

Faculty of Engineering

Ain Shams University

**REPORT LAB (1)**

|  |  |  |
| --- | --- | --- |
| **EMAIL** | **ID** | **NAME** |
| mennatallaashraf@gmail.com | **17p3051** | **Mennat – Allah Ashraf Fouad Fetouh** |

***Submitted to:***

**Dr. Islam El-Maddah**

**Eng. Adham Nour**

**Problem 1:  *Watermelon***

* ***Problem Description:***

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.

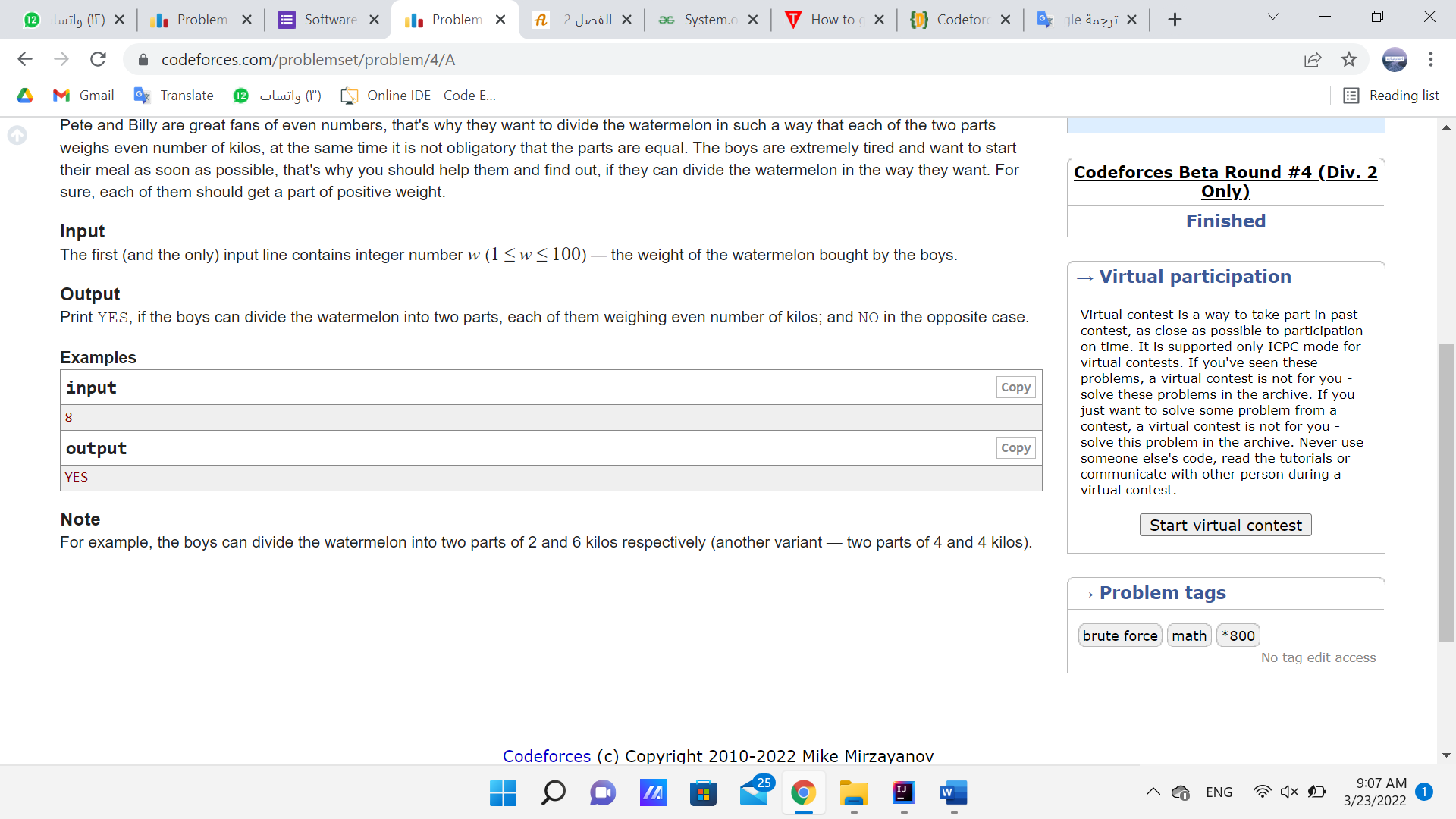
Pete and Billy are great fans of even numbers, that's why they want to divide the watermelon in such a way that each of the two parts weighs even number of kilos, at the same time it is not obligatory that the parts are equal. The boys are extremely tired and want to start their meal as soon as possible, 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 ≤ *w* ≤ 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.



