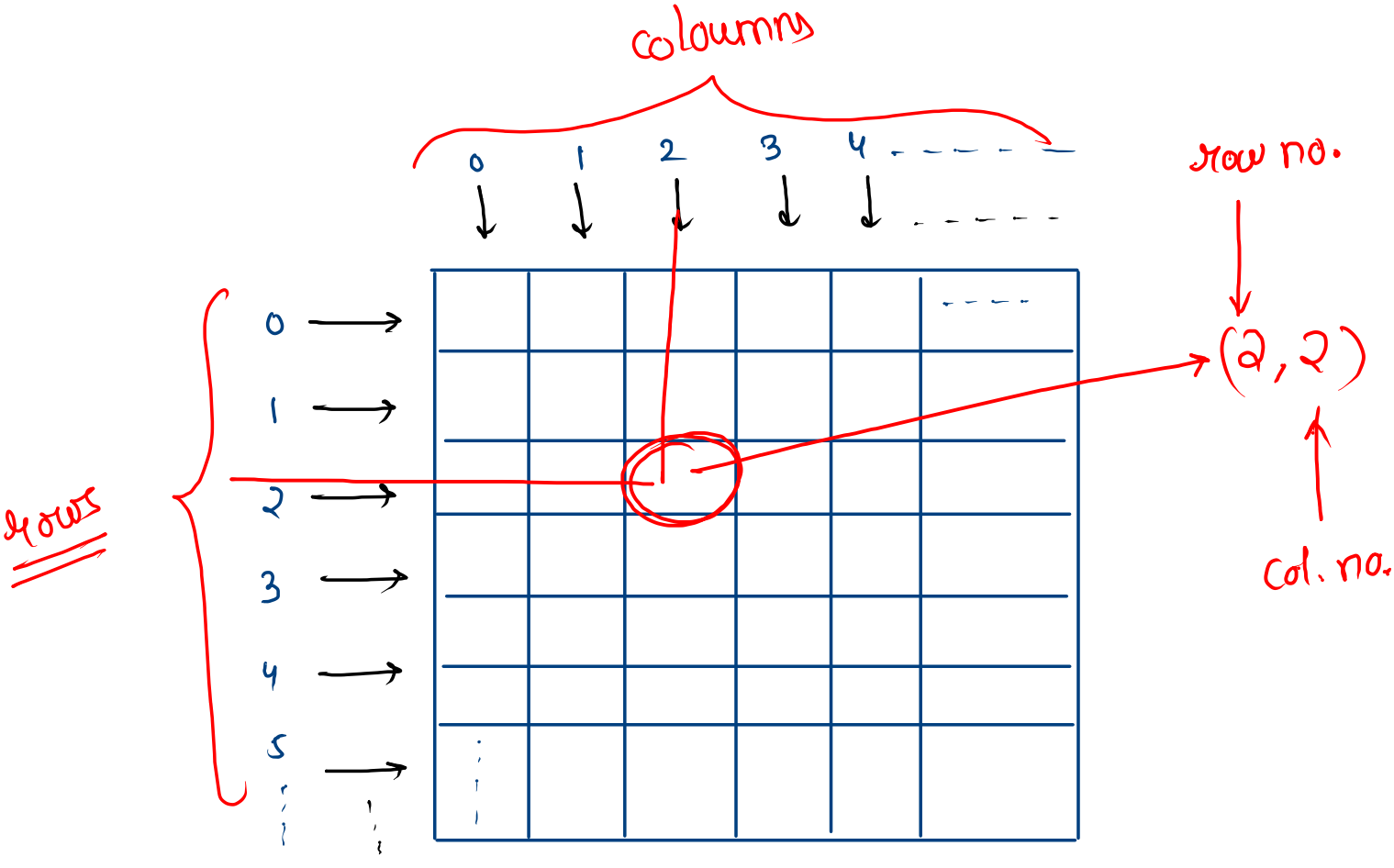
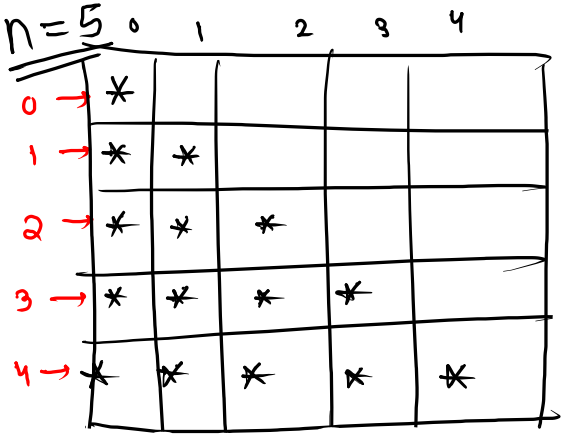
Structure of Grid

