n = 5

LD

RD

stars

↳ Coordinate

Pattern

<, <=, >, >=, +, -, *, /

n = 5

r \ c	0	1	2	3	4
0	*	-	-	-	*
1	-	*	-	*	-
2	-	-	*	-	-
3	-	*	-	*	-
4	*	-	-	-	*

RD

(0,4)
(1,3)
(2,2)
(3,1)
(4,0)
(r,c)

4
4
4
4
4

r+c == n-1

LD

(0,0)
(1,1)
(2,2)
(3,3)
(4,4)
(r,c)

r == c

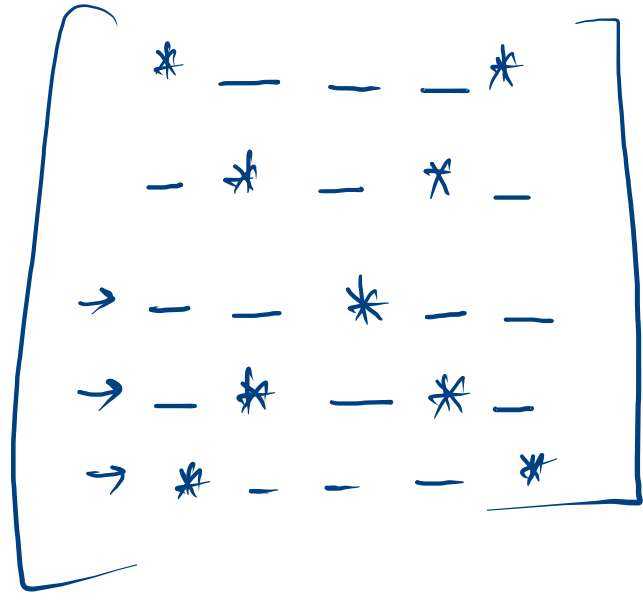
r c

n = 5 , n-1 = 4

```
int n = scn.nextInt(); // 5
```

```
for(int r = 0 ; r < n ; r++){
    for(int c = 0 ; c < n ; c++){
        if(r == c || (r+c) == n-1){
            System.out.print("*\t");
        }else{
            System.out.print("\t");
        }
    }
    System.out.println();
}
```

(< n) r	(< n) c
0	0 1 2 3 4 5
1	0 1 2 3 4
2	0 1 2 3 4 5
3	0 1 2 3 4 5
4	



r == c

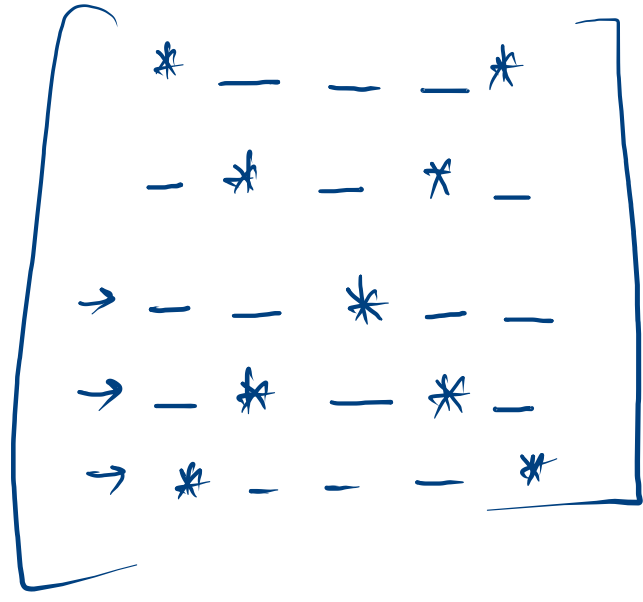
r + c == n - 1

n = 5 , n-1 = 4

```
int n = scn.nextInt(); // 5
```

```
for(int r = 0 ; r < n ; r++){
    for(int c = 0 ; c < n ; c++){
        if(r == c || (r+c) == n-1){
            System.out.print("*\t");
        }else{
            System.out.print("\t");
        }
    }
    System.out.println();
}
```

(< n) r	(< n) c
0	0 1 2 3 4 5
1	0 1 2 3 4
2	0 1 2 3 4 5
3	0 1 2 3 4 5
4	



r == c

r + c == n - 1

n=5

7

r==c

H.W.

