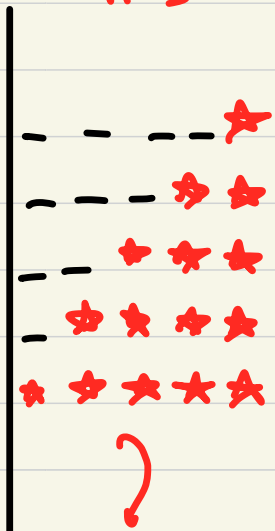
Q) n = 5

1 → 1
2 → 3
3 → 5
4 → 7
5 → 9

(i) → 2i - 1

Stars

n = 5



this pattern we already solved

Space → (n - row)

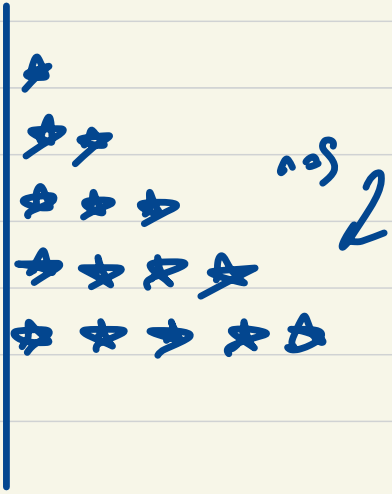
Stars → row

i

rows

1 → 5
2 → 4
3 → 3
4 → 2
5 → 1

x^{th} row → $n - x + 1$

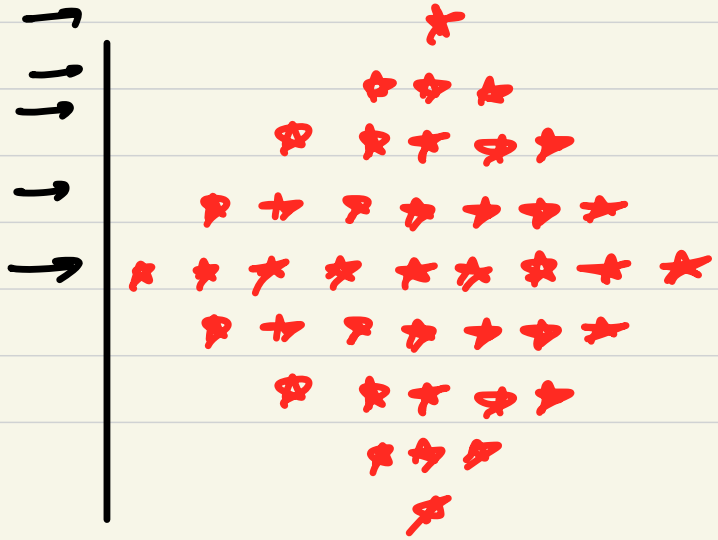


cols

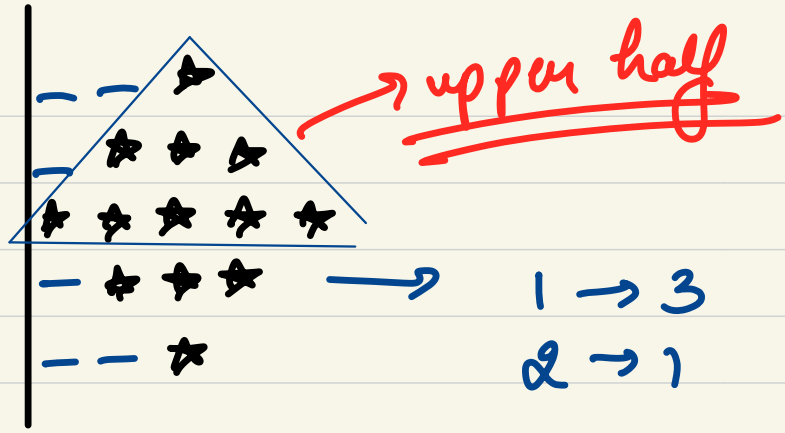
Q.

