

Pattern 3 - nxn star rectangle

Problem

Submissions

Leaderboard

Discussions

Take n as an integer input. Then print nxn star rectangle as mentioned below.

In each line, n stars should be printed.

And there should be n such lines.

```
for ( i → n ) // rows
    for ( j → n ) // colm
        print (*)
    } println();
```

```
Scanner scn= new Scanner(System.in);
int n = scn.nextInt();
for(int i = 1; i<=n ; i++){
    for(int j = 1; j<=n ; j++){
        System.out.print("*"); ✓
    }
    System.out.println(); ✓
}
```

Sample Input 0

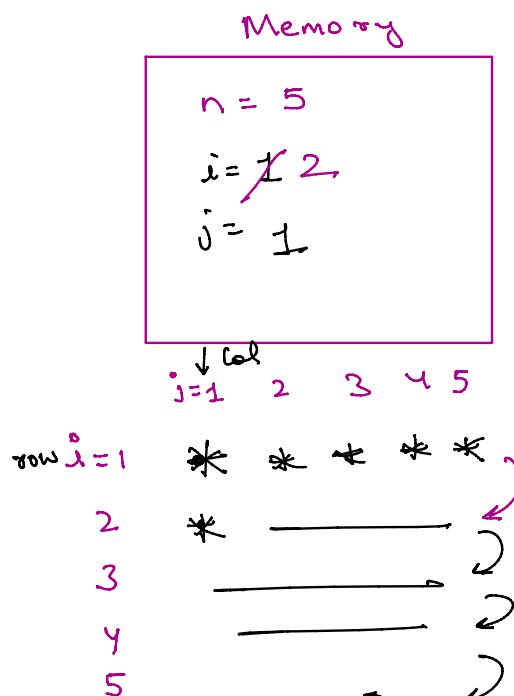
5

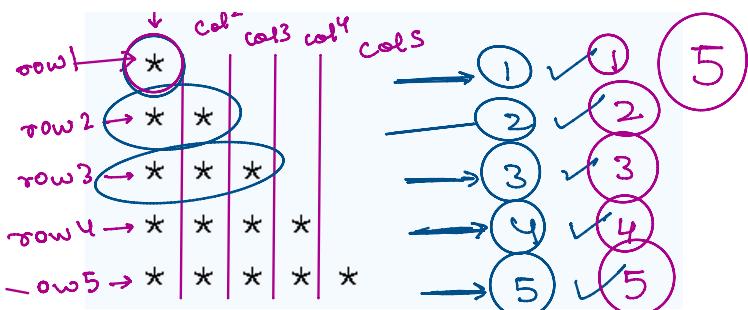
Sample Output 0



int star=1 row
for (i → n)
 for (i → star) {
 print (*);
 }

Memory
star=1, 2, 3, 4





```
for ( i → start ) {
    print (*);
}
println();
start++;
```

Memory
Star = 1/2/3/4

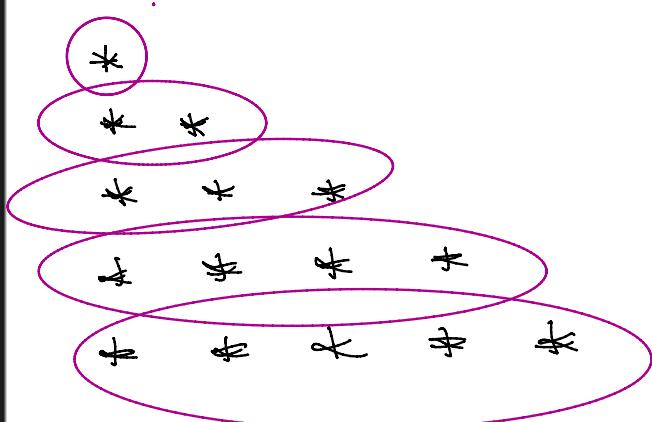
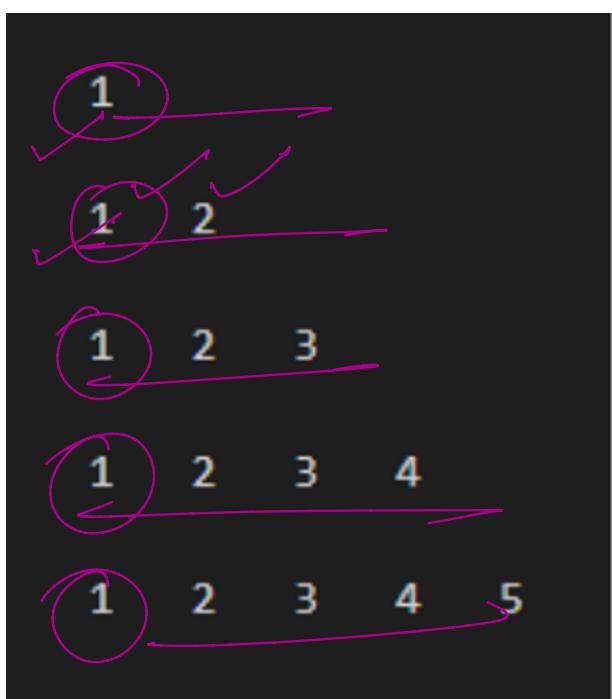
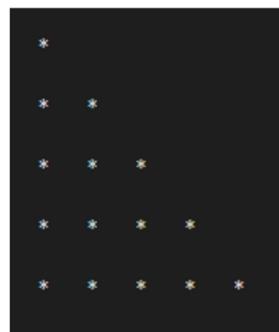
*
* *
* * *
* * * *

```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt(); → 5
int star = 1;
for(int i = 1; i <= n; i++){ // rows
    for(int j = 1; j <= star; j++) { // col
        System.out.print("*");
    }
    System.out.println();
    star++;
}
```

