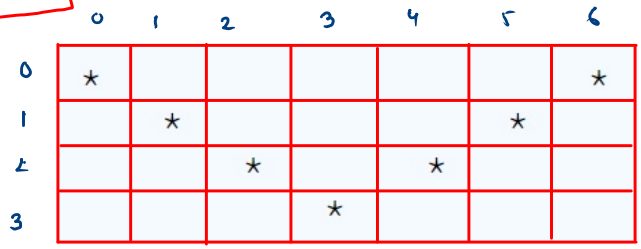
HW_Print V Pattern

n = 7



rows = $\frac{(n+1)}{2}$

indexes of stars = (0,0), (1,1), (2,2), (3,3) [i == j]

(2,4), (1,5), (0,6) [i + j == n - 1]

code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    int rows = (n + 1) / 2;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < n; j++) {
            if ( i == j || i + j == n - 1 ) {
                System.out.print("*\t");
            } else {
                System.out.print("\t");
            }
        }
        System.out.println();
    }
}
```

Find sum using a function

```java
public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int t = scn.nextInt();

        for (int i = 0; i < t; i++) {
            int x = scn.nextInt();
            int y = scn.nextInt();

            sum(x, y);
        }
    }

    public static void sum(int x, int y) {
        int ans = x + y;
        System.out.println(ans);
    }
}
```

⇒ Type of functions

parameters

↳ Paramaterized :-

```
public static void sum(int a, int b){
    Syso(a+b);
}
```

↳ Non-Paramaterized :-

```
public static void printHi(){
    Syso("Hi");
}
```

Code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);

    int t = scn.nextInt();
    for (int i = 0; i < t; i++) {
        int x = scn.nextInt();
        int y = scn.nextInt();

        findProduct(x, y);    // function calling
    }
}

public static void findProduct(int x, int y) {    // function declaration
    System.out.println(x * y);
}
```

Factorial of N

```
int n = 7;
```

factorial (n!) = n * (n-1) * (n-2) * (n-3) * ..... * 3 * 2 * 1

7! = 7 * 6 * 5 * 4 * 3 * 2 * 1

psudo code

```
n = 7;
ans = 1;
for( int i = 1 ; i <= n ; i++ ){
    ans = ans * i ;
}
```

ans = 1

i = 1, ans = 1

i = 2, ans = 1 * 2

i = 3, ans = 1 * 2 * 3

i = 4, ans = 1 * 2 * 3 * 4

i = 5, ans = 1 * 2 * 3 * 4 * 5

i = 6, ans = 1 * 2 * 3 * 4 * 5 * 6

i = 7, ans = 1 * 2 * 3 * 4 * 5 * 6 * 7

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    fact(n);
}

public static void fact(int n) {
    long ans = 1;
    for (int i = 1; i <= n; i++) {
        ans = ans * i;
    }
    System.out.println(ans);
}
```

Find nCr.

$$^{n}C_{r} = \frac{n!}{(n-r)! * r!}$$

$a = n!$

$b = (n-r)!$

$c = r!$

$$ans = \frac{a}{(b * c)}$$

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int r = scn.nextInt();

    findComb(n, r);
}

public static void findComb(int n, int r) {
    int a = fact(n);
    int b = fact(n - r);
    int c = fact(r);

    int ans = a / (b * c);
    System.out.println(ans);
}

public static int fact(int n) {
    int ans = 1;
    for (int i = 1; i <= n; i++) {
        ans = ans * i;
    }
    return ans;
}
```

⇒ Return type

↳ **Return type f^n :-** f^n which returns something (int, boolean, String, char, ...)

↳ **void type f^n :-** f^n which return nothing (void)

Note:- any question can be solved using return type f^n as well as void type f^n.