IT314 - Lab7

Name : Sumit Vaniya
ID : 202001085
Date : 12-04-2023

Section A :

Based on the input ranges, we can identify the following equivalence classes:

Valid dates: The input triple (day, month, year) that represent a valid date in the Gregorian calendar, such as (3, 4, 1995).
Invalid dates: The input triple (day, month, year) that represent an invalid date, such as (31, 2, 2022) or (29, 2, 1900).
Out of range dates: The input triple (day, month, year) that are outside the allowed ranges, such as (0, 5, 2010) or (15, 13, 2005). Based on these equivalence classes, we can design the following test cases:

Tester Action and Input Data Expected Outcome

Valid dates:

Calculate previous date for (15, 10, 2022) 14, 10, 2022
Calculate previous date for (1, 1, 2015) 31, 12, 2014
Calculate previous date for (31, 3, 2000) 30, 3, 2000

Invalid dates:

Calculate previous date for (29, 2, 2022) Invalid date
Calculate previous date for (31, 4, 2010) Invalid date
Calculate previous date for (30, 2, 2000) Invalid date

Out of range dates:

Calculate previous date for (0, 5, 2010) Invalid date
Calculate previous date for (15, 13, 2005) Invalid date
Calculate previous date for (31, 12, 1899) Invalid date

Boundary Value Analysis:

Using boundary value analysis, we can identify the following boundary test cases:

The earliest possible date: (1, 1, 1900)
The latest possible date: (31, 12, 2015)
The earliest day of each month: (1, 1, 2000), (1, 2, 2000), (1, 3, 2000),..., (1, 12, 2000)
The latest day of each month: (31, 1, 2000), (28, 2, 2000), (31, 3, 2000),..., (31, 12, 2000)
Leap year day: (29, 2, 2000)
Invalid leap year day: (29, 2, 1900)
One day before earliest date: (31, 12, 1899)
One day after latest date: (1, 1, 2016)

Based on these boundary test cases, we can design the following test cases:

Tester Action and Input Data Expected Outcome

Boundary Test Cases:

Calculate previous date for (1, 1, 1900) Invalid date
Calculate previous date for (31, 12, 2015) 30, 12, 2015
Calculate previous date for (1, 1, 2000) 31, 12, 1999
Calculate previous date for (31, 1, 2000) 30, 1, 2000
Calculate previous date for (29, 2, 2000) 28, 2, 2000

Calculate previous date for (29, 2, 1900) Invalid date
Calculate previous

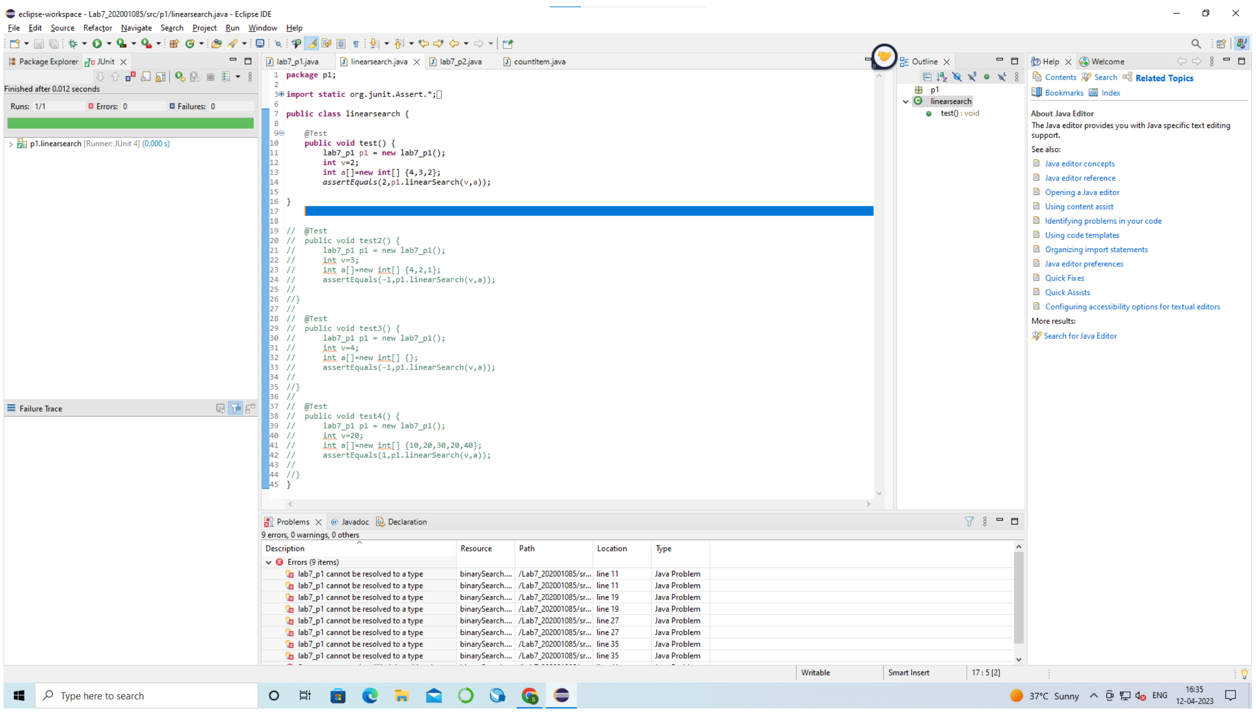
Programs

(P1). The function linearSearch searches for a value v in an array of integers a. If v appears in the array a, then the function returns the first index i, such that a[i] == v; otherwise, -1 is returned.

```
int linearSearch(int v, int a[]){
    int i = 0;
    while (i < a.length){
        if (a[i] == v)
        return(i);
        i++;
    }
    return (-1);
}
```

Test Cases :

(1) V = 2 , a={4,3,2} expected output = 2



eclipse-workspace - Lab7_202001085/src/p1/linearsearch.java - Eclipse IDE

JUnit: Finished after 0.012 seconds — Runs: 1/1 Errors: 0 Failures: 0

p1.linearsearch [Runner: JUnit 4] (0.000 s)

```java
package p1;

import static org.junit.Assert.*;

public class linearsearch {

    @Test
    public void test() {
        lab7_p1 p1 = new lab7_p1();
        int v=2;
        int a[]=new int[] {4,3,2};
        assertEquals(2,p1.linearSearch(v,a));

}


//  @Test
//  public void test2() {
//      lab7_p1 p1 = new lab7_p1();
//      int v=3;
//      int a[]=new int[] {4,2,1};
//      assertEquals(-1,p1.linearSearch(v,a));
//
//}
//
//  @Test
//  public void test3() {
//      lab7_p1 p1 = new lab7_p1();
//      int v=4;
//      int a[]=new int[] {};
//      assertEquals(-1,p1.linearSearch(v,a));
//
//}
//
//  @Test
//  public void test4() {
//      lab7_p1 p1 = new lab7_p1();
//      int v=20;
//      int a[]=new int[] {10,20,30,20,40};
//      assertEquals(1,p1.linearSearch(v,a));
//
//}
}
```

