Q.1 Count total number of subarrays with equal no. of 0's and 1's.

$$A = \left[\begin{array}{cccc} 1 & 0 & 0 & 1 \\ {\scriptstyle 0} & {\scriptstyle 1} & {\scriptstyle 2} & {\scriptstyle 3} \end{array}\right]$$

1, 0

0, 1

1, 0, 0, 1

$$A = \left[\begin{array}{ccccccc} 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ {\scriptstyle 0} & {\scriptstyle 1} & {\scriptstyle 2} & {\scriptstyle 3} & {\scriptstyle 4} & {\scriptstyle 5} & {\scriptstyle 6} \end{array}\right]$$

1, 0

0, 1

1, 0

1, 0, 0, 1

1, 1, 0, 0

1, 1, 0, 0, 1, 0

i) Replace 0's with -1, and then count total no. of subarrays having sum = 0.

$$\begin{array}{ccccccc} 1 & 1 & 1 & -1 & -1 & 1 & -1 \\ {\scriptstyle 0} & {\scriptstyle 1} & {\scriptstyle 2} & {\scriptstyle 3} & {\scriptstyle 4} & {\scriptstyle 5} & {\scriptstyle 6} \end{array}$$

subarray with 0 sum => count of 1's = $\underbrace{\text{count of -1's}}_{\text{count of 0's}}$

