

# How to Print Pattern in Java

**Java pattern program** enhances the coding skill, logic, and looping concepts. It is mostly asked in **Java interview** to check the logic and thinking of the programmer. We can print a **Java pattern program** in different designs. To learn the pattern program, we must have a deep knowledge of the Java loop, such as **for** loop **do-while** loop. In this section, we will learn **how to print a pattern in Java**.

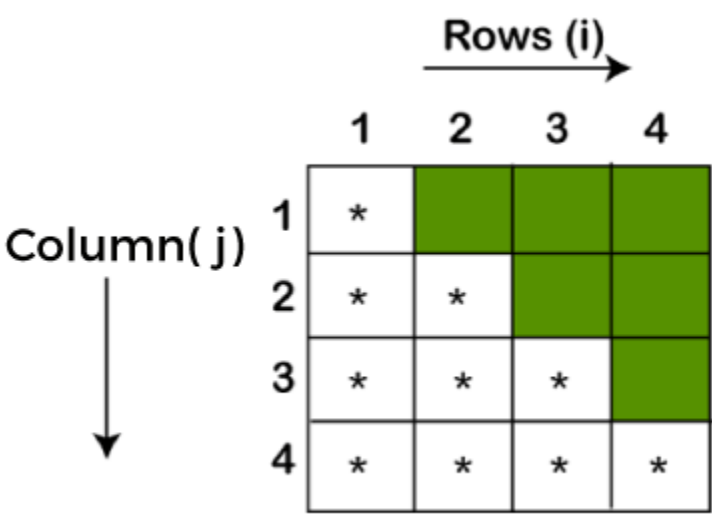
We have classified the **Java pattern program** into three categories:

- **Start Pattern**
- **Number Pattern**
- **Character Pattern**

Before moving to the pattern programs, let's see the approach.

Whenever you design logic for a pattern program, first draw that pattern in the blocks, as we have shown in the following image. The figure presents a clear look of the pattern.

Each pattern program has two or more than two loops. The number of the loop depends on the complexity of pattern or logic. The first for loop works for the row and the second loop works for the column. In the pattern programs, **Java for loop** is widely used.



In the above pattern, the **row** is denoted by **i** and the **column** is denoted by **j**. We see that the first row prints only a star. The second-row prints two stars, and so on. The **colored** blocks print the **spaces**.

Let's create the logic for the pattern, give above. In the following code snippet, we are starting row and column value from 0. We can also start it from 1, it's your choice.

1. **for(int i=0; i<row; i++)**
2. {
3. **for(int j=0; j<=i; j++)**
4. {
5. `System.out.print(" * ");`
6. }
7. `System.out.println();`

In the above code snippet, the first for loop is for row and the second for loop for columns.

Let's see the execution of the code step by step, for **n=4** (the number of rows we want to print).

## Iteration 1:

```
For i=0, 0<4 (true)
For j=0, j<=0 (true)
```

The first **print** statement prints a star at the first row and the second **println** statement prints the spaces and throws the cursor at the next line.

1. \*

Now the value of i and j is increased to 1.

## Iteration 2:

```
For i=1, 1<4 (true)
For j=1, 1<=1 (true)
```

The first **print** statement prints two stars at the second row and the second **println** statement prints the spaces and throws the cursor at the next line.

1. \*
2. \*\*

Now the value of i and j is increased to 2.

### Iteration 3:

```
For i=2, 2<4 (true)
For j=2, 2<=2 (true)
```

The first **print** statement prints three stars at the third row and the second **println** statement prints the spaces and throws the cursor at the next line.

1. \*
2. \*\*
3. \*\*\*

Now the value of i and j is increased to 3.

### Iteration 4:

```
For i=3, 3<4 (true)
For j=3, 3<=3 (true)
```

The first **print** statement prints four stars at the fourth row and the second **println** statement prints the spaces and throws the cursor at the next line.

1. \*
2. \*\*
3. \*\*\*
4. \*\*\*\*

Now the value of i and j is increased to 4.

```
For i=4, 4<4 (false)
```

The execution of the program will terminate when the value of i will be equal to the number of rows.

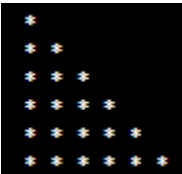
## Star Pattern

### 1. Right Triangle Star Pattern

1. **public class** RightTrianglePattern
2. {
3. **public static void** main(String args[])
4. {
5. //i for rows and j for columns
6. //row denotes the number of rows you want to print
7. **int** i, j, row=6;
8. //outer loop for rows
9. **for**(i=0; i<row; i++)
10. {
11. //inner loop for columns
12. **for**(j=0; j<=i; j++)
13. {
14. //prints stars
15. System.out.print("\* ");
16. }
17. //throws the cursor in a new line after printing each line
18. System.out.println();

```
19.}
20.}
21.}
```

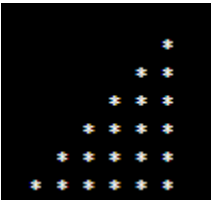
Output:



2. Left Triangle Star Pattern

```
1. public class LeftTrianglePattern
2. {
3.     public static void main(String args[])
4.     {
5.         //i for rows and j for columns
6.         //row denotes the number of rows you want to print
7.         int i, j, row = 6;
8.         //Outer loop work for rows
9.         for (i=0; i<row; i++)
10.        {
11.            //inner loop work for space
12.            for (j=2*(row-i); j>=0; j--)
13.            {
14.                //prints space between two stars
15.                System.out.print(" ");
16.            }
17.            //inner loop for columns
18.            for (j=0; j<=i; j++ )
19.            {
20.                //prints star
21.                System.out.print("* ");
22.            }
23.            //throws the cursor in a new line after printing each line
24.            System.out.println();
25.        }
26.    }
27. }
```

Output:



3. Pyramid Star Pattern

```
1. public class PyramidPattern
2. {
3.     public static void main(String args[])
4.     {
5.         //i for rows and j for columns
6.         //row denotes the number of rows you want to print
7.         int i, j, row = 6;
8.         //Outer loop work for rows
9.         for (i=0; i<row; i++)
10.        {
```

```

11. //inner loop work for space
12. for (j=row-i; j>1; j--)
13. {
14. //prints space between two stars
15. System.out.print(" ");
16. }
17. //inner loop for columns
18. for (j=0; j<=i; j++ )
19. {
20. //prints star
21. System.out.print("* ");
22. }
23. //throws the cursor in a new line after printing each line
24. System.out.println();
25. }
26. }
27. }

