```
import java.util.*;

class Main {
 public static void main(String args[]) {
 // Your code goes here

 /*
 Agenda :
 - 2 quizzes
 - 4-5 questions on Pattern printing
 */



 /*
 for(int i = 1; i<=10; i=i+2){
     System.out.println(i);
 } // 1 3 5 7 9
 */



 // B -> * N-1 times
 //for(int i=1; i<N; i++){
 //  System.out.print("*");
 // }

 // DRY RUN : N = 10
 //1, 2, 3, 4, 5, 6, 7, 8, 9  -> 9 numbers




 // C -> * N+1 times
 //for(int i=0; i<=N; i++){ System.out.print("*");}
 // DRY RUN : N = 10
 // 0, 1, 2,3 ,4 ,5 ,6 ,7 , 8, 9, 10 -> 11 numbers



 // A -> * N times
 //for(int i=1; i<=N; i++){ System.out.print("*");}
 // DRY RUN : N = 10
 // 1,2, 3, 4,5, 6, 7,8, 9, 10 -> 10



 // MATHEMATICAL NOTATION:

 // i takes values from 1-> 10 : Boundary numbers are included
 // i -> [1,  10]
 // i -> [9,  99]



 // i -> (1, 10 ] , i takes value of 10.
 // i -> [2, 10]

 // i -> (1, 10) : boundary numbers are excluded.
 // i : 2, 3,4 ,5,6,7,8,9
// i -> [1,  10] : 1, 2,3,4,5 ,6,7,8,9,10
// i-> [9, 99]: 90 => 99 - 9 + 1 => 91

// - - - - - - - - - - - -
//     L=9       R=99


 // [L, R] -> R - L +1



// Q. Print the following Pattern

// N = 3
/*

***
***
***

*/

// N = 4
/*

****
****
****
****

*/

// Open IDE -> Code the logic -> Check output.


Scanner s = new Scanner (System.in);
int n= s.nextInt();
// for(int i=1;i<=n;i++){
//  System.out.println("***");
// }


// Number of rows? N

// row number 1 -> print stars
// row number 2 -> print stars
// row number 3-> print stars

/*
for(int i = 1; i <= n; i++){
 // this logic prints ith row
 // in each row, I need to print n stars
 for(int j=1; j <= n; j++){
  System.out.print("*");
 }
 System.out.println();

}*/


/* Dry run :

N = 3.

i.    j.     output
1.    1        ***
   2.
   3
   4 ( break )

  2.    1        ***
   2.      ***
   3.
   4 ( break )


3.    1        ***
   2.      ***
   3.      ***
   4 ( break )

4.(break)

*/


// Q. Print the following Pattern

// N = 3

/*

***
**
*

*/

// N = 4

/*

****
***
**
*

*/
```

```
// N = 5

/*

*****
****
***
**
*

*/

/*
for(int i =1; i <= n; i++){
 // logic for ith row
 // [ 1, n - i + 1] => n - i + 1
 for(int j = 1; j <= (n - i + 1); j++){
  System.out.print("*");
 }
 System.out.println();
}*/

/*
for(int i =1; i <= n; i++){
 // logic for ith row
 // [ 1, n - i + 1] => n - i + 1
 for(int j = i; j <= n ; j++){
  System.out.print("*");
 }
 System.out.println();
}*/

// Now I just need to find out number
// of stars in ith row.

// Value known to me before executing
// internal for loop
// 'i', 'n'
// if I can figure out the number
// of stars in terms of n and i,
// I'm almost done!

/*


N = 5
i.     number of stars
1  5 -> 5(N) - 0 (1(i) - 1)
2  4 -> 5(N) - 1 (2(i) - 1)
3  3 -> 5(N) - 2 (3(i) - 1)
4  2 -> 5(N) - 3 (4(i) - 1)
5  1 -> 5(N) - 4 (5(i) - 1)

ith row -> N - ( i -1) -> N - i  + 1
ith row ->  N - i  + 1

n = 5
i = 5, 5 -5 +1 = 1 stars
i = 1, 5 - 1 + 1 = 5 stars
*/


// in ith row, need to print n-i+1 stars
// j = 1 to n-i +1
// j = i to n -> [i, n] -> n -i +1




// Q. Print the following pattern

// N = 3

/*

*
**
***

*/


// N = 4

/*

*
**
***
****

*/


// N = 5

/*

*
**
***
****
*****

*/

/*
for(int i = 1; i <= n; i++){
 // logic for every ith row
 // 'i' & 'n' are known to me.
 for(int j = 1 ; j <= i; j++){
  System.out.print("*");
 }
 System.out.println();
}*/


/*
N = 3

output
*
**
***



i.    j.
1.     1
    2 (break)

2.     1
    2
    3(break)

3.     1
    2
    3
    4(break)

4(break)

/*
N -> 5

i.    number of stars
1.     1(i)
2.     2(i)
3.     3(i)
4.     4(i)
5.     5(i)
*/

// Pattern with spaces ( - )

/*
// N = 3

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

// N = 4

*---*
```

```
*---*
*---*
*---*

// N = 5

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

*/

/*
for(int i = 1; i <= n; i++){
 // logic for ith row
 for(int j = 1; j <= n+1; j++){
  if( j == 1 || j == n+1){
   System.out.print("*");
  } else {
   System.out.print("-");
  }

 }
 System.out.println();
}*
 */
// each row -> * + some - + *

/*
for(int i = 1; i <= n; i++){
 // logic for ith row
 System.out.print("*");
 for(int j = 1; j <= n-1; j++){
  System.out.print("-");
 }
 System.out.print("*");
 System.out.println();
} */


/* DRY RUN

N = 3;

Output
*--*
*--*
*--*

i      j
1.     1
       2
    3(break)
```

```
2.     1
      2
   3(break)

3.     1
      2
   3(break)

4( break)*/



// Q. Pattern printing
/*

N =3

*--*
*-*
**

N = 4

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

N =5

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

*/

// each row -> * + some - + *

/*
for(int i = 1; i <= n; i++){
 // logic for ith row
 System.out.print("*");
 for(int j = 1; j <= n- i; j++){
  System.out.print("-");
 }
 System.out.print("*");
 System.out.println();
}*/


// N = 5
```

```
/*

i.    number of dashes(-)
1.      4 -> 5(N) - 1(i)
2.      3 -> 5(N) - 2(i)
3.      2 -> 5(N) - 3(i)
4.      1 -> 5(N) - 4(i)
5.      0 -> 5(N) - 5(i)
*/



for(int i = 1; i <= n; i++){
 // logic for ith row
 for(int j = 1; j <= n - i + 2; j++){
  if( j == 1 || j == n - i + 2){
   System.out.print("*");
  } else {
   System.out.print("-");
  }

 }
 System.out.println();
}



/*

N = 5
i -> how many items
1.    6 -> 5(N) - 1(i) + 2
2.    5 -> 5(N) - 2(i) + 2
3.    4 -> 5(N) - 3(i) + 2
4.    3 -> 5(N) - 4(i) + 2
5.    2 -> 5(N) - 5(i) + 2

//ith -> n - i + 2




// ith row -> n-i */




 }
}
```