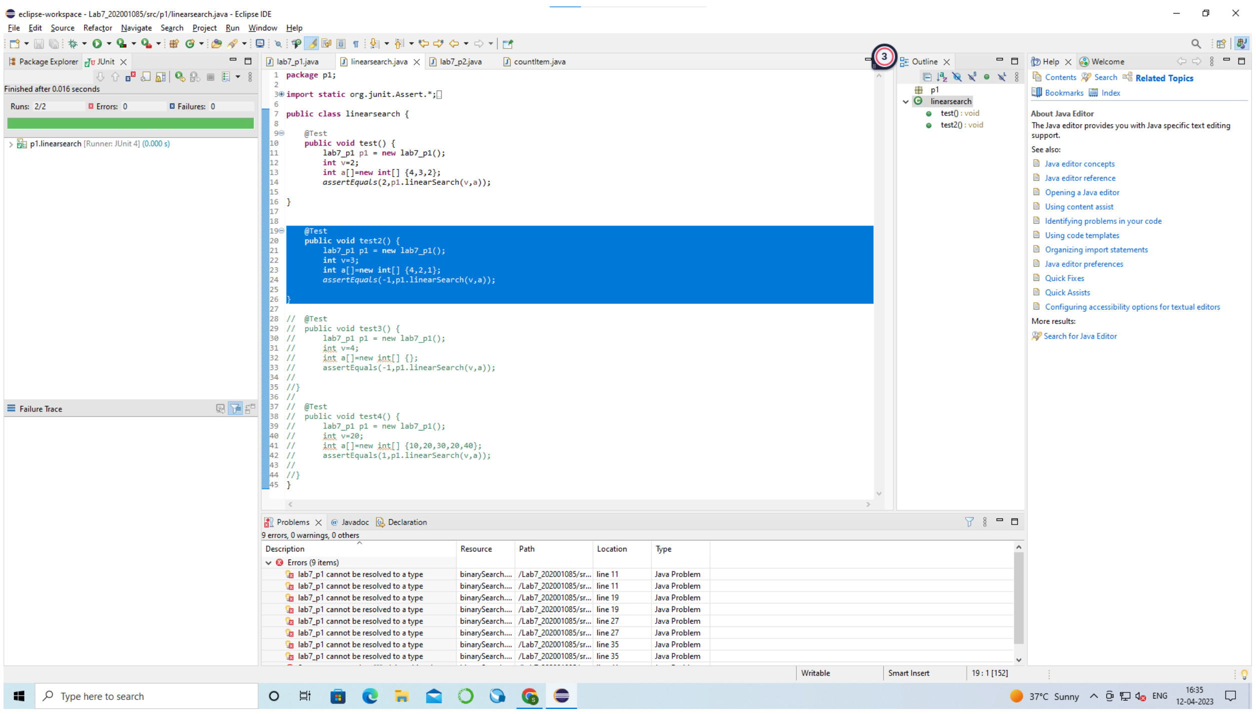
Problems: 9 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (9 items)				
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem

Writable | Smart Insert | 17 : 5 [2]

37°C Sunny 16:35 12-04-2023

(2) V = 2 , a={4,2,1} expected output = -1



eclipse-workspace - Lab7_202001085/src/p1/linearsearch.java - Eclipse IDE

JUnit: Finished after 0.016 seconds — Runs: 2/2 Errors: 0 Failures: 0

p1.linearsearch [Runner: JUnit 4] (0.000 s)

```java
package p1;

import static org.junit.Assert.*;

public class linearsearch {

    @Test
    public void test() {
        lab7_p1 p1 = new lab7_p1();
        int v=2;
        int a[]=new int[] {4,3,2};
        assertEquals(2,p1.linearSearch(v,a));

}


    @Test
    public void test2() {
        lab7_p1 p1 = new lab7_p1();
        int v=3;
        int a[]=new int[] {4,2,1};
        assertEquals(-1,p1.linearSearch(v,a));

}

//  @Test
//  public void test3() {
//      lab7_p1 p1 = new lab7_p1();
//      int v=4;
//      int a[]=new int[] {};
//      assertEquals(-1,p1.linearSearch(v,a));
//
//}
//
//  @Test
//  public void test4() {
//      lab7_p1 p1 = new lab7_p1();
//      int v=20;
//      int a[]=new int[] {10,20,30,20,40};
//      assertEquals(1,p1.linearSearch(v,a));
//
//}
}
```

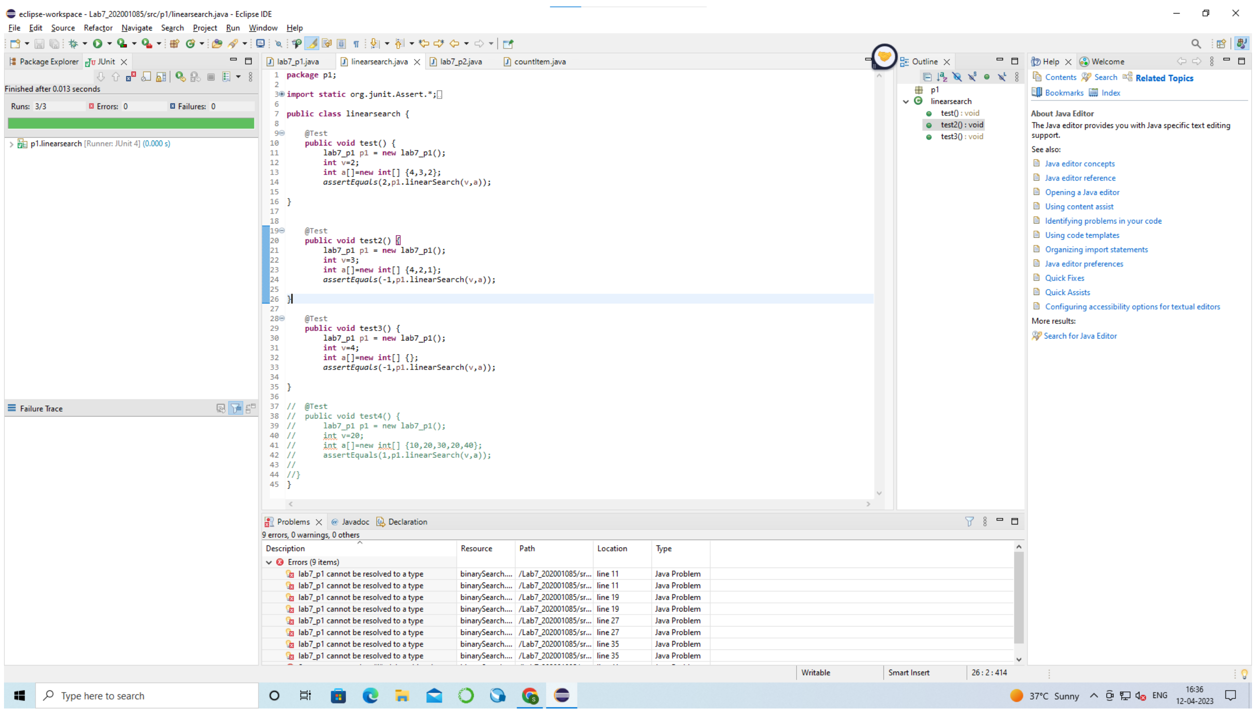
Problems: 9 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (9 items)				
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem

Writable | Smart Insert | 19 : 1 [152]

37°C Sunny 16:35 12-04-2023

(3) V = 2 , a={} expected output = -1



eclipse-workspace - Lab7_202001085/src/p1/linearsearch.java - Eclipse IDE

Finished after 0.013 seconds

Runs: 3/3 Errors: 0 Failures: 0

p1.linearsearch [Runner: JUnit 4] (0.000 s)

```java
package p1;

import static org.junit.Assert.*;

public class linearsearch {

    @Test
    public void test() {
        lab7_p1 p1 = new lab7_p1();
        int v=2;
        int a[]=new int[] {4,3,2};
        assertEquals(2,p1.linearSearch(v,a));

}


    @Test
    public void test2() {
        lab7_p1 p1 = new lab7_p1();
        int v=3;
        int a[]=new int[] {4,2,1};
        assertEquals(-1,p1.linearSearch(v,a));

}

    @Test
    public void test3() {
        lab7_p1 p1 = new lab7_p1();
        int v=4;
        int a[]=new int[] {};
        assertEquals(-1,p1.linearSearch(v,a));

}

//  @Test
//  public void test4() {
//      lab7_p1 p1 = new lab7_p1();
//      int v=20;
//      int a[]=new int[] {10,20,30,20,40};
//      assertEquals(1,p1.linearSearch(v,a));
//
//}
}
```

