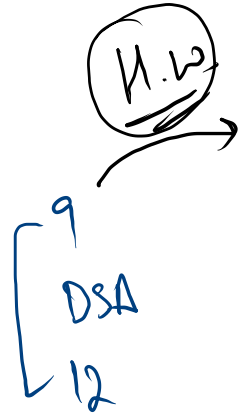
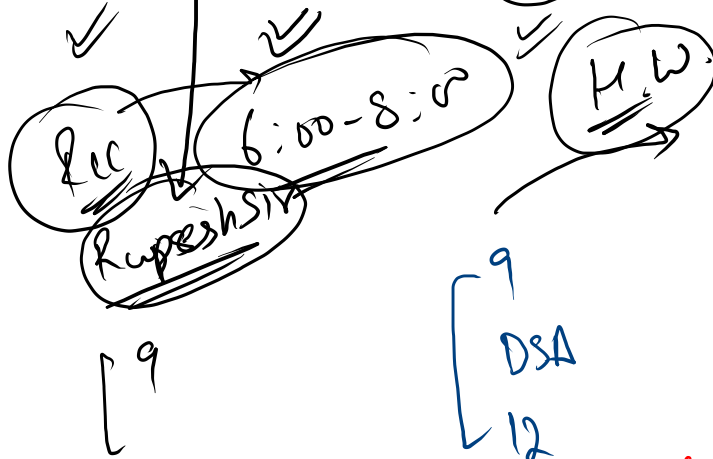


✓
Mon



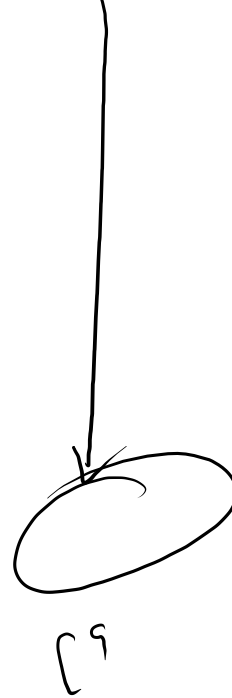
Tues



✓
Wed



Thurs



✓
Fri



Sat



- ① Nados → active → class updates
- ② Nados → H.W.

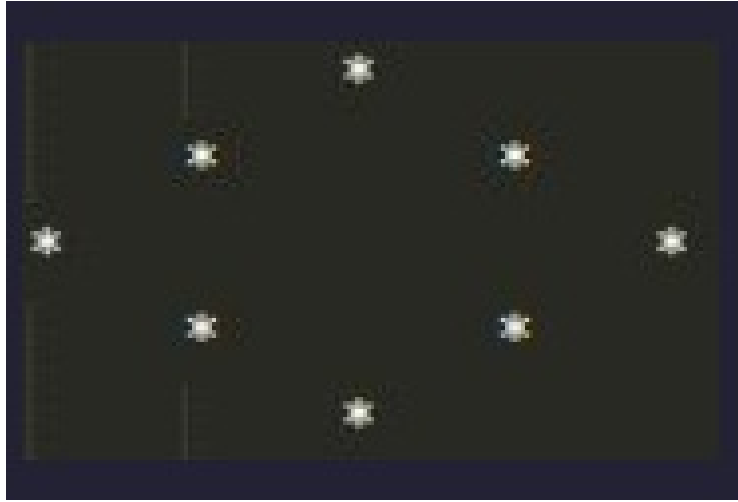
Dream → Comp/DSA kv

- 1. Pattern 10
- 2. Pattern 13
- 3. Pattern 15
- 4. Pattern 18
- 5. Pattern 19

inp/output

15 min ≈ 45 min

Pattern 10





$i = 1, \dots, \text{nstars}$

```
int row = 1 , nspaces = n/2 , nstars = 1;

while(row <= n){
    // code for each row
    //1. spaces
    for(int i = 1 ; i <= nspaces ; i++){
        System.out.print("\t");
    }

    //2. stars
    for(int i = 1 ; i <= nstars ; i++){
        System.out.print("*\t");
    }

    // move to new line
    System.out.println();

    // preparation for next row
    if(row <= n/2){ // first half
        nspaces = nspaces - 1;
        nstars = nstars + 2;
    }else{ // second half
        nspaces = nspaces + 1;
        nstars = nstars - 2;
    }
    row++;
}
```

