

⇒ Revision

- Programming language & Operators
- Variables
- Conditions and logical operators
- if else
- nested if else
- Switch statement
- Characters and Strings
- * → for loops
- * → while & do while loop
- * → Patterns
- * → functions (return statement)
- digit traversal (%10, /10) & number theory
- ★★ → Arrays (Printing, finding & searching, storing info, updating)
- * → Brute force (Permutation & Combination)
- * → Time & space Complexity

\Rightarrow ~~loops~~ (to perform some task certain no. of time)

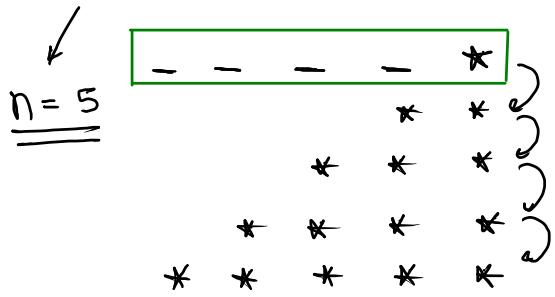
Syntax

1) `for (int i = 0 ; i < n ; i++) {
 // statement
}`

2) `int i = 0 ; // init.
while (i < n) { // cond
 // statement
 i++ ; // upgradat
}`

3) `int i = 0 ;
do {
 // statement
 i++ ;
} while (i < n);`

\Rightarrow Patterns



Template

```
int st = 1; }  
int sp = n-1; }  
for (int i=0; i<n; i++){  
    for (int j=0; j<sp; j++){  
        System.out.print(" ");  
    }  
    for (int j=0; j<st; j++){  
        System.out.print("*");  
    }  
    sp--; }  
    st++; }  
System.out.println();
```

HW_Print Inverted Right Angled Triangle Pattern

n=7

```
* * * * * * *
```

```
* * * * * *
```

```
* * * * *
```

```
* * * *
```

```
* * *
```

```
*
```

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
  
    int st = n;  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < st; j++) {  
            System.out.print("*");  
        }  
        st--;  
        System.out.println();  
    }  
}
```

Hw_Print Spaced Right-angled whole numbers

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
  
    int st = 1;  
    int sp = n - 1;  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < sp; j++) {  
            System.out.print(" ");  
        }  
        for (int j = 0; j < st; j++) {  
            System.out.print((j + 1) + " ");  
        }  
        sp--;  
        st++;  
        System.out.println();  
    }  
}
```

$n=5$, $st=1$, $sp=43210$

$i=0$,
 $i=1$,
 $i=2$,
 $i=3$,
 $i=4$,
 $i=5 < 5 \times$

3 4 5 6
-1

1 2 3 4 5

Hw_Print Inverted triangle

$n=7$

0	*	*	*	*	*	*	*
1		*	*	*	*	*	
2			*	*	*		
3				*			

$$\text{Rows} = \frac{n}{2} + 1$$

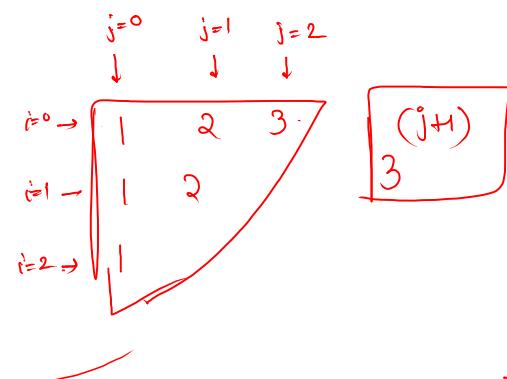
```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int st = n;
    int sp = 0;
    int row = (n / 2) + 1;
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < sp; j++) {
            System.out.print("\t");
        }
        for (int j = 0; j < st; j++) {
            System.out.print("*\t");
        }
        sp++;
        st -= 2;
        System.out.println();
    }
}
```

