Pattern-14: Increasing Letter Triangle Pattern

70

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

```
A
AB
ABC
ABCD
ABCDE
```

Here, N = 5.

Examples:

```
Input Format: N = 3
Result:
A
A B
A B C

Input Format: N = 6
Result:
A
A B
A B C
A B C D
A B C D E
A B C D E F
```

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 numbers inside the inner loop.

- Observe symmetry in the pattern or check if a pattern is a combination of two or more similar patterns.

In this pattern problem, instead of numbers, we have to print alphabets hence making the pattern look like a right-angled triangle. So, the outer loop will run for N rows and the inner loop will loop for i alphabets in each row where i is the row number. Alphabets in each row will start from A each time we enter a new row and will loop till (A+i)th alphabet in that row.

Code:

C++ Java

```java
class Main {

    static void pattern14(int N)
    {

        // Outer loop for the number of rows.
        for(int i=0;i<N;i++){

            // Inner loop will loop for i times and
            // print alphabets from A to A + i.
            for(char ch = 'A'; ch<='A'+i;ch++){
                System.out.print(ch + " ");

            }
            // As soon as the letters for each iteration are printed, we move to the
            // next row and give a line break otherwise all letters
            // 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;
        pattern14(N);
    }
}
```

Output

```
A
A B
A B C
A B C D
A B C D E
```