**Problem – 7 – Meter Surge Detection**

0- power ON

1- power surge detected

2- Tempering attempt

3- meter door open

4- battery low

5- signal loss

6- overload

..

**Problem Statement-**  
In a smart electricity metering system, each meter reports a **status code** in numeric form. This status code contains various flags encoded in **binary format**.

The **2nd bit (index 2, 0-based from the right)** of the status code indicates a **tampering attempt**.  
You are given the status codes of n meters. Your task is to **count how many meters** show a tampering attempt by checking if the 2nd bit is set in their status code.

**Example-**  
Input:

Enter the number of meters:

5

Enter the status codes (in number):

4 2 7 8 5

Output:

Number of meters with tempering attempt: 3

**Explanation-**  
Status codes in binary:

* 4 → 0100 → 2nd bit is set
* 2 → 0010 → 2nd bit not set
* 7 → 0111 → 2nd bit is set
* 8 → 1000 → 2nd bit not set
* 5 → 0101 → 2nd bit is set

So, **3 meters** have the 2nd bit set.

**Constraints:**

* Number of meters n > 0
* Status codes are non-negative integers

**Solution:**

import java.util.\*;

public class MeterSurgeDetection {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the number of meters: ");

        int n = scanner.nextInt();

        int [] inp = new int[n];

        System.out.println("Enter the status codes(in number): ");

        for (int i = 0; i < n; i++) {

            inp[i] = scanner.nextInt();

        }

        scanner.close();

        int count = 0;

        for (int j = 0; j < inp.length; j++) {

            int num = inp[j];

            if((num &(1<<2)) != 0){

                count++;

            }

        }

        System.out.println("Number of meters with tempering attempt: "+ count);

    }

}

**Complexities:**Time Complexity – O(N)  
Space Complexity – O(N)