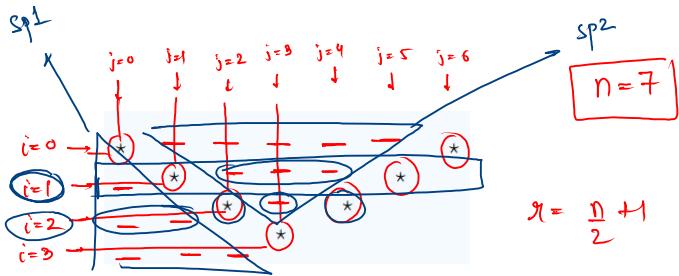
$n=2$

2	4
2	

$n=3$

3 9 81
3 9
3





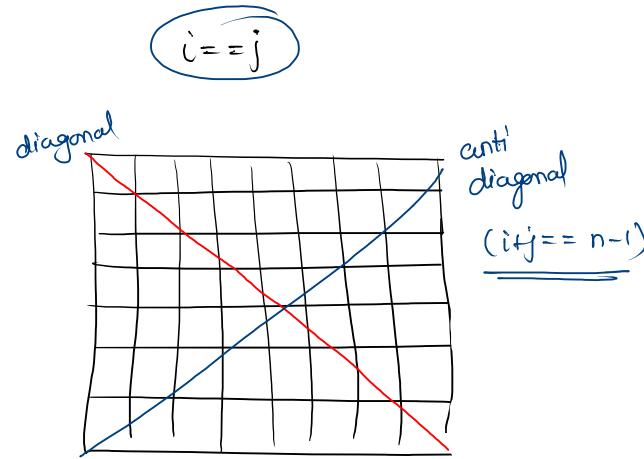
$$g = \frac{n}{2} + 1$$

$(0,0)$
 $(1,1)$
 $(2,2)$
 $(3,3)$

$i \quad j$
 $(2,4)$
 $(1,5)$
 $(0,6)$

$$\boxed{i+j = (n-1)} \quad \star \star$$

$$\left. \begin{array}{l} 2+4 = 6 \\ 1+5 = 6 \\ 0+6 = 6 \end{array} \right\}$$



Code

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
  
    int row = (n / 2) + 1;  
    int sp1 = 0;  
    int sp2 = n - 2;  
    for (int i = 0; i < row; i++) {  
        for (int j = 0; j < sp1; j++) {  
            System.out.print("\t");  
        }  
        → System.out.print("*\t");  
        for (int j = 0; j < sp2; j++) {  
            System.out.print("\t");  
        }  
        if (i != row - 1) {  
            System.out.print("*\t");  
        }  
        → sp1++;  
        → sp2 -= 2;  
        System.out.println();  
    }  
}
```

$$\underline{\underline{n=7}} \quad , \quad \underline{m_0\omega = 4}$$

Sp1 = ~~0 1 2 3 4~~

$$\underline{\underline{sp^2 = S \times X - 1}} - 3$$

The diagram illustrates a sequence of binary digits (0s and 1s) grouped by index i :

- $i=0$: $\star \text{ --- --- --- --- --- --- --- } \star$
- $i=1$: $\{ \text{---} \star \text{ --- --- --- --- } \star \}$
- $i=2$: $\{ \text{--- ---} \star \text{ --- } \star \}$
- $i=3$: $\{ \text{--- --- ---} \star \}$ (This row is circled in red)
- $i=4$: $\{ \}$

Below the sequence, the label $M_{0w} =$ is written.

$$M_{\text{GW}} = 4$$

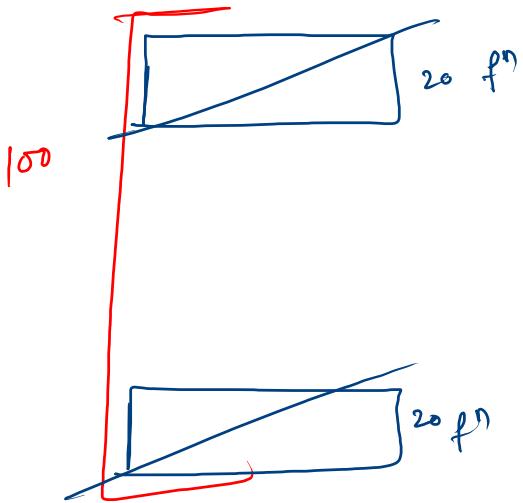
Code

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
  
    int row = (n / 2) + 1;  
    for (int i = 0; i < row; 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();  
    }  
}
```

\Rightarrow functions

(piece of code)

(reusability)



Syntax

```
int public static void fun1( ) { // declaration
    [ ] // statements ] 1000
    } return; ← Last line
```

```
fun1(); // calling
```

```
p. $ String fun1( int x ) {  
    if ( x == 5 ) {  
        return "Hello";  
    }  
    return " ";  
}
```

garbage value

```
String ans = fun1( 6 );  
if ( ans != " " )  
    System.out.println( ans );  
{
```

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = { 2, 3, 4, 3, 2, 1 };
    boolean ans = fun1('n');
    // if ( ans == true ) {
        System.out.println(ans);
    // }
}

public static boolean fun1(char n) {
    if ( n == '5' ) {
        System.out.println("Hi");
        return true;
    }
    return false;
}
```

var name
↓ ↓
str.toUpperCase();