Pattern 7

Pattern 8

n== 5

8

r+c == n-1

q

nst = 5

i = 1, 2, 3, 4, 5

* - - - *

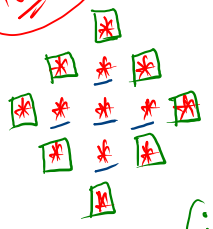
why ?

```
for(int i = 1 ; i <= nst ; i++){
    if(i == 1 || i == nst){
        System.out.print("*\t");
    }else{
        System.out.print("\t");
    }

}
```

row	nst	nsp
1	1	2
2	3	1
3	5	0
4	3	1
5	1	2

Diff =>

Pattern 5 → Pattern 6 (n=5)



space

star

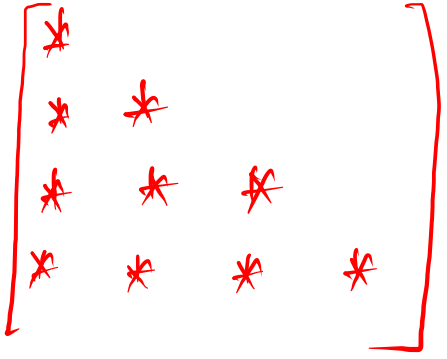
Preparation

i == 1 → first

i == nst → last

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

int row = 1 , nst = 1 , nsp = n/2;

while(row <= n){

    // code for each row

    for(int i = 1 ; i <= nsp ; i++){
        System.out.print("\t");
    }
    for(int i = 1 ; i <= nst ; i++){
        System.out.print("*\t");
    }

    // move to next row
    System.out.println();

    // preparation for next row
    if(row <= n/2){
        nsp = nsp - 1;
        nst = nst + 2;
    }else{
        nsp = nsp + 1;
        nst = nst - 2;
    }

    row++;
}
```

Pattern1



Step → no.

```
1
2   3
4   5   6
7   8   9   10
..
```

Structure wise → Part 1, (11)

Pattern 12

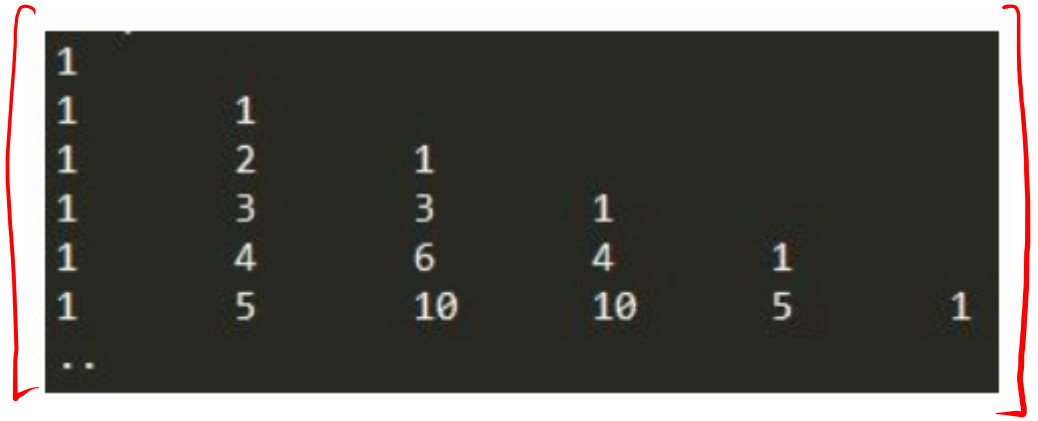
```
0
1       1
2       3       5
8       13      21      34
..
```

H.W -

Numbering

↳ fibonacci series

↳ third = first + second


1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
..

5

Pascal Triangle

Sample Output

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

Pascal Triangle
↳ Binomial

r → / n ↓	0	1	2	3	4
0	${}^0C_0=1$				
1	${}^1C_0=1$	${}^1C_1=1$			
2	${}^2C_0=1$	${}^2C_1=2$	${}^2C_2=1$		
3	${}^3C_0=1$	${}^3C_1=3$	${}^3C_2=3$	${}^3C_3=1$	
4	${}^4C_0=1$	${}^4C_1=4$	${}^4C_2=6$	${}^4C_3=4$	${}^4C_4=1$

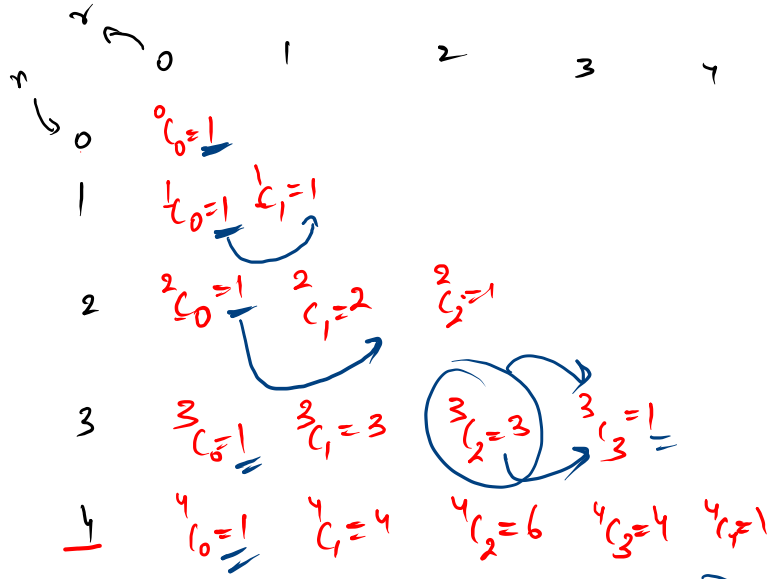
$$ {}^nC_r = \frac{n!}{(n-r)!\,r!} $$

$$ {}^2C_0 = \frac{2!}{(2-0)!\,0!} = \frac{2}{2\times 1} = 1 $$

$$ \frac{2!}{(2-1)!\,1!} = \frac{2}{1\cdot 1} \Rightarrow 2 $$