***Solution of Problem 1:***

1. ***The main code:***

***Watermelon.java***

public class Watermelon {  
 public int watermelon(int w) // w is weight  
 {  
 if (w<1 || w> 100) //1 ≤ w ≤ 100  
 {  
 System.*out*.println("WARNING:");  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*WRONG Entered weight\*\*\*\*\*\*\*\*");  
 System.*out*.println("The weight RANGE:(1 ≤ w ≤ 100)");  
 return -1;  
 }  
  
 else{  
 if (w%2 ==0 && w!=2)  
 {  
 System.*out*.println("YES");  
 return 1;  
 }  
  
 else  
 {  
 System.*out*.println("NO");  
 return 0;  
 }  
  
 }  
  
  
 }  
}

1. ***Test Cases:***

***WatermelonTest.java***

import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class WatermelonTest {

@Test  
 public void odd\_weight() {  
 Watermelon tester = new Watermelon();  
 *assertEquals*("NO",0,tester.watermelon(5));  
 }

@Test  
 public void even\_weight() {  
  
 Watermelon tester = new Watermelon();  
 *assertEquals*("YES",1,tester.watermelon(8));  
  
 }  
  
 @Test  
 public void weight\_equal\_2kilo() {  
 Watermelon tester = new Watermelon();  
 *assertEquals*("NO",0,tester.watermelon(2));  
 }  
  
 @Test  
 public void weight\_larger\_than\_100kilo() {  
 Watermelon tester = new Watermelon();  
 *assertEquals*(-1,tester.watermelon(150));  
  
 }  
  
 @Test  
 public void ZERO\_weight() {  
 Watermelon tester = new Watermelon();  
 *assertEquals*(-1,tester.watermelon(0)); }  
  
}

***Test Cases that fails:***

No test cases fails

***Test Cases that*** ***succeeded:***

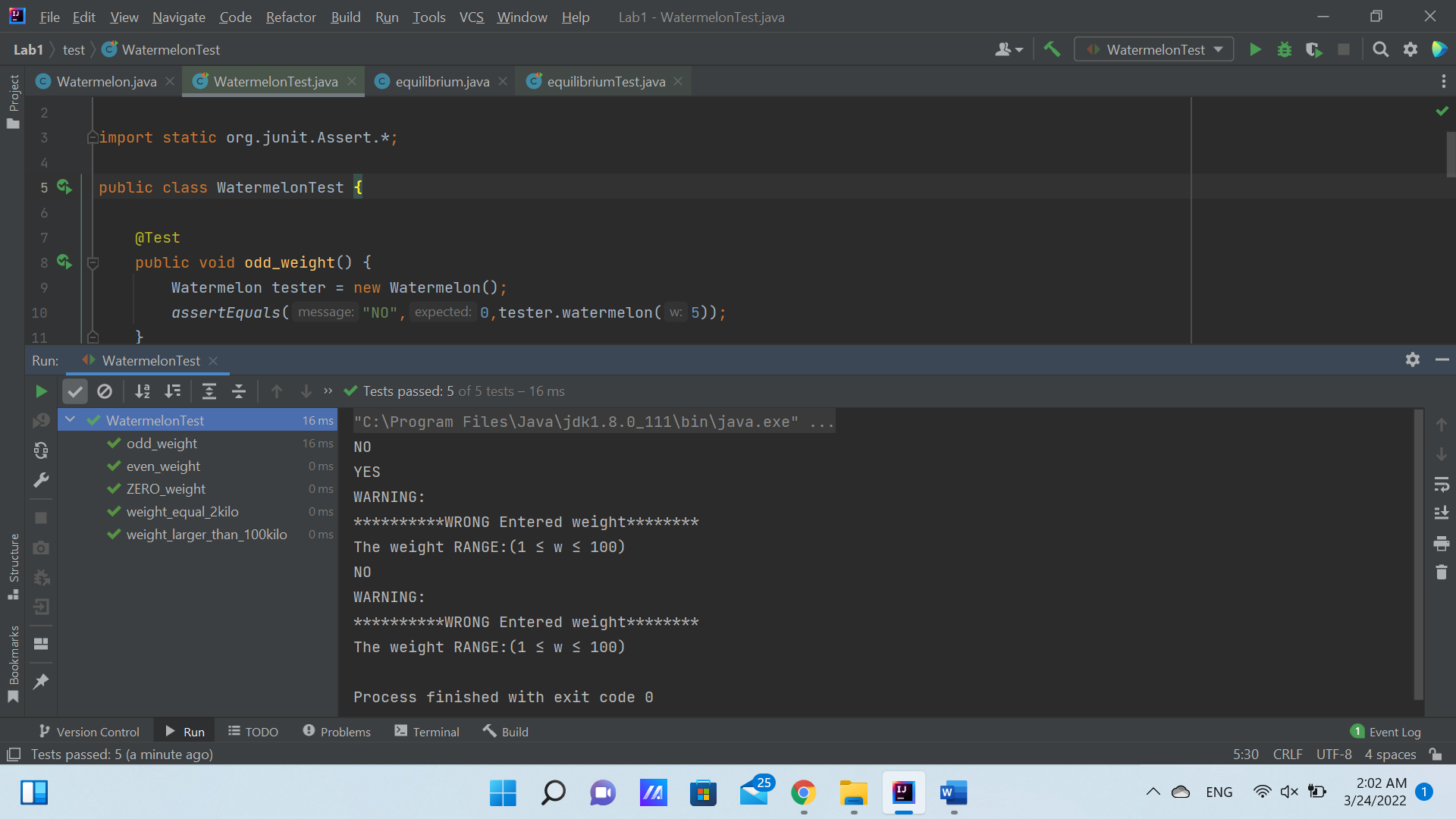
All test cases are succeeded.

