// Pattern Printing

N
↳ #{ No: of rows = N }
↳ In Every row: { row Increasing }
spaces stars

{ In i^{th} row spaces = // Spaces = {i-1} }
{ In i^{th} row stars = // Stars = { N-i+1 / N+1-i / N-(i-1) } }

N=1 *

row	Spaces	Stars
N=2 1 * *	0	2
2 — *	1	1
N=3 1 * * *	0	3
2 — * *	1	2
3 — — *	2	1
N=4 1 * * * *	0	4 = 4-0
2 — * * *	1	3 = 4-1
3 — — * *	2	2 = 4-2
4 — — — *	3	1 = 4-3
N=5 1 * * * * *	0	5 = 5-0
2 — * * * *	1	4 = 5-1
3 — — * * *	2	3 = 5-2
4 — — — * *	3	2 = 5-3
5 — — — — *	4	1 = 5-4

```
for(int i=1; i<=N; i++){
    // Spaces:
    for(j=1; j<=(i-1); j++){
        print(" ")
    }
    // Stars:
    for(j=1; j<=(N+1-i); j++){
        print("*")
    }
    println()
}
```

// Pattern - 11

→ given N:

↳ # N rows are present

↳ # In Each row : {1 - N}

In i^{th} row:

: left Stars : N+1-i

: space : 2*i - 2

: right Stars : N+1-i

N = 5

row i

	LS	SP	N+1-i
1	5	0	: 5
2	4	2	: 5+1-2 = 4
3	3	4	: 5+1-3 = 3
4	2	6	: 5+1-4 = 2
5	1	8	: 5+1-5 = 1

N = 4:

	SP
1	0
2	2
3	4
4	6

N = 6

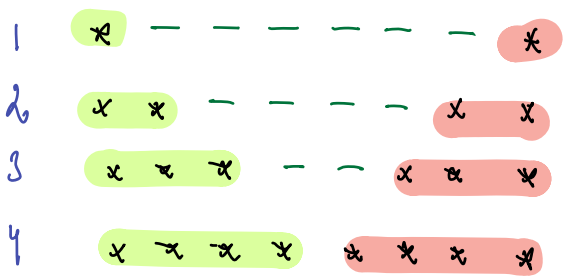
Pseudocode:

```
for(i=1; i<=N; i++){
    for(j=1; j<=(N+1-i); j++){
        print(*)
    }
    for(j=1; j<=(2*i-2); j++){
        print( )
    }
    for(j=1; j<=(N+1-i); j++){
        print(*)
    }
}
```

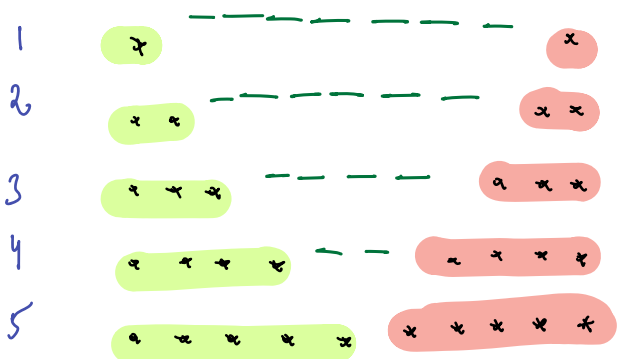
LS	SP
6	0 = 2*1-2
5	2 = 2*2-2
4	4 = 2*3-2
3	6 = 2*4-2
2	8 = 2*5-2
1	10 = 2*6-2

// Pattern-12 :

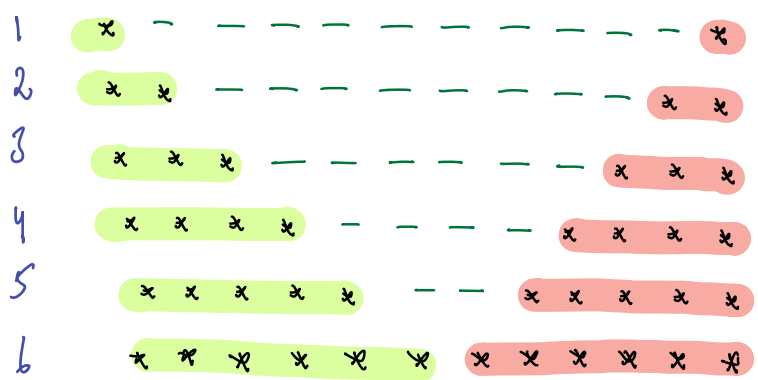
N=4:



N=5:



N=6



given N

↳ # N rows → {1, 2 .. N}

At ith row how many : 2N columns

: leftStars : i Stars

: Spam : 2N-2i

: rightStars : i Stars

In Every row how many columns we have = 2N + ?

LS + Spam + RS = 2N

i i

Spam = 2N - 2i

// Pattern-14

N=1 : 1

N=2 : 1 1
2 1 2

N=3 1 1
2 1 2
3 1 2 3

N=4 1 1
2 1 2
3 1 2 3
4 1 2 3 4

N=5 1 1
2 1 2
3 1 2 3
4 1 2 3 4
5 1 2 3 4 5

given N
↳ rows : N rows
↳ In i^{th} row : i Elements → [1-i]

Pseudocode :

```
for(i=1; i<=N; i++){
    for(j=1; j<=i; j++){
        print(j)
    }
    println()
}
```

Trace N = 4

i	j=1; j<=i; j++	output
1	[1 1]	1
2	[1 2]	1 2
3	[1 3]	1 2 3
4	[1 4]	1 2 3 4

// Pattern - 15 :

N = 1 → 1

N = 2 → 1
2 3

N = 3 → 1
2 3
4 5 6

N = 4 → 1
2 3
4 5 6
7 8 9 10

N = 5 → 1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

i = 5

↳ j = 1 ; j < 5 ; j++

N → N rows

In i^{th} row ⇒ i Elements

// Pseudo

```
int c = 1;
for(int i=1; i<=N; i++){
    for(int j=1; j<=i; j++){
        print(c)   c = c+1
    }
    print("\n")
}
```

optional :

a) i+j b) j+1 c) j

Check ?