

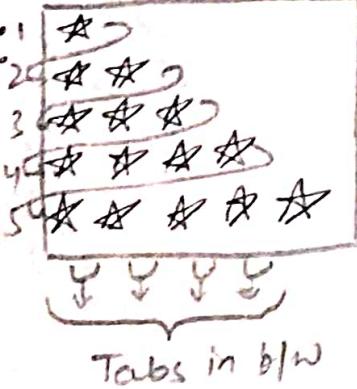
## PATTERN 1

I/P: 5

i \ j | 1 2 3 4 5

n = 5

O/P:



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

```
Scanner sc = new Scanner (System.in);
```

```
int n = sc.nextInt();
```

```
for (int i=1; i<=n; i++)
```

```
{ for (int j=1; j<=i; j++)
```

```
{ System.out.print ("* ");
```

```
}
```

```
System.out.println ();
```

```
}
```

```
}
```

## PATTERN 2

```
public static void main (String [] args)
```

```
{ Scanner s = new Scanner (System.in);
```

```
int n = s.nextInt();
```

```
for (int i=n; i>=1; i--)
```

```
{ for (int j=1; j<=i; j++) // MAGIC loop
```

```
{ System.out.print ("* \t");
```

```
}
```

```
System.out.println ();
```

```
}
```

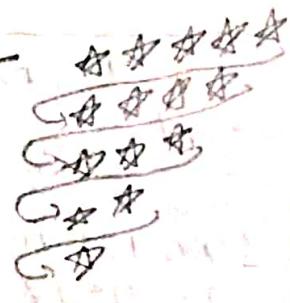
```
}
```

tab space



loop ends

n	i	j	
5	5	1	
	4	2	
	3	3	
	2	4	
	1	5	



### PATTERN 3

```

public static void main (String [] args)
{
    Scanner s = new Scanner (System.in);
    int n = s.nextInt(); → 5
    int sp = n-1; // Sp > Spaces 4
    int st = 1; // St = Star 1
    for (int i=1; i<=n; i++)
    {
        System.out.println(sp+" "+st);
        sp--;
        st++;
    }
}

```

Date: 8<sup>th</sup> Sept '2021  
lecture: 7

4 space 1 star
3 space 2 star
2 space 3 star
1 space 4 star
0 space 5 star

n	sp	st	i
5	4	1	1
	3	2	2
	2	3	3
	1	4	4
	0	5	5

loop ends

O/P: 4, 1  
3, 2  
2, 3  
1, 4  
0, 5

### FINAL CODE

```

public static void main (String [] args)
{
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int sp = n-1; // Space
    int st = 1; // Star
    for (int i=1; i<=n; i++)
    {
        for (int j=1; j<=sp; j++) // run till the
            // no. of spaces on
            // particular
            // line
        {
            System.out.print (" ");
            // Print the no. of space required
        }
        for (int j=1; j<=st; j++) // run till the no. of stars on particular
            // line
        {
            System.out.print ("★");
            // Print the no. of stars required
        }
        sp--; // Decrementing the spaces by 1 for next line.
        st++; // Incrementing the stars by 1 for next line.
        System.out.println (); // Adding a new line for output on
                               // console
    }
}

```

m	sp	st
5	4	1
	3	2
	2	3
	1	4
	0	5

O/P: 5

-----★
---★---
--★--★-
-★--★--★-
★--★--★--★-

## PATTERN 4

public static void main (String [] args)

```

    {
        Scanner s=new Scanner (System.in);
        int n=s.nextInt(); → 5
        int sp=n; ⚡ 4 3 2 1 0
        int st=0; ⚡ 1 2 3 4 5
        for (int i=1; i<=n; i++)
    }

```

```

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

```

→ NO. ALWAYS WILL BE ODD

## PATTERN 5 → ASSUMPTION



2Sp 1st  
1Sp 3St ↑ ↓  
0Sp 5St ↑ ↓  
1Sp 3St ↓ ↑  
2Sp 1st ↓ ↑

change  
in  
numbers

public static void main (String [] args)  
Scanner s=new Scanner (System.in);  
int n=s.nextInt();  
int sp=n/2;  
int st=1;

for (int i=1; i<=n; i++) {

System.out.println(sp+" "+st);

if (i<=n/2)

{ sp--; st=st+2; }

else {

sp++; st=st-2; }

	n	sp	st	i
5	5	2	1	+
2, 1	4	3	2	
1, 3	3	5	3	
0, 5	2	3	4	
1, 3	2	1	3	
2, 1	3	0	6	and { }

QFP

0Sp	5St
1Sp	4St
2Sp	3St
3Sp	2St
4Sp	1St

m	st	sp	i	j
5	5	0	2	2
	4	1	3	3
	3	2	4	4
	2	3	5	5
	1	4	6	6
	0	5	loop ends	

* * * *	*
- - * * *	*
- - - * *	*
- - - - *	*
- - - - -	

→ NO. ALWAYS WILL BE ODD

## PATTERN 5 → ASSUMPTION



2Sp 1st  
1Sp 3St ↑ ↓  
0Sp 5St ↑ ↓  
1Sp 3St ↓ ↑  
2Sp 1st ↓ ↑



Scanner s=new Scanner (System.in);  
int n=s.nextInt();  
int sp=n/2;  
int st=1;

for (int i=1; i<=n; i++) {

System.out.println(sp+" "+st);

if (i<=n/2)