import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class WatermelonTest {

@Test  
 public void odd\_weight() {  
 Watermelon tester = new Watermelon();  
 *assertEquals*("NO",0,tester.watermelon(5));  
 }  
  
 @Test  
 public void even\_weight() {  
  
 Watermelon tester = new Watermelon();  
 *assertEquals*("YES",1,tester.watermelon(8));  
  
 }

@Test  
 public void weight\_equal\_2kilo() {  
 Watermelon tester = new Watermelon();  
 *assertEquals*("NO",0,tester.watermelon(2));  
 }  
  
 @Test  
 public void weight\_larger\_than\_100kilo() {  
 Watermelon tester = new Watermelon();  
 *assertEquals*(-1,tester.watermelon(150));  
  
 }  
  
 @Test  
 public void ZERO\_weight() {  
 Watermelon tester = new Watermelon();  
 *assertEquals*(-1,tester.watermelon(0)); }  
  
}

***Test Cases “Run” window:***



**Problem 2:  *Young Physicist***

* ***Problem Description:***

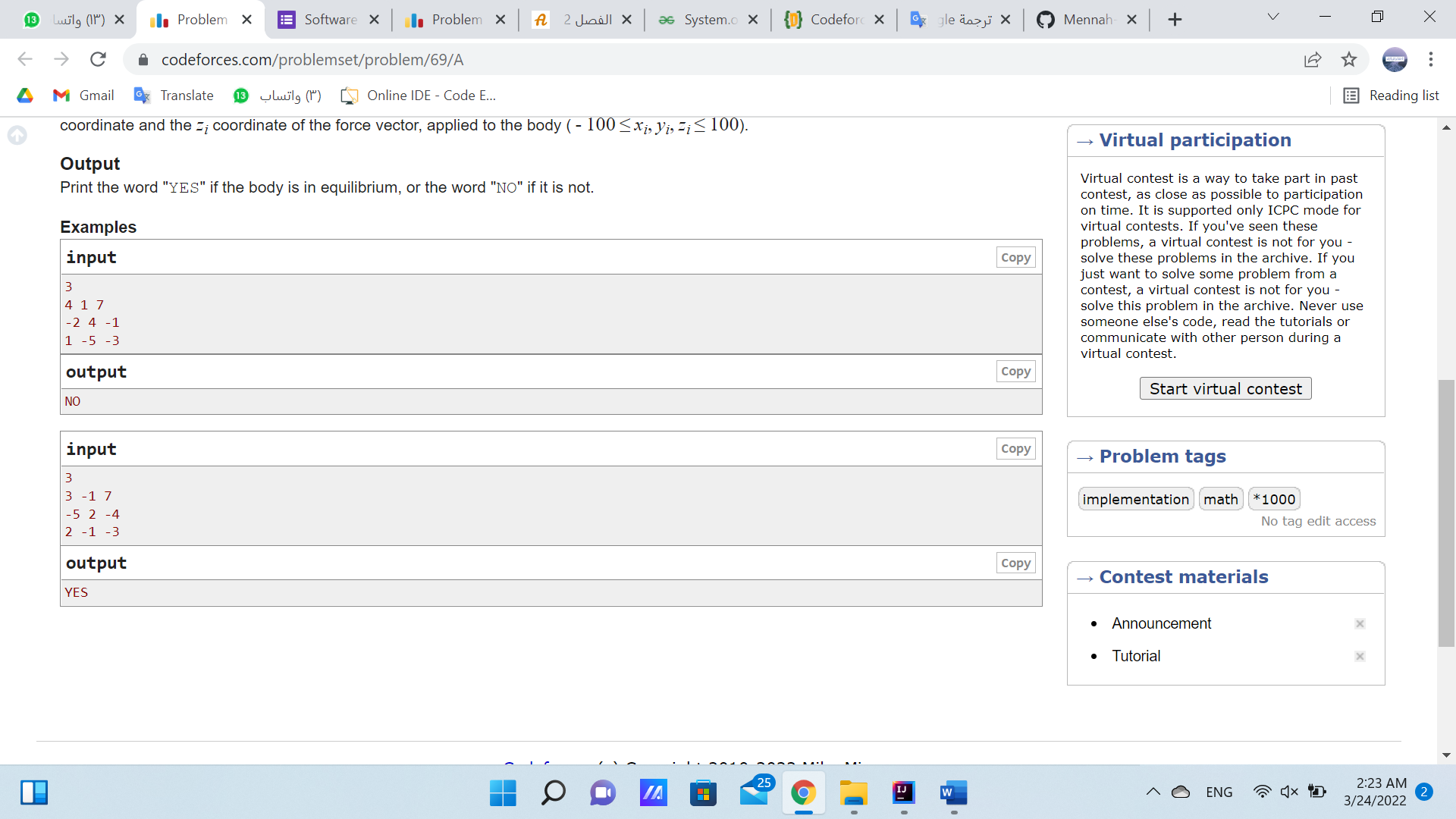
A guy named Vasya attends the final grade of a high school. One day Vasya decided to watch a match of his favorite hockey team. And, as the boy loves hockey very much, even more than physics, he forgot to do the homework. Specifically, he forgot to complete his physics tasks. Next day the teacher got 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 ≤ *n* ≤ 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.