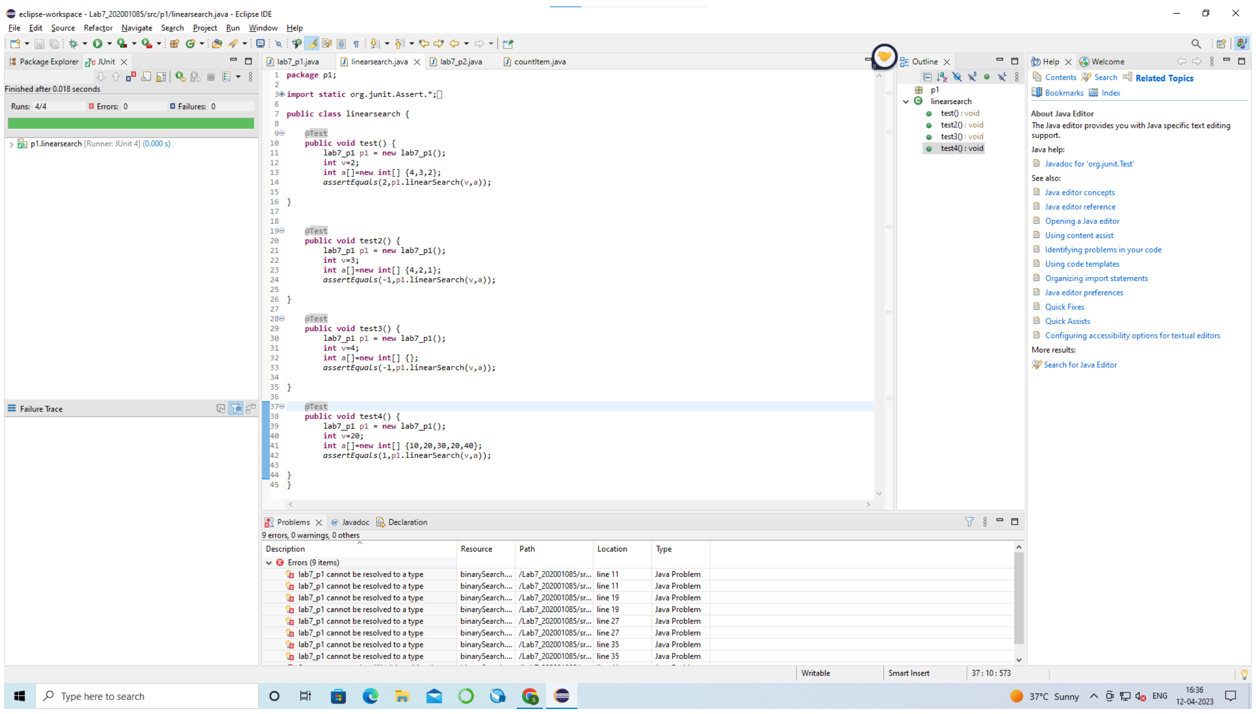
9 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (9 items)				
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem

Writable Smart Insert 26 : 2 : 414

37°C Sunny ENG 16:36 12-04-2023

(4) V = 2 , a={10,20,30,20,40} expected output = 1



eclipse-workspace - Lab7_202001085/src/p1/linearsearch.java - Eclipse IDE

Finished after 0.018 seconds

Runs: 4/4 Errors: 0 Failures: 0

p1.linearsearch [Runner: JUnit 4] (0.000 s)

```java
package p1;

import static org.junit.Assert.*;

public class linearsearch {

    @Test
    public void test() {
        lab7_p1 p1 = new lab7_p1();
        int v=2;
        int a[]=new int[] {4,3,2};
        assertEquals(2,p1.linearSearch(v,a));

}


    @Test
    public void test2() {
        lab7_p1 p1 = new lab7_p1();
        int v=3;
        int a[]=new int[] {4,2,1};
        assertEquals(-1,p1.linearSearch(v,a));

}

    @Test
    public void test3() {
        lab7_p1 p1 = new lab7_p1();
        int v=4;
        int a[]=new int[] {};
        assertEquals(-1,p1.linearSearch(v,a));

}

    @Test
    public void test4() {
        lab7_p1 p1 = new lab7_p1();
        int v=20;
        int a[]=new int[] {10,20,30,20,40};
        assertEquals(1,p1.linearSearch(v,a));

}
}
```

9 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (9 items)				
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem

Writable Smart Insert 37 : 10 : 573

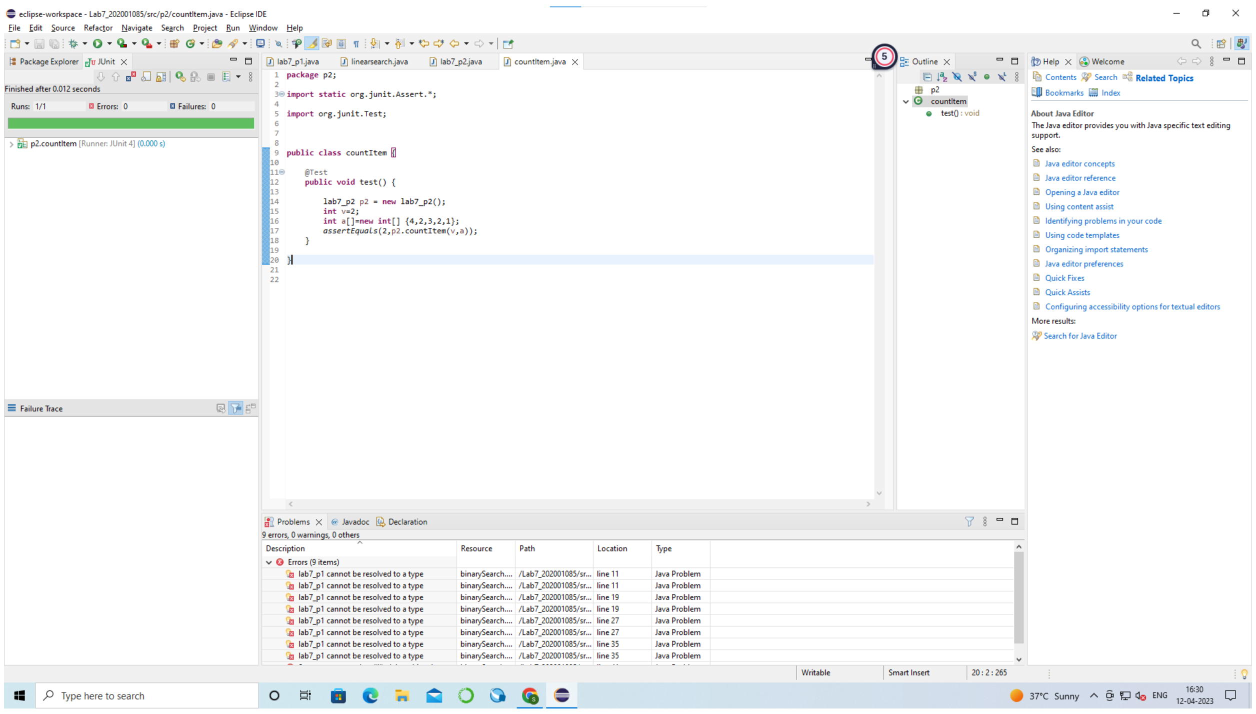
37°C Sunny ENG 16:36 12-04-2023

(P2). The function countItem returns the number of times a value v appears in an array of integers a.

```
int countItem(int v, int a[]){
    int count = 0;
    for (int i = 0; i < a.length; i++){
        if (a[i] == v)
        count++;
    }
    return (count);
}
```

Test Cases :

(1)   v=2 , a={4,2,3,2,1} expected output = 2


eclipse-workspace - Lab7_202001085/src/p2/countItem.java - Eclipse IDE

JUnit: Finished after 0.012 seconds

Runs: 1/1 Errors: 0 Failures: 0

p2.countItem [Runner: JUnit 4] (0.000 s)

```java
package p2;

import static org.junit.Assert.*;

import org.junit.Test;



public class countItem {

    @Test
    public void test() {

        lab7_p2 p2 = new lab7_p2();
        int v=2;
        int a[]=new int[] {4,2,3,2,1};
        assertEquals(2,p2.countItem(v,a));
    }

}
```