{ sp--; st=st+2; }

else {

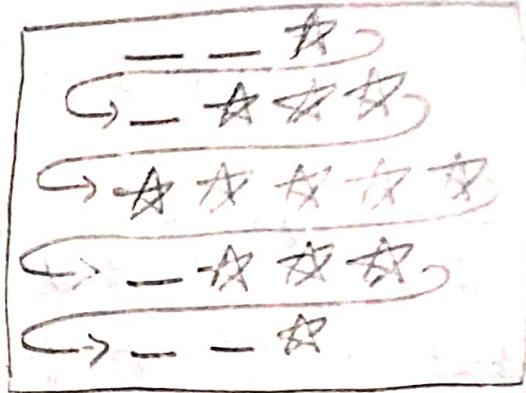
sp++; st=st-2; }

```
public static void main (String [] args)
```

```
{
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int sp = n/2; // spaces
    int st = 1; // stars
    for (int i=1; i<=n; i++)
    {
        for (int j=1; j<=sp; j++) // printing space
        {
            System.out.print(" ");
        }
        for (int j=1; j<=st; j++) // printing stars
        {
            System.out.print("*");
        }
        if (i <= n/2)
        {
            sp--; // decrementing sp by 1
            st = st + 2; // incrementing st by 2
        }
        else
        {
            sp++; // incrementing sp by 1
            st = st - 2; // decrementing st by 2
        }
        System.out.println(); // newline
    }
}
```

n	i	j	sp	st
5	1	1	2	1
	2	2	1	3
	3	3	0	5
	4	4	1	3
	5	5	2	1

6 loop end.



DIAMOND PATTERN = SQUARE IN SHAPE  
HOLLOW DIAMOND PATTERN = RECTANGLE IN SHAPE

## PATTERN 6

①	★ ★ ★ - - ★ ★ ★
②	★ ★ - - - ★ ★
③	★ - - - - ★
④	★ ★ - - - ★ ★
⑤	★ ★ ★ - - ★ ★ ★

3st 1sp 3st  
2st 3sp 2st  
1st 5sp 1st  
0st 3sp 2st  
3st 1sp 3st

### CONSTRAINTS

1 <= n <= 10

n = odd

```

public static void main (String [] args)
{
    Scanner s = new Scanner (System.in);
    int n = s.nextInt(); → 5
    int st = n/2 + 1;
    int sp = 1;
    for (int i=1; i<=n; i++)
    {
        System.out.println(st+" "+sp+" "+st);
        if (i<=n/2)
        {
            st--;
            sp=sp+2;
        } else
        {
            st++;
            sp=sp-2;
        }
    }
}

```

n	sp	st	
5	1	3	
	3	2	
	5	X	
	3	2	
	1	3	

O/P

5	3, 1, 3
	2, 3, 2
	1, 5, 1
	2, 3, 2
	3, 1, 3

### FINAL CODE

```

public static void main (String [] args)
{
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int st = n/2 + 1;
    int sp = 1;
    for (int i=1; i<=n; i++)
    {
        //stars for (int j=1; j<=st; j++)
        loop: {
            System.out.print("*\t");
            for (int j=1; j<=sp; j++)
                System.out.print(" \t");
            for (int j=1; j<=st; j++)
                System.out.print("*\t");
            if (i<=n/2)
            {
                st--;
                sp+=2;
            } else
            {
                st++;
                sp-=2;
            }
        }
    }
}

```



```

System.out.println();
}
}

```

## PATTERN 7

① public static void main (String [] args) {  
 Scanner s = new Scanner (System.in);  
 int n = s.nextInt();  
 for (int i=1; i<=n; i++) {  
 for (int j=1; j<=n; j++) {  
 if (System.out.print ("\* \t"),  
 System.out.println());  
 }
 }
}

j ① ② ③ ④ ⑤

② Scanner s = new Scanner (System.in);  
 int n = s.nextInt();  
 for (int i=1; i<=n; i++) {  
 for (int j=1; j<=n; j++) {  
 if (i==j)  
 System.out.print ("\* \t");  
 else  
 System.out.print (" \t");  
 System.out.println();  
 }
 }
}

O/P :

```

* * * * *
*   *   *
*     * 
*   *   *
* * * * *

```

O/P

```

A
-
-   *
- -   *
- - -   *
- - - -   *

```

FINAL CODE (OPTIMIZED) → PASTER

```

public static void main (String [] args) {  

    Scanner s = new Scanner (System.in);  

    int n = s.nextInt();  

    for (int i=1; i<=n; i++) {  

        for (int j=1; j<=n; j++) {  

            if (i==j){  

                System.out.print ("* \t");  

                break;  

            } else {  

                System.out.print ("\t");  

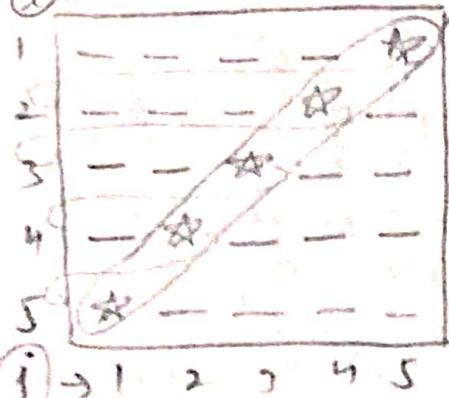
            } System.out.println();  

        }
    }
}

