process. So we can say that every time the function calls itself with a simpler version of the original problem. The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called a recursive function.

It has basically two parts:

- A precondition that is used to stop this recursive call (base Condition).
- A function that is capable of calling itself (recursive call).

Need of Recursion

Recursion is an amazing technique with the help of which we can reduce the length of our code and make it easier to read and write. It has certain advantages over the iteration technique which will be discussed later. A task that can be defined with its similar subtask, recursion is one of the best solutions for it. For example; The Factorial of a number.

Terminating Condition:

Just like how we have a condition in an iterative statement to terminate the loop similarity, it must have a terminating condition to terminate the recursive call; otherwise, it will result in an overflow error as it will go inside an infinite recursive call.

Steps of any generic recursive function:

- **Base case :** this is the smallest subproblem whose answer is known or can be calculated. Beyond this point one needs not to make further calls to the function.
- **Recursive calls:** this is dividing the major problem into smaller chunks to assemble answers for the main, the original problem.