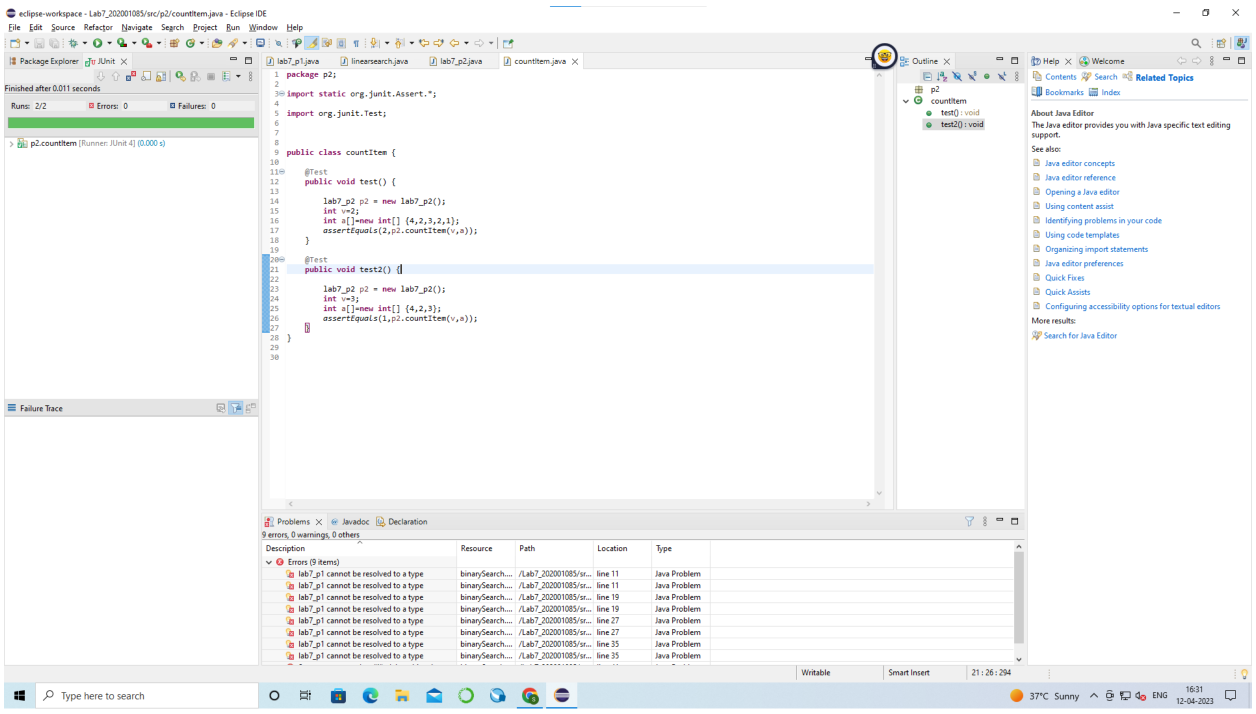
Problems: 9 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (9 items)				
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem

Writable | Smart Insert | 20 : 2 : 265

37°C Sunny ENG 16:30 12-04-2023

(2) v=3 , a={4,2,3} expected output =1


eclipse-workspace - Lab7_202001085/src/p2/countItem.java - Eclipse IDE

JUnit: Finished after 0.011 seconds

Runs: 2/2 Errors: 0 Failures: 0

p2.countItem [Runner: JUnit 4] (0.000 s)

```java
package p2;

import static org.junit.Assert.*;

import org.junit.Test;



public class countItem {

    @Test
    public void test() {

        lab7_p2 p2 = new lab7_p2();
        int v=2;
        int a[]=new int[] {4,2,3,2,1};
        assertEquals(2,p2.countItem(v,a));
    }

    @Test
    public void test2() {

        lab7_p2 p2 = new lab7_p2();
        int v=3;
        int a[]=new int[] {4,2,3};
        assertEquals(1,p2.countItem(v,a));
    }
}
```

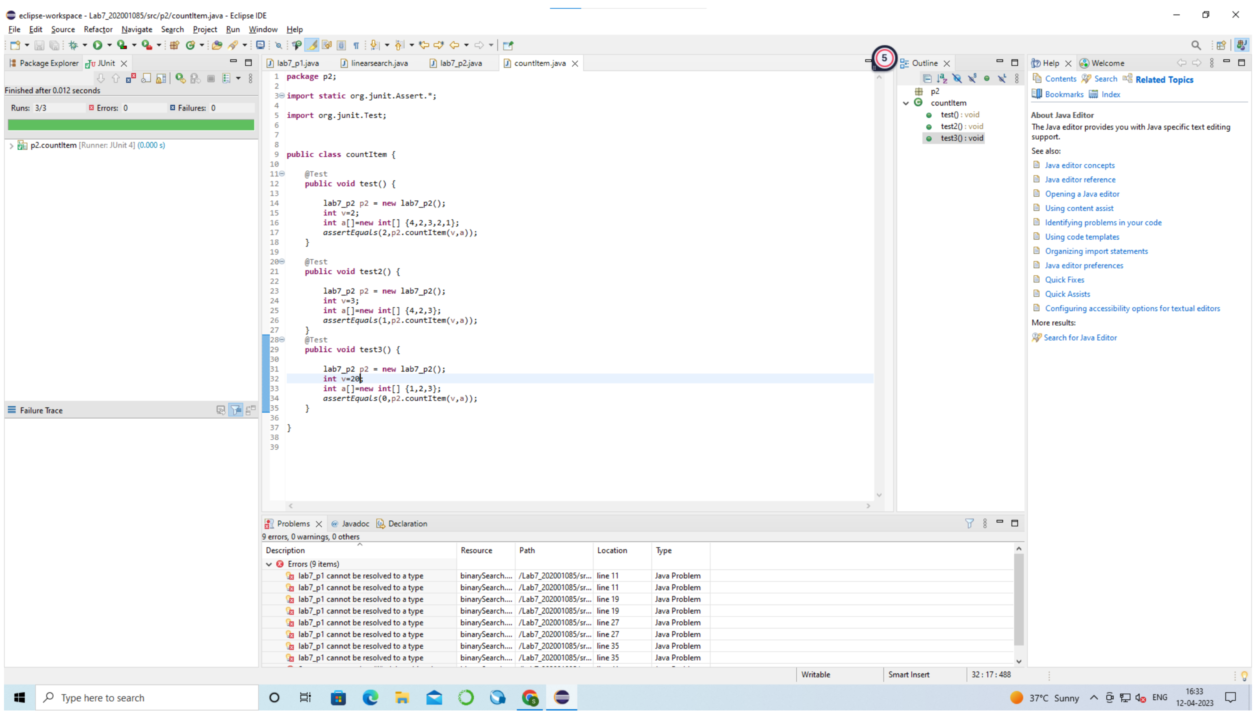
Problems: 9 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (9 items)				
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem

Writable | Smart Insert | 21 : 26 : 294

37°C Sunny ENG 16:31 12-04-2023

(3) v= 20 , a= {1,2,3} expected output =0



eclipse-workspace - Lab7_202001085/src/p2/countItem.java - Eclipse IDE

Finished after 0.012 seconds

Runs: 3/3 Errors: 0 Failures: 0

p2.countItem [Runner: JUnit 4] (0.000 s)