```

## PATTERN 8

Q+



$$i+j = n+1$$

```

public static void main (String [] args)
{
    Scanner s = new Scanner (System.in);
    int n = s.nextInt ();
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            if (i + j == n + 1)
                System.out.print ("# \t");
            else
                System.out.print ("\t");
        }
        System.out.println ();
    }
}

```

PATTERN 9

$\star$	$(0, 1)$	-	-	$\star$	$(1, 5)$
-	$\star$	$(2, 2)$	-	$\star$	$(2, 4)$
-	-	$\star$	$(3, 3)$	-	-
-	$\star$	$(4, 2)$	-	$\star$	$(4, 4)$
$\star$	$(5, 1)$	-	-	$\star$	$(5, 5)$

$$\text{Let } c = 100 \\ n = \text{odd}$$

THIS PATTERN IS COMBINATION OF PATTERN 7 + PATTERN 8

NOTE: WE WILL USE (OR) OPERATOR. FOR CONDITIONS.

for (int j = 0; j < n; j++) prints the stars in the row

YA FIR

$i \neq j$  ( $i+j = n+1$ )

```
System.out.print("A\\t");
```

30

System.out.print("\t");

29

stem.out.println(); // Change

`System.out.println();` Change the row  
on output console

CONDITION  
OF  
PATTERN 9

PATTERN 9

卷之三

八

Re you

at congo

卷之三

卷之三

## PATTERN 10

Date : 9<sup>th</sup> Sept '2021  
lecture : 8



OUTER SPACE	INNER SPACE
2	-1
1	1
0	3
1	1
2	-1

Code

```
public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int os = n/2; //Outer spacing > half of n
    int is = -1; //Same as 0; not to print anything
    for (int i=1; i<=n; i++) {
        System.out.print(os + " " + is);
        if (i <= n/2) {
            os--;
            is = is + 2;
        } else {
            os++;
            is = is - 2;
        }
    }
}
```

```
if (i <= n/2) {
    os--;
    is += 2;
} else {
    os++;
    is -= 2;
}
System.out.println();
```

(O/P : 5)

2, -1  
1, 1

0, 3  
1, 1

2, -1

FINAL CODE

```
public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int os = n/2;
    int is = -1;
    for (int i=1; i<=n; i++) {
        for (int j=1; j<=os; j++)
            System.out.print(" \t");
        System.out.print("* \t");
        for (int j=1; j<=is; j++)
            System.out.print(" \t");
        if (i>1 && i<n) {
            System.out.print("* \t");
        }
    }
}
```

## PATTERN 11 → CONSTRAINTS: $1 \leq n \leq 44$

1			
2	3		
4	5	6	
7	8	9	10

*			
*	*		
*	*	*	
*	*	*	*

FINAL CODE 3

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

```
public static void main(String[] args)
{
    Scanner s = new Scanner(System.in);
    int n = s.nextInt();
    int val = 1;
    for (int i=1; i<n; i++)
    {
        for (int j=1; j<=i; j++)
        {
            System.out.print(val + "\t");
            val++;
        }
        System.out.println();
    }
}
```

## PATTERN 12 → CONSTRAINTS: $1 \leq n \leq 5$

0			
1	1		
2	3	5	
8	13	21	34

```
public static void main(String[] args)
{
    Scanner s = new Scanner(System.in);
    int n = s.nextInt();
    int a = 0;
    int b = 1;
    for (int i=1; i<n; i++)
    {
        for (int j=1; j<=i; j++)
        {
            System.out.print(a + "\t");
        }
        System.out.println();
    }
}
```

↓  
FIBONACCI PATTERN

0	a	0	④
1	b	1	
1	c	2	
2	d	3	
3	e	4	
5	f		
8	g		
13	h		
21	i		
34	j		

```
for (int j=1; j<=i; j++)
{
    System.out.print(a + "\t");
    int c = a+b;
    a = b;
    b = c;
}
System.out.println();
```

PATTERN 13 → CONSTRAINTS:  $1 \leq n \leq 10$

1												
1	1											
1	2	1										
1	3	3	1									
1	4	6	4	1								

DESIRED OUTPUT PATTERN

TO ACHIEVE OUR DESIRED PATTERN WE WILL WORK IN STAGES.

★	★	★	★	★	★	★	★	★	★	★	★	★
★	★	★	★	★	★	★	★	★	★	★	★	★
★	★	★	★	★	★	★	★	★	★	★	★	★
★	★	★	★	★	★	★	★	★	★	★	★	★
★	★	★	★	★	★	★	★	★	★	★	★	★
★	★	★	★	★	★	★	★	★	★	★	★	★

STAGE ① ←

```

Scanner s = new Scanner(System.in);
int n = s.nextInt();
for (int i=1; i<=n; i++) {
    for (int j=1; j<=i; j++) {
        System.out.print("*\t");
    }
    System.out.println();
}
    
```

$i \downarrow \quad j \rightarrow \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5$

0	1	1	1	1	1
1	1	2	1	1	1
2	1	3	3	1	1
3	1	4	6	4	1
4	1	5	10	10	5
5	1	6	15	20	15

${}^5C_0 \quad {}^5C_1 \quad {}^5C_2 \quad {}^5C_3 \quad {}^5C_4 \quad {}^5C_5$

[1, 5] = 5 Bands  
 1, 2, 3, 4, 5

[0, 5] = 5 Bands  
 0, 1, 2, 3, 4

HAR ROW KI PHELI  
 AUR AKHRI VALUE ①  
 HAI AND YEH  
 SYMMETRICAL HAI

We know: Binomial theorem

$${}^nC_0 = {}^nC_n = 1$$

$${}^nC_k = {}^nC_{n-k}$$

$${}^nC_k = \frac{n!}{k!(n-k)!}$$

# ROW - C - CALL

$i = 0$  se  $i < n$  tak jayega.  
 $j = 0$  se  $j \leq i$  tak jayega  $\rightarrow ? \rightarrow kn \cdot JAYEGA$