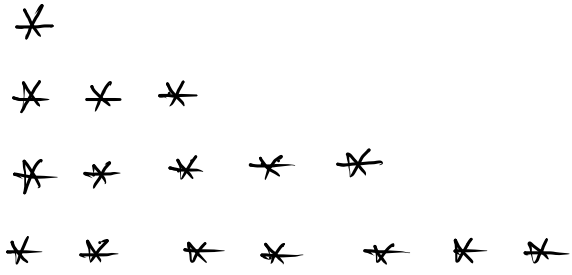


Template for Pattern Questions




```
int st = 1;
for(int i=0; i<n; i++){
    for(int j=0; j<st; j++){
        Syso("*" + " ");
    }
    st++;
    Sysoln();
}
```

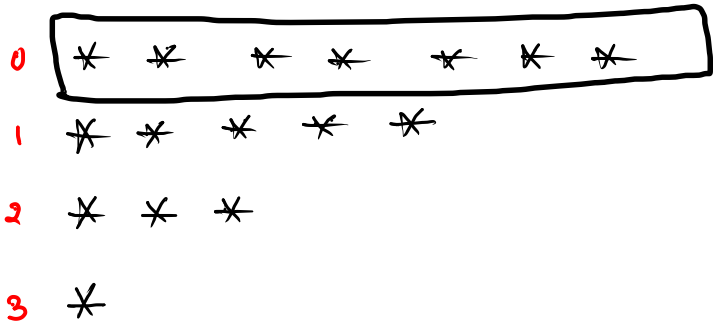
Quel



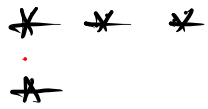
```
int st = 1;
for(int i=0; i<n; i++){
    for(int j=0; j<st; j++){
        Syso("*" + " ");
    }
    st += 2;
    SysoIn();
}
```

Quel

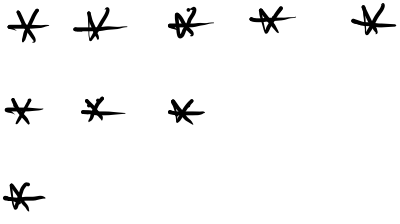
$n = 4$ ✓



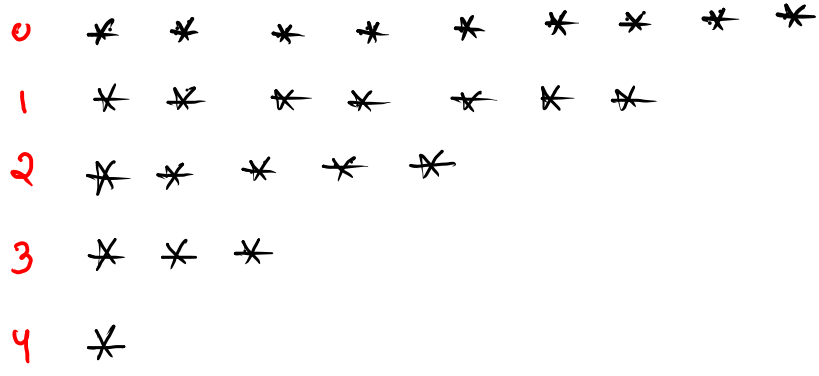
$n = 2$



$n = 3$



$n = 5$



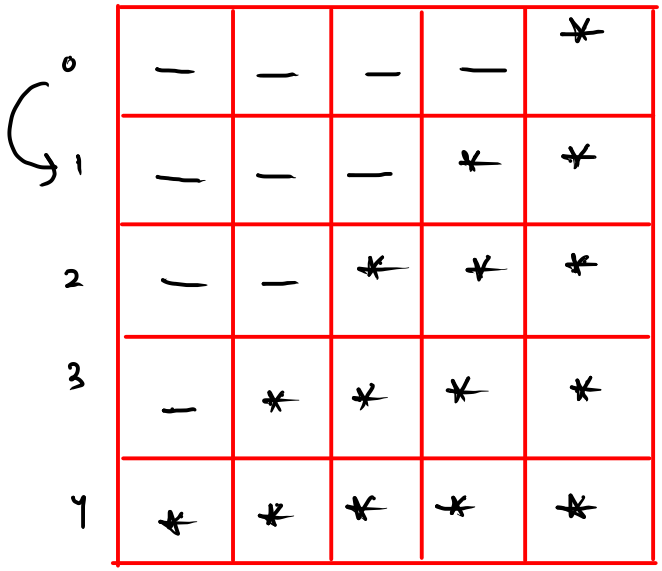
n = 7 → 13
n = 6 → 11
n = 5 → 9
n = 4 → 7
n = 3 → 5
n = 2 → 3
n = 1 → 1