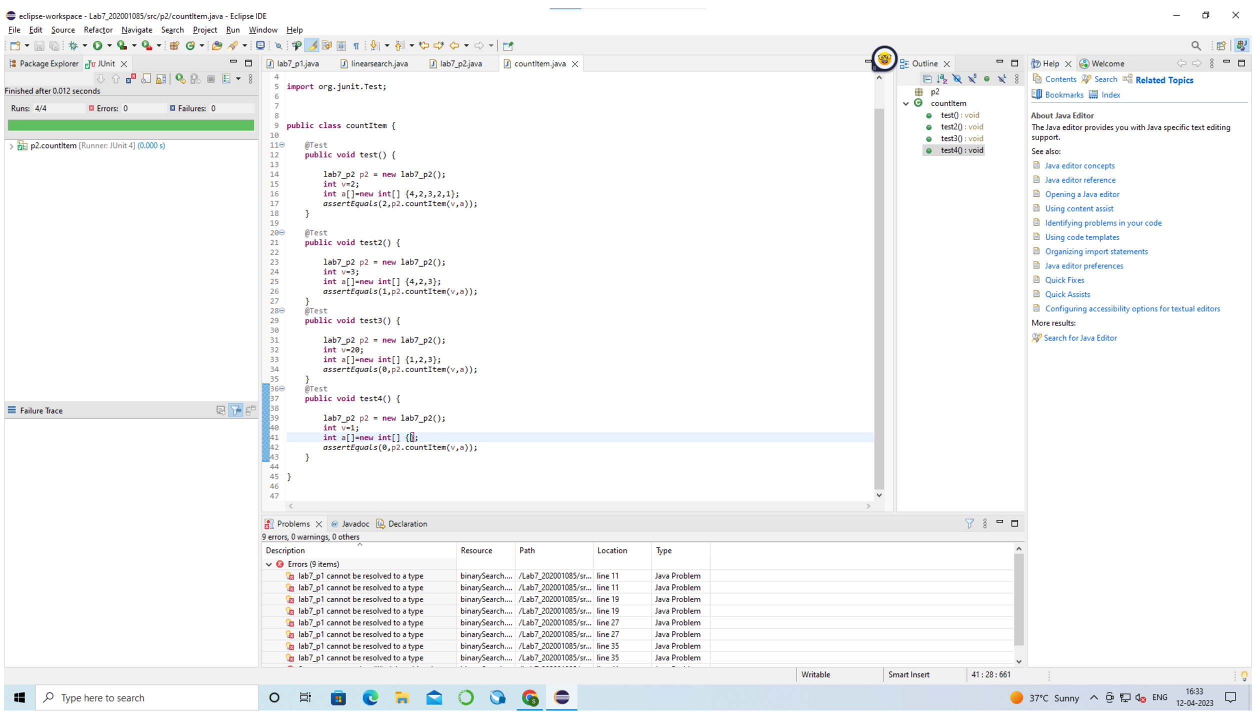
```java
package p2;

import static org.junit.Assert.*;

import org.junit.Test;



public class countItem {

    @Test
    public void test() {

        lab7_p2 p2 = new lab7_p2();
        int v=2;
        int a[]=new int[] {4,2,3,2,1};
        assertEquals(2,p2.countItem(v,a));
    }

    @Test
    public void test2() {

        lab7_p2 p2 = new lab7_p2();
        int v=3;
        int a[]=new int[] {4,2,3};
        assertEquals(1,p2.countItem(v,a));
    }
    @Test
    public void test3() {

        lab7_p2 p2 = new lab7_p2();
        int v=20;
        int a[]=new int[] {1,2,3};
        assertEquals(0,p2.countItem(v,a));
    }

}
```

9 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (9 items)				
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem

Writable Smart Insert 32 : 17 : 488

(4) v=1 , a ={} , expected output = 0


eclipse-workspace - Lab7_202001085/src/p2/countItem.java - Eclipse IDE

Finished after 0.012 seconds

Runs: 4/4 Errors: 0 Failures: 0

p2.countItem [Runner: JUnit 4] (0.000 s)

```java
import org.junit.Test;



public class countItem {

    @Test
    public void test() {

        lab7_p2 p2 = new lab7_p2();
        int v=2;
        int a[]=new int[] {4,2,3,2,1};
        assertEquals(2,p2.countItem(v,a));
    }

    @Test
    public void test2() {

        lab7_p2 p2 = new lab7_p2();
        int v=3;
        int a[]=new int[] {4,2,3};
        assertEquals(1,p2.countItem(v,a));
    }
    @Test
    public void test3() {

        lab7_p2 p2 = new lab7_p2();
        int v=20;
        int a[]=new int[] {1,2,3};
        assertEquals(0,p2.countItem(v,a));
    }
    @Test
    public void test4() {

        lab7_p2 p2 = new lab7_p2();
        int v=1;
        int a[]=new int[] {};
        assertEquals(0,p2.countItem(v,a));
    }

}
```

9 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (9 items)				
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 11	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 19	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 27	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem
lab7_p1 cannot be resolved to a type	binarySearch....	/Lab7_202001085/sr...	line 35	Java Problem

Writable Smart Insert 41 : 28 : 661

(P3). The function binarySearch searches for a value v in an ordered array of integers a. If v appears in the array a, then the function returns an index i, such that a[i] == v; otherwise, -1 is returned.

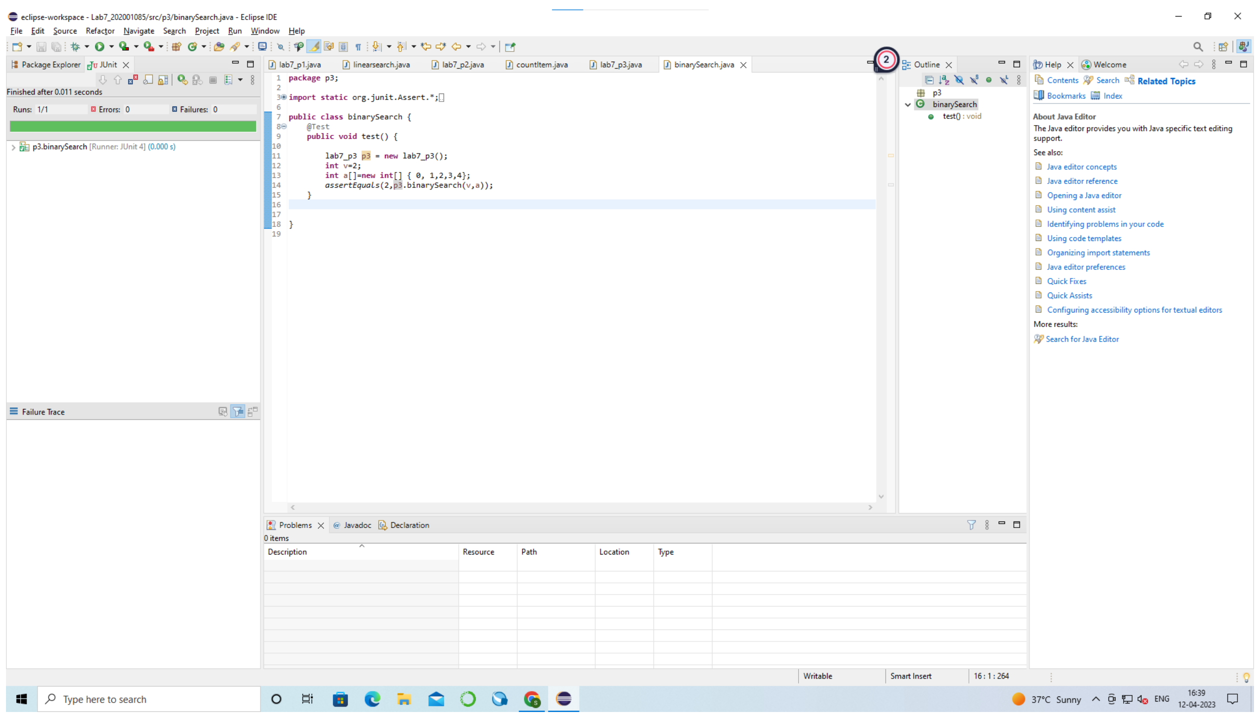
Assumption: the elements in the array a are sorted in non-decreasing order.

```
int binarySearch(int v, int a[]){
    int lo,mid,hi;
    lo = 0;
    hi = a.length-1;
    while (lo <= hi){
        mid = (lo+hi)/2;
        if (v == a[mid])
            return (mid);
        else if (v < a[mid])
```

```
            hi = mid-1;
        else
            lo = mid+1;
    }
    return(-1);
}
```

Test Cases :