0	1	0	0	0
1	1	1	0	0
2	1	2	1	0
3	1	3	3	1
4	1	4	6	4
5	1	5	10	10
	0	1	2	3
	4	5		

\* 1<sup>st</sup> row me  $j$  ne 2 Bande point kri hei  
ek nahi!  
\* 2<sup>nd</sup> row me jab  $i = 2$  toh  $j$  ne 3 Bande  
point kri hai!  
ISI LIYE  $j = 0$  se  $j < i$  tak jayega  
Total  $(i+1)$  bande point honge.  
WASA HO BHI RAHA!

- \* NON  $i$  and  $j$  control me agaye hai!
- \* Har row ka phela banda toh  ${}^n C_0$  hai! Becoz  $j=0$
- \* Toh  ${}^n C_0 = 1$  (ALWAYS). HAMESHA  $VAL = \text{value ko}$
- $i$  leke challenge.

Formula

$${}^n C_{k+1} = \frac{{}^n C_k \cdot (n-k)}{k+1}$$

According to our code :

$${}^i C_{j+1} = \frac{{}^i C_j \cdot (i-j)}{j+1}$$

ISS FORMULA SE  
HUM  
 ${}^5 C_0$  se  ${}^5 C_1$   
Calculate kr  
Saktey hai!

$$\rightarrow {}^5 C_1 = \frac{{}^5 C_0 \cdot (5-0)}{0+1} = 5$$

```

public static void main (String [] args)
{
    Scanner s = new Scanner (System.in);
    int n = s.nextInt ();
    for (int i=0; i<n; i++)
    {
        int icj = 1; //  ${}^n C_0 = 1$ 
        for (j=0; j<=i; j++)
        {
            System.out.print (icj + " \t");
            int icjpl = icj * (i-j) / (j+1); //  ${}^n C_k = \frac{{}^n C_{k-1}}{k+1}$ 
            icj = icjpl;
        }
        System.out.println ();
    }
}

```

④ PATTERN 14 → CONSTRAINTS:  $1 \leq n \leq 10$

$$\begin{aligned} x * 1 &= x \\ x * 2 &= 2x \\ x * 10 &= 10x \end{aligned}$$

Q/P: (5)

$$\begin{aligned} 5x * 1 &= 5 \\ 5x * 2 &= 10 \end{aligned}$$

$$5x * 10 = 50$$

```

public static void main (String [] args)
{
    Scanner s = new Scanner (System.in);
    int n = s.nextInt ();
    for (int i=1; i<=n; i++)
    {
        int v = x * ja;
        System.out.println (x + " x " + ja + "=" + v);
    }
}

```

③ PATTERN 15 → CONSTRAINTS: LEN 6 | MODE 1

DESIRED OUTPUT

-	-	1			
-	2	3	2		
3	4	5	4	3	
-	2	3	2		
-	-	1			

TO REACH OUR DESIRED OUTPUT PATTERN WE NEED TO BUILD LOGIC STATE BY STATE.

INITIAL STATE → FINAL STATE  
LOGIC BUILDING

-	-	★			
-	★	★	★		
★	★	★	★	★	
★	★	★			
-	-	★			

-	-	1	2	3	2
-	1	1	1	1	1
-	-	1	1	1	1
-	-	1			
-	-				

-	-	1	2	2	2
-	2	3	3	3	3
3	4	4	4	4	4
-	-	5			
-	-				

① BEGIN

②

③

-	-	1			
-	2	3	2		
3	4	5	4	3	
-	2	3	2		
-	-	1			

⑥ FINAL

-	-	1			
-	2	3	4		
3	4	5	6	7	
-	2	3	4		
-	-	1			

⑤

-	-	1			
-	2	2	2		
3	3	3	3	3	
-	2	2	2		
-	-	2			

④

-	-	★			
-	★	★	★		
★	★	★	★	★	
★	★	★	★		
-	-	★			

←

LET'S START, FOCUS ON SHAPE.  
BEGIN ①

```
int n = s.nextInt();
int sp = n/2;
int st = 1;
for(int i=1; i<=n; i++){
    for(int j=1; j<=sp; j++){
        System.out.print("\t");
    }
    for(int j=1; j<=st; j++){
        System.out.print("A\t");
    }
    if(i<=n/2){
        sp--;
        st+=2;
    } else {
        sp++;
        st-=2;
    }
    System.out.println();
}
```

**2**

- - 1
- 1 1 1
1 1 1 1 1
- 1 1 1 1 1
- - 1

Adding `(val++)` above the last `System.out.println();`  
we will get

**3**

- - 1 2
- 2 2 2 3
3 3 3 3 3
- 4 4 4 5
- - 5

`if (i <= n/2) {  
 sp--;  
 st += 2;  
 val++;  
}  
else {  
 sp++;  
 st -= 2;  
 val--;`