***Solution of Problem 2:***

1. ***The main code:***

***equilibruim.java***

public class equilibrium {  
 public int equilibrium (int n, int[][] arr) //n is number of lines  
 {  
 if (n<1 || n> 100) //1 ≤ n ≤ 100  
 {  
 System.*out*.println("WARNING:");  
 System.*out*.println("\*\*\*\*\*\*\* WRONG NO. OF LINES ENTERED \*\*\*\*\*\*\*");  
 System.*out*.println("NO. OF LINES (n) RANGE: (1 ≤ n ≤ 100)");  
 return -1;  
 }  
  
 else  
 {  
 int [] sum ={0,0,0};  
  
 for(int i=0; i < n ; i++) //i = row  
 {  
 for (int j=0; j < 3 ; j++) //j = column  
 {  
 int x = arr[i][j];  
 sum[j] = +x; // Sum of column items  
 }  
 }

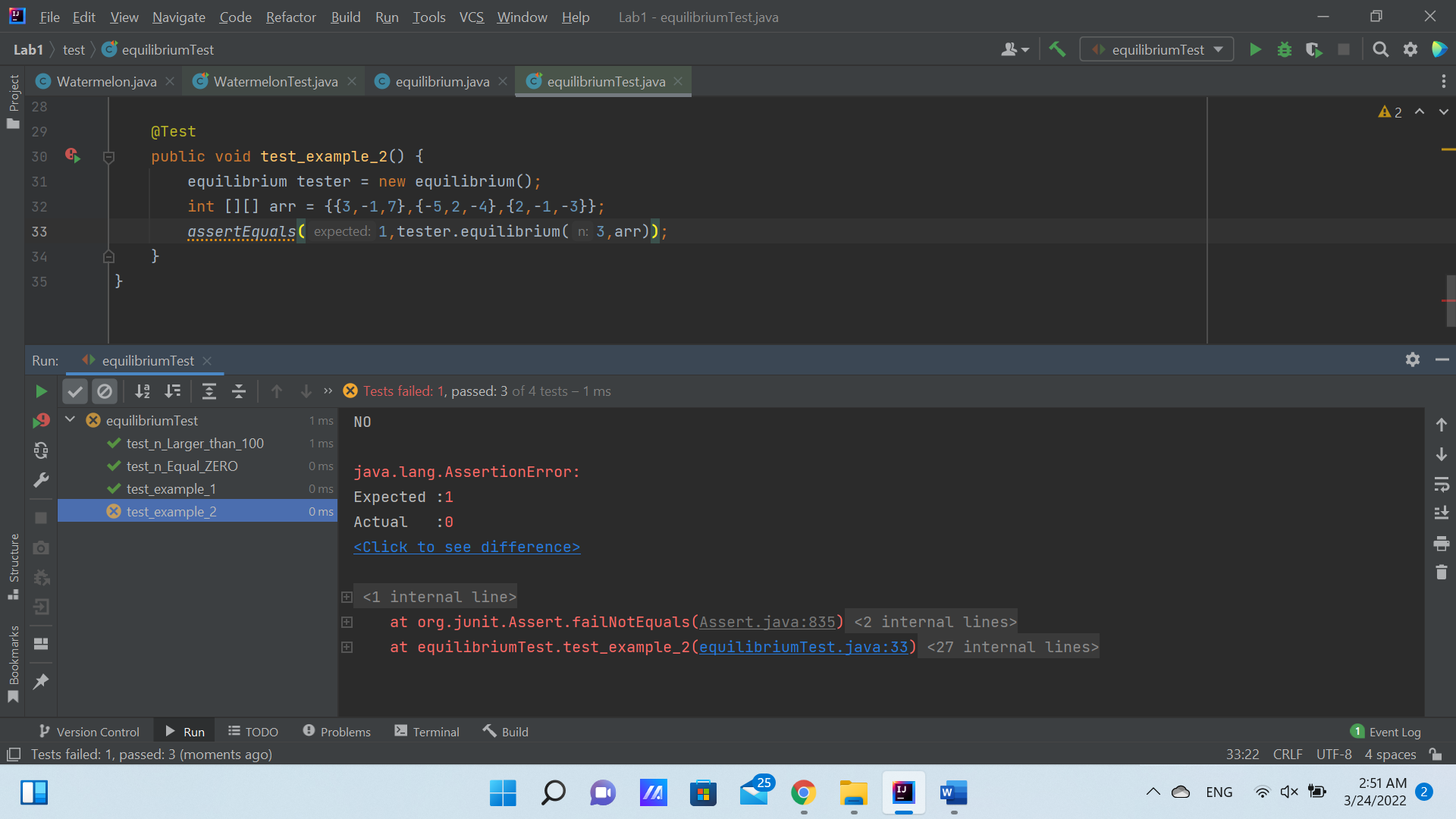
if (sum[0]==0 && sum[1]==0 && sum[2]==0){  
 System.*out*.println("YES");  
 return 1;  
 }  
  