```
for(int n = 0 ; n < inp ; n++){
    for(int r = 0 ; r <= n ; r++){

    }
}
```

n	r
0	0
1	0, 1
2	0, 1, 2
3	0, 1, 2, 3
4	0, 1, 2, 3, 4



$$^{n}C_r \Rightarrow \frac{n!}{r!\,(n-r)!}$$

$$^{n}C_{r+1} = {}^{n}C_r \left(\frac{n-r}{r+1}\right) \qquad \left(^{3}C_2 = 3\right)$$

$$\left(^{3}C_3 = {}^{3}C_2 \left(\frac{3-2}{2+1}\right) = 3\left(\frac{1}{3}\right) = 1\right)$$

$n = 3$

$nFact = 1 \times 1 = 1 \times 2 = 2 \times 3 = 6$

$k = 1, 2, 3$

```
int nFact = 1;
for(int k = 1 ; k <= n ; k++){
    nFact = nFact * k;
}
```

$3! \Rightarrow 6$

```java
for(int n = 0 ; n < inp ; n++){
    for(int r = 0 ; r <= n ; r++){

        int nFact = 1;
        for(int k = 1 ; k <= n ; k++){
            nFact = nFact * k;
        }

        int rFact = 1;
        for(int k = 1 ; k <= r ; k++){
            rFact = rFact * k;
        }

        int nMrFact = 1;
        for(int k = 1 ; k <= n-r ; k++){
            nMrFact = nMrFact * k;
        }