**4**

- - 1 2
- 2 2 2 3
3 3 3 3 3
- 2 2 2
- - 1

THIS PART OF CODE WILL GIVE ME

**5** OUTPUT, TOH COLUMN PE FOCUS,  
AB CHAIYE MERERO (CVAL) banaliya jo val se start hoga.

cval → 1st row → val = 1 → toh cval = 1 se start hoga  
                   2nd row → val = 2 → toh cval = 2  
                   3rd row → val = 3 → toh cval = 3  
                   4th row → val = 2 → toh cval = 2  
                   5th row → val = 1 → toh cval = 1

Andhe cval ko print krenge aur usse increase krenge  
 Row pe khelega val aur column pe khelega huma  
 cval, vo val ki value se start hoga aur increase hoga.

```

int n = s.nextInt();
int sp = n/2;
int st = 1;
int val = 1;
for(int i=1; i<=n; i++) {
    for(int j=1; j<=sp; j++) {
        System.out.print("\t");
    }
    for(int j=1; j<=st; j++) {
        System.out.print(" " + val + "\t");
    }
    if (i <= n/2) {
        sp--;
        st += 2;
    } else {
        sp++;
        st -= 2;
    }
    System.out.println();
}
  
```

```

int eval = val;
for (int j=1; j <= st; j++) {
    System.out.print(eval + " * ");
    eval++;
}

```

ISS CHANGE IN  
CODE SE HUME  
⑤ Output milega

⑤

→ AB HUME CHAIYE  
IS CODE KA.

→ TOH HUM GK CHOTA SA CHANGE  
KARENKY.

-	-	1
-	2	3
3	4	5
6	7	
-	2	3
-	-	1

ISS CODE CHAIYE  
KA USE KRKE  
HUME HUMARA  
DESIRED PATTERN  
MILENA

-	-	1
-	2	3
3	4	5
4	3	2
-	2	3
-	-	1

FINAL OUTPUT

```

int eval = val;
for (int j=1; j <= st; j++) {
    System.out.print(eval + " * ");
    if (j <= st/2) {
        eval++;
    } else {
        eval--;
    }
}

```

⑥ FINAL OUTPUT

→ FINAL

CODE



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

Date: 10 Sept '2021  
lecture: 9

**PATTERN 16**

1	-	-	-	-	-	1
1	2	-	-	-	2	D
1	2	3	-	3	2	L
1	2	3	4	3	2	1

$$\boxed{\text{Total columns} = 2^n - 1}$$

1, 5, 1  
2, 3, 2  
3, 1, 3  
4, -1, 4

中———中  
中中———中中  
中中中———中中中  
中中中中中中中中

$\downarrow$   
last row che  
if ( $i == n$ ) {  
    st --;  
}

☆ - - - - ☆  
☆☆ - - ☆☆  
☆☆☆ - ☆☆☆  
☆☆☆☆☆☆☆

$$\begin{array}{r}
 1 - \underline{\quad \quad \quad} \\
 12 - \underline{\quad \quad} 2 \\
 123 - \underline{3} 2 \\
 123 \qquad \qquad \qquad \underline{3} 2
 \end{array}$$

error

CONSTRAINTS:  $1 \leq n \leq 10$

No. of spaces in row 1 =  $2n - 3$

n rows

```

public static void main (String [] args) {
Scanner s = new Scanner (System.in);
int n = s.nextInt();
int sp = 2n-3;
int st = 1;
for (int i=1; i<=n; i++) {
    for (int j=1; j<=st; j++)
        System.out.print ("*\\t");
    for (int j=1; j<=sp; j++)
        System.out.print ("\\t");
    for (int j=1; j<=st; j++)
        System.out.print ("*\\t");
    st++;
    sp -= 2;
    System.out.println();
}
}

```

```
for (int i=1; i<n; i++) {
    int val = 1;
    for (int j=1; j <= st; j++) {
        System.out.print(val + "\t");
    }
}
```

```
for (int j=1; j<=sp; j++) {
```

```
System.out.print("At ");
```

if ( $i == n$ ) { // last row check  
    st --;

for (int j=1; j<=8t; j++) {  
    val--;  
    cout << point (\*va

## FINAL CODE

```

public static void main (String [] args) {
Scanner s=new Scanner (System.in);
int n = s.nextInt();
int sp = 2 * n - 3;
int st = 1;
for (int i=1; i<=n; i++) {
    int val=1;
    for (int j=1; j<=st; j++)
        System.out.print ("val + " + "\t");
    val++;
    for (int j=1; j<=sp; j++)
        System.out.print ("\t");
    if (i==n) {           Error Resolving
        st--;
        val--;             here
    }
    for (int j=1; j<=st; j++)
        val--;
    System.out.print ("val + " + "\t");
}
st++;
sp = 2;
System.out.println();
}

```



CONSTRAINTS:  $1 \leq n \leq 10$   
 $n \rightarrow \text{odd}$

## PATTERN 17



2 Sp	1 *
2 Sp	2 *
0 Sp	5 *
2 Sp	2 *
2 Sp	1 *

$n/2$  spaces  
 Stars  $\uparrow$  by 1 till  $n/2+1$  row  
 Stars  $\downarrow$  by 1 after  $n/2$  row  
 Special row

```

public static void main (String [] args)
{
    Scanner s = new Scanner (System.in);
    int n = s.nextInt ();
    int sp = n / 2; ] Initialization & variable to store space count
    int st = 1; ] of sp and st // variable to store star count
    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= sp; j++) // printing whitespace
        {
            System.out.print ("\\t");
        }
        for (int j = 1; j <= st; j++) // printing stars
        {
            System.out.print ("*\\t");
        }
        if (i <= n / 2) // checking if less than or
                         equal to middle row
        {
            st++; } // ① star till middle row
        else
        {
            st--; } // decreasing stars post
                         middle row.
        System.out.println (); // changing the
                         row
    }
}

