

# Recursion (function called itself).

- \* while the function is not finished executing it will remain in stack.
- \* when a function finished executing it is removed from stack and flow of program is restore where the function called.

\* Base condition <sup>second condition</sup> where our recursion stop making now calls.

\* Every call of function will take space memory. Even a same function different function does not matter.

\* If No base condition  $\rightarrow$  function call will keep happening. Stack will be filled again and again. Memory of computer will exceed to limit & Stack overflow take place. (bhot Sara valley tak aagaye)

Q: Print Number from 5 to 1.

```
public class Recursion {  
    public static void main PrintName (int n) {
```

```
        if (n == 0) {  
            return;
```

base case

```
    }
```

```
        System.out.println(n);  
        printNumb (n - 1);
```

Print

recursion

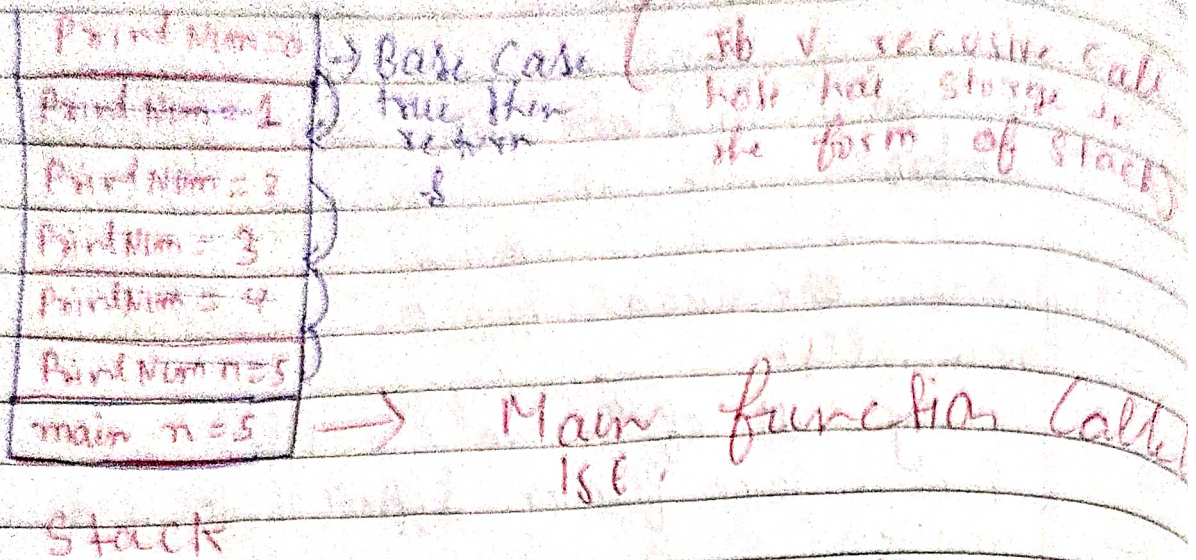
```
}
```

```
public static void main (String args) {
```

```
    int n = 5;  
    printNumb (n); // - 5 } function call
```



\* when ever function is called itself and recursion take place it store in the form of stack



\* always analyse base case.

Q2 print Sum of first n natural numbers.

- ① main fn → 1
- ② Base Case → n (Sum)
- ③ "koam" work → calculate sum

Take value in parameter whose value is change in level call.

Code

```

Public class Recursion 2
function [ Public static void PrintSum (int i, int sum) {
    if (i == n) {
        Sum += i;
        S = O.P (Sum);
    }
}
    
```

Teacher's Signature.....



```

    }
    return;

    Sum += i;
    printSum( i+1, n, Sum );
    }

```

```

P.S.V.M( String args[]) {
    printSum(1, 5, 0);
}
}

```

read the 15)	P.S.i = 5 n = 5 s = 15	✓
base 4	P.S.i = 4 n = 5 s = 10	✓
can 3	P.S.i = 3 n = 5 s = 6	
2	P.S.i = 2 n = 5 s = 3	✓
1	P.S.i = 1 n = 5 s = 1	✓
	main i = 1 n = 5 s = 0	

Q) Hello print using function.

```

Public class Main {
    P.S.V.m( String[] args) {
        message kaho();
    }
}

```

```

Static void message () {
    S.O.P \n ("Hello world");
    message();
}

```