Memory
n = 5
i = 1/2/3/4/5
j = 1
Star = 1/2/3/4/5

1. 2. 3. 4. 5

1. *
- 2. * *
- 3. * * *
- 4. * * *
- 5. * * *



```
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int star = 1;
for(int i = 1; i <= n; i++){
    for(int j = 1; j <= star; j++) {
        System.out.print("*");
    }
}
```

? ?



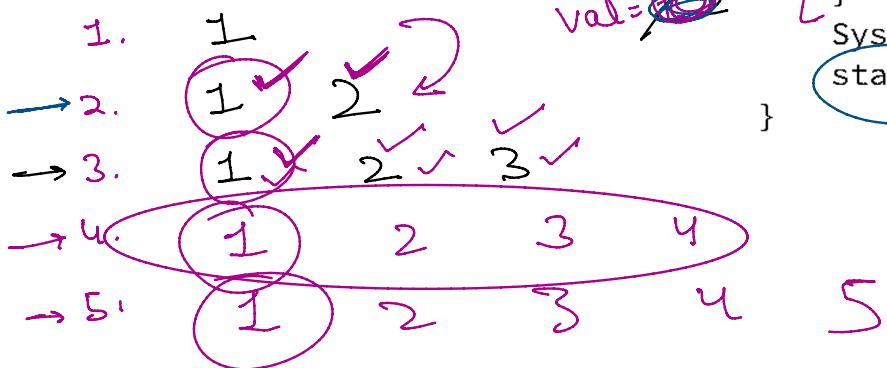
```

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

```

Diagram illustrating the execution of the nested loops:

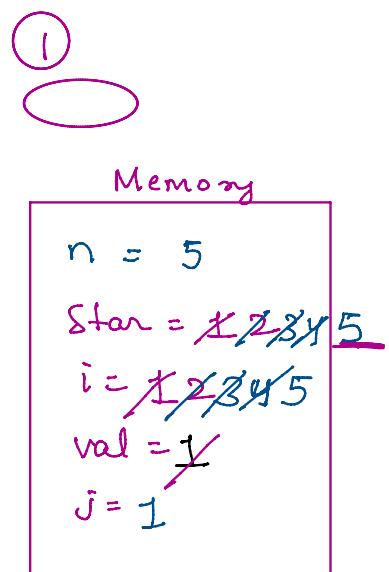
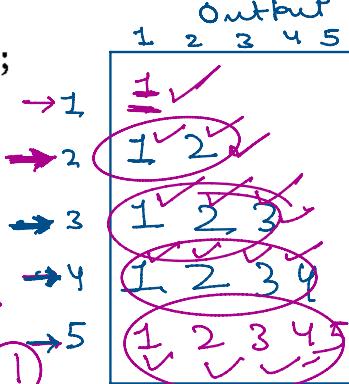
- Iteration 1: $i=1$, $star=1$. Prints 1 asterisk (*).
- Iteration 2: $i=2$, $star=2$. Prints 2 asterisks (* *).
- Iteration 3: $i=3$, $star=3$. Prints 3 asterisks (* * *).
- Iteration 4: $i=4$, $star=4$. Prints 4 asterisks (* * * *).
- Iteration 5: $i=5$, $star=5$. Prints 5 asterisks (* * * * *).