```

Adding this block  
of code to previous  
code we can reach  
our desired pattern



```

for (int j = 1; j <= sp; j++)
{
    if (i == n / 2 + 1)
    {
        System.out.print ("*\\t");
    }
    else
    {
        System.out.print ("\\t");
    }
}

```



PATTERN 18 → CONSTRAINTS :  $1 \leq n \leq 10 \therefore n = \text{odd}$ )

★	★	★	★	★	★	★	★	★	★	★
-	★	-	-	-	★	-	-	-	-	-
-	-	★	-	-	★	-	-	-	-	-
-	-	-	★	-	-	-	-	-	-	-
-	-	-	★	★	★	-	-	-	-	-
-	★	★	★	★	★	-	-	-	-	-
★	★	★	★	★	★	★	★	★	★	★

Phle  
Yeh  
Banao  
fir  
Desired  
Pattern  
obtain  
Karo

★	★	★	★	★	★	★	★	★	★	★
-	★	-	★	-	★	-	★	-	★	-
-	-	★	-	★	-	★	-	★	-	-
-	-	-	★	-	-	-	★	-	-	-
-	-	-	★	★	-	-	★	-	-	-
-	★	★	★	★	-	-	★	-	-	-
★	★	★	★	★	★	★	★	★	★	★

AFTER this, we will remove  
the stars from upper half  
of the hourglass pattern.

# Start

★	★	★	★	★	★	★
-	★	★	★	★	★	-
-	-	★	★	-	-	-
-	-	-	★	-	-	-
-	-	-	★	★	-	-
-	★	★	★	★	-	-
★	★	★	★	★	★	★

sp	st
0	7
1	5
2	3
3	1
2	3
1	5
0	7

```
public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    int n = s.nextInt();
    int st = n; // there is n stars in 1st row
    int sp = 0; // there is no spaces in 1st row.
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= sp; j++) // printing spaces
            System.out.print (" ");
        for (int j = 1; j <= st; j++)
            System.out.print ("*");
        }
    }
}
```

if the row is b/w middle and  
first row and if star is  
not the first or last of the  
row

changing  
to this

```
for (int j = 1; j <= st; j++)
{
    if (i > 1 && i <= n / 2 && j > 1 && j < st)
        System.out.print ("\t");
    else
        System.out.print ("* \t");
}
}

Yields
```

★	★	★	★	★	★	★
-	★	-	-	★	-	-
-	-	★	-	★	-	-
-	-	-	★	-	-	-
-	-	-	★	★	-	-
-	★	★	★	★	-	-
★	★	★	★	★	★	★

```
for (int j = 1; j <= st; j++)
{
    System.out.print ("* \t");
}
}

Yields
```

```
if (i <= n / 2) {
    sp++;
    st -= 2;
} else {
    sp--;
    st += 2;
}

if (int j = 1; j <= st; j++) {
    if (i > 1 && i <= n / 2) {
        if (j == 1 || j == st) {
            sys.out.print ("* \t");
        } else {
            sys.out.print ("\t");
        }
    } else {
        sys.out.print ("* \t");
    }
}

Yields
```

```
System.out.println ();
}

Yields
```

## PATTERN 19

CONSTRAINTS:  $1 \leq n \leq 10$   
 $n = \text{odd}$

*	-	-	-	*
*	-	-	-	*
*	-	*	-	*
*	*	-	*	*
*	-	-	-	*

- = 1st and last column All \*'s in all the row's
- = \*'s on diagonals below the middle row. ( $n/2$ )
- =  $|j| = 1 \text{ || } |j| = n$  → condition for 1st and last column of all rows
- =  $|i| = |j| \text{ || } |i| + |j| = 1$  → condition for diagonals  
 ↳ But need only stars below middle row ( $n/2$ )  
 we will place \* & check  $\boxed{\text{if } (i > n/2)}$

```

public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    int n = s.nextInt();
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (j == 1 || j == n) {
                System.out.print("*\t");
            } else if (i > n/2) && (i == j || i + j == n + 1) {
                System.out.print("*\t");
            } else {
                System.out.print("\t");
            }
        }
        System.out.println();
    }
}
  
```

print \*'s      Else  
 ↳ printing space

### OUTPUT

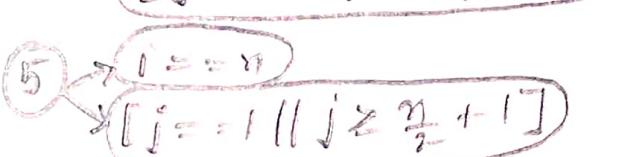
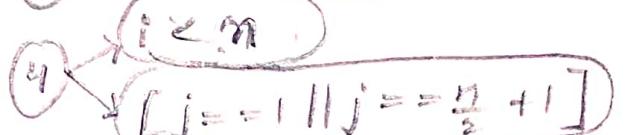
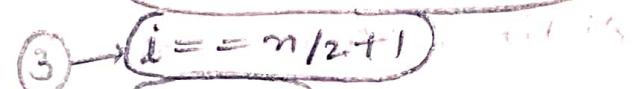
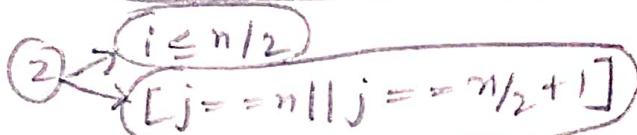
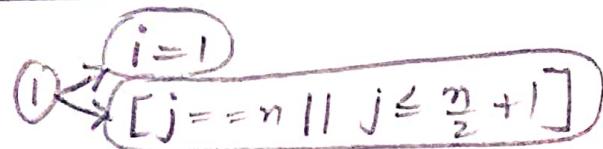
*	-	-	-	*
*	-	-	-	*
*	-	*	-	*
*	*	-	*	*
*	-	-	-	*