0	1	2	3	4	5	6
1	1	1	0	0	1	0

replace 0's with -1

0	1	2	3	4	5	6
1	1	1	-1	-1	1	-1

count subarrays having sum=0

	0	1	2	3	4	5	6
	1	1	1	-1	-1	1	-1
sum: 0	1	2	3	2	1	2	1

0 →	1
1 →	3
2 →	3
3 →	1

sum vs freq

count = 1 + 1 + 2 + 2

0 1 2 3

1 −1 −1 1

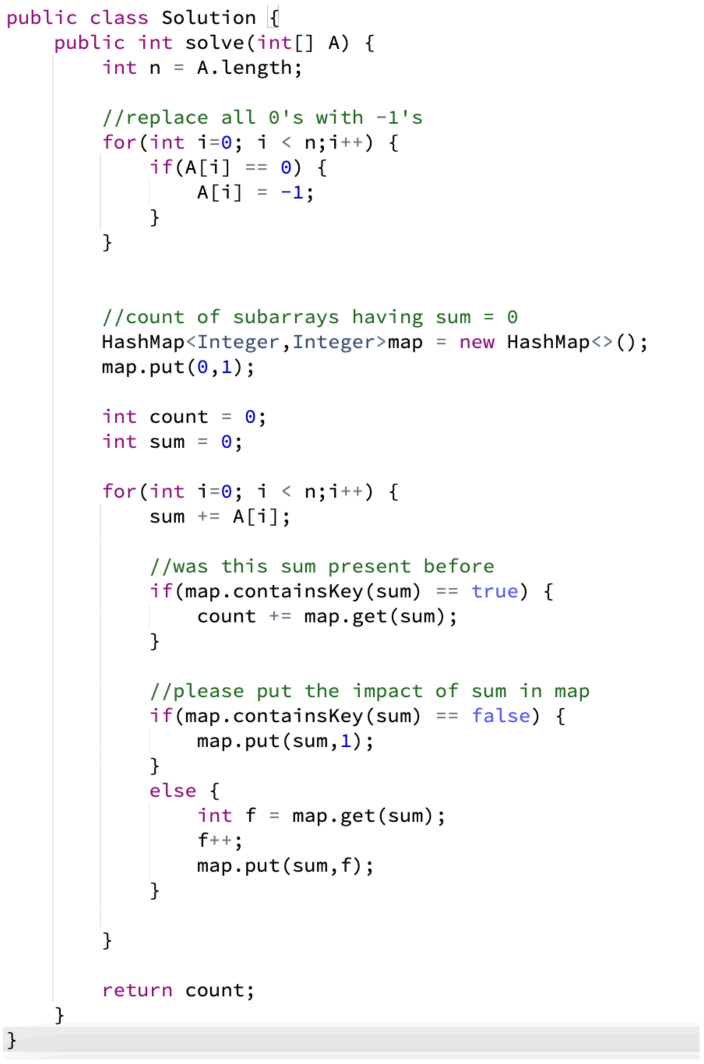
sum : 0 1 0 −1 0

count = 0 + 1 + 2

0 → 3

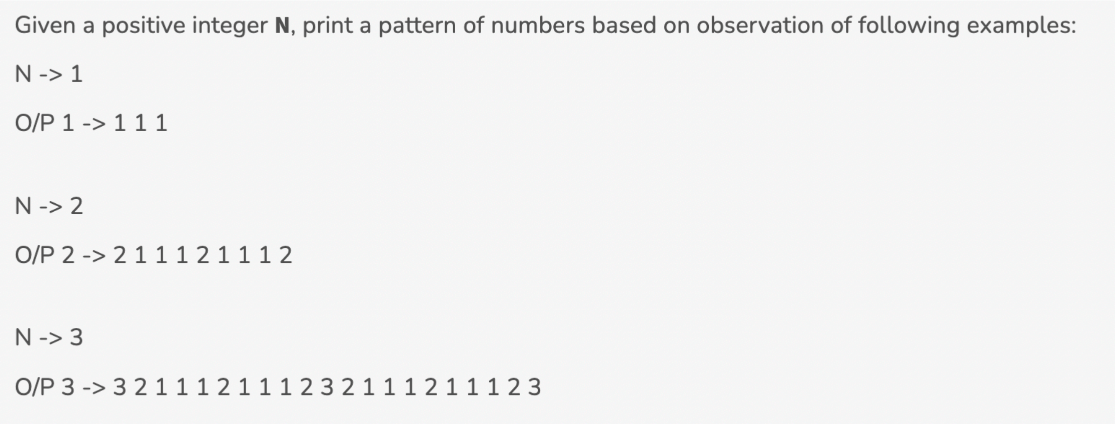
1 → 1

−1 → 1



```java
public class Solution {
    public int solve(int[] A) {
        int n = A.length;

        //replace all 0's with -1's
        for(int i=0; i < n;i++) {
            if(A[i] == 0) {
                A[i] = -1;
            }
        }


        //count of subarrays having sum = 0
        HashMap<Integer,Integer>map = new HashMap<>();
        map.put(0,1);

        int count = 0;
        int sum = 0;

        for(int i=0; i < n;i++) {
            sum += A[i];

            //was this sum present before
            if(map.containsKey(sum) == true) {
                count += map.get(sum);
            }

            //please put the impact of sum in map
            if(map.containsKey(sum) == false) {
                map.put(sum,1);
            }
            else {
                int f = map.get(sum);
                f++;
                map.put(sum,f);
            }

        }

        return count;
    }
}
```

Q. 2 Print zigzag.



Given a positive integer **N**, print a pattern of numbers based on observation of following examples:

N -> 1

O/P 1 -> 1 1 1

N -> 2

O/P 2 -> 2 1 1 1 2 1 1 1 2

N -> 3

O/P 3 -> 3 2 1 1 1 2 1 1 1 2 3 2 1 1 1 2 1 1 1 2 3

$pzz(1) = 1 \; 1 \; 1$

$$pzz(2) = 2 \; \underbrace{1 \; 1 \; 1}_{pzz(1)} \; 2 \; \underbrace{1 \; 1 \; 1}_{pzz(1)} \; 2$$

$$pzz(3) = 3 \; \underbrace{2 \; 1 \; 1 \; 1 \; 2 \; 1 \; 1 \; 1 \; 2}_{pzz(2)} \; 3 \; \underbrace{2 \; 1 \; 1 \; 1 \; 2 \; 1 \; 1 \; 1 \; 2}_{pzz(2)} \; 3$$

$pzz(n) \Rightarrow$ SOPln (n)

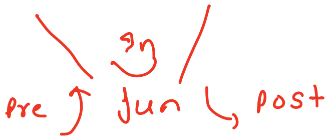
pzz (n-1)

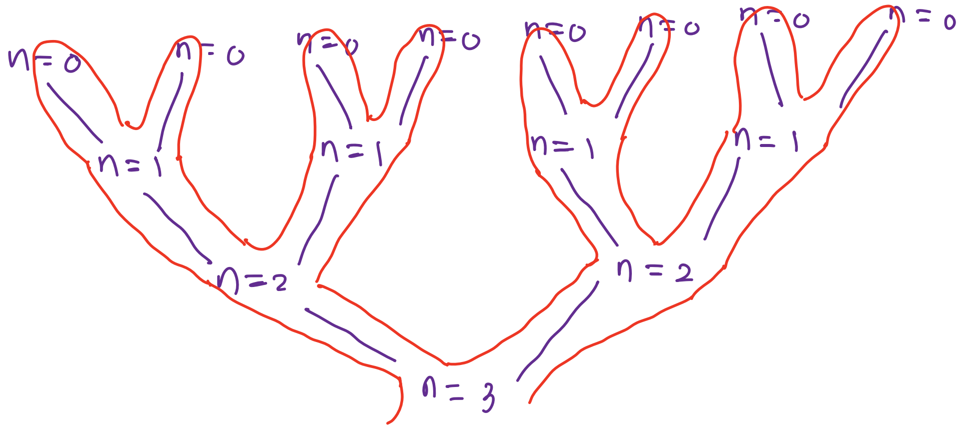
SOPln (n)

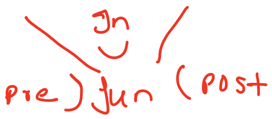
pzz (n-1)

SOPln (n)