(1) v=2 , a= { 0, 1,2,3,4} expected output = 2



eclipse-workspace - Lab7_202001085/src/p3/binarySearch.java - Eclipse IDE

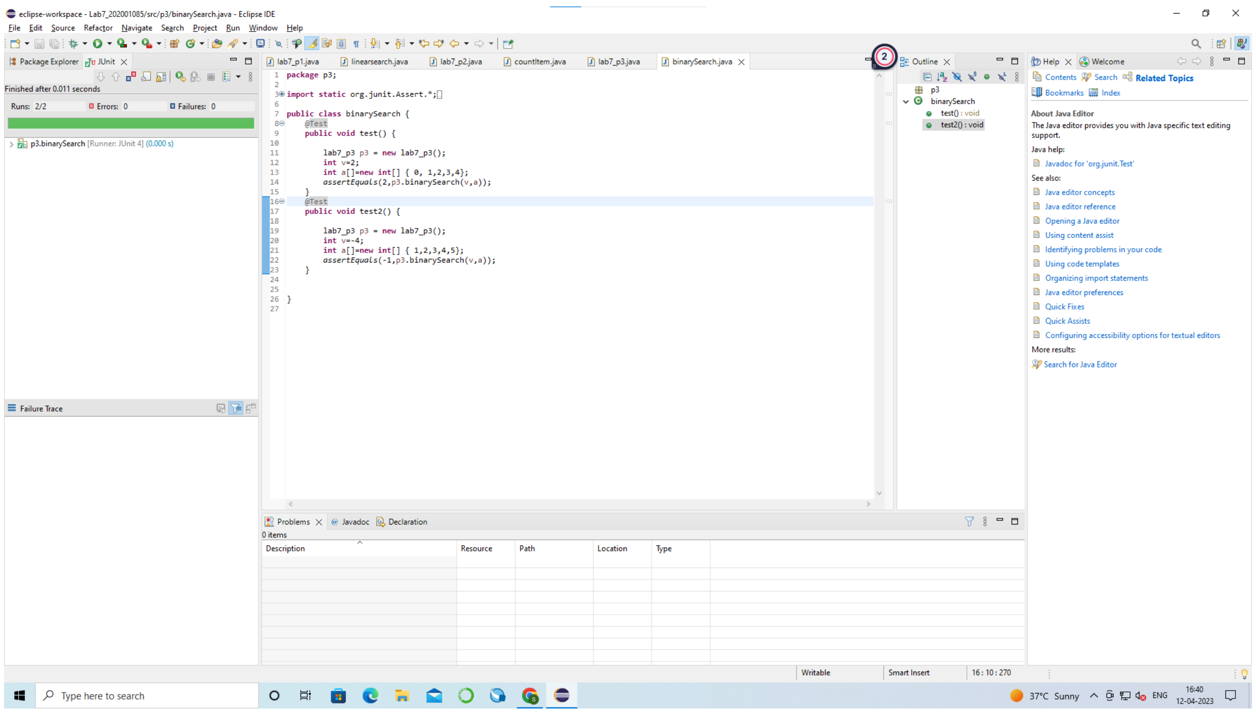
Finished after 0.011 seconds

Runs: 1/1 Errors: 0 Failures: 0

p3.binarySearch [Runner: JUnit 4] (0.000 s)

```java
package p3;

import static org.junit.Assert.*;

public class binarySearch {
    @Test
    public void test() {

        lab7_p3 p3 = new lab7_p3();
        int v=2;
        int a[]=new int[] { 0, 1,2,3,4};
        assertEquals(2,p3.binarySearch(v,a));
    }


}
```

(2) v= -4 , a= {1,2,3,4,5 } expected output = -1



eclipse-workspace - Lab7_202001085/src/p3/binarySearch.java - Eclipse IDE

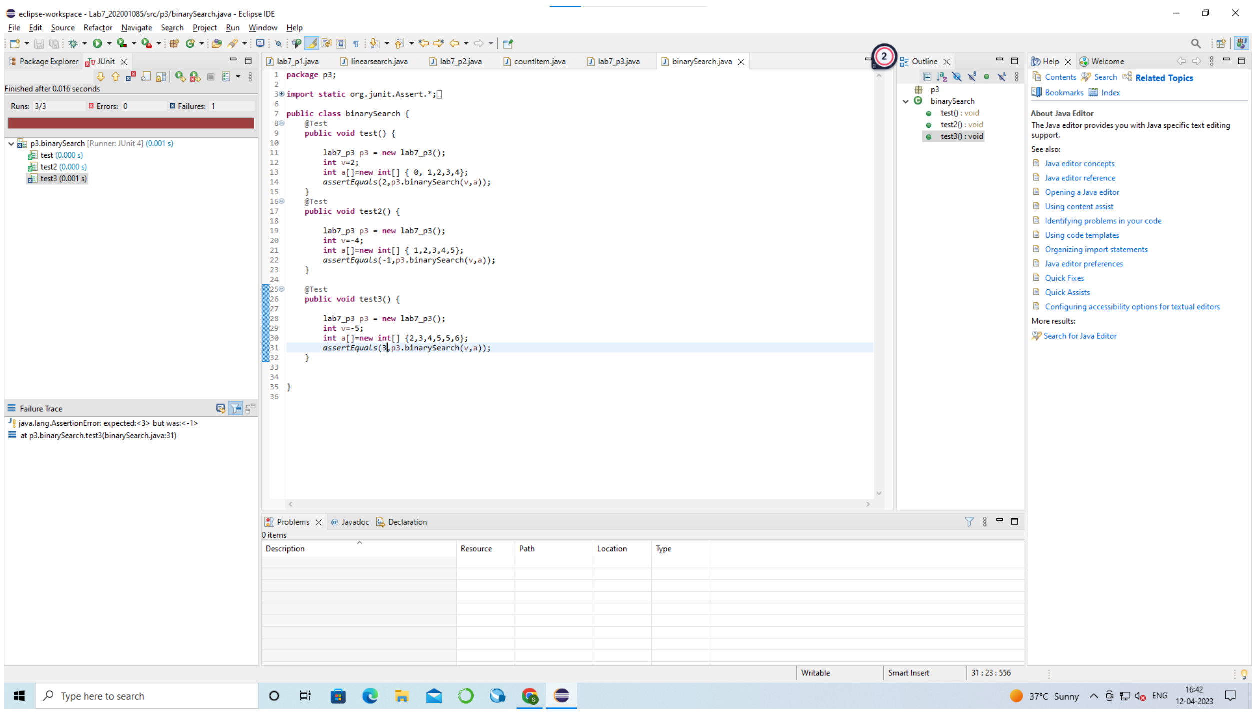
Finished after 0.011 seconds

Runs: 2/2 Errors: 0 Failures: 0

p3.binarySearch [Runner: JUnit 4] (0.000 s)

```java
package p3;

import static org.junit.Assert.*;

public class binarySearch {
    @Test
    public void test() {

        lab7_p3 p3 = new lab7_p3();
        int v=2;
        int a[]=new int[] { 0, 1,2,3,4};
        assertEquals(2,p3.binarySearch(v,a));
    }
    @Test
    public void test2() {

        lab7_p3 p3 = new lab7_p3();
        int v=-4;
        int a[]=new int[] { 1,2,3,4,5};
        assertEquals(-1,p3.binarySearch(v,a));
    }


}
```

(3) v=5 , a= {2,3,4,5,5,6} expected output = 3 or 4



eclipse-workspace - Lab7_202001085/src/p3/binarySearch.java - Eclipse IDE

Finished after 0.016 seconds

Runs: 3/3 Errors: 0 Failures: 1

```
p3.binarySearch [Runner: JUnit 4] (0.001 s)
    test (0.000 s)
    test2 (0.000 s)
    test3 (0.001 s)
```

```java
package p3;

import static org.junit.Assert.*;

public class binarySearch {
    @Test
    public void test() {

        lab7_p3 p3 = new lab7_p3();
        int v=2;
        int a[]=new int[] { 0, 1,2,3,4};
        assertEquals(2,p3.binarySearch(v,a));
    }
    @Test
    public void test2() {

        lab7_p3 p3 = new lab7_p3();
        int v=-4;
        int a[]=new int[] { 1,2,3,4,5};
        assertEquals(-1,p3.binarySearch(v,a));
    }

    @Test
    public void test3() {

        lab7_p3 p3 = new lab7_p3();
        int v=-5;
        int a[]=new int[] {2,3,4,5,5,6};
        assertEquals(3,p3.binarySearch(v,a));
    }


}
```