        int nCr = nFact / (nMrFact*rFact);
        System.out.print(nCr+"\t");
    }
    System.out.println();
}
```

$$\frac{{}^{n}C_{r}}{{}^{n}C_{r+1}} \Rightarrow \frac{n!}{r!(n-r)!} \div \frac{n!}{(r+1)!(n-r-1)!}$$

$$\Rightarrow \frac{n!}{r!(n-r)!} \times \frac{(r+1)!(n-r-1)!}{n!}$$

$$\Rightarrow \frac{(r+1)!}{r!} \times \frac{(n-r-1)!}{(n-r)!}$$

$$\Rightarrow \frac{(r+1).r!}{r!} \times \frac{(n-r-1)!}{(n-r)(n-r-1)!} \Rightarrow \frac{(r+1)}{n-r}$$

$$5! \Rightarrow 5 \times 4!$$

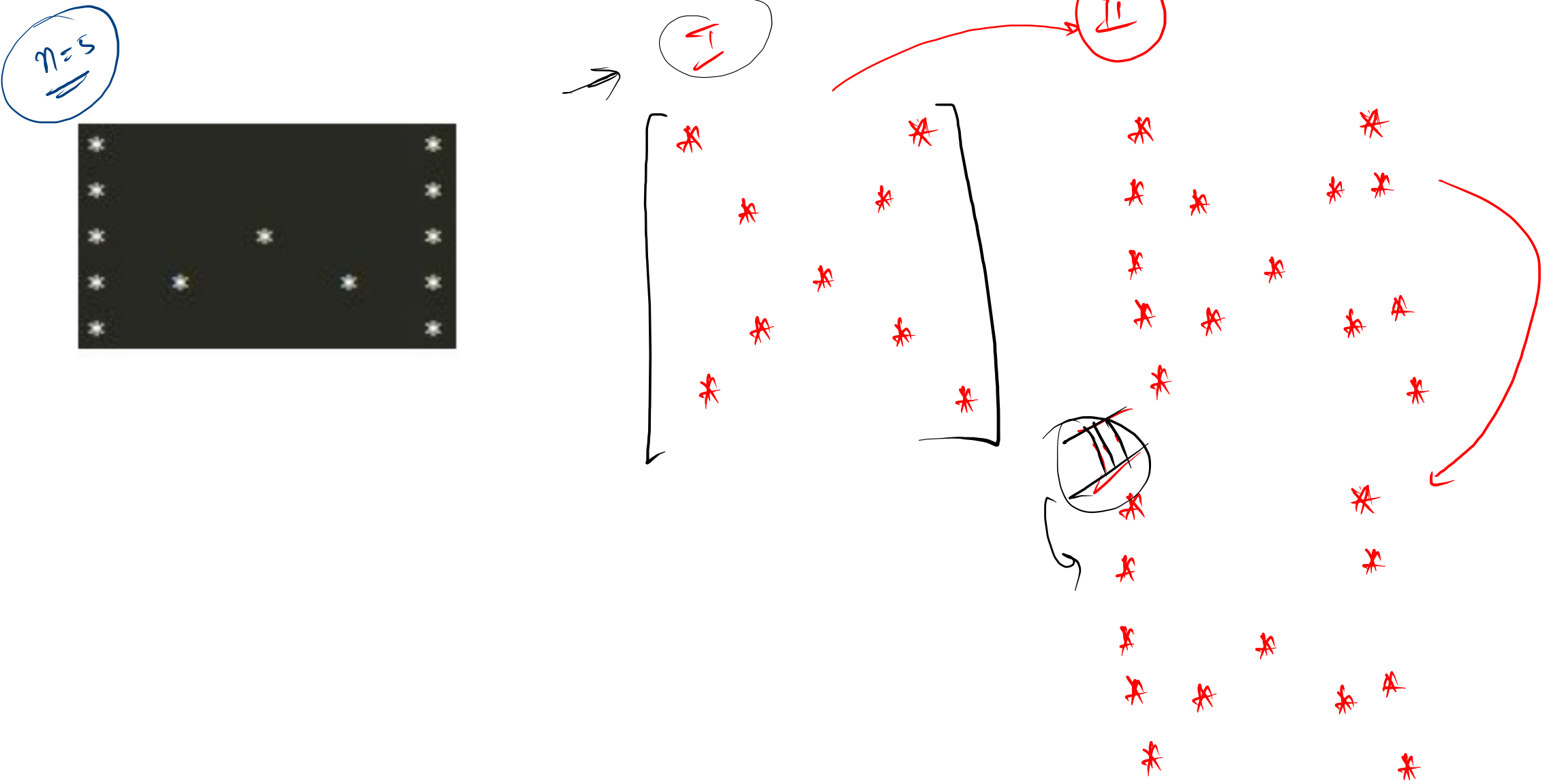
$$\frac{{}^{n}C_{r}}{{}^{n}C_{r+1}} = \frac{(r+1)}{(n-r)}$$

$$\frac{(n-r)}{r+1} \quad {}^{n}C_{r} = {}^{n}C_{r+1}$$

```java
int inp = scn.nextInt();