$n = 3$

↳ n will be always odd



$n = 5$


upper half
1 → 3
2 → 1

Total → n-1 rows

row → x

space → x

1 → 7
2 → 5
3 → 3
4 → 1

$(2x(n-x)-1)$ → stars

$n-1$

x ↑

$a - b$

$10 - 1$

$10 - 2$

$10 - 3$

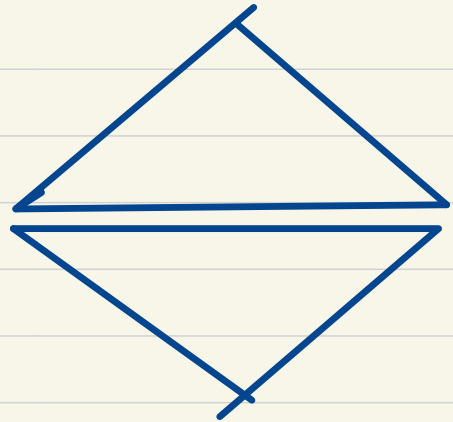
$10 - 4$

$10 - 5$

$10 - 6$

upperTriangle(n) ←

→ lowerTriangle(n) ←



Butterfly

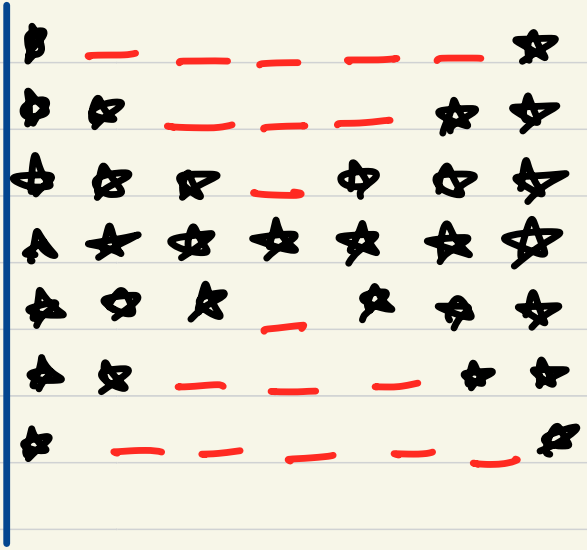
Q$_n$

n = 3

1	★ _ _ _ ★	1 → 3
2	★ ★ _ ★ ★	2 → 1
3	★ ★ ★ ★ ★	5 - 2(1) = 3
4	★ ★ _ ★ ★	
5	★ _ _ _ ★	5 - 2(2) = 1

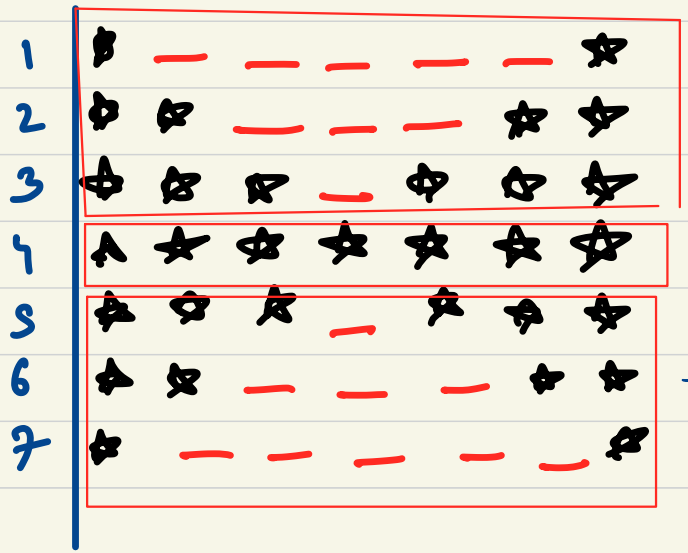
n → odd

n = 7



n = 7

→ n rows printing

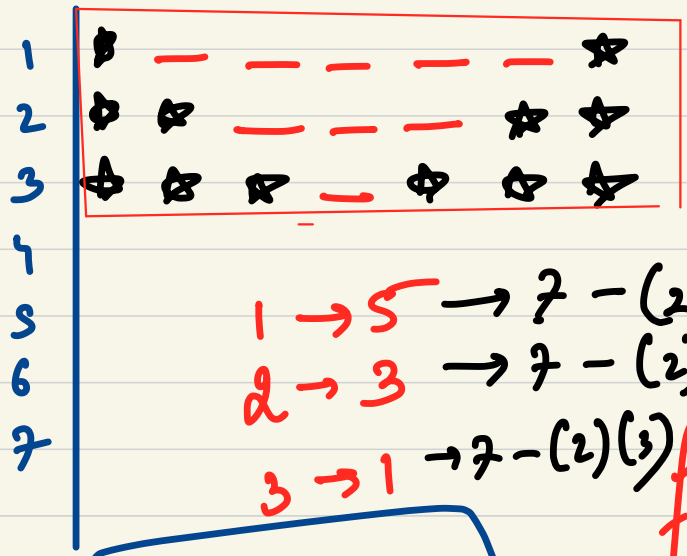


→ upper

→ middle

→ lower

n = 7



1 → 5 → 7 − (2)(1)

2 → 3 → 7 − (2)(2)

3 → 1 → 7 − (2)(3)

Spaces → n − 2 × row

In the upper component you've 3 rows for n = 7

for a given value n, we have $\frac{n-1}{2}$ rows

In the upper component, the no. of stars before space & after space is same as the row no.

