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 Discretion from Codeforces:

2- My Approaches to Fix it:

```
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:

```
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 Discretion from Codeforces:

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;

    }
}
```

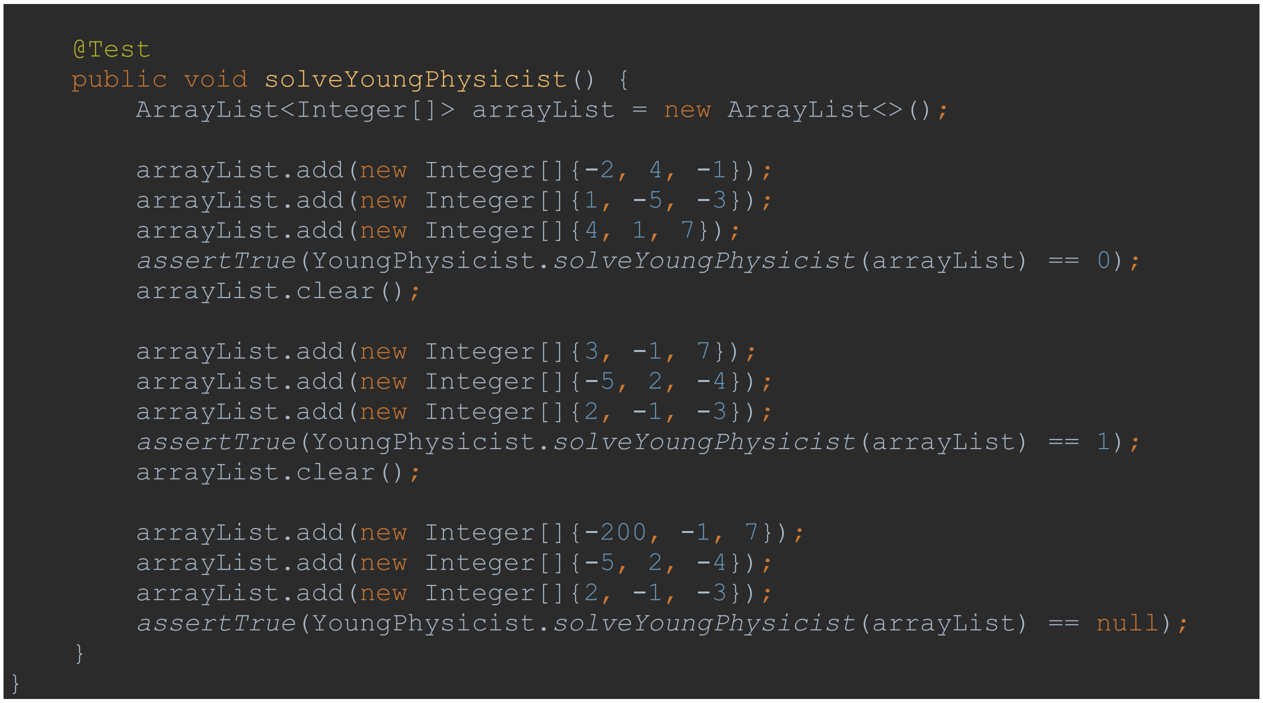
3- The test cases that succeeded:

```
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