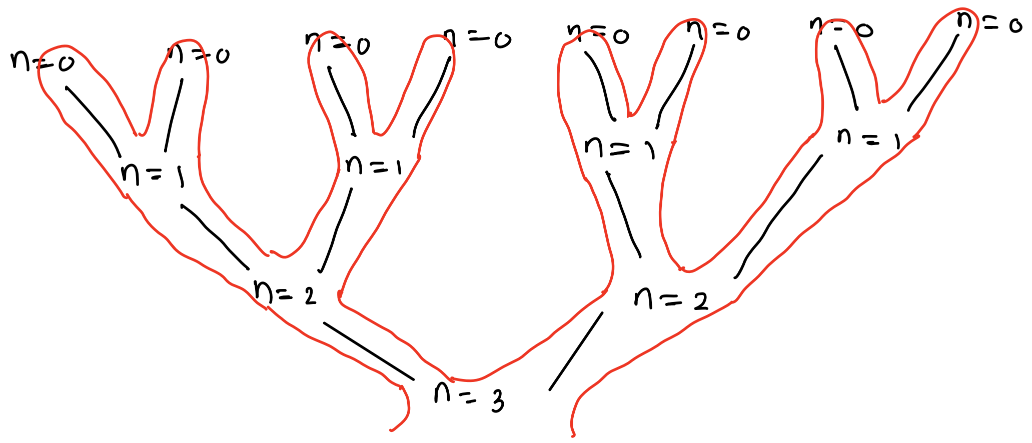
```
void   pzz (int  n)  {
        if(n == 0) {
            return;
        }

        sop(n + " ");  → pre

        pzz (n-1);

        sop(n + " ");  → In

        pzz (n-1);

        sop(n + " ");  → post

}
```



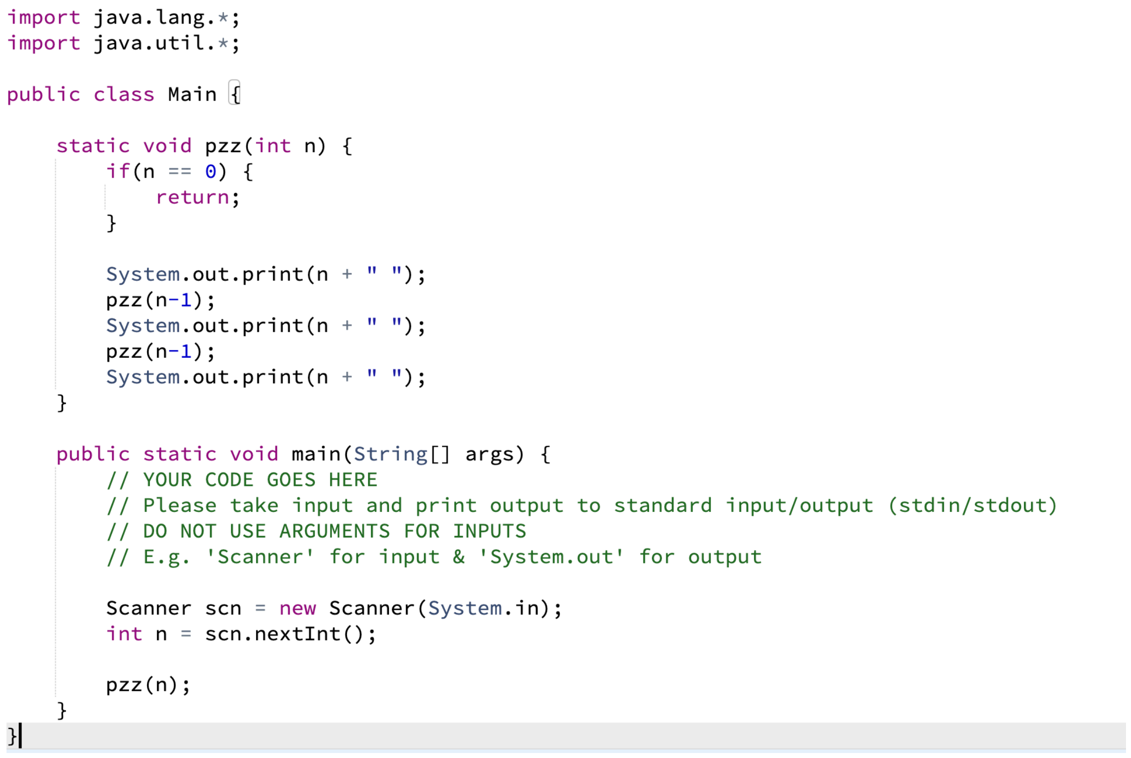




O/P   3 2 1 1 1 2 1 1 1 2 3 2 1 1 1 2 1 1 1 2 3



O|P: 3 2 1 1 1 2 1 1 1 2 3 2 1 1 1 2 1 1 1 2 3



```
import java.lang.*;
import java.util.*;

public class Main {

    static void pzz(int n) {
        if(n == 0) {
            return;
        }

        System.out.print(n + " ");
        pzz(n-1);
        System.out.print(n + " ");
        pzz(n-1);
        System.out.print(n + " ");
    }

    public static void main(String[] args) {
        // YOUR CODE GOES HERE
        // Please take input and print output to standard input/output (stdin/stdout)
        // DO NOT USE ARGUMENTS FOR INPUTS
        // E.g. 'Scanner' for input & 'System.out' for output

        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();

        pzz(n);
    }
}
```

i) class by class revision

ii) topic by topic revision