

----- Pattern problems in **JAVA** -----

1) Solid rectangle

output format:

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

// logic approach: each row, 1 to 'c' columns and next row so on.. (use nested for loops)

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int r,c,i,j;
        r = sc.nextInt();
        c = sc.nextInt();
        for(i=1;i<=r;i++){
            for(j=1;j<=c;j++){
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

// 4 rows 5 cols

output:

```
4 5
* * * * *
* * * * *
* * * * *
* * * * *
```

2)Hollow rectangle

output format:

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

```
/*
(1,1)(1,2)(1,3)(1,4)(1,5)
* * * * *
(2,1)*       * (2,5)
(3,1)*       * (3,5)
* * * * *
(4,1)(4,2)(4,3)(4,4)(4,5)
*/
```

// logic approach: observe i=1 || i==row(4), j=1 || j=col(5) only place "*" ; remaining places print spaces->except the condition

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int r,c,i,j;
        r = sc.nextInt();
        c = sc.nextInt();
        for(i=1;i<=r;i++){
            for(j=1;j<=c;j++){
                if(i==1 || i==r || j==1 || j==c){
                    System.out.print("* ");
                } else{
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```

// 4 rows, 5 cols

output:

```
4 5
* * * * *
*       *
*       *
*       *
* * * * *
```

– Try all co-ordinates (Q1,Q2,Q3,Q4) pyramid shapes (with * pattern and number patterns)

3)Half pyramid (Quadrant Q1)

Output format:

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

```
*      row 1: i=1 and 1 *
* *    row 2: i=2 and 1 to 2 * (i.e,1 to i)
* * *  row 3: i=3 and 1 to 3 *
* * * * row 4: i=4 and 1 to 4 *
* * * * * row 5: i=5 and 1 to 5 *
```

// logic approach: rows(given); row 1: 1 '*' , row 2: 2 '*' and so on...

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int r,i,j;
        r = sc.nextInt();
        for(i=1;i<=r;i++){
            for(j=1;j<=i;j++){
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

// 5 rows

output:

```
5
*
* *
* * *
* * * *
* * * * *
```

4) Inverted half pyramid (Quadrant Q4)

Output format:

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

// approach logic

```
* * * * *   i=1 -> 1st row: print r no of '*'
* * * *     i=2 -> 2nd row: decrement 1 star each time ..
* * *
* *
*
```

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int r,i,j;
        r = sc.nextInt();
        for(i=r;i>=1;i--){
            for(j=1;j<=i;j++){
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

// 5 rows

output:

```
5
* * * * *
* * * *
* * *
* *
*
```

5) Inverted and rotated half pyramid (simply, left side half pyramid) -> (Quadrant Q2)

Output format:

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

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int r,i,j,k;
        r = sc.nextInt();
        for(i=1;i<=r;i++){
            for(j=r-1;j>=i;j--){ // r-1 spaces ...
                System.out.print(" ");
            }
            for(k=1;k<=i;k++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

// 5 rows

output:

```
5
 *
 **
***
****
*****
```

6)Inverted triangle (Quadrant Q3)

Output format:

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

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int r,i,j,k;
        r = sc.nextInt();
        for(i=r;i>=1;i--){
            for(j=r;j>=i;j--){
                System.out.print(" ");
            }
            for(k=1;k<=i;k++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

output:

```
5
*****
****
***
**
*
```

7)Pyramid pattern (hint Q2)

output format:

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

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int r,i,j,k;
        r = sc.nextInt();
        for(i=1;i<=r;i++){
            for(j=r-1;j>=i;j--){
                System.out.print(" ");
            }
            for(k=1;k<=i;k++){
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

output:

```
5
  *
 * *
* * *
* * * *
* * * * *
```

7)Reverse Pyramid pattern (hint Q3)

output format:

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

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int r,i,j,k;
        r = sc.nextInt();
        for(i=r;i>=1;i--){
            for(j=r-1;j>=i;j--){
                System.out.print(" ");
            }
            for(k=1;k<=i;k++){
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

output:

```
5
* * * * *
* * * *
* * *
* *
*
```


8)Floyd's triangle

output format:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int i,j,num = 1,n;
        n = sc.nextInt();
        for(i=1;i<=n;i++){
            for(j=1;j<=i;j++){
                System.out.print(num+" ");
                num++;
            }
            System.out.println();
        }
    }
}
```

output:

```
5
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

9) 0-1 Triangle

output format:

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

1	(1,1)	-->(i+j)=(1+1)=2
0 1	(2,1) (2,2)	-->(i+j)=(2+1)=3 (2+2)=4
1 0 1	(3,1) (3,2) (3,3)	-->(i+j)=(3+1)=4 (3+2)=5 (3+3)=6
0 1 0 1	(4,1) (4,2) (4,3) (4,4)	-->(i+j)=(4+1)=5 (4+2)=6 (4+3)=7 (4+4)=8
1 0 1 0 1	(5,1) (5,2) (5,3) (5,4) (5,5)	-->(i+j)=(5+1)=6 (5+2)=7 (5+3)=8 (5+4)=9 (5+5)=10

// logic approach: observe sum of (i+j)=2,4,6.. even nums position placed "1"; odd:"0"

// (i+j)%2==0 ? "1" : "0"

code:

```
import java.util.*;
public class Patterns{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int n,i,j;
        n = sc.nextInt();
        for(i=1;i<=n;i++){
            for(j=1;j<=i;j++){
                if((i+j)%2==0){
                    System.out.print("1 ");
                } else{
                    System.out.print("0 ");
                }
            }
            System.out.println();
        }
    }
}
```

output:

```
5 // rows
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

10)Pascal Triangle

output format:

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
```

// apply formula -> $(nCr) = (n!) / (n-r)! * (r!)$

/*

0C0

1C0 1C1

2C0 2C1 2C2

3C0 3C1 3C2 3C3 (and so on.. rows)

*/

code:

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

```
public class Main{
```

```
    // factorial of a number
```

```
    public static int findFactorial(int num){
```

```
        if(num==0 || num==1) return 1;
```

```
        return num*findFactorial(num-1);
```

```
    }
```

```
    // apply formula  $(nCr) = (n!) / (n-r)! * (r!)$ 
```

```
    public static int applyFormula(int n,int r){
```

```
        return findFactorial(n)/(findFactorial(n-r)*findFactorial(r));
```

```
    }
```

```
    public static void main(String[] args){
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int rows,i,j;
```

```
        rows = sc.nextInt();
```

```
        // Number of rows
```

```
        for(i=0;i<rows;i++){
```

```
            // print (rows-i) spaces
```

```
            for(j=0;j<(rows-i);j++){
```

```
                System.out.print(" ");
```

```
            }
```

```
            // print pascal triangle nums
```

```
            for(j=0;j<=i;j++){
```

```
                System.out.print(applyFormula(i,j)+" ");
```

```
            }
```

```
            System.out.println();
```

```
        }
```

```
    }
```

```
}
```

output:

6 // number of rows

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```