

Pattern - 9: Diamond Star Pattern

Problem Statement: Given an integer **N**, print the following pattern :

```

    *
  ***
 *****
*****
*****
*****
  ***
    *

```

Here, $N = 5$.

Examples:

Input Format: $N = 3$

Result:

```

 *
 ***
*****
*****
 ***
  *

```

Input Format: $N = 6$

Result:

```

  *
 ***
*****
*****
*****
*****
*****
 *****
 *****
 *****
 *****
  *

```

Solution

Disclaimer: *Don't jump directly to the solution, try it out yourself first.*

[Problem Link](#)

Approach:

There are 4 general rules for solving a pattern-based question :

- We always use nested loops for printing the patterns. For the outer loop, we count the number of lines/rows and loop for them.
- Next, for the inner loop, we focus on the number of columns and somehow connect them to the rows by forming a logic such that for each row we get the required number of columns to be printed.
- We print the '*' inside the inner loop.
- Observe symmetry in the pattern or check if a pattern is a combination of two or more similar patterns or not.

This pattern is just a mixture of the last two patterns (erect pyramid and inverted pyramid). Firstly, we will print the erect pyramid and then an inverted pyramid below it.

Code:

C++Java

```
class Main {  
  
    static void erect_pyramid(int N)  
    {  
        // This is the outer loop which will loop for the rows.  
        for (int i = 0; i < N; i++)  
        {  
            // For printing the spaces before stars in each row  
            for (int j =0; j<N-i-1; j++)  
            {  
                System.out.print(" ");  
            }  
        }  
    }  
}
```

```

        // For printing the stars in each row
        for(int j=0;j< 2*i+1;j++){

            System.out.print("*");
        }

        // For printing the spaces after the stars in each row
        for (int j =0; j<N-i-1; j++)
        {
            System.out.print(" ");
        }

        // As soon as the stars for each iteration are printed, we
        move to the
        // next row and give a line break otherwise all stars
        // would get printed in 1 line.
        System.out.println();
    }
}

static void inverted_pyramid(int N)
{
    // This is the outer loop which will loop for the rows.
    for (int i = 0; i < N; i++)
    {
        // For printing the spaces before stars in each row
        for (int j =0; j<i; j++)
        {
            System.out.print(" ");
        }
    }
}

```

```

        // For printing the stars in each row
        for(int j=0;j< 2*N -(2*i +1);j++){

            System.out.print("*");

        }

        // For printing the spaces after the stars in each row
        for (int j =0; j<i; j++)
        {
            System.out.print(" ");
        }

        // As soon as the stars for each iteration are printed, we
        move to the
        // next row and give a line break otherwise all stars
        // would get printed in 1 line.
        System.out.println();
    }
}

public static void main(String[] args) {

    // Here, we have taken the value of N as 5.
    // We can also take input from the user.
    int N = 5;
    erect_pyramid(N);
    inverted_pyramid(N);
}
}

```

Output

