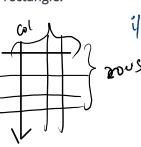


Pattern 7 - Print a hollow m by n star rectangle.



if  $p \rightarrow m$  (col) = 5  
 $\rightarrow n$  (row) = 4

5x4

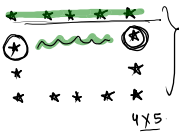


m=6

n=4



m=5  
n=4



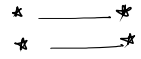
for ( 4 - (n) )  
 {  
 if ( 1<sup>st</sup> row & last row  
 all stars



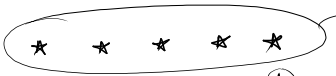
2 stars

m-2

else



m=5  
n=4



5-2 = 3



m=5  
n=4

rows = 1 x 3 x 4 x 5

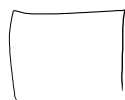
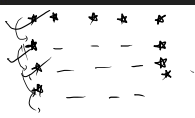
1 ≤ 4  
 2 ≤ 4  
 3 ≤ 4  
 4 ≤ 4

rows = 1 2 3 4 5  
 col = 1 2 3 4 5

col ≤ m

6 ≤ 5

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int m = sc.nextInt(); //rows
    int n = sc.nextInt(); //cols
    for(int rows = 1; rows <= m; rows++){
        if(rows == 1 || rows == m){
            for(int col = 1; col <= n; col++){
                System.out.print("*");
            }
        }
        else{
            System.out.print(" ");
            for(int col = 1; col <= n; col++){
                System.out.print(" ");
            }
        }
        System.out.println();
    }
}
```



Pattern 8 - Print a hollow square without top

i/p  $\rightarrow$   $\textcircled{n} = 5$



Obs.

Total rows =  $\textcircled{n}$

for (             $\textcircled{n}$  )

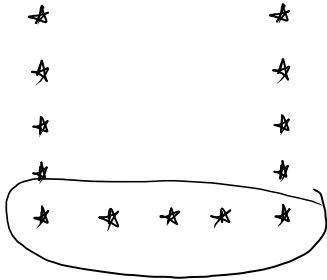
{

$\textcircled{2}$

rows

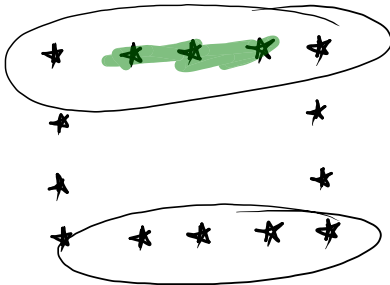
1. last rows
2. other.

}



1st

$m = 5$   
 $n = 4$

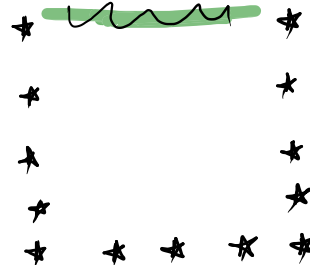


2nd

(Square)

$m = 5$   
cols

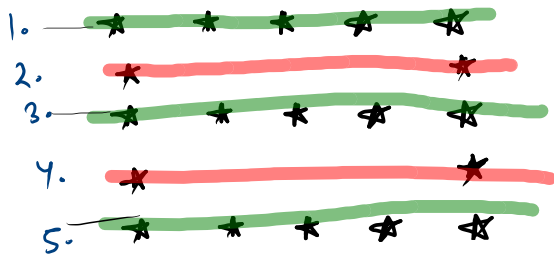
$\textcircled{n = 5}$



# Square Ladder

$n=5$

rows = 1 —  $5/n$



✓ 1. Obs.

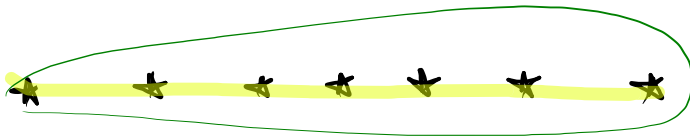
$n$  rows

2.

odd → complete star.  
 even → space in b/w.

$n=7$

1. ✓



2.



3.



4.



5.



6.



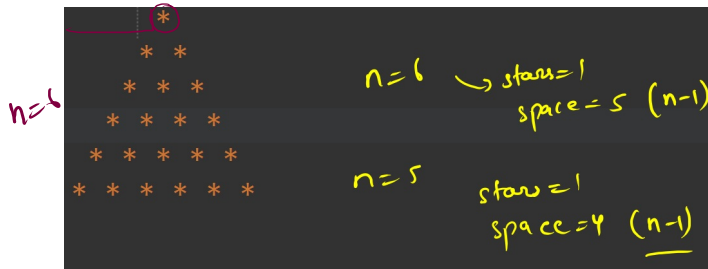
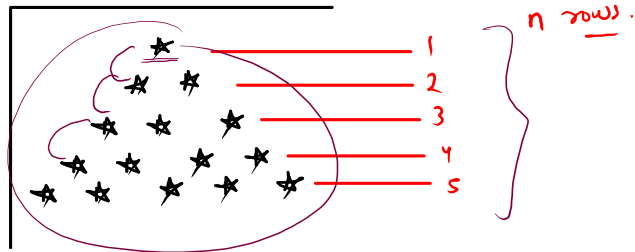
7.



# GKSTR24 Pattern\_7\_Pyramid

n=5

star 8 = 1



```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        //init cond
        int stars = 1;
        int spaces = n-1;

        for(int rows = 1; rows <= n; rows++){
            //spaces
            for(int csp = 1 ; csp <= spaces; csp++){
                System.out.print(" ");
            }
            //stars
            for(int cst = 1; cst <= stars; cst++){
                System.out.print("* ");
            }
            stars++;
            spaces--;
            System.out.println();
        }
    }
}
```

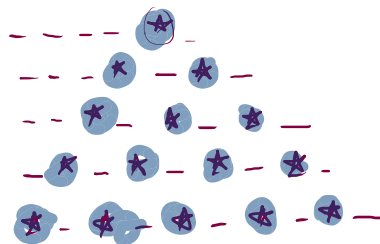
n=5.

rows = 1 2 3 4 5

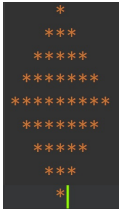
stars = 1 2 3 4 5

sp = 4 3 2 1 0

5 ≤ n ✓



# Point Diamond:



$n=5$

9 nos.

1. 1st no. of nos =  $(2 \times n) - 1$  ✓

2. init

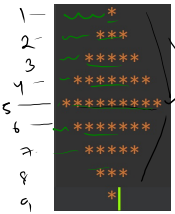
spaces =  $n-1$

stars = 1

3. for (total no. of rows)

$n=3$

5 rows



$n=5$

UH

LH

$row > n$

$row < n$

$< =$

$<$

upper half

star += 2

space --

lower half

star -= 2

space ++

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

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

    if(rows < n){
        //upper half
        star += 2;
        space--;
    }
    else{
        //lower half
        star -= 2;
        space++;
    }
}
```

$n=3$

star = 1 ✓ 1 ≤ 5

space = 2 ✓ 2 ≤ 5

star = 3 ✓ 3 ≤ 5

star = 4 ✓ 4 ≤ 5

star = 5 ✓ 5 ≤ 5

star = 6 ✗ 6 > 5

1 < 3

2 < 3

3 < 3

4 < 3

5 < 3