n=5

row

nspaces

nstars

row	nspaces	nstars
1	2	1
→ 2	1	3
3	0	5

---*

→ * - - - *

```
int n = scn.nextInt();
int row = 1, nspaces = n/2, nstars = 1;

while(row <= n){
    // code for each row
    //1. spaces
    for(int i = 1 ; i <= nspaces ; i++){
        System.out.print("\t");
    }
    //2. stars
    for(int i = 1 ; i <= nstars ; i++){
        if(i == 1 || i == nstars){
            System.out.print("*\t");
        }else{
            System.out.print("\t");
        }
    }

    // move to new line
    System.out.println();

    // preparation for next row
    if(row <= n/2){ // first half
        nspaces = nspaces - 1;
        nstars = nstars + 2;
    }else{ // second half
        nspaces = nspaces + 1;
        nstars = nstars - 2;
    }
    row++;
}
```

n=5

n/2

---*

- * * *

* * * * *

- * * *

- _ *

- _

n/2

nstars

---*

- - * * *

* * * * *

* * * * *

* * * * *

* * *

*

$n=7$

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

	c_1	c_2	c_3	c_4	c_5	c_6	c_7
R_1	*	*	*	*			*
R_2				*			*
R_3				*			*
R_4	*	*	*	*	*	*	*
R_5	*			*			
R_6	*			*			
R_7	*			*	*	*	*

Annotations: $n=7$ is written above the grid. $\frac{n}{2}+1$ is written above the grid with an arrow pointing to c_4 . A green line is drawn across the top of the grid, starting from c_1 and ending at c_7 . A green line is drawn across the bottom of the grid, starting from c_4 and ending at c_7 . The grid is divided into five horizontal sections labeled A, B, C, D, and E on the right side. Section A is the first row (R_1). Section B is the next two rows (R_2, R_3). Section C is the next row (R_4). Section D is the next two rows (R_5, R_6). Section E is the last row (R_7).

if($r == 1$)

if($r > 1$ && $r < \frac{n}{2} + 1$)

if($r == \frac{n}{2} + 1$)

if($r > \frac{n}{2} + 1$ && $r < n$)

if($r == n$)

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

```
for(int r = 1 ; r <= n ; r++){
```

```
    for(int c = 1 ; c <= n ; c++){
```

```
        if(r == 1){ // A
```

```
        }else if(r > 1 && r < n/2 + 1){ // B
```

```
        }else if(r == n/2 + 1){ // C
```

```
        }else if(r > n/2 + 1 && r < n){ // D
```

```
        }else if(r == n){ // E
```

```
        }
```

```
    }
```

```
}
```

R_1
↑

R_2
↑

R_3
↑

R_4
↑

R_5 R_6
↑

R_7
↑


```

int n = scn.nextInt();

for(int r = 1 ; r <= n ; r++){
    for(int c = 1 ; c <= n ; c++){
        if(r == 1){ // A
            if(c == n || (c >= 1 && c <= n/2 + 1)){
                System.out.print("*\t");
            }else{
                System.out.print("\t");
            }
        }else if(r > 1 && r < n/2 + 1){ // B
            if(c == n/2+1 || c == n){
                System.out.print("*\t");
            }else{
                System.out.print("\t");
            }
        }else if(r == n/2 + 1){ // C
            System.out.print("*\t");
        }else if(r > n/2 + 1 && r < n){ // D
            if(c == n/2+1 || c == 1){
                System.out.print("*\t");
            }else{
                System.out.print("\t");
            }
        }else if(r == n){ // E
            if(c == 1 || (c >= n/2 + 1 && c <= n)){
                System.out.print("*\t");
            }else{
                System.out.print("\t");
            }
        }
    }
    System.out.println();
}

```

$R=1$ $C=1$ 2 3 4 5 6 7

$R=2$

$R=3$

$R=4$

$R=5$

$R=6$ * * * * - - *

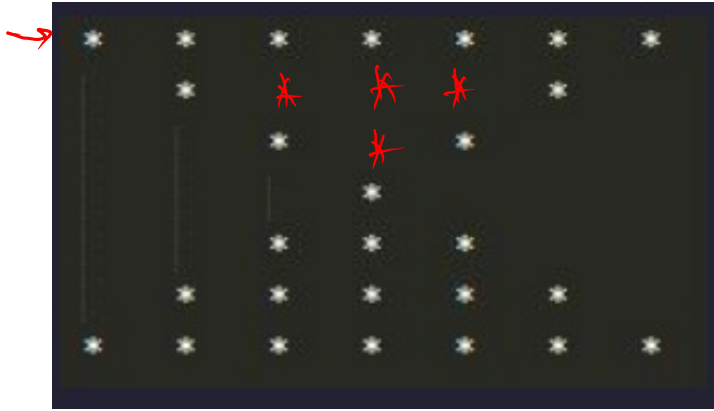
$R=7$

\downarrow \downarrow
 2 3 4 5 6 7
 * *

Pattern → 18 if $(r > 1 \ \& \ r \leq \frac{n+1}{2})$
H.W. ~~I~~

Nabos

II



(Similar to pattern 5)