for(int n = 0 ; n < inp ; n++){
    int nCr = 1;
    for(int r = 0 ; r <= n ; r++){
        System.out.print(nCr+"\t");
        nCr = ((n-r)*nCr)/(r+1);
    }
    System.out.println();
}
```

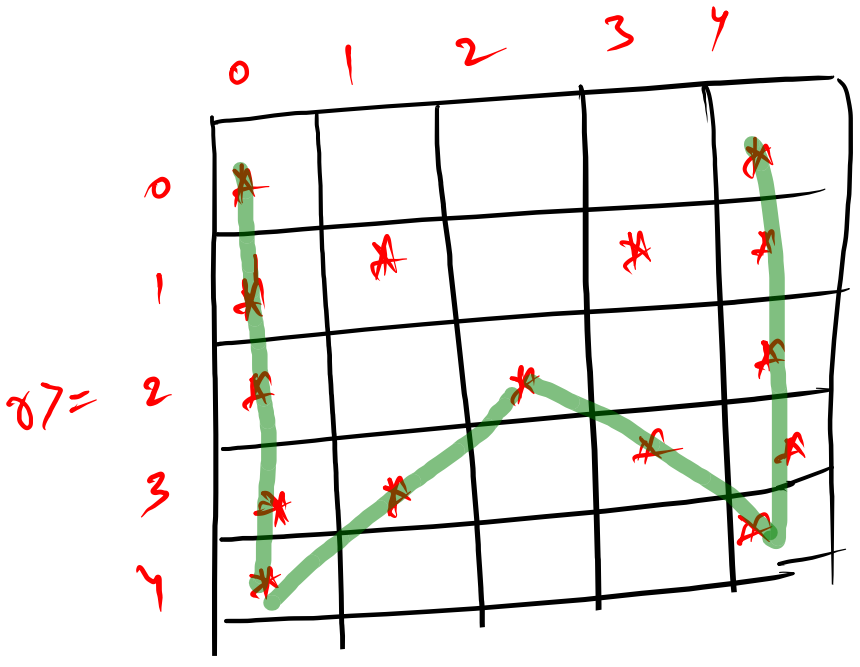
