Pattern - 11: Binary Number Triangle Pattern

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

Here, N = 5.

Examples:

```
Input Format: N = 3
Result:
1
01
101

Input Format: N = 6
Result:
1
01
101
0101
10101
010101
```

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 or not.

In this problem, we have to print binary digits alternatively in each row and column as shown in the examples. Let's say that the first row starts with the binary digit '1', the second row must start with '0' and then the 3rd row with '1' again, and so on. Similar is the case for the columns as well. Initially, we declare a start variable and set it to 1 for the first row. For even no. of rows, the start variable is 1 and for odd it is 0. Now for the inner loop, the numbers are printed i times (i is the row index) alternatively by simply subtracting the start variable from 1 after each iteration.

Code:

C++Java

```
class Main {

   static void pattern11(int N)
{
    // First row starts by printing a single 1.
      int start =1;

      // Outer loop for the no. of rows
      for(int i=0;i<N;i++){

          // if the row index is even then 1 is printed first
          // in that row.
          if(i%2 ==0) start = 1;

          // if odd, then the first 0 will be printed in that row.
          else start = 0;
```

```
            // We alternatively print 1's and 0's in each row by using
            // the inner for loop.
            for(int j=0;j<=i;j++){
                System.out.print(start);
                start = 1-start;
            }


        // As soon as the 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.
        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;
        pattern11(N);
    }
}
```

Output

```
1
01
101
0101
10101
```