LAB 1

(Young Physicist / Watermelon)

Karen Alber Farid
19P8948

16/04/2022

—

Software Testing

—

Eng. Adham

Young Physicist Pb

First Trial:

```
public class Young_Physicist {

        public String y_p(int n, int x, int y, int z) {
           int sum_x = 0, sum_y = 0, sum_z = 0;
            String result;
            if (n >= 1 && n <= 100 && x <= 100 && x >= -100 && y <= 100 && y >= -100 && z
<= 100 && z >= -100) {
                while (n-- >= 1) {
                    sum_x += x;
                    sum_y += y;
                    sum_z += z;
                }
            }
            else
              result = "out of bounds!!";



            if (sum_x == 0 && sum_y == 0 && sum_z == 0)
                result = "Yes";
            else
                result = "No";
            return result;
        }

}


import org.junit.Test;
import static org.junit.Assert.*;

public class Young_Physicist_test {

    @Test
    public void y_p() {
        Young_Physicist yp = new Young_Physicist();
         assertEquals("No", yp.y_p(3 ,3,0,0) );      //3  4 1 7  -2 4 -1  1 -5 -3
         assertEquals("Yes", yp.y_p(3 ,0,0,0) );     //3  3 -1 7  -5 2 -4  2 -1 -3

    }

}
```

Some Modified Steps:

Regarding the testing, I have tried to assert x, y and z with the number times of n but every time I insert more numbers, the compiler do not accept as it only accepts the same number of the function parameters regardless the repeat of the rest code as I wrote a while loop that takes values of x, y and z with the number of times of n. So I decided to try if the parameters take only the n value and the rest I will enter them manually what will the code be like?!
And that was the result:

Second Trial:

```
import java.util.Scanner;
public class Young_Physicist {

        public String y_p(int n) {
            int x; int y; int z;
            int sum_x = 0, sum_y = 0, sum_z = 0;
            String result;
            Scanner sc = new Scanner(System.in);
            if(n >= 1 && n <= 100) {
                while (n-- >= 1) {
                    x = sc.nextInt();
                    y = sc.nextInt();
                    z = sc.nextInt();
                    if(x <= 100 && x >= -100 && y <= 100 && y >= -100 && z <= 100 && z >=
-100)
                    {
                        sum_x += x;
                        sum_y += y;
                        sum_z += z;
                    }
                    else
                        result = "out of bounds!!";
                }
            }

            if (sum_x == 0 && sum_y == 0 && sum_z == 0)
                result = "Yes";
            else
                result = "No";
            return result;
        }

}
```

Some Modified Steps:

Here, I updated the main code to solve the previous problem but I couldn't update the testing code accordingly. So I decided to change the main code again to be as follows:

Third Trial:

```
public class Young_Physicist {

        public String y_p(int n, int[] arr) {
            int sum_x = 0, sum_y = 0, sum_z = 0;
            String result;
            if(n >= 1 && n <= 100) {
                for (int i = 0; i < 3 * n; i++) {
                    if (arr[i] >= -100 && arr[i] <= 100) {
                        for (int j = 0; j < 3; j++) {
                            sum_x += arr[3*j];
                            sum_y += arr[3*j + 1];
                            sum_z += arr[3*j + 2];
                        }
                    }
                }
            }
            else
                return result = "out of bounds!!";



            if (sum_x == 0 && sum_y == 0 && sum_z == 0)
                result = "Yes";
            else
                result = "No";
            return result;
        }
}

import org.junit.Test;
import static org.junit.Assert.*;

public class Young_Physicist_test {

    @Test
    public void y_p() {
        Young_Physicist yp = new Young_Physicist();
        int arr[] ={4,1,7,-2,4,-1,1,-5,-3};
         assertEquals("No", yp.y_p(3, arr));    //3  4 1 7  -2 4 -1  1 -5 -3
        int zrr[] ={3,-1,7,-5,2,-4,2,-1,-3};
         assertEquals("Yes", yp.y_p(3, zrr));     //3  3 -1 7  -5 2 -4  2 -1 -3
        int krr[] ={101,-102,0,-99,2,-4,2,99,-3};
        assertEquals("out of bounds!!", yp.y_p(101, krr));
        int vrr[] ={101,-102,99,0,100,-90,-101,2,-9};
```

```
        assertEquals("out of bounds!!", yp.y_p(102, vrr));
    }

}
```

Comment:

In the third trial, I tried to solve all the issues faced above and that was my final code.

Codeforces Submission:

```
import java.util.Scanner;
public class codeforces_yp {
      public static void main(String[] args){
         Scanner sc = new Scanner(System.in);
         int n = sc.nextInt();
         int x = 0, y = 0, z = 0;
         while(n-- >= 1){
            x = x + sc.nextInt();
            y = y + sc.nextInt();
            z = z + sc.nextInt();
         }
         if(x==0 && y==0 && z == 0)
            System.out.println("YES");
         else
            System.out.println("NO");
      }
   }
```

LINK: https://codeforces.com/contest/69/submission/149759420

Watermelon Pb

First Trial:

```
public class Watermelon {

    public String W_M(int w){
        String result;
        if(w >= 1 && w <= 100) {
            if (w % 2 == 0) {
                if (w == 1 || w == 2)
                    result = "NO";
                else
                    result = "YES";
            } else
                result = "NO";
        }
        else
            result = "out of bounds!!";
        return result;
    }
}
```

```
import org.junit.Test;
import static org.junit.Assert.*;

public class Watermelon_test {

    @Test
    public void W_M(){
        Watermelon wm = new Watermelon();
        assertEquals("YES" , wm.W_M(8));
        assertEquals("out of bounds!!" , wm.W_M(0));
        assertEquals("NO" , wm.W_M(1));
        assertEquals("NO" , wm.W_M(2));
        assertEquals("NO" , wm.W_M(9));
        assertEquals("YES" , wm.W_M(100));
        assertEquals("out of bounds!!" , wm.W_M(102));

    }
}
```

Some modifications done over the code:

I changed in the code assigned on codeforces so as to remove the Scanner input java function and try asserting the values manually to test the code. This above code is the final code reached so far.

Codeforces Submission:

```
import java.util.Scanner;
public class Watermelon {

   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      int w = sc.nextInt();
      if(w%2 == 0){
         if(w <= 2)
            System.out.println("NO");
         else
            System.out.println("YES");
      }
      else
         System.out.println("NO");
   }
}
```

LINK: https://codeforces.com/contest/4/submission/149768787