PATTERN 20 → CONSTRANTS :  $1 \leq n \leq 10$   $n = \text{odd}$

## SWASTIKA

$n=7$

$i$	$j$	1	2	3	$n$	5	6	7
1	-	*	*	*	*	-	-	*
2	-	-	-	*	-	-	*	*
3	-	-	-	*	-	-	*	*
4	*	*	*	*	*	*	*	*
5	*	-	-	*	-	-	-	-
6	*	-	-	*	-	-	-	-
7	*	-	-	*	*	*	*	*



public static void main (String [] args) {

Scanner s = new Scanner (System.in);

int n = s.nextInt();

for (int i=1; i<=n; i++)

{ for (int j=1; j<=n/2; j++)

{ if (i==1) { // ① kya app  
st row ho? }

if (j==n || j<=n/2+1) {

sys0.print ("\*\\t");

} else {

sys0.print ("\t");

} else if (i<=n/2) { // ②  
row ho? }

if (j==n || j==n/2+1) {

sys0.print ("\*\\t");

} else {

sys0.print ("\t");

} else if (i==n/2+1) { // ③  
middle row ho? }

sys0.print ("\*\\t");

}

else if (i>n) { // ④  
last row ho? }

if (j==1 || j==n/2+1) {

sys0.print ("\*\\t");

} else {

sys0.print ("\t");

} else if (j==1 || j>n/2+1) { // 7th row ho? }

else { // ⑤ }

if (j==1 || j>n/2+1) {

sys0.print ("\*\\t");

} else {

sys0.print ("\t");

sys0.println();

8

### ③ PATTERN 21 $n=4$

4	4	4	4	4	4	4	4
4	3	3	3	3	3	3	4
4	3	2	2	2	3	4	
4	3	2	1	2	3	4	
4	3	2	2	2	3	4	
4	3	3	3	3	3	4	
4	4	4	4	4	4	4	

(1 to  $n-1$ )  
 1st loop  
 [ ]  
 (n to n)  
 2nd loop  
 [ ]  
 (n+1 to 2n-1)  
 3rd loop  
 [ ]

$n = 3$

3	3	3	3	3
3	2	2	2	3
3	2	1	2	3
3	2	2	2	3
3	3	3	3	3

BREAK ROW → BREAK COLUMN

[1st loop (1 to n-1) → [b.no. → 1-1  
 Constant → 1-2  
 I.no. → 1-3]  
 [2nd loop (n) → [n-1 → 2-1  
 2-2  
 2-3]  
 [3rd loop (n+1 to 2n-1) → [D.no. → 3-1  
 Constant → 3-2  
 I.no. → 3-3]