(2 * n − 1)

```
int st = 2*n-1;
for(int i=0; i<n; i++){
    for(int j=0; j<st; j++){
        Syso("*"+" ");
    }
    st-=2;
    Sysoln();
}
```

GKSTR20 Pattern_5

n = 5



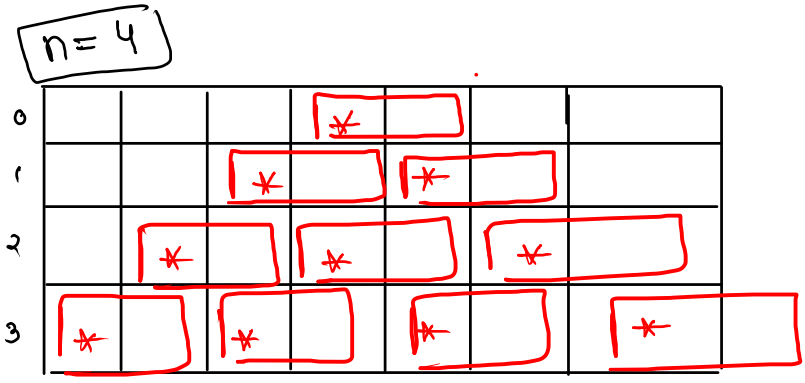

```
int st = 1;
int sp = n-1;
for(int i=0; i<n; i++){
    for(int j=0; j<sp; j++){
        Syso(" ");
    }
    for(int j=0; j<st; j++){
        Syso("*");
    }
    st++; sp--;
    Sysoln();
}
```

code

```java
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("*");
        }
        sp--;
        st++;
        System.out.println();
    }
}
```

GKSTR24 Pattern_7_Pyramid




```
int st=1;
int sp=n-1;
for(int i=0; i<n; i++){
    for(int j=0;j<sp;j++){
        Syso(" ");
    }
    for(int j=0;j<st;j++){
        Syso("* ");
    }
    sp--;
    st++;
    sysoln();
}
```

```java
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("* ");
        }
        sp--;
        st++;
        System.out.println();
    }
}
```

GKSTR29_Pattern_12_Diamond

i/p n=5

row = (2*n)-1;

9

sp--;
st+=2;

sp++;
st-=2;

