**RECURSION**

Finding Factorial of a number (using call stack)

Every time function calling itself -> there is a different call stack being made

Recursion is function calling itself

And it can be defined as a solution of problem is a combine solution of its sub problem

Q what if you want to print factorial

Int fact(int n)

{

If(n==0)

Return 1;

Int sa=fact(n-1);

Int result=n\*sa;

Return result;

}

Q what if you want to print 1 to n

Int print1(int n)

{

If(n==0)

Return ;

Print1(n-1);

Sop(n);

}

Q what if you want to print n to 1

Int print1(int n)

{

If(n==0)

Return ;

Sop(n);

Print1(n-1);

}

Q what if you want to print n to 1 and 1 to n

Int print1(int n)

{

If(n==0)

Return ;

Sop(n);

Print1(n-1);

Sop(n);

}

Q find element in an array

Boolean Check(int arr[],int element){

If(arr.length==0)

Return false;

If(arr[0]==element)

Return true;

Else{

Int []sa=new int[arr.length-1];

For(int i=1;i<arr.length;i++)

Sa[i-1]=arr[i];

}

Boolean result=check(sa,element);

Return result;

}

Basically in every function calling array is reduced by 1 in size with a loss of first element

Alternative good solution

Boolean Check(int arr[],int element){

If(arr.length==0)

Return false;

If(arr[0]==element)

Return true;

Else{

boolean result=check(sa,element);

}

}