```

**Output:**



#### 4. Diamond Shape Pattern

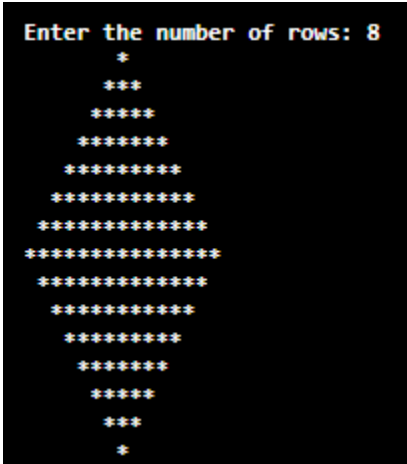
```

1. import java.util.Scanner;
2. public class DiamondPattern
3. {
4. public static void main(String args[])
5. {
6. int row, i, j, space = 1;
7. System.out.print("Enter the number of rows you want to print: ");
8. Scanner sc = new Scanner(System.in);
9. row = sc.nextInt();
10. space = row - 1;
11. for (j = 1; j<= row; j++)
12. {
13. for (i = 1; i<= space; i++)
14. {
15. System.out.print(" ");
16. }
17. space--;
18. for (i = 1; i <= 2 * j - 1; i++)
19. {
20. System.out.print("*");
21. }
22. System.out.println("");
23. }
24. space = 1;
25. for (j = 1; j<= row - 1; j++)
26. {
27. for (i = 1; i<= space; i++)
28. {
29. System.out.print(" ");
30. }
31. space++;

```

```
32. for (i = 1; i<= 2 * (row - j) - 1; i++)
33. {
34. System.out.print("*");
35. }
36. System.out.println("");
37. }
38. }
39. }
```

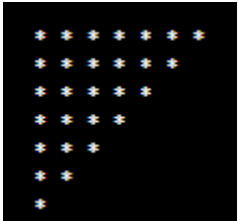
Output:



5. Downward Triangle Star Pattern

```
1. public class DownwardTrianglePattern
2. {
3.     public static void main(String[] args)
4.     {
5.         int rows=7;
6.         //inner loop
7.         for (int i= rows-1; i>=0 ; i--)
8.         {
9.             //outer loop
10.            for (int j=0; j<=i; j++)
11.            {
12.                //prints star and space
13.                System.out.print("*" + " ");
14.            }
15.            //throws the cursor in the next line after printing each line
16.            System.out.println();
17.        }
18.    }
19. }
```

Output:



6. Mirrored Right Triangle Star Pattern

```
1. public class MirroredRightTrianglePattern
2. {
3.     public static void main(String[] args)
4.     {
5.         int n=7;
6.         //inner loop
7.         for (int i= 0; i<= n; i++)
```

```
8. {
9. //outer loop
10. for (int j=1; j<=n-i; j++)
11. {
12. System.out.print(" ");
13. }
14. for (int k=0; k<=i; k++)
15. {
16. System.out.print("*");
17. }
18. System.out.println("");
19. }
20. }
21. }
```

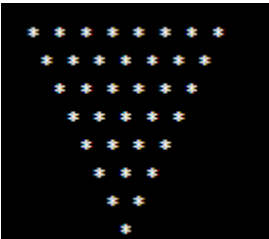
Output:



7. Reverse Pyramid Star Pattern

```
1. public class ReversePyramidPattern
2. {
3. public static void main(String[] args)
4. {
5. int rows=8;
6. for (int i= 0; i<= rows-1; i++)
7. {
8. for (int j=0; j<=i; j++)
9. {
10. System.out.print(" ");
11. }
12. for (int k=0; k<=rows-1-i; k++)
13. {
14. System.out.print("*" + " ");
15. }
16. System.out.println();
17. }
18. }
19. }
```

Output:

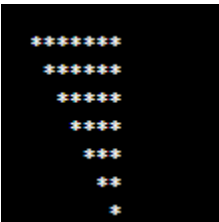


8. Right Down Mirror Star Pattern

```
1. public class RightDownMirrorPattern
2. {
3. public static void main(String args[])
4. {
5. int row=7;
```

```
6. for (int i= row; i>= 1; i--)
7. {
8. for (int j=row; j>i;j--)
9. {
10. System.out.print(" ");
11.}
12. for (int k=1;k<=i;k++)
13. {
14. System.out.print("*");
15.}
16. System.out.println("");
17.}
18.}
19.}
```

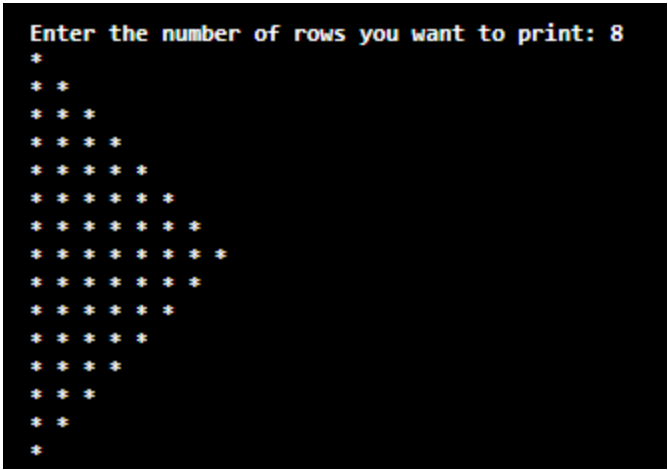
Output:



9. Right Pascal's Triangle

```
1. import java.util.Scanner;
2. public class RightPascalTrianglePattern
3. {
4. public static void main(String[] args)
5. {
6. int i, j, rows;
7. Scanner sc = new Scanner(System.in);
8. System.out.print("Enter the number of rows you want to print: ");
9. rows = sc.nextInt();
10. for (i= 0; i<= rows-1; i++)
11. {
12. for (j=0; j<=i; j++)
13. {
14. System.out.print("*" + " ");
15.}
16. System.out.println("");
17.}
18. for (i=rows-1; i>=0; i--)
19. {
20. for(j=0; j <= i-1;j++)
21. {
22. System.out.print("*" + " ");
23.}
24. System.out.println("");
25.}
26.}
27.}
```

Output:

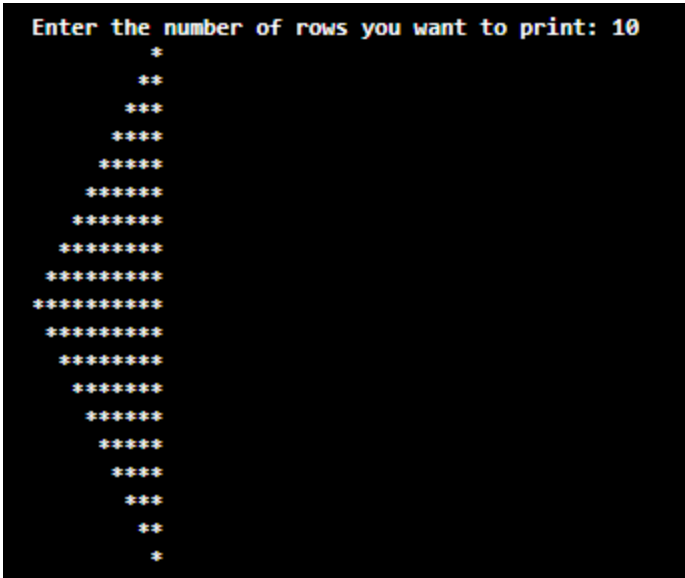


10. Left Pascal's Triangle

```
1. import java.util.Scanner;
2. public class LeftPascalTrianglePattern
3. {
4.     public static void main(String[] args)
5.     {
6.         int i, j, k, rows;
7.         Scanner sc = new Scanner(System.in);
8.         System.out.print("Enter the number of rows you want to print: ");
9.         rows = sc.nextInt();
10.        for (i= 1; i<= rows ; i++)
11.        {
12.            for (j=i; j <rows ;j++)
13.            {
14.                System.out.print(" ");
15.            }
16.            for (k=1; k<=i;k++)
17.            {
18.                System.out.print("*");
19.            }
20.            System.out.println("");
21.        }
22.        for (i=rows; i>= 1; i--)
23.        {
24.            for(j=i; j<=rows;j++)
25.            {
26.                System.out.print(" ");
27.            }
28.            for(k=1; k<i ;k++)
29.            {
30.                System.out.print("*");
31.            }
32.            System.out.println("");
33.        }
34.        sc.close();
35.    }
36. }
```

Output:

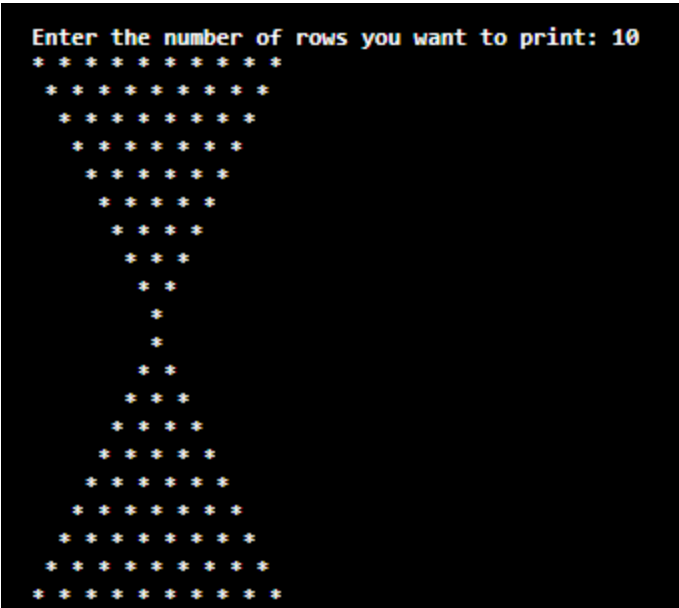




11. Sandglass Star Pattern

```
1. import java.util.Scanner;
2. public class SandglassPattern
3. {
4.     public static void main(String[] args)
5.     {
6.         int i, j, k, n;
7.         Scanner sc = new Scanner(System.in);
8.         System.out.print("Enter the number of rows you want to print: ");
9.         n = sc.nextInt();
10.        for (i= 0; i<= n-1 ; i++)
11.        {
12.            for (j=0; j<i; j++)
13.            {
14.                System.out.print(" ");
15.            }
16.            for (k=i; k<=n-1; k++)
17.            {
18.                System.out.print("*" + " ");
19.            }
20.            System.out.println("");
21.        }
22.        for (i= n-1; i>= 0; i--)
23.        {
24.            for (j=0; j<i; j++)
25.            {
26.                System.out.print(" ");
27.            }
28.            for (k=i; k<=n-1; k++)
29.            {
30.                System.out.print("*" + " ");
31.            }
32.            System.out.println("");
33.        }
34.        sc.close();
35.    }
36. }
```

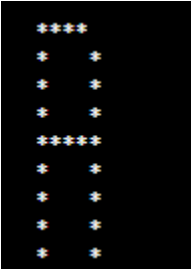
Output:



12. Alphabet Star Pattern

```
1. import java.util.*;
2. public class AlphabetPattern
3. {
4.     public static void main(String[] args)
5.     {
6.         int i, j, n=8;
7.         // Outer for loop for number of lines
8.         for (i = 0; i<=n; i++)
9.         {
10.            // Inner for loop for logic execution
11.            for (j = 0; j<= n / 2; j++)
12.            {
13.                // prints two vertical lines
14.                if ((j == 0 || j == n / 2) && i != 0 ||
15.                // print first line of alphabet
16.                i == 0 && j != n / 2 ||
17.                // prints middle line
18.                i == n / 2)
19.                    System.out.print("*");
20.                else
21.                    System.out.print(" ");
22.            }
23.            System.out.println();
24.        }
25.    }
26. }
```

Output:

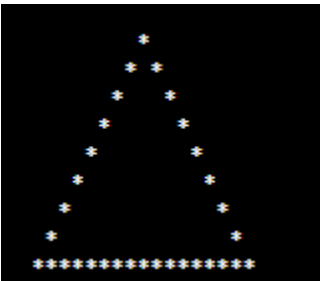


13. Triangle Star Pattern

```
1. import java.util.Scanner;
2. public class TrianglePattern
3. {
4.     public static void main(String[] args)
5.     {
6.         int i, j, k, rows=9;
7.         for (i=1; i<= rows ; i++)
```

```
8. {
9.  for (j = i; j < rows ; j++)
10. {
11. System.out.print(" ");
12. }
13. for (k = 1; k <= (2*i -1) ;k++)
14. {
15. if(k==1 || i == rows || k==(2*i-1))
16. {
17. System.out.print("*");
18. }
19. else
20. {
21. System.out.print(" ");
22. }
23. }
24. System.out.println("");
25. }
26. }
27. }
```

Output:



14. Down Triangle Pattern

```
1. import java.util.Scanner;
2. public class DownTrianglePattern
3. {
4.  public static void main(String[] args)
5.  {
6.  int i, j, k, rows=9;
7.  for (i=rows; i>= 1 ; i--)
8.  {
9.  for (j = i; j<rows ; j++)
10. {
11. System.out.print(" ");
12. }
13. for (k = 1; k <= (2*i -1) ;k++)
14. {
15. if( k==1 || i == rows || k==(2*i-1))
16. {
17. System.out.print("*");
18. }
19. else
20. {
21. System.out.print(" ");
22. }
23. }
24. System.out.println("");
25. }
26. }
27. }
```

