CSE 338 Software Testing, Validation and Verification

Lab Report 1

Submitted to:

Prof. Dr. Islam Ahmed El-Maddah

Eng. Adham Nour

Submitted by:

Kerollos Wageeh Youssef 19P3468

CESS G2 S3 Junior

Watermelon

1- The Problem description from Codeforces:

One hot summer day Pete and his friend Billy decided to buy a watermelon. They chose the biggest and the ripest one, in their opinion. After that the watermelon was weighed, and the scales showed w kilos. They rushed home, dying of thirst, and decided to divide the berry, however they faced a hard problem.

Pete and Billy are great fans of even numbers, that's why they want to divide the watermelon in such a way that each of the two parts weighs even number of kilos, at the same time it is not obligatory that the parts are equal. The boys are extremely tired and want to start their meal as soon as possible, that's why you should help them and find out, if they can divide the watermelon in the way they want. For sure, each of them should get a part of positive weight.

Input:

The first (and the only) input line contains integer number w $(1 \leq w \leq 100)$ — the weight of the watermelon bought by the boys.

Output:

Print YES, if the boys can divide the watermelon into two parts, each of them weighing even number of kilos; and `NO` in the opposite case.

2- My Approaches to Fix it:

```java
import java.util.Scanner;

public class Watermelon {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int weight = input.nextInt();
        if(solveWatermelon(weight) == 1) System.out.println("YES");
        else System.out.println("NO");
    }
    public static Integer solveWatermelon(int weight){
        if(weight < 1 || weight > 100) return null;
        else if(weight % 2 == 0 && weight > 2) return 1;
        else return 0;
    }
}
```

3- The test cases that succeeded:

```java
import org.junit.Test;

import static org.junit.Assert.*;

public class WatermelonTest {

    @Test
    public void solveWatermelon() {
        assertTrue(Watermelon.solveWatermelon(-1) == null);
        assertTrue(Watermelon.solveWatermelon(0) == null);
        assertTrue(Watermelon.solveWatermelon(110) == null);
        assertTrue(Watermelon.solveWatermelon(1) == 0);
        assertTrue(Watermelon.solveWatermelon(100) == 1);
        assertTrue(Watermelon.solveWatermelon(2) == 0);
        assertTrue(Watermelon.solveWatermelon(3) == 0);
        assertTrue(Watermelon.solveWatermelon(4) == 1);
        assertTrue(Watermelon.solveWatermelon(5) == 0);
        assertTrue(Watermelon.solveWatermelon(6) == 1);
        assertTrue(Watermelon.solveWatermelon(7) == 0);
        assertTrue(Watermelon.solveWatermelon(50) == 1);
    }
}
```

4- My Github Repo that contains your solution:
https://github.com/KerollosWageeh/TestingLab1

Young Physicist

1- The Problem description from Codeforces:

A guy named Vasya attends the final grade of a high school. One day Vasya decided to watch a match of his favorite hockey team. And, as the boy loves hockey very much, even more than physics, he forgot to do the homework. Specifically, he forgot to complete his physics tasks. Next day the teacher got very angry at Vasya and decided to teach him a lesson. He gave the lazy student a seemingly easy task: You are given an idle body in space and the forces that affect it. The body can be considered as a material point with coordinates (0; 0; 0). Vasya had only to answer whether it is in equilibrium. "Piece of cake" — thought Vasya, we need only to check if the sum of all vectors is equal to 0. So, Vasya began to solve the problem. But later it turned out that there can be lots and lots of these forces, and Vasya can not cope without your help. Help him. Write a program that determines whether a body is idle or is moving by the given vectors of forces.

Input:

The first line contains a positive integer n ($1 \le n \le 100$), then follow n lines containing three integers each: the xi coordinate, the yi coordinate and the zi coordinate of the force vector, applied to the body (- 100 ≤ xi, yi, zi ≤ 100).

Output:

Print the word "YES" if the body is in equilibrium, or the word "NO" if it is not.

2- My Approaches to Fix it:

```
import java.util.ArrayList;
import java.util.Scanner;

public class YoungPhysicist {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int x, y, z, n;
        n = input.nextInt();
        ArrayList<Integer[]> arrayList = new ArrayList<>();
        while(n-- > 0)
        {
            x = input.nextInt();
            y = input.nextInt();
            z = input.nextInt();
            arrayList.add(new Integer[]{x, y, z});
        }
        if(solveYoungPhysicist(arrayList) == 1) System.out.println("YES");
        else System.out.println("NO");
    }
    public static Integer solveYoungPhysicist(ArrayList<Integer[]>
```

```
arrayList){

        int x = 0, y = 0, z = 0;
        for (Integer[] integers : arrayList) {
            x += integers[0];
            y += integers[1];
            z += integers[2];
            if(x < -100 || x > 100 || y < -100 || y > 100 || z < -100 || z >
100) return null;
        }
        if(x == 0 && y == 0 && z == 0) return 1;
        return 0;

    }
}
```

3- The test cases that succeeded:

```
import org.junit.Test;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;

import static org.junit.Assert.*;

public class YoungPhysicistTest {

    @Test
    public void solveYoungPhysicist() {
        ArrayList<Integer[]> arrayList = new ArrayList<>();

        arrayList.add(new Integer[]{-2, 4, -1});
        arrayList.add(new Integer[]{1, -5, -3});
        arrayList.add(new Integer[]{4, 1, 7});
        assertTrue(YoungPhysicist.solveYoungPhysicist(arrayList) == 0);
        arrayList.clear();

        arrayList.add(new Integer[]{3, -1, 7});
        arrayList.add(new Integer[]{-5, 2, -4});
        arrayList.add(new Integer[]{2, -1, -3});
        assertTrue(YoungPhysicist.solveYoungPhysicist(arrayList) == 1);
        arrayList.clear();

        arrayList.add(new Integer[]{-200, -1, 7});
        arrayList.add(new Integer[]{-5, 2, -4});
        arrayList.add(new Integer[]{2, -1, -3});
        assertTrue(YoungPhysicist.solveYoungPhysicist(arrayList) == null);
    }
}
```

4- My Github Repo that contains your solution:
https://github.com/KerollosWageeh/TestingLab1