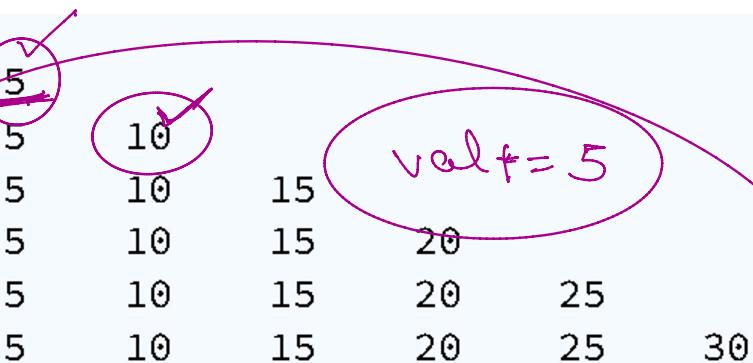
```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int star = 1;
for(int i = 1; i<=n ; i++){
    int val = 1;
    for(int j=1 ; j<=star ; j++){
        System.out.print(val + " ");
        val++;
    }
    System.out.println();
    star++;
}

```



$val = 5$



```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int star = 1;
for(int i = 1; i<=n ; i++){
    int val = 5;
    for(int j=1 ; j<=star ; j++){
        System.out.print(val + " ");
        val+=5;
    }
    System.out.println();
    star++;
}

```

```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt();
int star = 1; 6 <= 5
for(int i =1 ; i<=n ; i++){
    int val = 5;
    for(int j = 1 ; j<= star; j++){
        System.out.print(val + "\t");
        val+=5;
    }
    System.out.println(); ✓
    star++; ✓
}

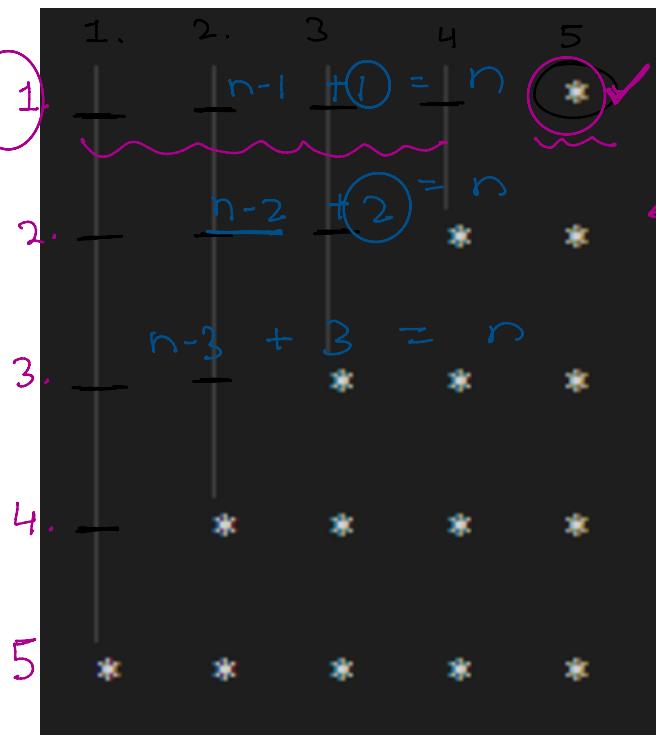
```

