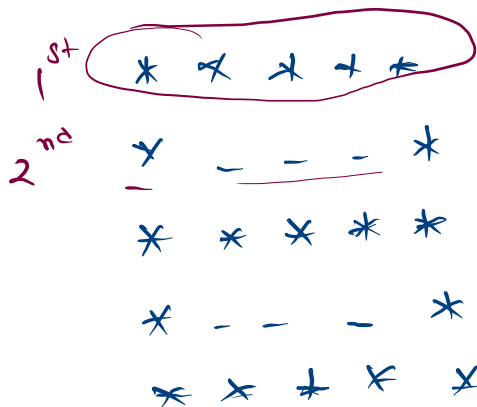


$$n=5$$



1 0
2
3
4
5

now →

odd →
even →

all
* — n-2 — *

$n=3$
 $n \text{ rows}$

$star = 1$
 $space = n - 1$

$n \text{ rows}$
 $space \quad (star)$
 $star++$
 $space--$

Diamond.

$$\text{row?} = 2n - 1$$

Sample Input 0

$n=5$

Sample Output 0

$n=5$

```

1      *
2     ***
3    *****
4   *****
5  *****
6  *****
7   *****
8    *****
9     *
    
```

Annotations on output:

- Row 1: $n=5$
- Row 2: $***$ (3 stars)
- Row 3: $*****$ (5 stars)
- Row 4: $*****$ (5 stars)
- Row 5: $*****$ (5 stars)
- Row 6: $*****$ (5 stars)
- Row 7: $*****$ (5 stars)
- Row 8: $*****$ (5 stars)
- Row 9: $*$ (1 star)

$n=3$

Diagram for $n=3$:

```

      *
     * *
    * * *
   * * *
  * * *
 * * *
* * *
    
```

Annotations:

- 3 row (circled)
- 2 row (underlined)
- $n + (n-1) = 2n - 1$
- $2 \times 3 = 6$
- Row 1: $*$
- Row 2: $* * *$
- Row 3: $* * * *$
- Row 4: $* * *$
- Row 5: $* * *$
- Row 6: $* * *$
- Row 7: $* * *$

space ~ ~ ~ star ~ ~ ~

$$2n-1$$

$$2(5)-1$$

$$= 9$$

$$n + n - 1$$

$$5 + 5 - 1$$

$$= 9$$

Upper half

star = 1
space = n - 1



star += 2
Space --;

lower half

star -= 2
Space++;

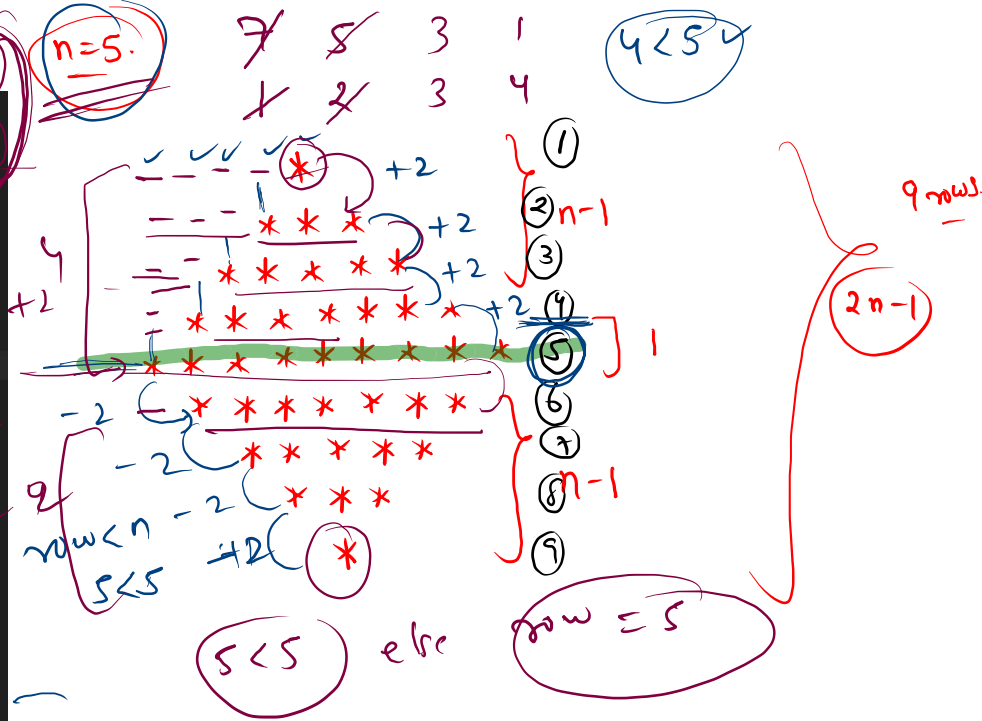
```

int star = 1;
int space = n-1;

for(int row = 1; row <= (2*n)-1; row++){
    for(int csp = 1; csp <= space; csp++){
        System.out.print(" ");
    }
    for(int cst = 1; cst <= star; cst++){
        System.out.print("*");
    }

    if(row < n){
        star += 2;
        space--;
    }
    else{
        star -= 2;
        space++;
    }
    System.out.println();
}