Failure Trace

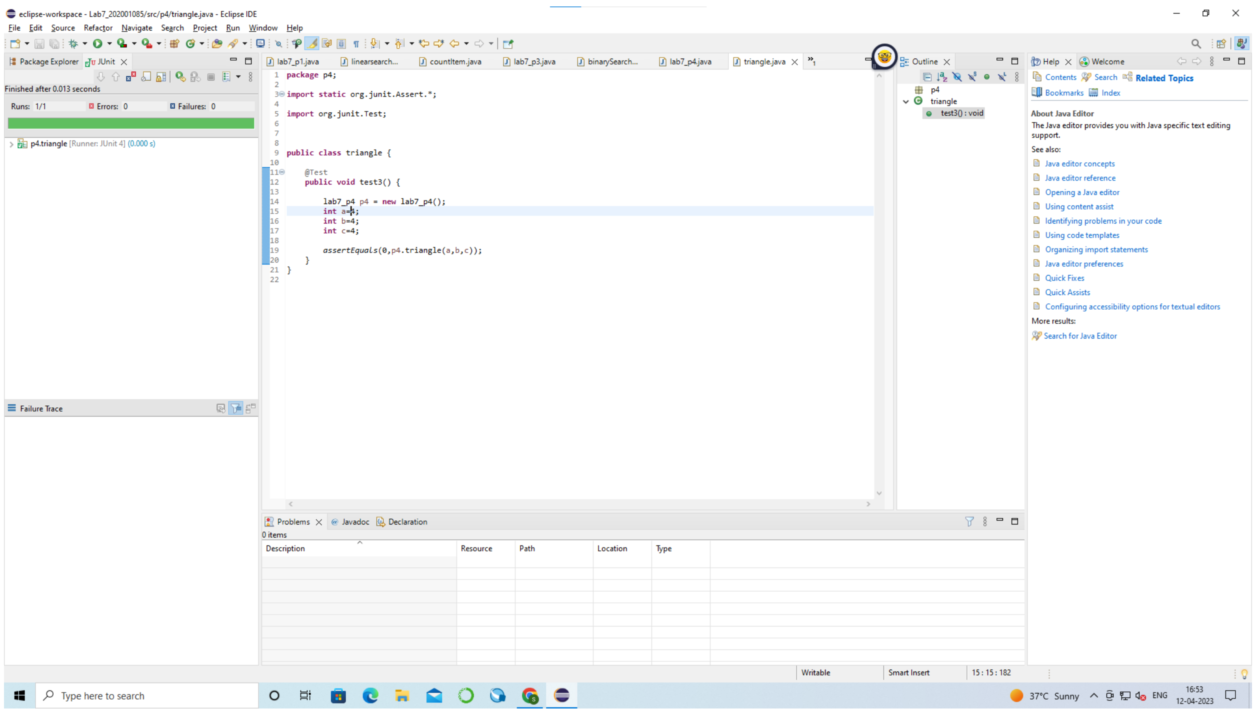
```
java.lang.AssertionError: expected:<3> but was:<-1>
at p3.binarySearch.test3(binarySearch.java:31)
```

(P4). The following problem has been adapted from The Art of Software Testing, by G. Myers (1979). The
function triangle takes three integer parameters that are interpreted as the lengths of the sides of a
triangle. It returns whether the triangle is equilateral (three lengths equal), isosceles (two lengths equal),
scalene (no lengths equal), or invalid (impossible lengths).

```
final int EQUILATERAL = 0;
final int ISOSCELES = 1;
final int SCALENE = 2;
final int INVALID = 3;

int triangle(int a, int b, int c){
    if (a >= b+c || b >= a+c || c >= a+b)
        return(INVALID);
    if (a == b && b == c)
        return(EQUILATERAL);
    if (a == b || a == c || b == c)
        return(ISOSCELES);
return(SCALENE);
}
```

Test Cases :

(1) a=4,b=4,c=4 expected output = EQUILATERAL

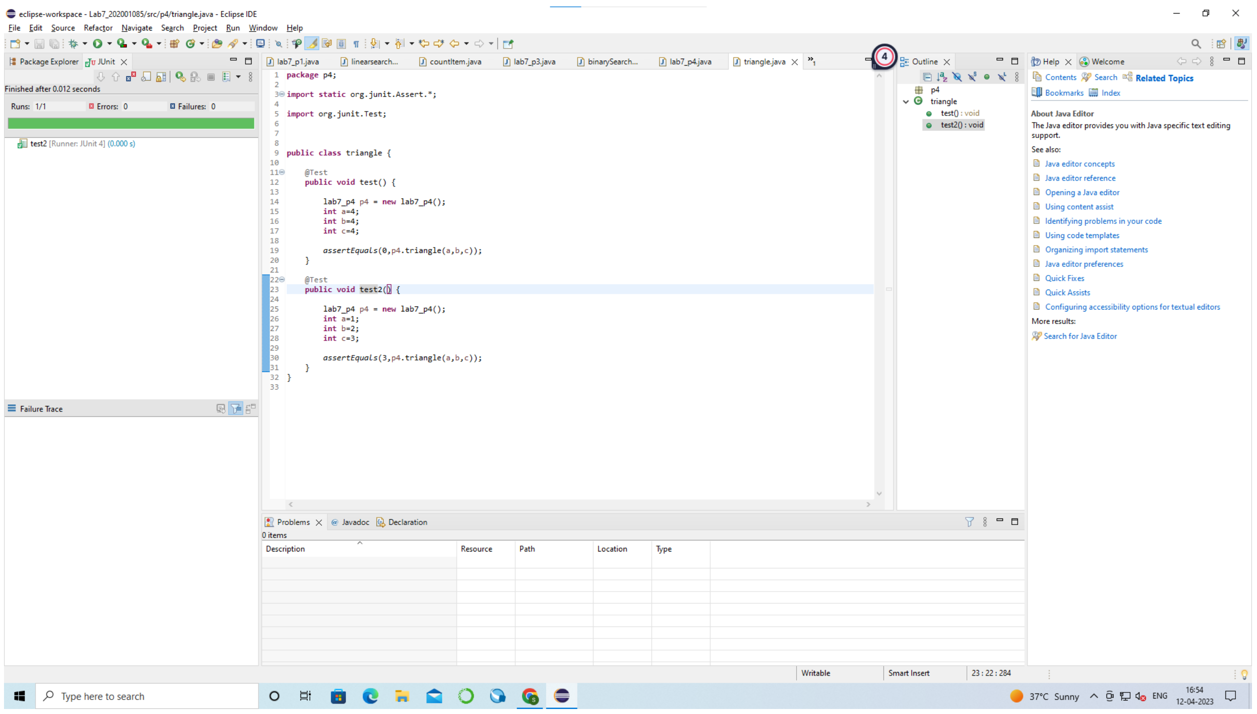


eclipse-workspace - Lab7_202001085/src/p4/triangle.java - Eclipse IDE

JUnit: Finished after 0.013 seconds — Runs: 1/1 Errors: 0 Failures: 0 — p4.triangle [Runner: JUnit 4] (0.000 s)

```java
package p4;

import static org.junit.Assert.*;

import org.junit.Test;



public class triangle {

    @Test
    public void test3() {

        lab7_p4 p4 = new lab7_p4();
        int a=4;
        int b=4;
        int c=4;

        assertEquals(0,p4.triangle(a,b,c));
    }
}
```