Output

1.	5
2.	5 10
3.	5 10 15
4.	5 10 15 20
5.	5 10 15 20 25

Memory

n = 5
star = 1 2 3 4 5 6
i = 1 2 3 4 5 6
val = 5 10 15 20 25 30
j = 1 2 3 4 5 6



$$n-1 = 5 \times 5$$

4 spaces
3 spaces
2 spaces
1 space
0 space

1 star
2 star
3 star
4 star
5 star

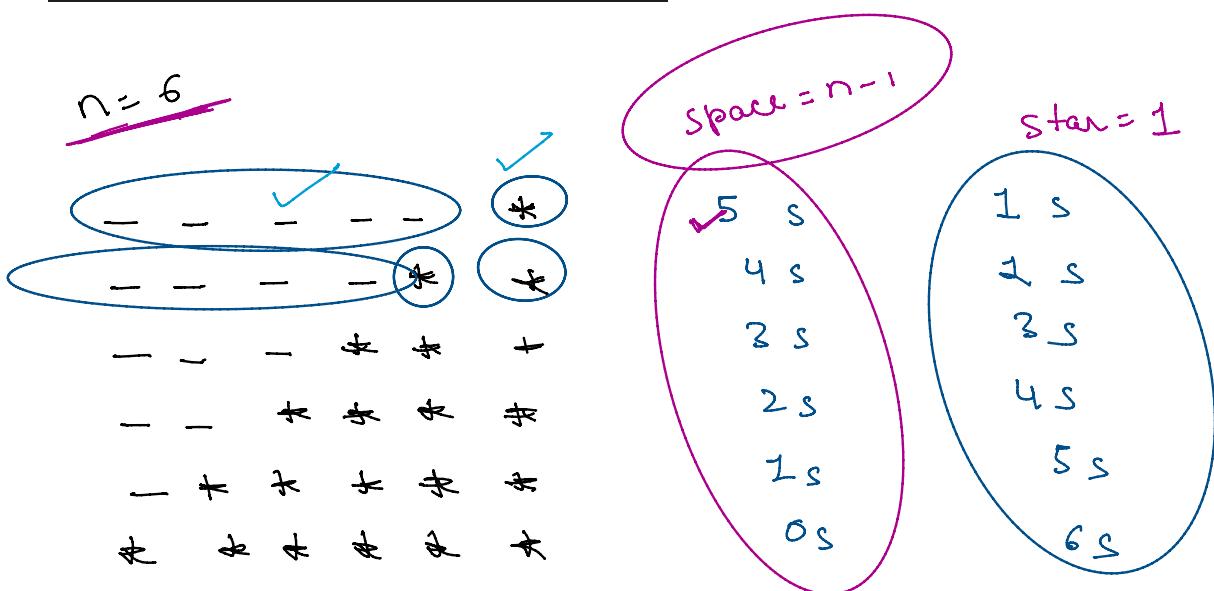
Space decreased by 1

star is getting increased by 1

5 * * * *

↓
Space decreased
by 1

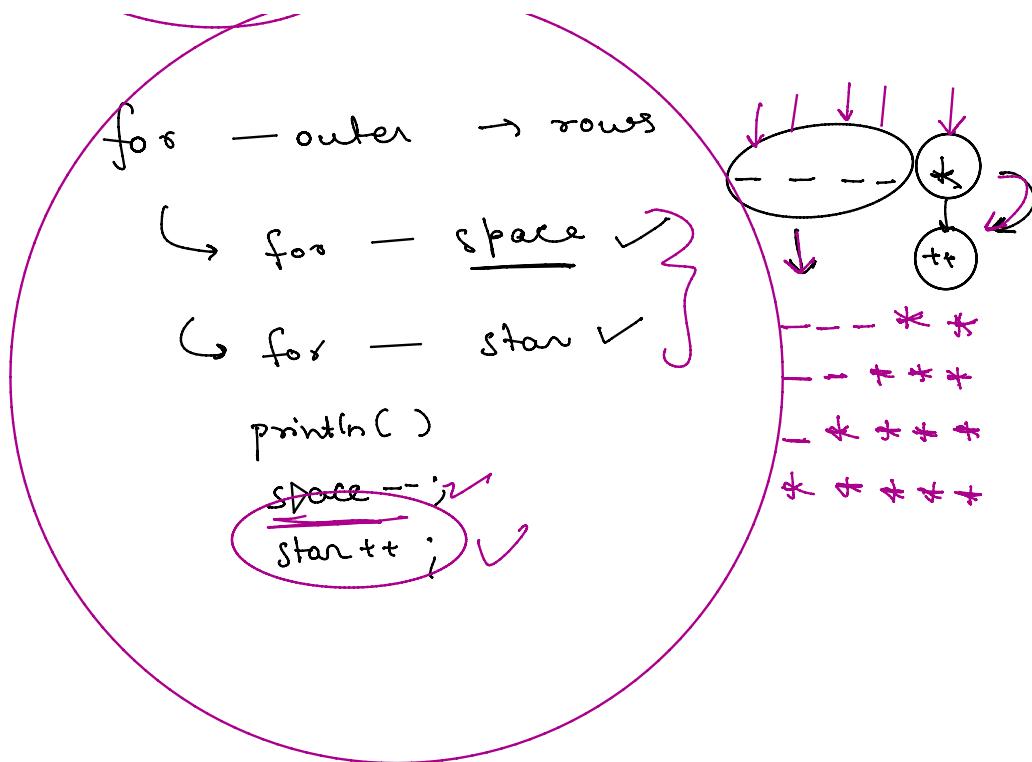
Star is decreasing
increased by 1



Scanner class

$n \rightarrow 5$

star = 1 }
Space = $n - 1$ ✓ }
for (i = 1 → n) // rows }
 for (1 → space) }
 print (" "); } space } col
 for (1 → star) } star }
 print (" * "); }
 }
 println(); }
 star++;
 Space--; }
 , , , ,



```

Scanner scn = new Scanner(System.in);
int n = scn.nextInt(); 5
int star = 1;
int space = n - 1;
for(int i = 1; i <= n; i++) {
  for(int j = 1; j <= space; j++) {
    System.out.print(" ");
  }
  for(int j = 1; j <= star; j++) {
    System.out.print("*");
  }
  System.out.println(); ✓
  space--;
  star++;
}
  
```

Memory

n=5
Star = 1 2 3 4 5 6
Space = 4 3 2 1 0 -1

1. - - - - *
2. - - - * *
3. - - * * *
4. - * * * *
5. * * * * *