 else  
 {  
 System.*out*.println("NO");  
 return 0;  
 }  
  
 }  
 }  
}

1. ***Test Cases:***

***equilibruimTest.java***

import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class equilibriumTest {  
  
 @Test  
 public void test\_n\_Equal\_ZERO() {  
 equilibrium tester = new equilibrium();  
 int [][] arr = {{4,1,7},{-2,4,-1},{1,-5,-3}};  
 *assertEquals*(-1,tester.equilibrium(0,arr));  
 }  
  
 @Test  
 public void test\_n\_Larger\_than\_100() {  
  
 equilibrium tester = new equilibrium();  
 int [][] arr = {{4,1,7},{-2,4,-1},{1,-5,-3}};  
 *assertEquals*(-1,tester.equilibrium(200,arr));  
 }  
  
 @Test  
 public void test\_example\_1() {  
 equilibrium tester = new equilibrium();  
 int [][] arr = {{4,1,7},{-2,4,-1},{1,-5,-3}};  
 *assertEquals*("NO",0,tester.equilibrium(3,arr));  
 }  
  
 @Test  
 public void test\_example\_2() {  
 equilibrium tester = new equilibrium();  
 int [][] arr = {{3,-1,7},{-5,2,-4},{2,-1,-3}};  
 *assertEquals*(1,tester.equilibrium(3,arr));  
 }  
}

***Test Cases that fails:***



All other test cases are **succeeded**.

***How to fix the test case that failed?***

1. Add to the main code ***“equilibruim.java”*** the following codeafter the for loop and before the internal if-statement in the else part:

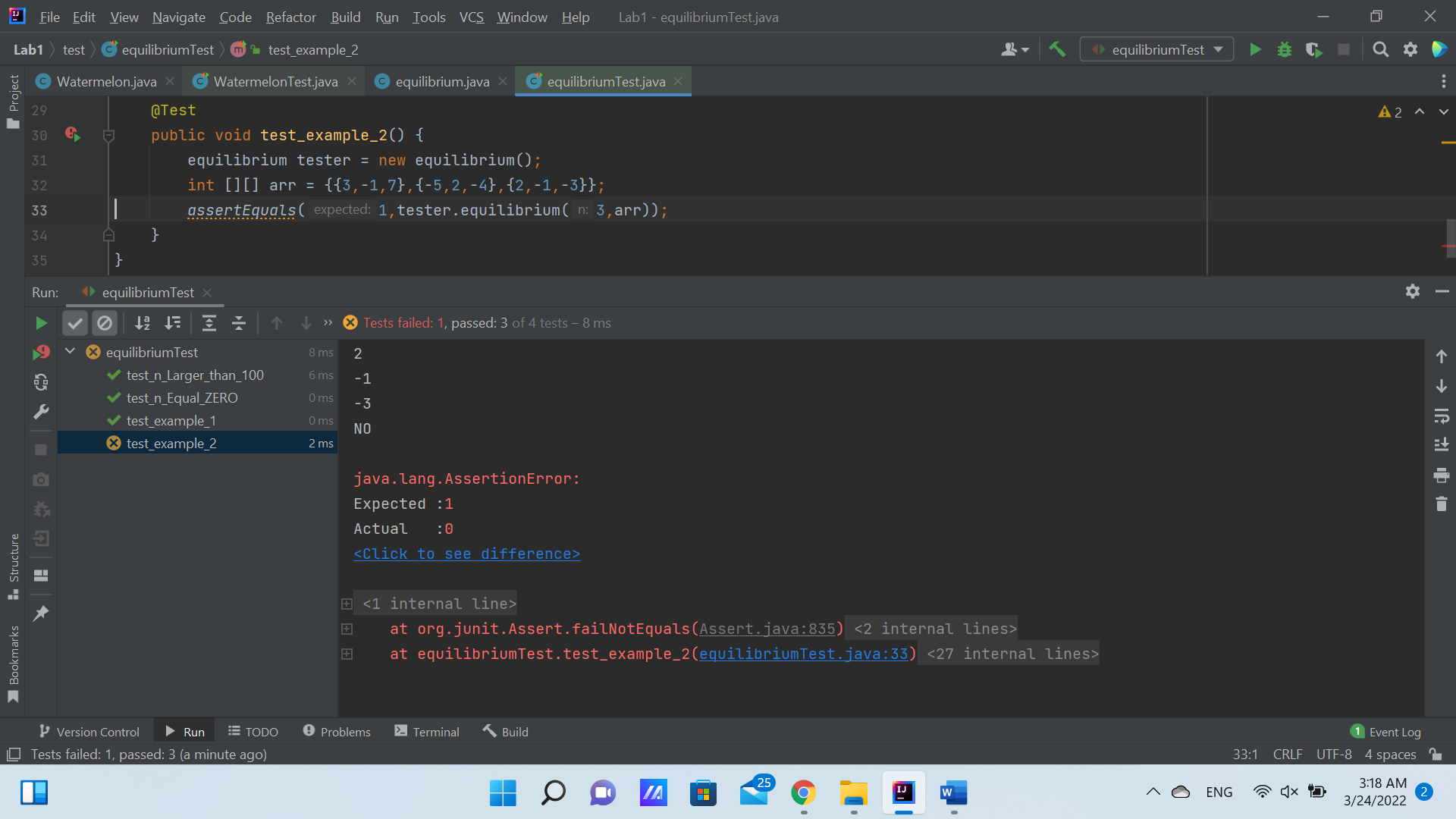
System.*out*.println(sum[0]);  
System.*out*.println(sum[1]);  
System.*out*.println(sum[2]);

This will show the sum array content.

The main code ***“equilibruim.java”*** will be:

public class equilibrium {  
 public int equilibrium (int n, int[][] arr) //n is number of lines  
 {  
 if (n<1 || n> 100) //1 ≤ n ≤ 100  
 {  
 System.*out*.println("WARNING:");  
 System.*out*.println("\*\*\*\*\*\*\* WRONG NO. OF LINES ENTERED \*\*\*\*\*\*\*");  
 System.*out*.println("NO. OF LINES (n) RANGE: (1 ≤ n ≤ 100)");  
 return -1;  
 }  
  
 else  
 {  
 int [] sum ={0,0,0};  
  
 for(int i=0; i < n ; i++) //i = row  
 {  
 for (int j=0; j < 3 ; j++) //j = column  
 {  
 int x = arr[i][j];  
 sum[j] = +x; // Sum of column items  
 }  
 }  
  
 System.*out*.println(sum[0]);  
 System.*out*.println(sum[1]);  
 System.*out*.println(sum[2]);  
  
  
 if (sum[0]==0 && sum[1]==0 && sum[2]==0){  
 System.*out*.println("YES");  
 return 1;  
 }  
  
 else  
 {  
 System.*out*.println("NO");  
 return 0;  
 }  
  
 }  
 }  
}

Then, save changes and run again the test cases:



**The content of sum array**

This content results from adding only the first 2 elements on column without adding the last element. Thus, the for loop doesn’t work as expected.

