Pattern - 6: Inverted Numbered Right Pyramid

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

Here, N = 5.

Examples:

```
Input Format: N = 3
Result:
1 2 3
1 2
1

Input Format: N = 6
Result:
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

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.

In this pattern, we run the outer loop for N times as we have to print N rows and since we have to print a right-angled triangle/pyramid which must be inverted, so the inner loop will run from 1 to (N-i)th integer in every row till we reach the Nth row where only '1' would be left to get printed. For eg: in the 1st-row numbers from 1 to N get printed, in the 2nd-row numbers from 1 to (N-1) get printed, and so on.

Code:

C++Java

```
#include <bits/stdc++.h>
using namespace std;

void pattern6(int N)
{
    // This is the outer loop which will loop for the rows.
    for (int i = 0; i < N; i++)
    {
        // This is the inner loop which loops for the columns
        // no. of columns = (N - row index) for each line here
        // as we have to print an inverted pyramid.
        // (N-j) will give us the numbers in a row starting from 1 to N-i.
        for (int j =N; j>i; j--)
        {
            cout <<N-j+1<<" ";
        }
```

```
        // As soon as numbers for each iteration are printed, we move to the
        // next row and give a line break otherwise all numbers
        // would get printed in 1 line.
        cout << endl;
    }
}

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

    pattern6(N);

    return 0;
}
```

Output

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```