Syntax of any generic recursive code is as follows:

```
methodName(paramters){
    if(baseConditions)return;
    methodName(changedParameters);
}
```

This will get more clear when we will solve some problems based on recursion.

Q1. given an integer 'n'. Find the factorial of 'n'.

Input 1: n = 5
Output 1: 120
Explanation : 5! = 5*4*3*2*1 = 120

Input 1: n = 3
Output 1: 6
Explanation : 5! = 3*2*1 = 6

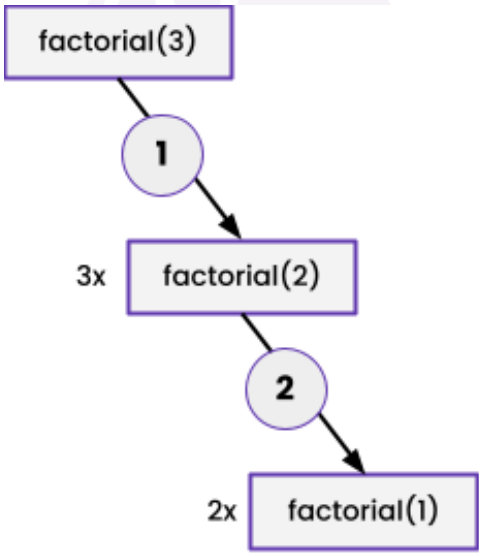
Solution :

Code : LP_Code1.java

Output:

```
Enter the number : 6
The factorial is : 720
```

Approach :



- For any number 'n', we can write its factorial as 'n' multiplied by the factorial of its previous number.
- This is a statement we know to be true. So we recursively call our function to return to us the factorial of the previous number in order to calculate the factorial of the current number by the help of this statement.
- Lets' calculate the factorial of 5 using the recursion code we wrote:
  factorial(5) will call for factorial(4)
  factorial(4) will call for factorial(3)
  factorial(3) will call for factorial(2)

factorial(2) will call for factorial(1)
Now we know the value of factorial(1) to be 1. So we simply return the value i.e. our base case.
Now factorial(1) will return 1. Using this value factorial(2) will be calculated.
factorial(2) = 2 * factorial(1)= 2 * 1 = 2
Similarly,
factorial(3) = 3 * factorial(2) = 3 * 2 = 6
factorial(4) = 4 * factorial(3) = 4 * 6 = 24
factorial(5) = 5 * factorial(4) = 5 * 24 = 120

At the end we have factorial(5)= 120, the desired output.

Q2: given an integer n. Find the nth fibonacci number.

Input1 : n = 7
Output : 13
Explanation : 0 1 1 2 3 5 8 13 21. . . . . .

Input1 : n = 5
Output : 5
Explanation : 0 1 1 2 3 5 8 13 21. . . . . .

Solution:
Code : LP_Code2.java

Output :

```
Enter the number : 6
The nth fibonacci number is : 8
```

Approach :
In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation
Fn = Fn-1 + Fn-2
Here F0=0,F1=1, F2=1 .....
Lets find out the 5th fibonacci term :