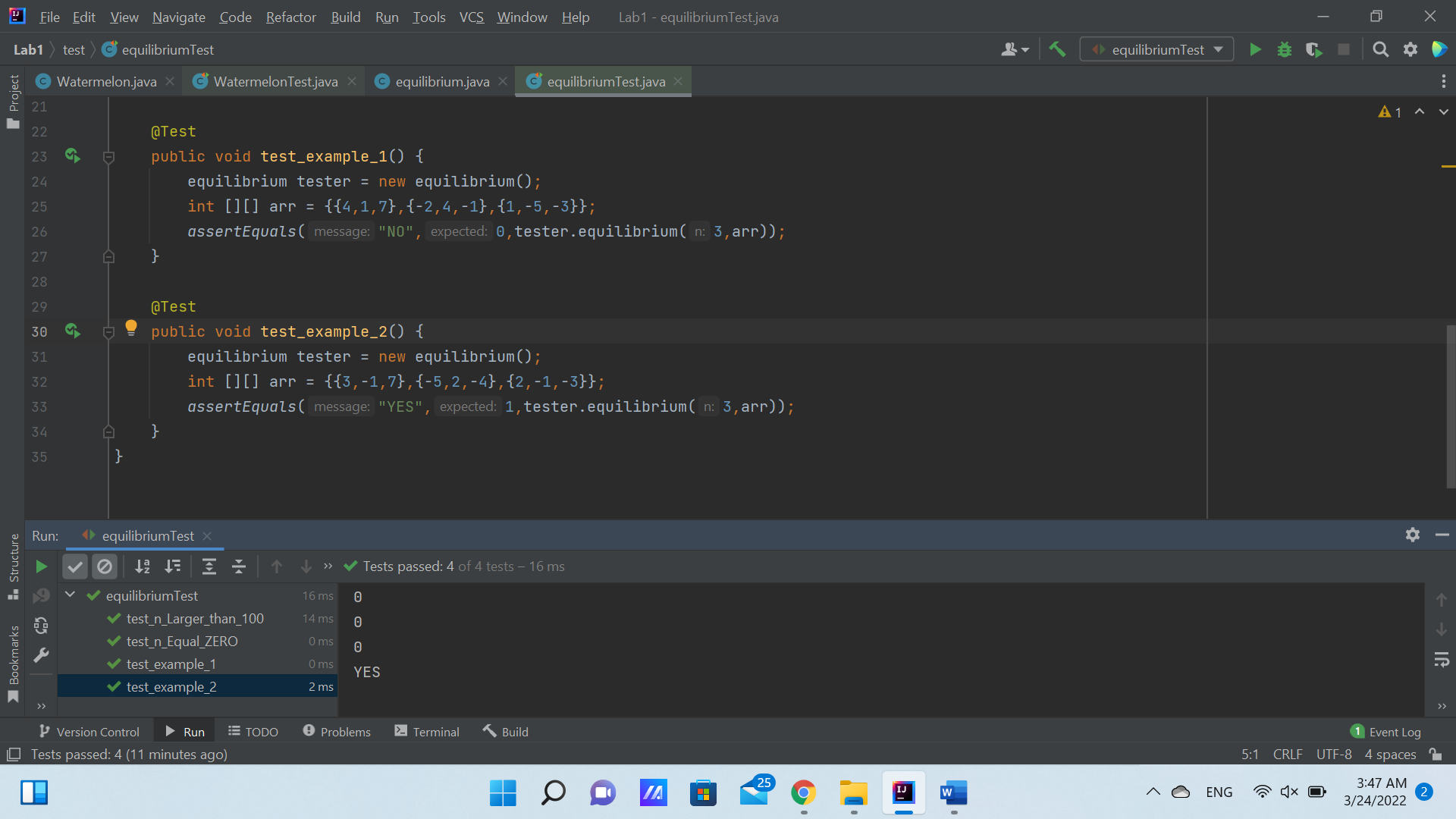
* After reviewing the for-loop code again, and change the code from :

sum[j] =+ x; // Sum of column items

To:

sum[j] += x; // Sum of column items

Then, it runs as expected and all test cases are succeeded.



The main code ***“equilibruim.java”*** after editing is:

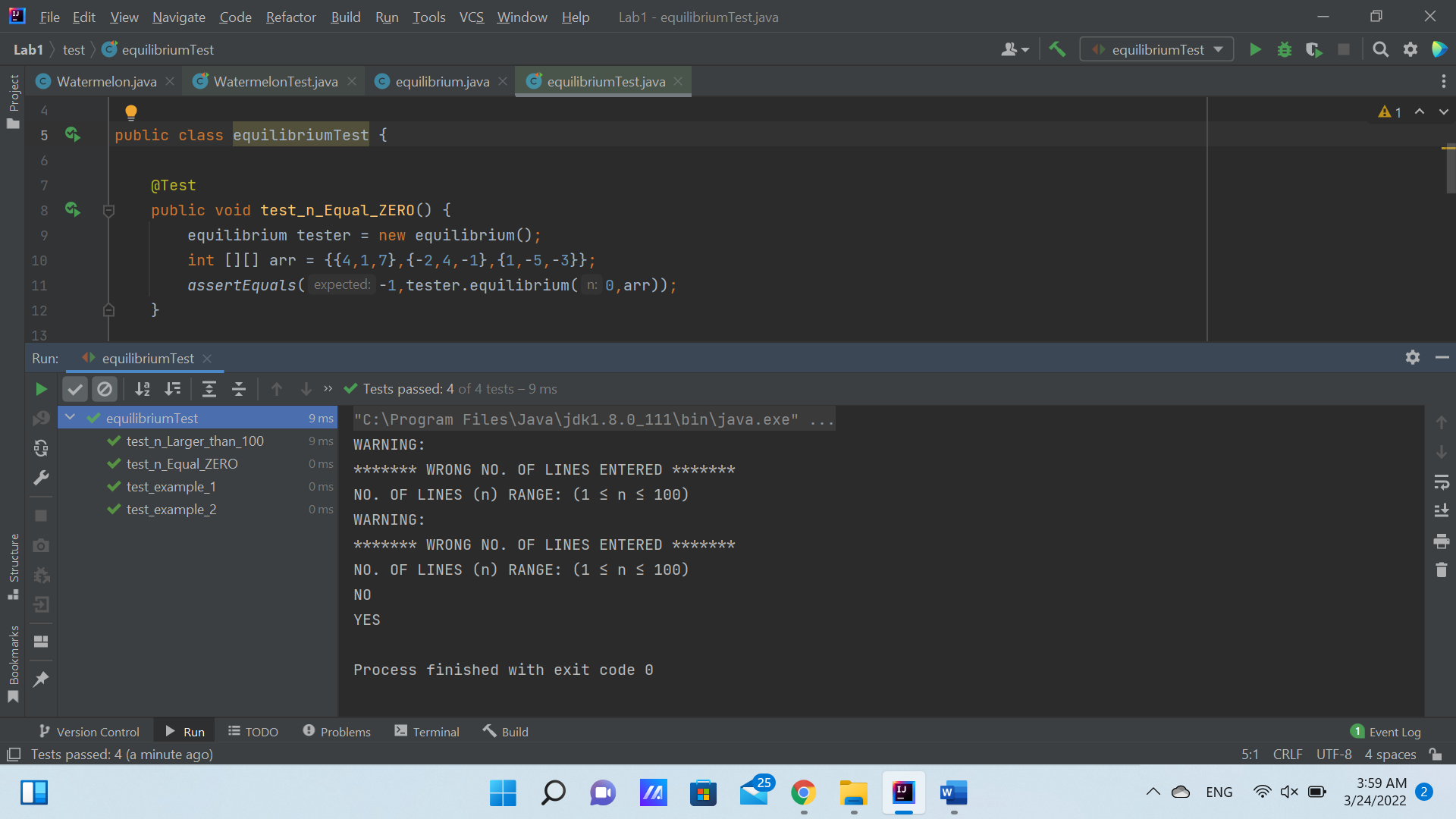
public class equilibrium {  
 public int equilibrium (int n, int[][] arr) //n is number of lines  
 {  
 if (n<1 || n> 100) //1 ≤ n ≤ 100  
 {  
 System.*out*.println("WARNING:");  
 System.*out*.println("\*\*\*\*\*\*\* WRONG NO. OF LINES ENTERED \*\*\*\*\*\*\*");  
 System.*out*.println("NO. OF LINES (n) RANGE: (1 ≤ n ≤ 100)");  
 return -1;  
 }  
  
 else  
 {  
 int [] sum ={0,0,0};  
  
 for(int i=0; i < n ; i++) //i = row  
 {  
 for (int j=0; j < 3 ; j++) //j = column  
 {  
 int x = arr[i][j];  
 sum[j] += x; // Sum of column items  
 }  
 }  
  
  
 if (sum[0]==0 && sum[1]==0 && sum[2]==0){  
 System.*out*.println("YES");  
 return 1;  
 }  
  
 else  
 {  
 System.*out*.println("NO");  
 return 0;  
 }  
  
 }  
 }  
}

The code of test cases ***“equilibruimTest.java”*** is:

import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class equilibriumTest {  
  
 @Test  
 public void test\_n\_Equal\_ZERO() {  
 equilibrium tester = new equilibrium();  
 int [][] arr = {{4,1,7},{-2,4,-1},{1,-5,-3}};  
 *assertEquals*(-1,tester.equilibrium(0,arr));  
 }

@Test  
 public void test\_n\_Larger\_than\_100() {  
  
 equilibrium tester = new equilibrium();  
 int [][] arr = {{4,1,7},{-2,4,-1},{1,-5,-3}};  
 *assertEquals*(-1,tester.equilibrium(200,arr));  
 }  
  
 @Test  
 public void test\_example\_1() {  
 equilibrium tester = new equilibrium();  
 int [][] arr = {{4,1,7},{-2,4,-1},{1,-5,-3}};  
 *assertEquals*("NO",0,tester.equilibrium(3,arr));  
 }  
  
 @Test  
 public void test\_example\_2() {  
 equilibrium tester = new equilibrium();  
 int [][] arr = {{3,-1,7},{-5,2,-4},{2,-1,-3}};  
 *assertEquals*("YES",1,tester.equilibrium(3,arr));  
 }  
}

***Test Cases “Run” window:***



Finally, all test cases are **succeeded**.

***GitHub Repo:***

My GitHub Repo contains solution & codes of this lab report problems. Link is below:

<https://github.com/Mennah-Ashraf/Testing-LabReport1.git>

**Important Note:**

Recently, I have been transferred to be with Group 2 in this course labs. Thus, the deadline is been extended for me.