leftstars = rowno

rightstars = rowno

Spaces → n − 2*rowno

```
for (let row = 1; row <= (n-1)/2; row += 1) {
    let str = "";
    let leftstars = row;
    for (let i = 1; i <= leftstars; i += 1) {
        str += "*";
    }
    let spaces = n - 2 * row;
    for (let j = 1; j <= spaces; j += 1) {
        str += " ";
    }
    let rightstars = row;
    for (let k = 1; k <= rightstars; k += 1) {
        str += "*";
    }
    console.log(str);
}
```

left stars

spaces

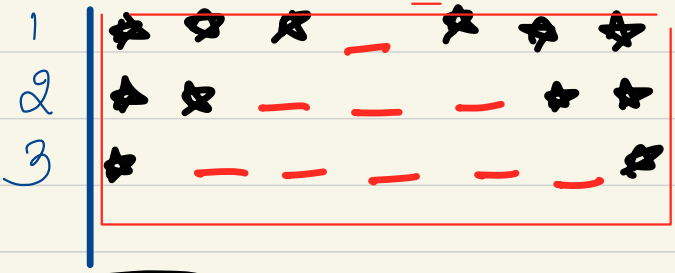
right stars

Middle component always consists of one line with n Stars.

```
function middle (n) {
    let str = "";
    for (let i=1 ; i<=n ; i+=1){
            str += "*";
    }
    console.log(str);
}
```

$\left(\frac{n-1}{2}\right)$ rows

n=7



→ lower

leftstars =

$\left(\frac{7-1}{2}\right) + \frac{6}{2} \rightarrow 3$

$\left(\frac{n-1}{2}\right) - row + 1$

$3 - row + 1$

2row − 1

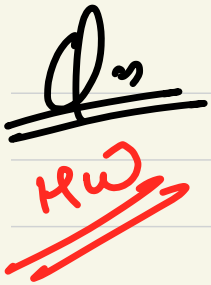
1 → 3
2 → 2
3 → 1

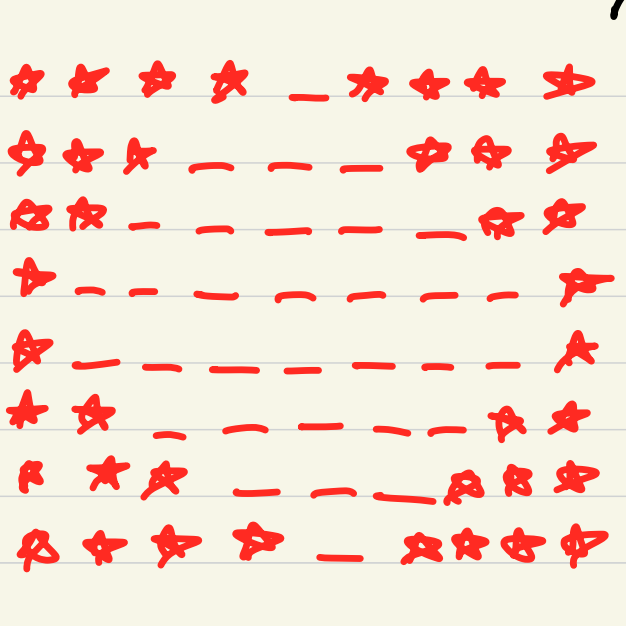
leftstars → $\left(\frac{n-1}{2}\right) - row + 1$