n	r	$\frac{n_{C_r}}{\times 0}$ $\left(\frac{n-r}{r+1}\right).n_{C_r}$
0	0	
1	0 1	$\frac{1 \cdot 1}{1} = 1$
2	0, 1, 2	1 2 1
3	0, 1, 2, 3	1 3 3 1
4		

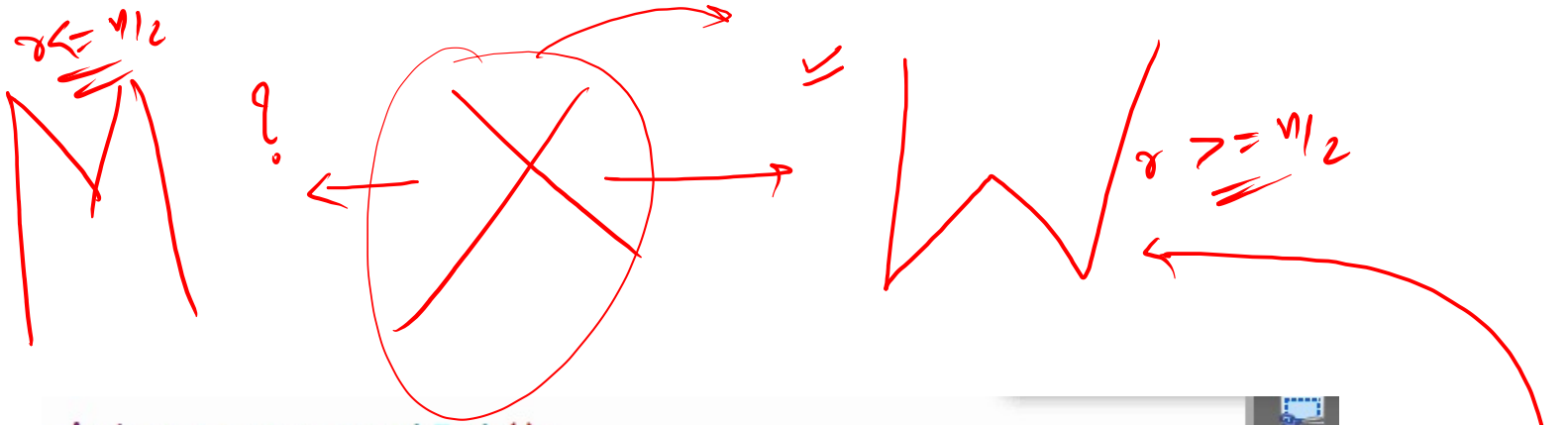
→ 1
→ 1 1
→ 1 2 1
