Print stars in a single line.

→ i/p → 5

fixed.

3

7

Sys (" ** * ** ")

```java
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();

        for(int i = 1; i <= n; i++){
            System.out.print("*");
        }

    }
}
```

Pattern 2 - Print n x 12 star rectangle

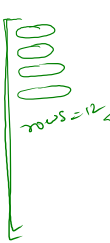
Take n as an integer input and then print a star rectangle such that each line has n stars.
Also, there are 12 such lines.

ample Output

rows. = 12

12 lines.

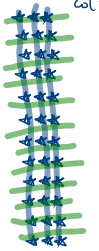
Six → star → line

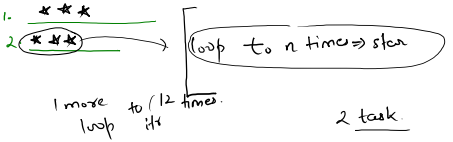


rows = 12

if n = 3

col = 3





Statement: –

we have to print 12 rows & in each row we have to print n stars.

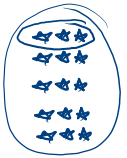
```java
import java.io.*;
import java.util.*;

public class Solution {

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

        int rows = 12;
        int cols = n;

        for(int r = 1; r <= rows; r++){
            //do your task for each row

            for(int c = 1; c <= cols; c++){
                System.out.print("*");
            }
            System.out.println();
        }

    }
}
```

rows = 12

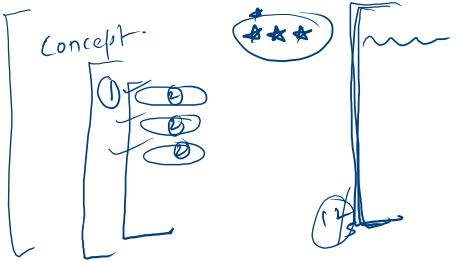
cols = 3

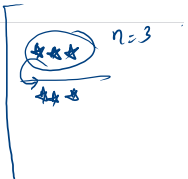
r = 1 1 ≤ 5

C = 1

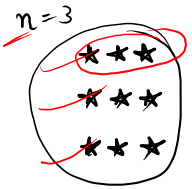


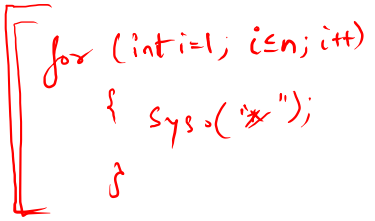
Concept.



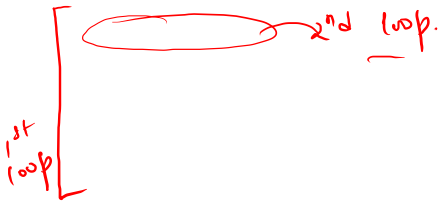


Pattern 3 - nxn star rectangle

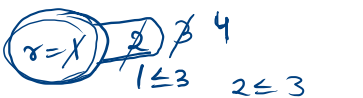


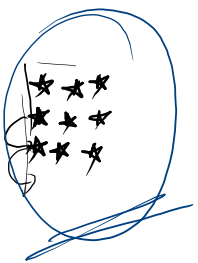


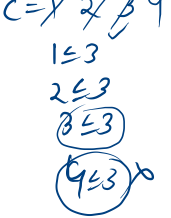
n × n
r c




```java
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();

        for(int r = 1; r <= n; r++){
            for(int c = 1; c <= n; c++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```







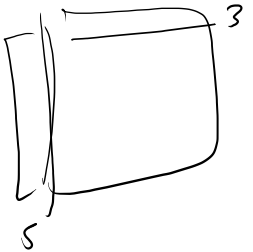
n=3.

GKSTR19 Pattern_4

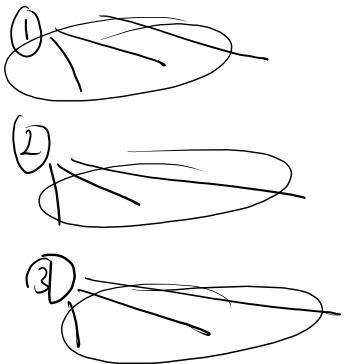
no. of rows = no. of stars in each row.

for (r=[1 — n] rows.)
  for (— r)
    {
    —
    }
}

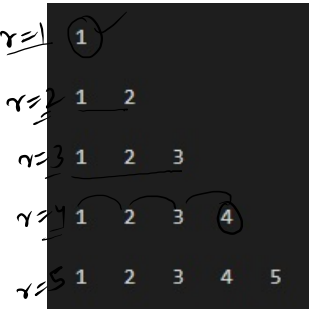
n=3

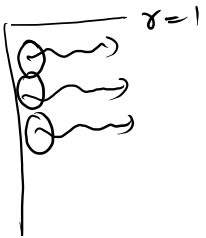


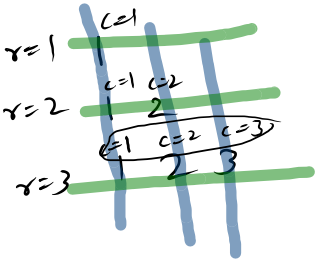
5



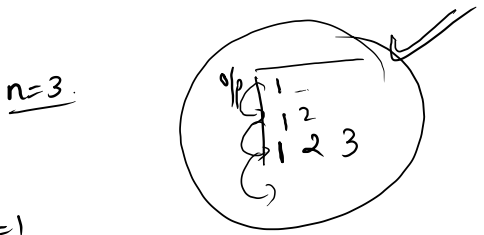
GKSTR17 Pattern 2

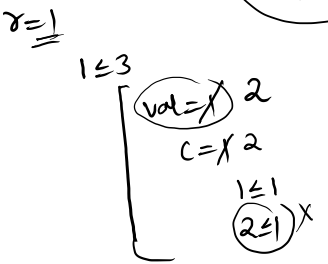


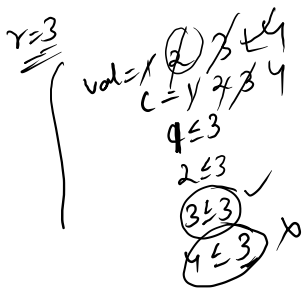


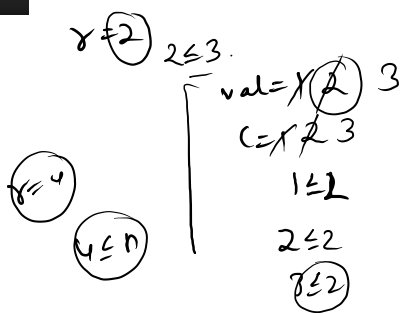