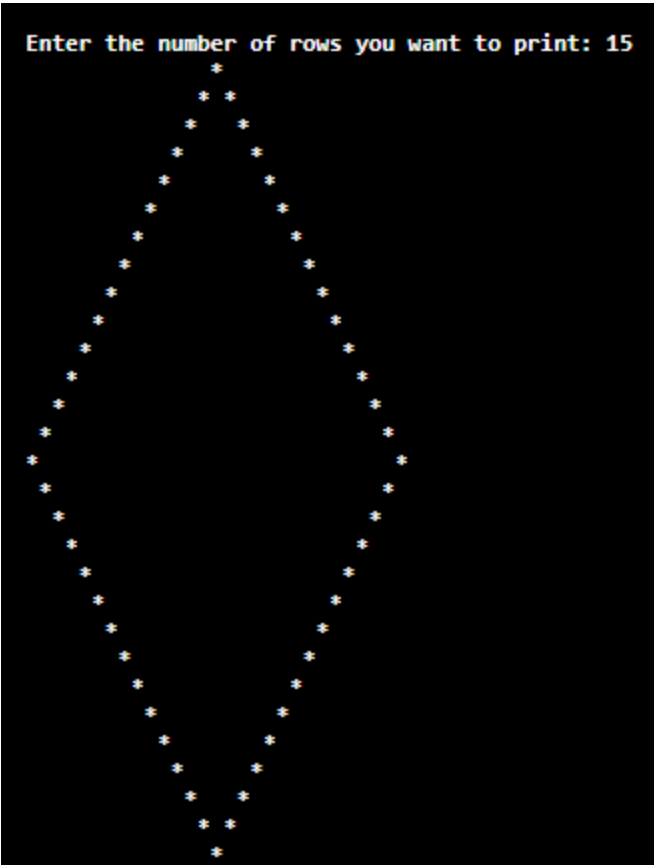
Output:



### 15. Diamond Star Pattern

```
1. import java.util.*;
2. public class DiamondPattern
3. {
4.     public static void main(String[] args)
5.     {
6.         Scanner sc = new Scanner(System.in);
7.         System.out.println("Enter the number of rows you want to print: ");
8.         int rows = sc.nextInt();
9.         for (i=1; i<= rows ; i++)
10. {
11.     for (j = rows; j > i ; j--)
12. {
13.         System.out.print(" ");
14.     }
15.     System.out.print("*");
16.     for (k = 1; k < 2*(i - 1) ;k++)
17. {
18.         System.out.print(" ");
19.     }
20.     if( i==1)
21. {
22.         System.out.println("");
23.     }
24.     else
25. {
26.         System.out.println("*");
27.     }
28. }
29. for (i=rows-1; i>= 1 ; i--)
30. {
31.     for (int j = rows; j > i ; j--)
32. {
33.         System.out.print(" ");
34.     }
35.     System.out.print("*");
36.     for (int k = 1; k < 2*(i - 1) ;k++)
37. {
38.         System.out.print(" ");
39.     }
40.     if(i==1)
41.         System.out.println("");
42.     else
43.         System.out.println("*");
44.     }
45. }
46. }
```

Output:

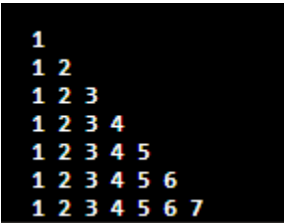


## Number Pattern

### 1. Pattern-1

```
1. public class Pattern1
2. {
3.     public static void main(String args[])
4.     {
5.         int i,j,number, n=7;
6.         //loop for rows
7.         for(i=0; i<n; i++)
8.         {
9.             number=1;
10.        //loop for columns
11.        for(j=0; j<=i; j++)
12.        {
13.            //prints num
14.            System.out.print(number+ " ");
15.            //incrementing the value of number
16.            number++;
17.        }
18.        //throws the cursor at the next line after printing each row
19.        System.out.println();
20.    }
21. }
22. }
```

Output:

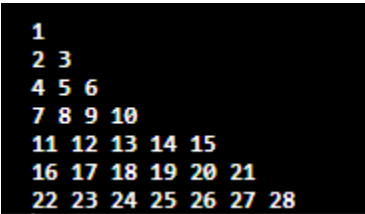


### 2. Pattern-2

```
1. public class Pattern2
2. {
3.     public static void main(String[] args)
4.     {
```

```
5. int i, j, k = 1;
6. //inner loop
7. for (i = 1; i <= 7; i++)
8. {
9. //outer loop
10. for (j = 1; j < i + 1; j++)
11. {
12. //prints the value of k
13. System.out.print(k++ + " ");
14. }
15. //throws the cursor at the next line
16. System.out.println();
17. }
18. }
19. }
```

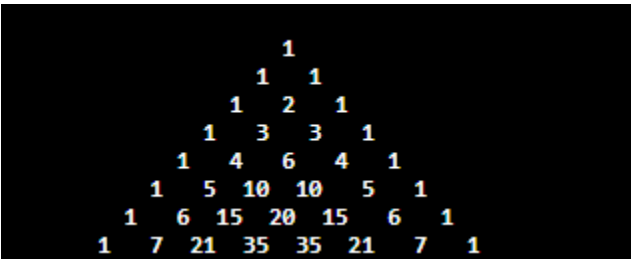
Output:



3. Pattern-3

```
1. public class Pattern3
2. {
3. public static void main(String[] args)
4. {
5. int n = 8; //n is the number of rows you want to print
6. for (int i = 0; i < n; i++)
7. {
8. int number = 1;
9. System.out.printf("%" + (n - i) * 2 + "s", "");
10. for (int j = 0; j <= i; j++)
11. {
12. System.out.printf("%4d", number);
13. number = number * (i - j) / (j + 1);
14. }
15. System.out.println();
16. }
17. }
18. }
```

Output:

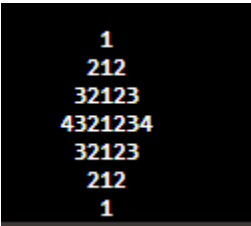


4. Pattern-4

```
1. public class Pattern4
2. {
3. public static void main(String[] args)
4. {
5. for (int i = 1; i <= 4; i++)
```

```
6. {
7.  int n = 8;
8.  for (int j = 1; j <= n - i; j++)
9.  {
10. System.out.print(" ");
11. }
12. for (int k = i; k >= 1; k--)
13. {
14. System.out.print(k);
15. }
16. for (int l = 2; l <= i; l++)
17. {
18. System.out.print(l);
19. }
20. System.out.println();
21. }
22. for (int i = 3; i >= 1; i--)
23. {
24. int n = 10;
25. for (int j = 0; j <= n - i; j++)
26. {
27. System.out.print(" ");
28. }
29. for (int k = i; k >= 1; k--)
30. {
31. System.out.print(k);
32. }
33. for (int l = 2; l <= i; l++)
34. {
35. System.out.print(l);
36. }
37. System.out.println();
38. }
39. }
40. }
```

Output:

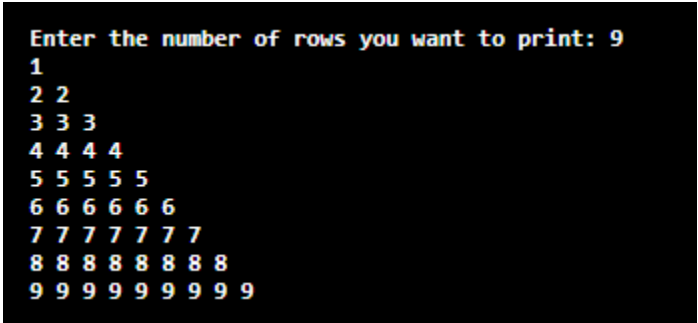


5. Pattern-5

```
1. import java.util.*;
2. public class Pattern5
3. {
4.  public static void main(String[] args)
5.  {
6.  int i, j, rows;
7.  Scanner sc = new Scanner(System.in);
8.  System.out.print("Enter the number of rows you want to print: ");
9.  rows = sc.nextInt();
10. for (i = 1; i <= rows; i++)
11. {
12.  for (j = 1; j <= i; j++)
13. {
```

```
14. System.out.print(i+ " ");
15. }
16. System.out.println();
17. }
18. }
19. }
```

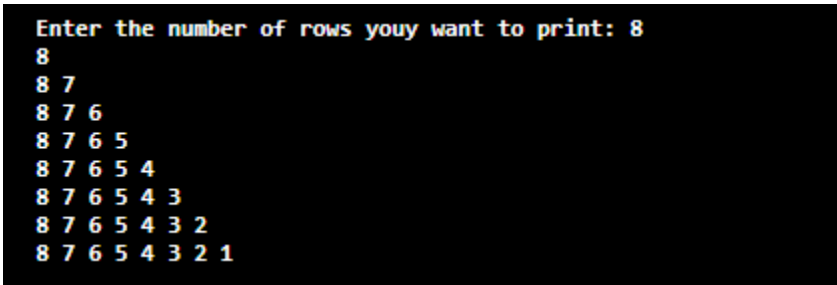
Output:



6. Pattern-6

```
1. import java.util.*;
2. public class Pattern6
3. {
4.     public static void main(String[] args)
5.     {
6.         int i, j, rows;
7.         Scanner sc = new Scanner(System.in);
8.         System.out.print("Enter the number of rows youy want to print: ");
9.         rows = sc.nextInt();
10.        for (i = rows; i >= 1; i--)
11.        {
12.            for (j = rows; j >= i; j--)
13.            {
14.                System.out.print(j+ " ");
15.            }
16.
17.            System.out.println();
18.        }
19.    }
20. }
```

Output:



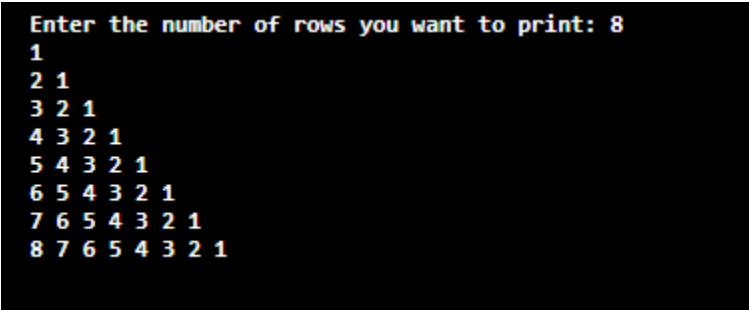
7. Pattern-7

```
1. import java.util.Scanner;
2. public class Pattern7
3. {
4.     public static void main(String[] args)
5.     {
6.         int i, j, n;
7.         Scanner sc = new Scanner(System.in);
8.         System.out.print("Enter the number of rows you want to print: ");
9.         n = sc.nextInt();
10.        for (i = 1; i <= n; i++)
```



```
11. {
12. for (j = i; j >= 1; j--)
13. {
14. System.out.print(j+ " ");
15. }
16. System.out.println();
17. }
18. }
19. }
```

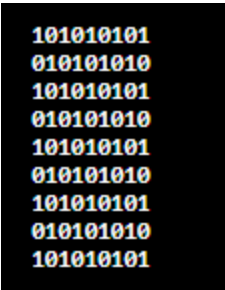
Output:



8. Pattern-8

```
1. public class Pattern8
2. {
3. public static void main(String[] args)
4. {
5. int rows=9; //number of rows to print
6. for (int i = 1; i <= rows; i++)
7. {
8. int num;
9. if(i%2 == 0)
10. {
11. num = 0;
12. for (int j = 1; j <= rows; j++)
13. {
14. System.out.print(num);
15. num = (num == 0)? 1 : 0;
16. }
17. }
18. else
19. {
20. num = 1;
21. for (int j = 1; j <= rows; j++)
22. {
23. System.out.print(num);
24. num = (num == 0)? 1 : 0;
25. }
26. }
27. System.out.println();
28. }
29. }
30. }
```

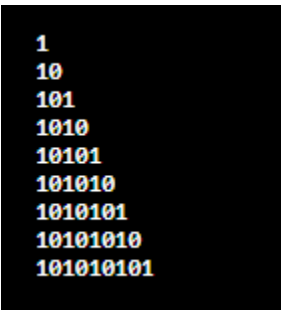
Output:



9. Pattern-9

```
1. import java.util.Scanner;
2. public class Pattern9
3. {
4.     public static void main(String[] args)
5.     {
6.         int i, j, rows=9;
7.         for (i = 1; i <= rows; i++)
8.         {
9.             for (j = 1; j <= i; j++)
10.            {
11.                if(j%2 == 0)
12.                {
13.                    System.out.print(0);
14.                }
15.                else
16.                {
17.                    System.out.print(1);
18.                }
19.            }
20.            System.out.println();
21.        }
22.    }
23. }
```

Output:



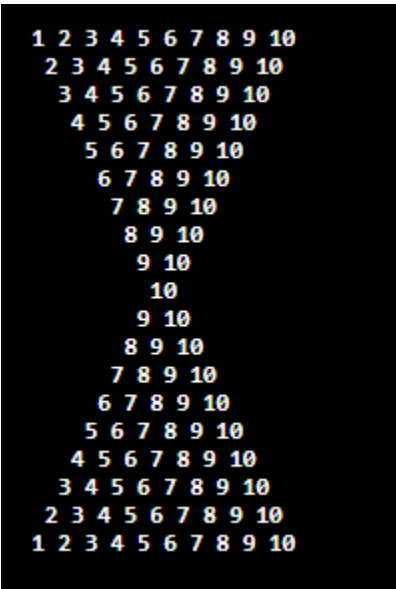
10. Pattern-10

```
1. public class Pattern10
2. {
3.     public static void main(String[] args)
4.     {
5.         int n = 10;
6.         for (int i = 1; i <= n; i++)
7.         {
8.             for (int j = 1; j < i; j++)
9.             {
10.                System.out.print(" ");
11.            }
12.            for (int k = i; k <= n; k++)
13.            {
14.                System.out.print(k+" ");
15.            }