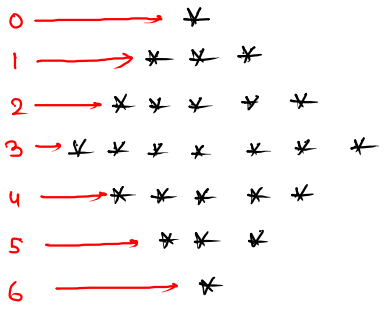
if(i < n)
first half
else
second h

---

n=4

row = 7

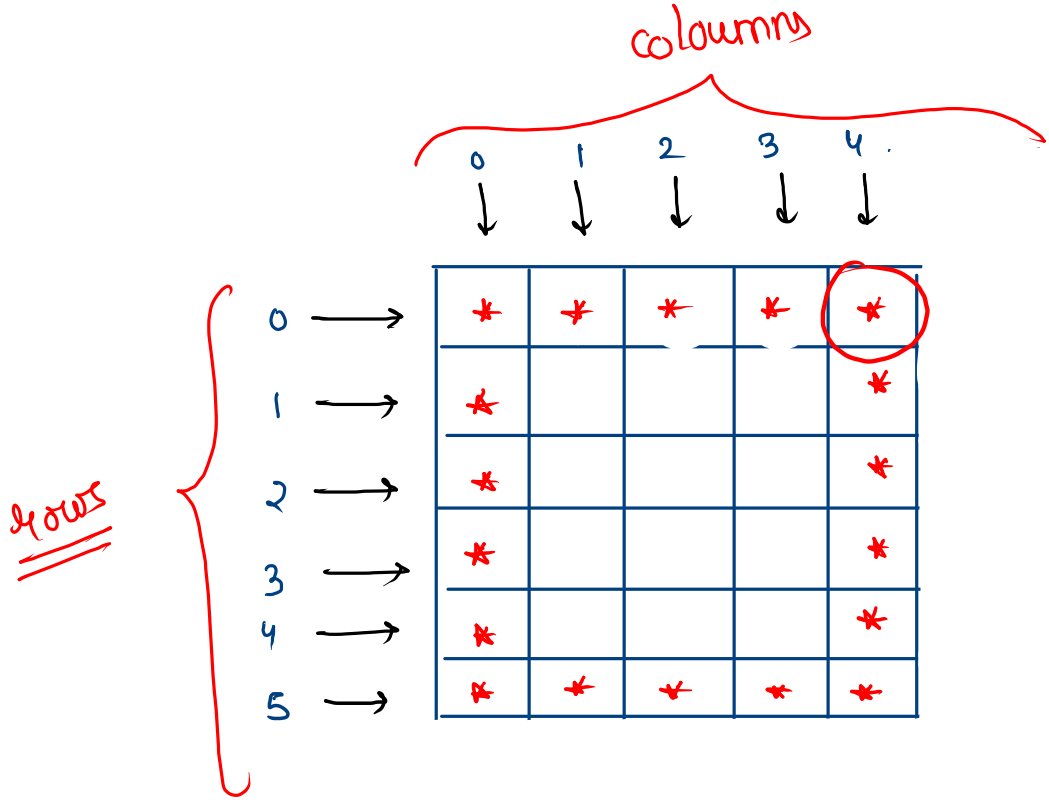
mid = row/2
= 7/2
= 3



first senario

second senario

indexing



logic

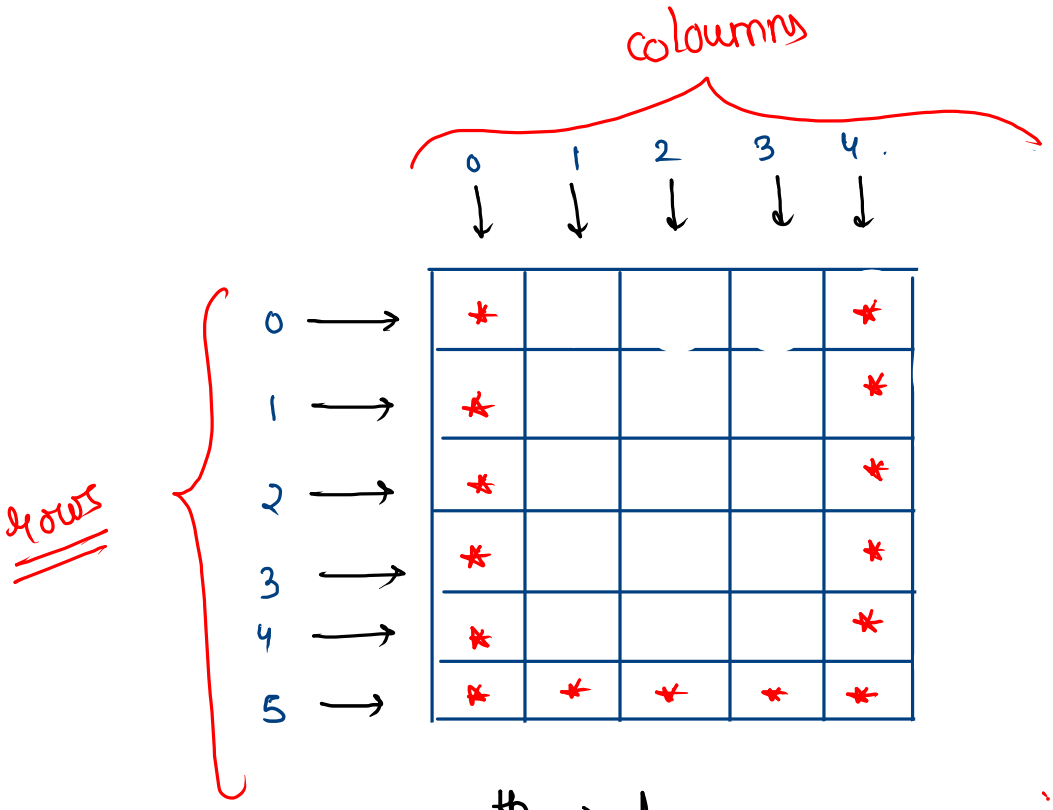
1) 0^{th} index row
2) 0^{th} index col.
3) $(r-1)^{th}$ index row
4) $(c-1)^{th}$ index col.

Note:- if any of the 4 senarios are correct then print ★ , else space

Code

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

    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            if ( i == 0 || i == row - 1 || j == 0 || j == col - 1 ) {
                System.out.print("*");
            } else {
                System.out.print(" ");
            }
        }
        System.out.println();
    }
}
```



1) ~~0th index row~~ $i == 0$

2) 0th index col. $j == 0$

3) $(n-1)^{th}$ index row $i == (n-1)$

4) $(m-1)^{th}$ index col. $j == (m-1)$

code

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

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

Pattern 9 - Square Ladder with top and bottom