```java
import java.io.*;
import java.util.*;

public class Solution {

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

        for(int r = 1; r <= n; r++){
            int val = 1;
            for(int c = 1; c <= r; c++){
                System.out.print(val + " ");
                val++;
            }
            System.out.println();
        }

    }
}
```

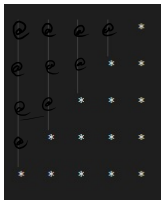








GKSTR20 Pattern_5



i/p → n=5.

```
for(int rows = 1; rows <= n; rows++){
    int spaces = n-rows;
    int stars = rows;
}
```

1st row

n-1 ⇒ @/" "

1 = star

n-3 ⇒ @/"'

3 ⇒ star

2nd row

n-2 ⇒ @/"'

2 = star

4th row

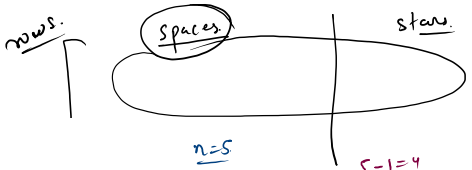
n-4 ⇒ @/"'

4 ⇒ star

5th row

n-5 ⇒ @/"'

5 ⇒ star




```
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();

        for(int rows = 1; rows <= n; rows++){
            int spaces = n-rows;
            int stars = rows;

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

n=5

n-rows

5-1=4

4 space 1 star

rows=1

rows=2

rows=3

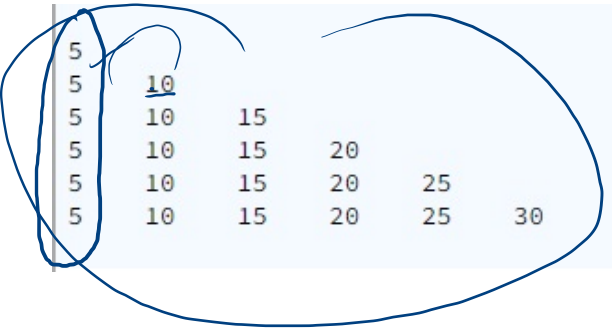
rows=4

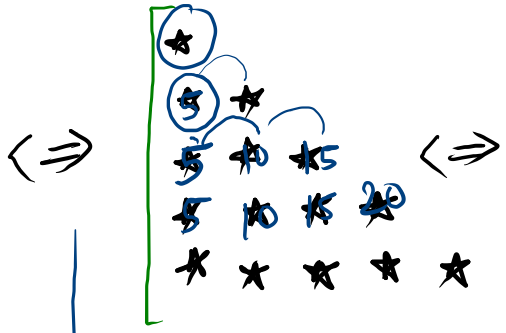
rows=5

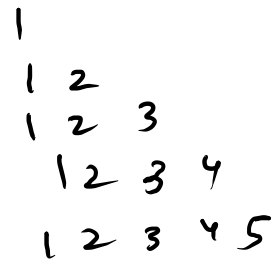
n-3=2

Pattern 6 - Right triangle of 5 multiples

```
5
5    10
5    10    15
5    10    15    20
5    10    15    20    25
5    10    15    20    25    30
```







5

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

    for(int r = 1; r <= n; r++){
        int val = 5;
        for(int c = 1; c <= r; c++){
            System.out.print(val + " ");
            val += 5;
        }
        System.out.println();
    }

}
```