```

```
16. System.out.println();
17. }
18. for (int i = n-1; i >= 1; i--)
19. {
20.     for (int j = 1; j < i; j++)
21.     {
22.         System.out.print(" ");
23.     }
24.     for (int k = i; k <= n; k++)
25.     {
26.         System.out.print(k+" ");
27.     }
28. System.out.println();
29. }
30. }
31. }
```

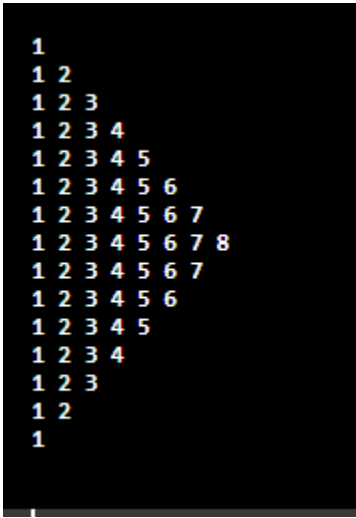
Output:



11. Pattern-11

```
1. public class Pattern11
2. {
3.     public static void main(String[] args)
4.     {
5.         int rows=8;
6.         //Prints upper half pattern
7.         for (int i = 1; i <= rows; i++)
8.         {
9.             for (int j = 1; j <= i; j++)
10.            {
11.                System.out.print(j+" ");
12.            }
13.            System.out.println();
14.        }
15.        //prints lower half pattern
16.        for (int i = rows-1; i >= 1; i--)
17.        {
18.            for (int j = 1; j <= i; j++)
19.            {
20.                System.out.print(j+" ");
21.            }
22.            System.out.println();
23.        }
24.    }
25. }
```

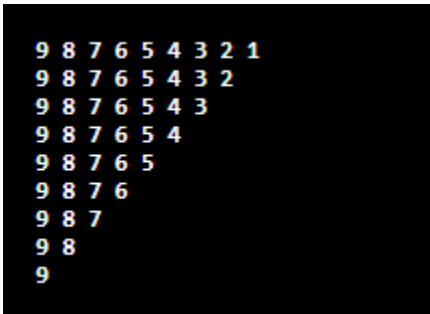
Output:



12. Pattern-12

- 1. `public class` Pattern12
- 2. {
- 3. `public static void` main(String[] args)
- 4. {
- 5. `int` rows=9;
- 6. `for` (`int` i = 1; i <= rows; i++)
- 7. {
- 8. `for` (`int` j = rows; j >= i; j--)
- 9. {
- 10. System.out.print(j+ " ");
- 11. }
- 12. System.out.println();
- 13. }
- 14. }
- 15. }

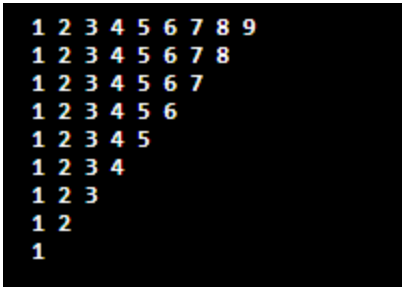
Output:



13. Pattern-13

- 1. `public class` Pattern14
- 2. {
- 3. `public static void` main(String[] args)
- 4. {
- 5. `int` i, j, rows=9;
- 6. `for` (i = rows; i >= 1; i--)
- 7. {
- 8. `for` (j = 1; j <= i; j++)
- 9. {
- 10. System.out.print(j+ " ");
- 11. }
- 12. System.out.println();
- 13. }
- 14. }
- 15. }

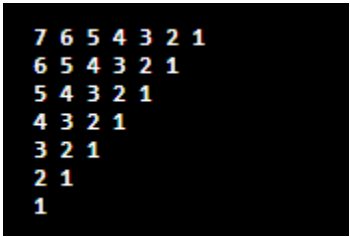
Output:



14. Pattern-14

- 1. `public class` Pattern14
- 2. {
- 3. `public static void` main(String[] args)
- 4. {
- 5. `int` rows=7, i, j;
- 6. `for` (i = rows; i >= 1; i--)
- 7. {
- 8. `for` (j = i; j >= 1; j--)
- 9. {
- 10. System.out.print(j+ " ");
- 11. }
- 12. System.out.println();
- 13. }
- 14. }
- 15. }

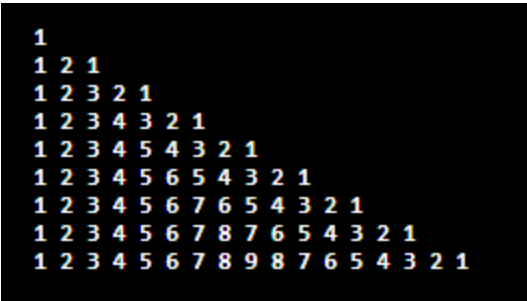
Output:



15. Pattern-15

- 1. `public class` Pattern15
- 2. {
- 3. `public static void` main(String[] args)
- 4. {
- 5. `int` i, j, rows=9;
- 6. `for` (i = 1; i <= rows; i++)
- 7. {
- 8. `//Prints first half of the row`
- 9. `for` (j = 1; j <= i; j++)
- 10. {
- 11. System.out.print(j+ " ");
- 12. }
- 13. `//Prints second half of the row`
- 14. `for` (j = i-1; j >= 1; j--)
- 15. {
- 16. System.out.print(j+ " ");
- 17. }
- 18. System.out.println();
- 19. }
- 20. }
- 21. }

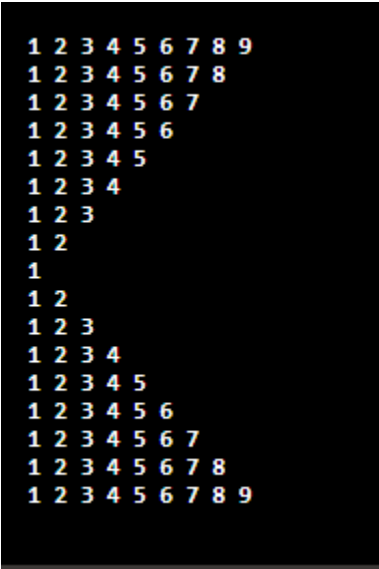
Output:



16. Pattern-16

```
1. public class Pattern16
2. {
3.     public static void main(String[] args)
4.     {
5.         int i, j, rows=9;
6.         //Prints upper half pattern
7.         for (i = rows; i >= 1; i--)
8.         {
9.             for (j = 1; j <= i; j++)
10.            {
11.                System.out.print(j+ " ");
12.            }
13.            System.out.println();
14.        }
15.        //Prints lower half pattern
16.        for (i = 2; i <= rows; i++)
17.        {
18.            for (j = 1; j <= i; j++)
19.            {
20.                System.out.print(j+ " ");
21.            }
22.            System.out.println();
23.        }
24.    }
25. }
```

Output:

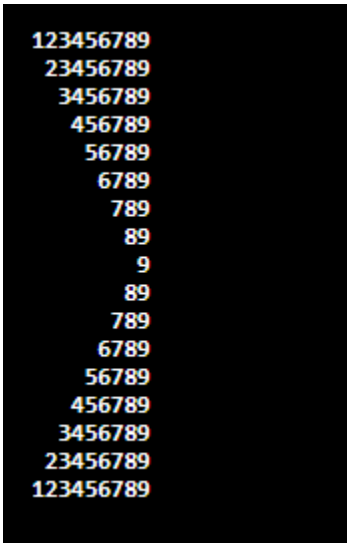


17. Pattern-17

```
1. public class Pattern17
2. {
3.     public static void main(String[] args)
4.     {
5.         int rows=9;
6.         //Prints upper half pattern
7.         for (int i = 1; i <= rows; i++)
8.         {
```

```
9. //Prints i spaces at the beginning of each row
10. for (int j = 1; j < i; j++)
11. {
12. System.out.print(" ");
13. }
14. //Prints i to rows value at the end of each row
15. for (int j = i; j <= rows; j++)
16. {
17. System.out.print(j);
18. }
19. System.out.println();
20. }
21. //Prints lower half pattern
22. for (int i = rows-1; i >= 1; i--)
23. {
24. //Prints i spaces at the beginning of each row
25. for (int j = 1; j < i; j++)
26. {
27. System.out.print(" ");
28. }
29. //Prints i to rows value at the end of each row
30. for (int j = i; j <= rows; j++)
31. {
32. System.out.print(j);
33. }
34. System.out.println();
35. }
36. }
37. }
```

Output:

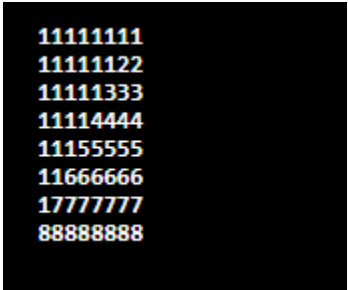


18. Pattern-18

```
1. public class Pattern18
2. {
3. public static void main(String[] args)
4. {
5. int rows=8;
6. for (int i = 1; i <= rows; i++)
7. {
8. for (int j = 1; j <= rows-i; j++)
9. {
10. System.out.print(1);
11. }
12. for (int j = 1; j <= i; j++)
13. {
```

```
14. System.out.print(i);
15. }
16. System.out.println();
17. }
18. }
19. }
```

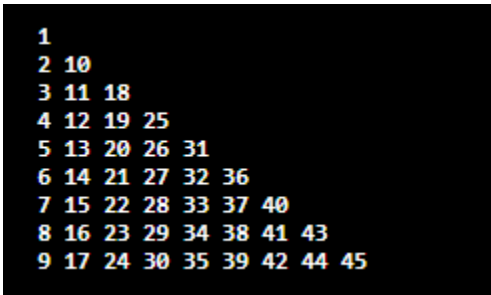
Output:



19. Pattern-19

```
1. public class Pattern19
2. {
3.     public static void main(String args[])
4.     {
5.         int rows=9;
6.         for (int i = 1; i <= rows; i++)
7.         {
8.             int num = i;
9.             for (int j = 1; j <= i; j++)
10.            {
11.                System.out.print(num+" ");
12.                num = num+rows-j;
13.            }
14.            System.out.println();
15.        }
16.    }
17. }
```

Output:



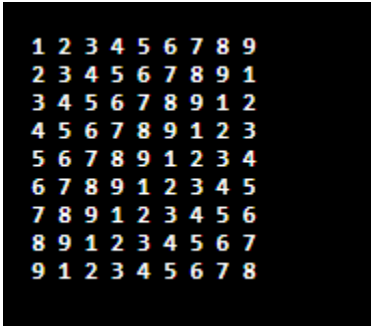
20. Pattern-20

```
1. public class Pattern20
2. {
3.     public static void main(String[] args)
4.     {
5.         int i, j, k, rows=9;
6.         for(i=1;i< rows+1 ;i++)
7.         {
8.             for(j=i; j < rows+1 ;j++)
9.             {
10.                System.out.print(j + " ");
11.            }
12.            for(k=1; k < i ;k++)
13.            {
```



```
14. System.out.print(k + " ");
15. }
16. System.out.println();
17. }
18. }
19. }
```

Output:

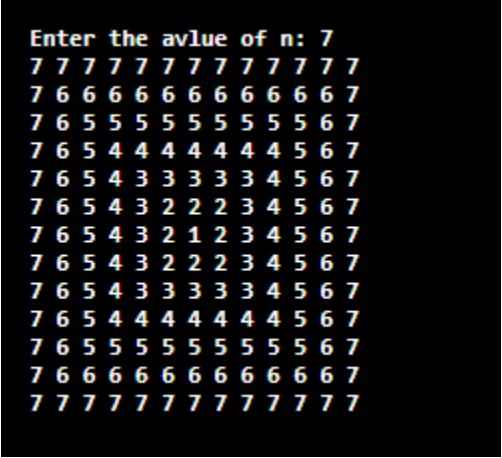


21. Pattern-21

```
1. import java.util.Scanner;
2. public class Pattern21
3. {
4.     public static void main(String[] args)
5.     {
6.         int i, j, min, n; //n is the number up to you want to print
7.         System.out.print("Enter the value of n: ");
8.         Scanner sc= new Scanner(System.in);
9.         n=sc.nextInt();
10.        //loo for upper left part
11.        for (i = 1; i <= n; i++)
12.        {
13.            for (j = 1; j <= n; j++)
14.            {
15.                min = i < j ? i : j;
16.                System.out.print(n - min + 1 + " ");
17.            }
18.            //loop for upper right part
19.            for (j = n - 1; j >= 1; j--)
20.            {
21.                min = i < j ? i : j;
22.                System.out.print(n - min + 1 + " ");
23.            }
24.            System.out.println();
25.        }
26.        //loop for lower left part
27.        for (i = n - 1; i >= 1; i--)
28.        {
29.            for (j = 1; j <= n; j++)
30.            {
31.                min = i < j ? i : j;
32.                System.out.print(n - min + 1 + " ");
33.            }
34.            //loop for lower right part
35.            for (j = n - 1; j >= 1; j--)
36.            {
37.                min = i < j ? i : j;
38.                System.out.print(n - min + 1 + " ");
39.            }
40.            System.out.println();
}
```