Pattern
5

- while
- ① code for each row
 - ② next line
 - ③ preparation for next row
 - ④ row++;

Row \rightarrow num

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

num

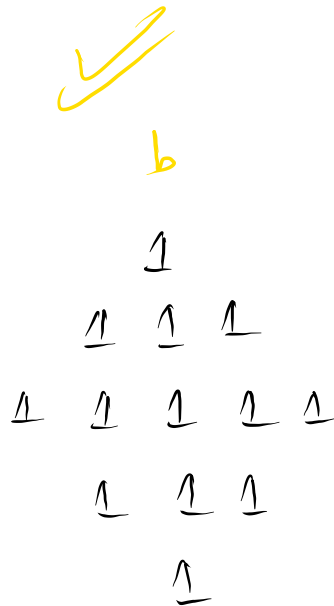
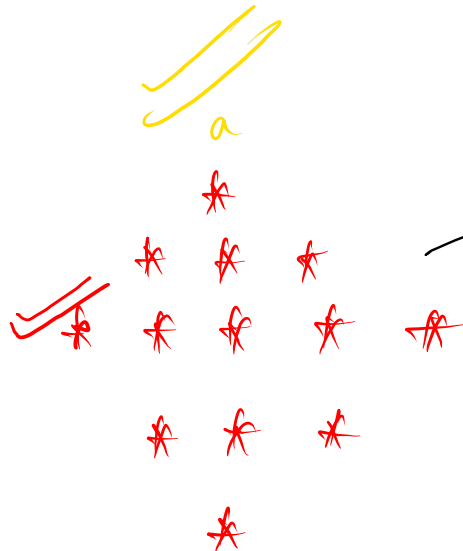
R₁ 1 $\downarrow +1$

R₂ 2 $\downarrow +1$

R₃ 3 $\downarrow +1$

R₄ 2 $\downarrow -1$

R₅ 1 $\downarrow -1$



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

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

```
int row = 1 , nspaces = n/2 , nstars = 1 , num = 1;
```

```
while(row <= n){
    // code for each row
    //1. spaces
    for(int i = 1 ; i <= nspaces ; i++){
        System.out.print("\t");
    }
    //2. stars
    int tmp = num;
    for(int i = 1 ; i <= nstars ; i++){
        System.out.print(tmp+"\t");
        if(i <= nstars/2){
            tmp = tmp + 1;
        }else{
            tmp = tmp - 1;
        }
    }

    // move to new line
    System.out.println();

    // preparation for next row
    if(row <= n/2){ // first half
        nspaces = nspaces - 1;
        nstars = nstars + 2;
        num = num + 1;
    }else{ // second half
        nspaces = nspaces + 1;
        nstars = nstars - 2;
        num = num - 1;
    }
    row++;
}
```

row nsp nst num n=5

row	nsp	nst	num
1	2	1	1
2	1	3	2
→ 3	0	5	3

i = 1 ↓ ↓ ↓ ↓ ↓

* * * * *

(3) (4) (5)

P₁ - - 1

P₂ → - 2 3 2

P₃ → 3 4 5 4 3

P₄ →

P₅

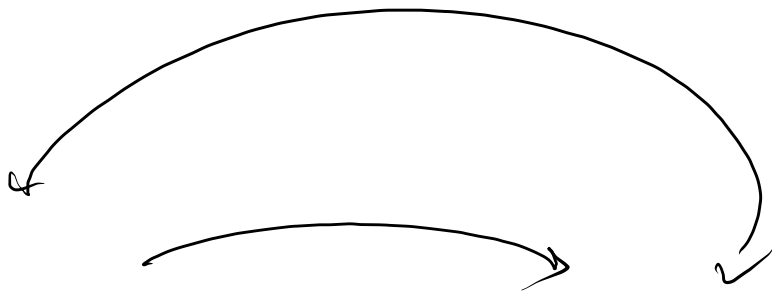
1 + 2 = 3

1 + 1 + 1 = 3

$n=4$

I

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



II

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

$n=5$

	c_1	c_2	c_3	c_4	c_5
$R_1 \rightarrow$	*	*	*		*
$R_2 \rightarrow$			*		*
	*	*	*	*	*
	*		*		
	*		*	*	*

1, 2, 3, 5

$$\left(\overline{c} \geq 1 \ \&\& \ \overline{c} \leq \frac{3}{n/2+1} \right)$$

1 2 3 4 5

$$(c == n \quad || \quad c == n/2 + 1)$$