```



n → 5

3

7

9 rows

5 rows

13 rows

2n-1

functions.

$$f(x) = x^2$$

$$f(2) = 4.$$

$$f(x) = 2x + 9$$

2 3 4

$$f(2) = 13$$

p. s.

void or int

return type

what is
funⁿ will
return

{

body,

}

sum

name of
funⁿ

(parameters.)
(optional).

WAP to find factorial.

factorial.

$$(6!)$$

$$\begin{aligned} &= 6 \times 5 \times 4 \times 3 \times 2 \times 1 \\ \text{or} & 1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720 \end{aligned}$$

$$\underline{5!}$$

$$\hookrightarrow 1 \times 2 \times 3 \times 4 \times 5 = 120$$

$$\begin{aligned} 4! &= 1 \times 2 \times 3 \times 4 \\ &= 24 \end{aligned}$$

$$n! = 1 \times 2 \times 3 \times \dots \times (n-1) \times n$$

factorial.

```
public static void main(String[] args) {  
    int n = 4;  
  
    int ans = 1;  
  
    for(int i = 1; i <= n; i++){  
        ans *= i;  
    }  
    System.out.println(ans);  
}
```

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

$$\begin{matrix} n=6 \\ r=2 \end{matrix} = \frac{6!}{4!}$$

factorial.

$$f(x) = 1 \times 2 \times 3 \dots \times x$$

Diagram illustrating the factorial function $f(x)$. The input x is circled in red, with an arrow pointing to the word "value" below it. The entire expression $1 \times 2 \times 3 \dots \times x$ is enclosed in a blue box, with an arrow pointing to the word "ans." below it.

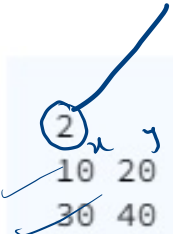
factorial (n) =

$$as = 1 \times 2 \times 3 \times 4 \times 5 \dots n$$

Diagram illustrating the factorial calculation for $n=5$. The sequence of numbers 1, 2, 3, 4, 5 is shown. Above the numbers 1, 2, 3, 4, 5 are small circles containing the letters 'i', 'i', 'i', 'i', 'i' respectively. An arrow points from the circle above 4 to the circle above 5, indicating the iterative multiplication process. The final result 120 is circled in red.

```
public class Solution {  
    //own function  
    public static int factorial(int n){  
        //logic  
        int ans = 1;  
        for(int i = 1; i <= n; i++){  
            ans *= i;  
        }  
        return ans;  
    }  
  
    public static void main(String[] args) {  
        Scanner scn = new Scanner(System.in);  
        int n = scn.nextInt();  
  
        int fn = factorial(n);    // fn = 720  
  
        System.out.println(fn);  
    }  
}
```


Sum using functions:-



2	x	y
10	20	
30	40	

Sample Output 0

30
70

TC 1

x	y
10	20

= 30

TC 2

x	y
30	40

= 70

$$f(x) = x^2$$

→ ans