41.}  
42.}  
43.}

Output:

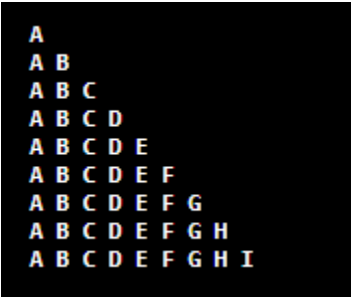


## Character Pattern

### 1. Right Triangle Alphabetic Pattern

```
1. public class RightAlphabeticPattern
2. {
3.     public static void main(String[] args)
4.     {
5.         int alphabet = 65; //ASCII value of capital A is 65
6.         //inner loop for rows
7.         for (int i = 0; i <= 8; i++)
8.         {
9.             //outer loop for columns
10.            for (int j = 0; j <= i; j++)
11.            {
12.                //adds the value of j in the ASCII value of A and prints the corresponding alphabet
13.                System.out.print((char) (alphabet + j) + " ");
14.            }
15.            System.out.println();
16.        }
17.    }
18. }
```

Output:

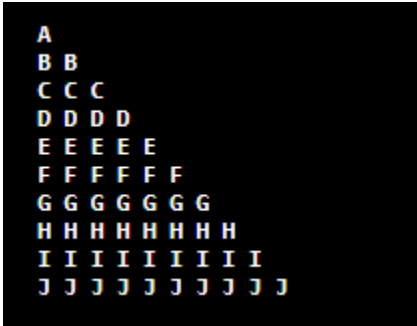


### 2. Repeating Alphabet Pattern

```
1. public class RepeatingPattern
2. {
3.     public static void main(String[] args)
4.     {
5.         int letter = 65; //ASCII value of capital A is 65
6.         //inner loop for rows
7.         for (int i = 0; i <= 9; i++)
8.         {
9.             //outer loop for columns
10.            for (int j = 0; j <= i; j++)
```

```
11. {
12. //prints the character
13. System.out.print((char) letter + " ");
14. }
15. letter++;
16. System.out.println();
17. }
18. }
19. }
```

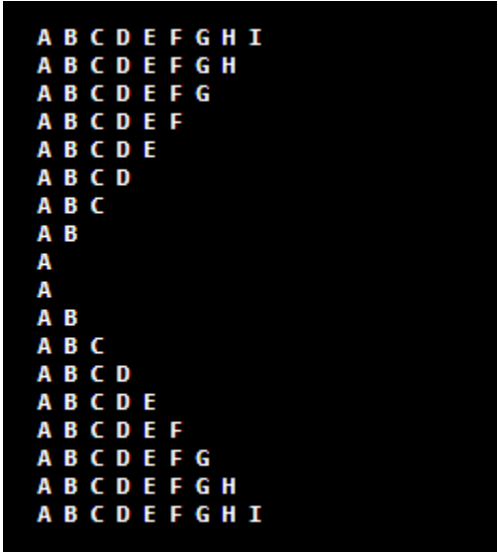
Output:



3. K-shape Alphabet Pattern

```
1. public class KshapePattern
2. {
3.     public static void main(String[] args)
4.     {
5.         for (int i = 8; i >= 0; i--)
6.         {
7.             int alphabet = 65;
8.             for (int j = 0; j <= i; j++)
9.             {
10. System.out.print((char) (alphabet + j) + " ");
11. }
12. System.out.println();
13. }
14. for (int i = 0; i <= 8; i++)
15. {
16. int alphabet = 65;
17. for (int j = 0; j <= i; j++)
18. {
19. System.out.print((char) (alphabet + j) + " ");
20. }
21. System.out.println();
22. }
23. }
24. }
```

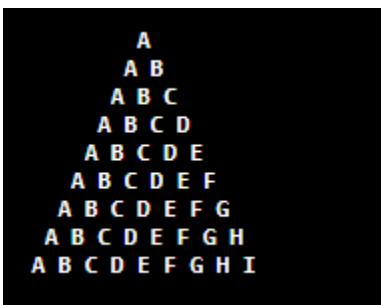
Output:



#### 4. Triangle Character Pattern

```
1. public class TriangleCharacterPattern
2. {
3.     public static void main(String[] args)
4.     {
5.         for (int i = 0; i <= 8; i++)
6.         {
7.             int alphabet = 65;
8.             for (int j = 8; j > i; j--)
9.             {
10.                System.out.print(" ");
11.            }
12.            for (int k = 0; k <= i; k++)
13.            {
14.                System.out.print((char) (alphabet + k) + " ");
15.            }
16.            System.out.println();
17.        }
18.    }
19. }
```

Output:



#### 5. Diamond Character Pattern

```
1. import java.util.Scanner;
2. public class DiamondCharacterPattern
3. {
4.     public static void main(String[] args)
5.     {
6.         char[] alphabet = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
7.         int alphabet_number = 0;
8.         String[] diamond = new String[26]; // array of strings
9.         System.out.print("Enter a Character between A to Z : ");
10.        Scanner reader = new Scanner(System.in);
11.        try
12.        {
13.            char user_alphabet = reader.next("[A-Z]").charAt(0);
14.            // search for letter number in the array letter
15.            for (int i = 0; i < alphabet.length; i++)
16.            {
17.                if (letter[i] == user_alphabet)
18.                {
19.                    alphabet_number = i;
20.                    break;
21.                }
22.            }
23.            //construct diamond
24.            for (int i = 0; i <= alphabet_number; i++)
25.            {
26.                diamond[i] = "";
```

```
27. //add initial spaces
28. for (int j = 0; j < alphabet _number - i; j++)
29. {
30. diamond[i] += " ";
31. }
32. // add alphabet
33. diamond[i] += alphabet
34. //add space between letters
35. if (alphabet[i] != 'A')
36. {
37. for (int j = 0; j < 2 * i - 1; j++)
38. {
39. diamond[i] += " ";
40. }
41. // add alphabet
42. diamond[i] += alphabet[i];
43. }
44. // Draw the first part of the diamond
45. System.out.println(diamond[i]);
46. }
47. for (int i = alphabet _number - 1; i >= 0; i--)
48. {
49. // Draw the second part of the diamond
50. // prints the diamondArray in the reverse order
51. System.out.println(diamond[i]);
52. }
53. }
54. catch (Exception e)
55. {
56. e.printStackTrace();
57. }
58. finally
59. {
60. reader.close();
61. }
62. }
63. }
```

**Output:**

Enter a Character between A to Z : P

```

      A
    B B
  C   C
 D   D
E   E
F   F
G   G
H   H
I   I
J   J
K   K
L   L
M   M
N   N
O   O
P   P
O   O
N   N
M   M
L   L
K   K
J   J
I   I
H   H
G   G
F   F
E   E
D   D
C   C
  B B
    A
```