→ 1 3 3 1


η=5
I
II

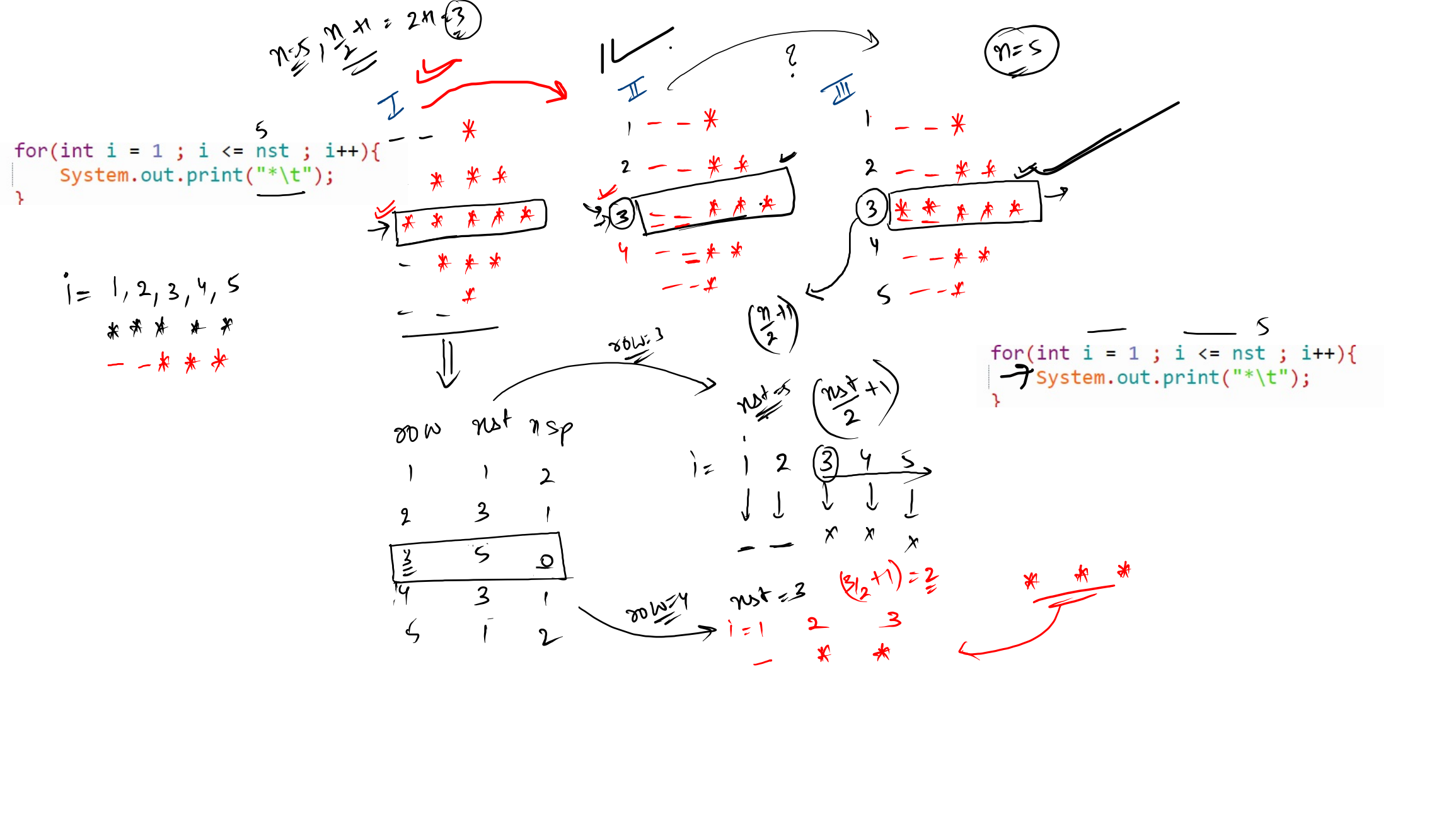
$S \rightarrow 2$

$n \rightarrow n/2$



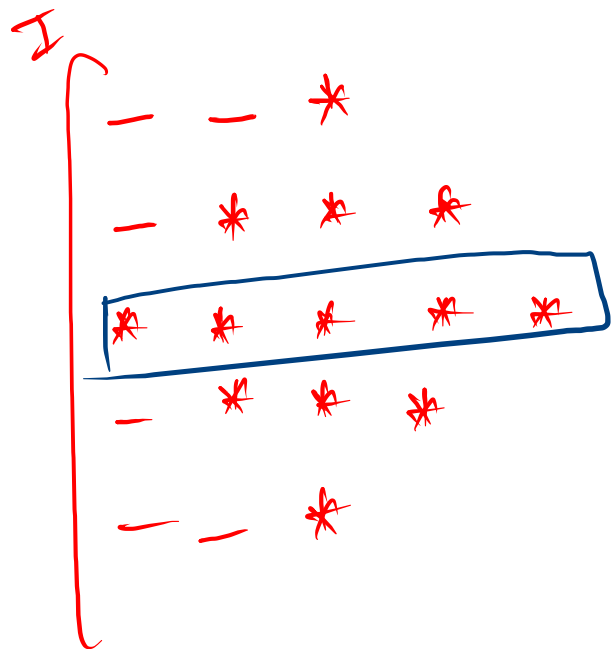


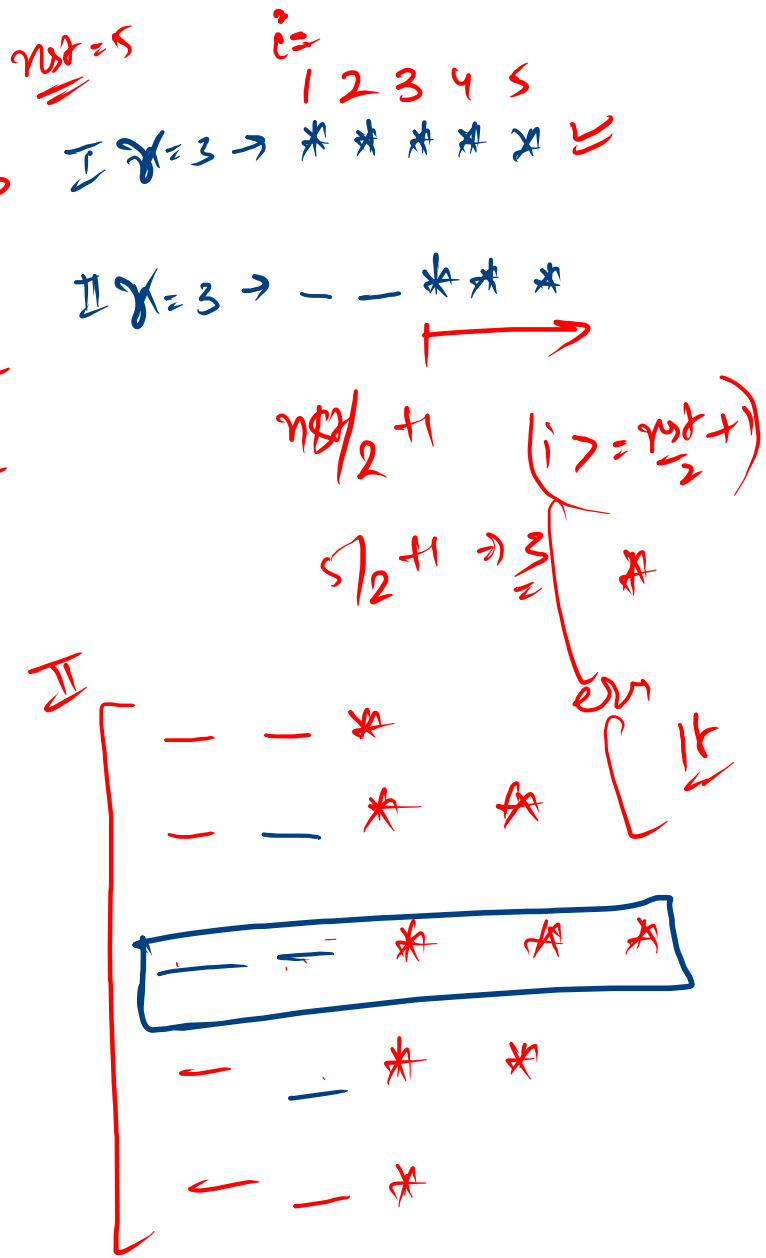
```java
int n = scn.nextInt();