FOCUS  
X  
GENERIC USE

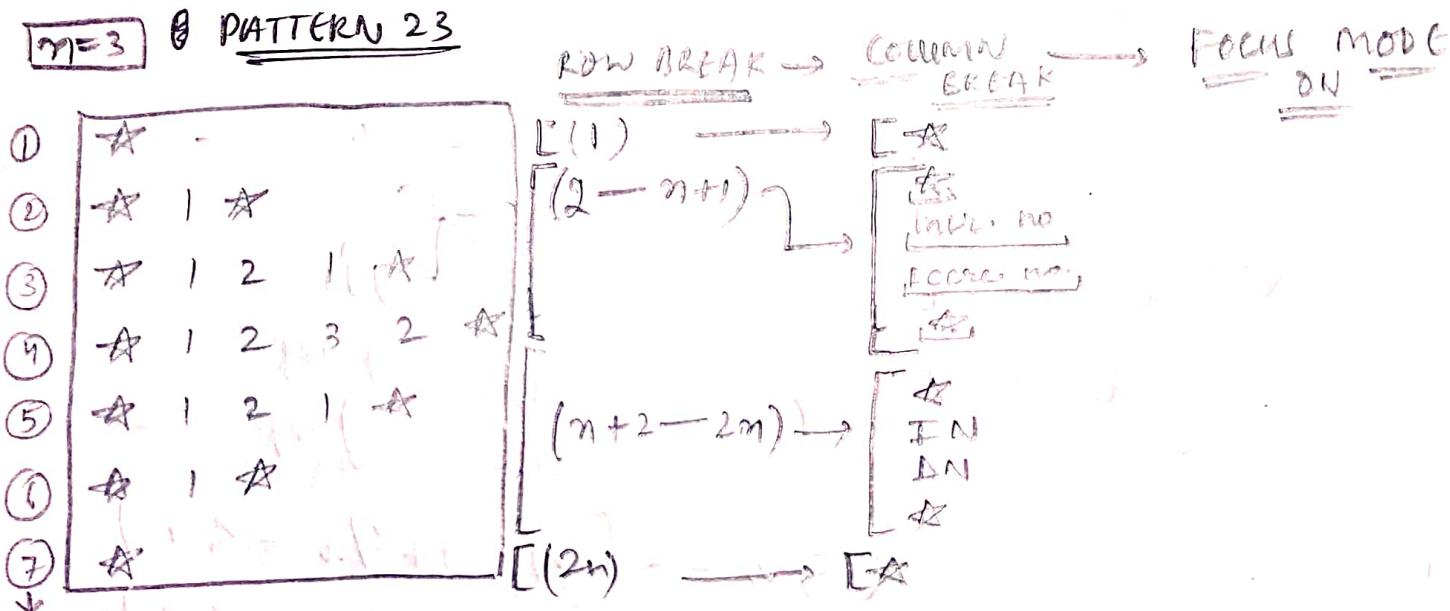
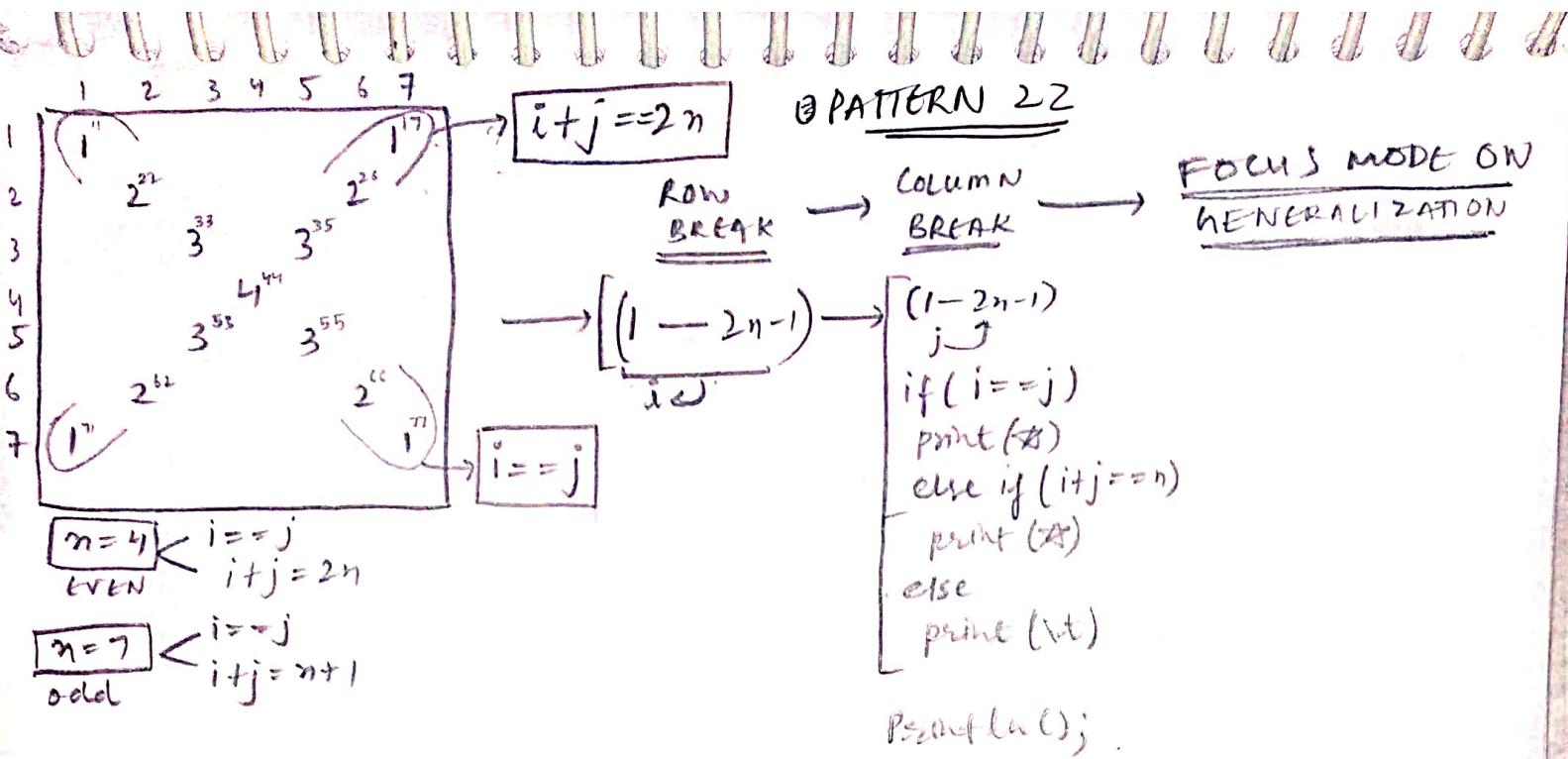
① for (int i=1; i <= n-1; i++)  
 Row break  
 [Decreasing] i=1 D L I  
 [Constant] i=2 X ③ ② X  
 [Increasing] i=3 ③, ② ① ②, ③  
 Print 2n

### ② Break column

3 3 3 3 3       $n=3$   
 3 2 2 2 3  
 3 2 1 2 1

### DECONSTRUCT PATTERN

- ① BREAK ROW ✓
- ② BREAK COLUMN ✓
- ③ GENERALIZE IT! ✓
- FOCUS MODE ON!



HOW TO REVIEW → Ques. ↗ Understand ↗  
 ↗ Code it ↗ Reproduce it ↗  
 ↗ Implement New Ques.

**RECALL**

nardeep.sandhu@gmail.com