rightstars → $\left(\frac{n-1}{2}\right) - row + 1$

Spaces → $2 \times row - 1$

n=7



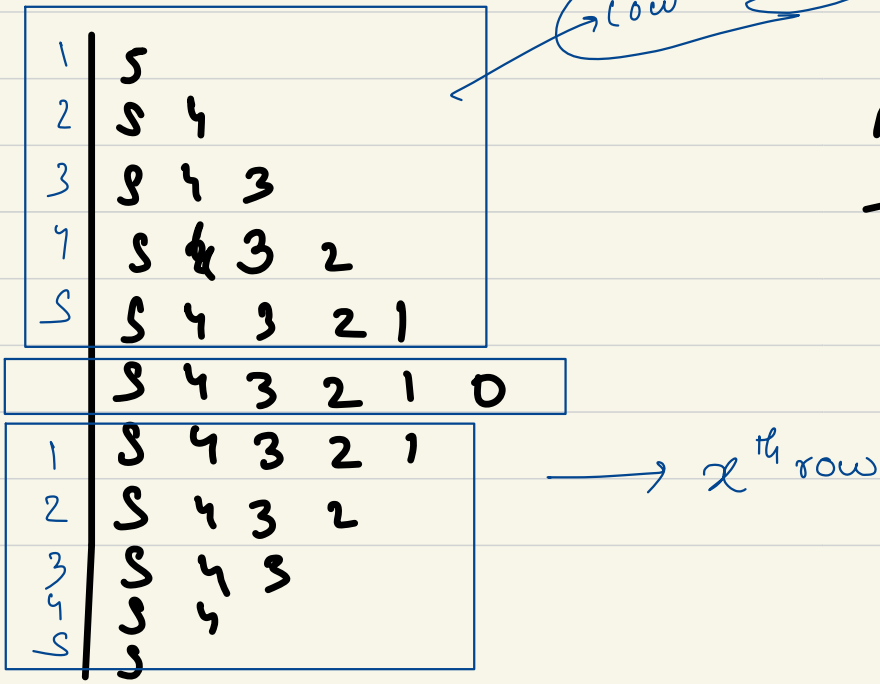


Qn

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

n = 4

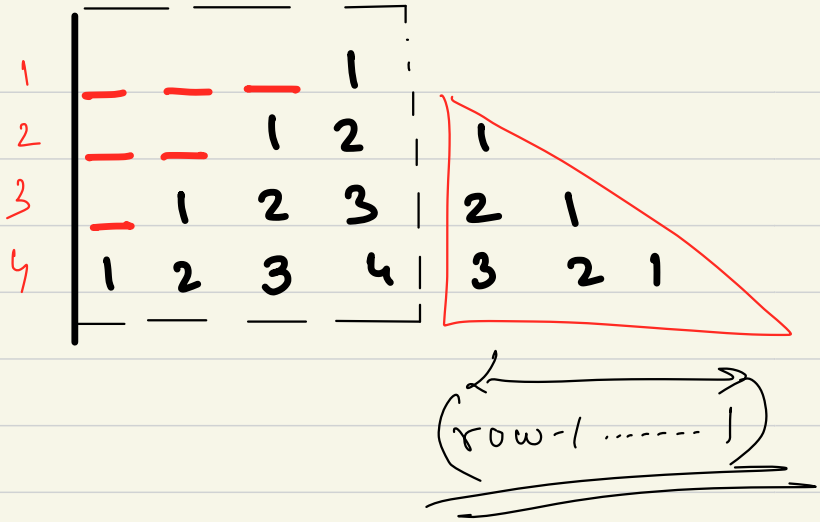
Qn



n = 5

→ x^{th} row → n − row + 1 numbers

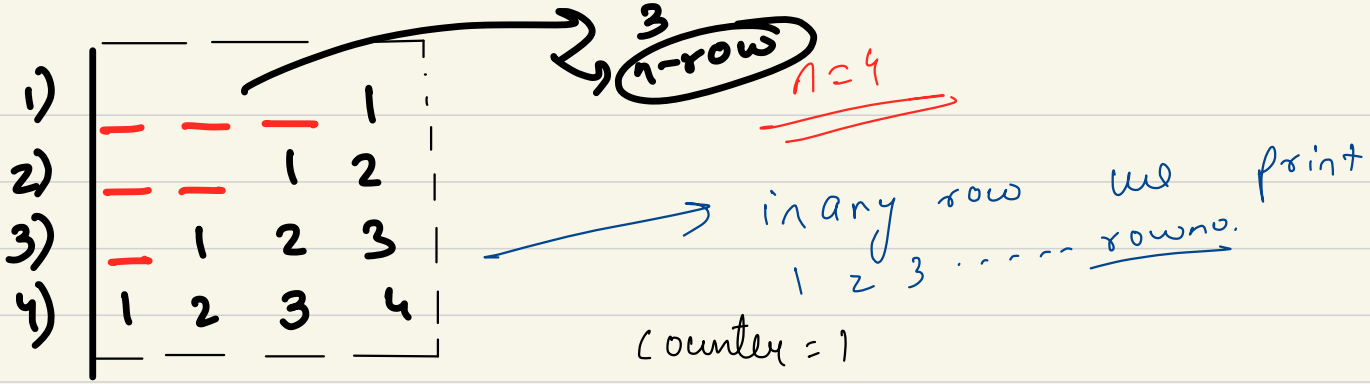
Q)



n=4

row → How many no. to print

↳ row-1



for every row we have a task, to print some spaces, followed by numbers.

for any count row → spaces ≥ n-row

```
for (let row=1; row <= n; row++) {
    let str="";
    let spaces = n-row;
    for (let i=1; i<=spaces; i++) {
        str += " ";
    }
    let counter = 1;
    for (let j=1; j<=row; j++){
        str += counter;
        counter++;
    }
    let counter2 = row-1
    for (let k=1; k<=row-1; k++){
        str += counter2
        counter--;
    }
}
```

row=3

counter=1 2 3 4

—12321 j=1 2 3 4

counter=0 1 2

k<=1 2 3

it will be repeated for every row