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

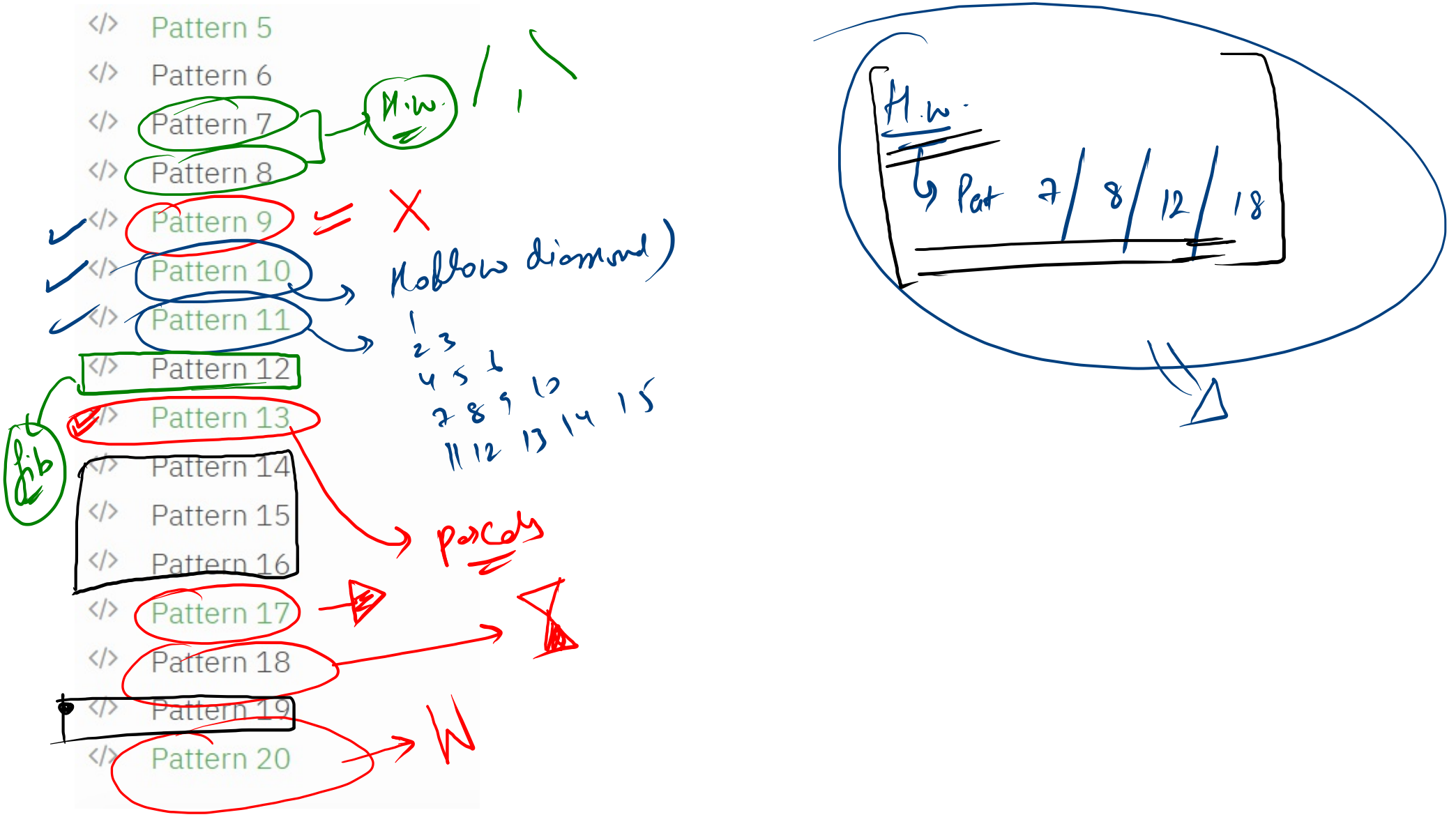

for(int i = 1 ; i <= nst ; i++){
System.out.print("*\t");
}
i= 1, 2, 3, 4, 5
row nst nsp
1 1 2
2 3 1
3 5 0
4 3 1
5 1 2
row: 3
row: 4
nst = 5
(n/2 +1)
i= 1 2 3 4 5
nst = 3
(3/2 +1) = 2
i = 1 2 3
for(int i = 1 ; i <= nst ; i++){
System.out.print("*\t");
}

```java
int row = 1 , nst = 1 , nsp = n/2;

while(row <= n){

    // code for each row

    for(int i = 1 ; i <= nsp ; i++){
        System.out.print("  ");
    }
    for(int i = 1 ; i <= nst ; i++){
        System.out.print("* ");
    }

    // move to next row
    System.out.println();

    // preparation for next row
    if(row <= n/2){
        nsp = nsp - 1;
        nst = nst + 2;
    }else{
        nsp = nsp + 1;
        nst = nst - 2;
    }

    row++;
}
```

row	nst	nsp
1	1	2
2	3	1
3	5	0
4	3	1
5	1	2






Pattern 5
Pattern 6
Pattern 7
Pattern 8
Pattern 9
Pattern 10
Pattern 11
Pattern 12
Pattern 13
Pattern 14
Pattern 15
Pattern 16
Pattern 17
Pattern 18
Pattern 19
Pattern 20
M.W.
X
Hollow diamond)
1
2 3
4 5 1
2 8 9 10
11 12 13 14 15
fib
pascals
M.W.
Pat 7/ 8/ 12/ 18