(2) a=1,b=2,c=3 expected output= INVALID

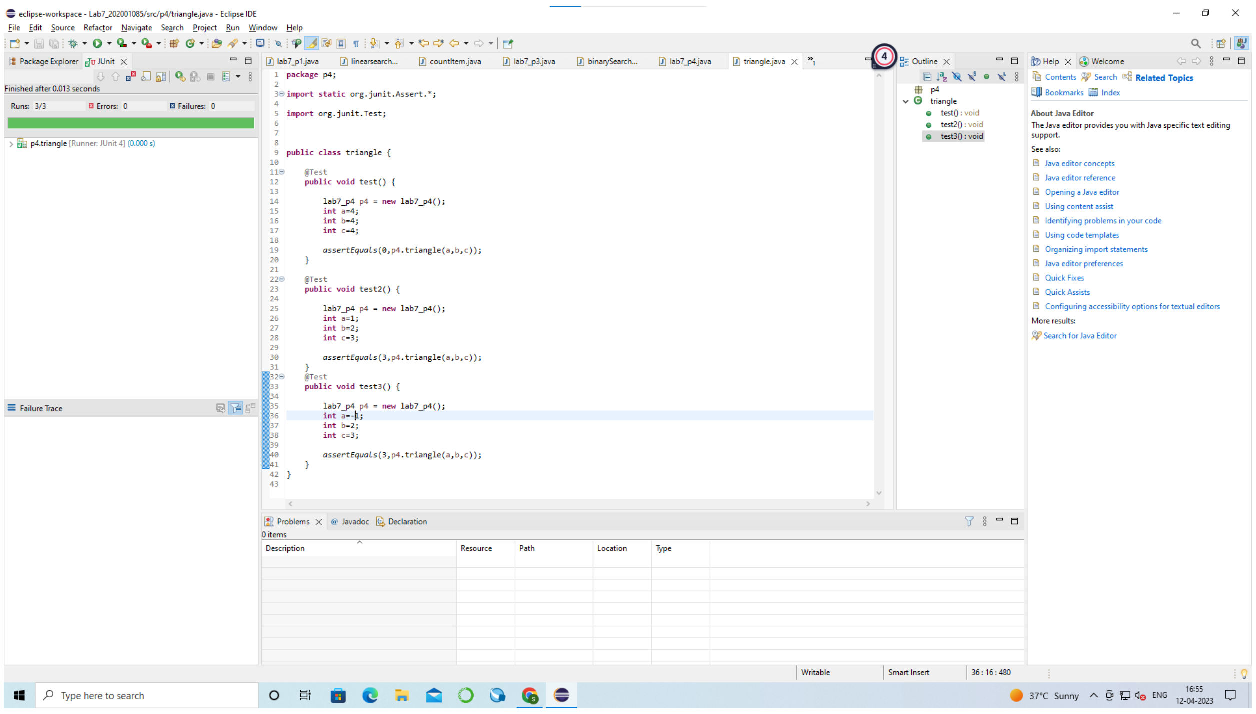


eclipse-workspace - Lab7_202001085/src/p4/triangle.java - Eclipse IDE

JUnit: Finished after 0.012 seconds — Runs: 1/1 Errors: 0 Failures: 0 — test2 [Runner: JUnit 4] (0.000 s)

```java
package p4;

import static org.junit.Assert.*;

import org.junit.Test;



public class triangle {

    @Test
    public void test() {

        lab7_p4 p4 = new lab7_p4();
        int a=4;
        int b=4;
        int c=4;

        assertEquals(0,p4.triangle(a,b,c));
    }

    @Test
    public void test2() {

        lab7_p4 p4 = new lab7_p4();
        int a=1;
        int b=2;
        int c=3;

        assertEquals(3,p4.triangle(a,b,c));
    }
}
```

(3) a=-1,b=2,c=3 expected output = INVALID



eclipse-workspace - Lab7_202001085/src/p4/triangle.java - Eclipse IDE

Finished after 0.013 seconds

Runs: 3/3 Errors: 0 Failures: 0

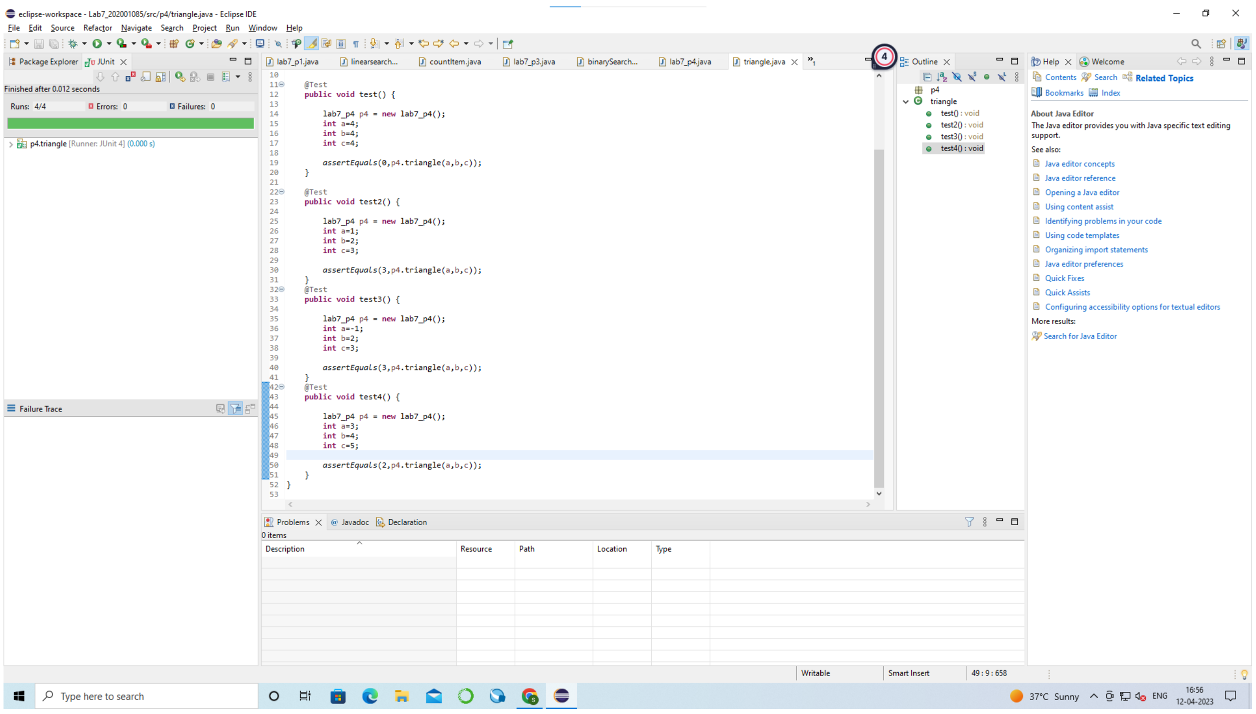
p4.triangle [Runner: JUnit 4] (0.000 s)

```java
package p4;

import static org.junit.Assert.*;

import org.junit.Test;



public class triangle {

    @Test
    public void test() {

        lab7_p4 p4 = new lab7_p4();
        int a=4;
        int b=4;
        int c=4;

        assertEquals(0,p4.triangle(a,b,c));
    }

    @Test
    public void test2() {

        lab7_p4 p4 = new lab7_p4();
        int a=1;
        int b=2;
        int c=3;

        assertEquals(3,p4.triangle(a,b,c));
    }
    @Test
    public void test3() {

        lab7_p4 p4 = new lab7_p4();
        int a=-1;
        int b=2;
        int c=3;

        assertEquals(3,p4.triangle(a,b,c));
    }
}
```

Problems: 0 items

Writable | Smart Insert | 36 : 16 : 480

(4) a=3,b=4,c=5 expected output = SCALENE



eclipse-workspace - Lab7_202001085/src/p4/triangle.java - Eclipse IDE

Finished after 0.012 seconds

Runs: 4/4 Errors: 0 Failures: 0

p4.triangle [Runner: JUnit 4] (0.000 s)

```java
    @Test
    public void test() {

        lab7_p4 p4 = new lab7_p4();
        int a=4;
        int b=4;
        int c=4;

        assertEquals(0,p4.triangle(a,b,c));
    }

    @Test
    public void test2() {

        lab7_p4 p4 = new lab7_p4();
        int a=1;
        int b=2;
        int c=3;

        assertEquals(3,p4.triangle(a,b,c));
    }
    @Test
    public void test3() {

        lab7_p4 p4 = new lab7_p4();
        int a=-1;
        int b=2;
        int c=3;

        assertEquals(3,p4.triangle(a,b,c));
    }
    @Test
    public void test4() {

        lab7_p4 p4 = new lab7_p4();
        int a=3;
        int b=4;
        int c=5;

        assertEquals(2,p4.triangle(a,b,c));
    }
}
```

Problems: 0 items

Writable | Smart Insert | 49 : 9 : 658

(P6). Consider again the triangle classification program (P4) with a slightly different specification: The program
reads floating values from the standard input. The three values A, B, and C are interpreted as
representing the lengths of the sides of a triangle. The program then prints a message to the standard output
that states whether the triangle, if it can be formed, is scalene, isosceles, equilateral, or right angled.
Determine the following for the above program:

a) Identify the equivalence classes for the system

Equivalence Classes:
EC1: All sides are positive, real numbers.
EC2: One or more sides are negative or zero.
EC3: The sum of the lengths of any two sides is less than or equal to the length of the remaining side (impossible lengths).
EC4: The sum of the lengths of any two sides is greater than the length of the remaining side (possible lengths).

b) Identify test cases to cover the identified equivalence classes. Also, explicitly mention which test case would cover which equivalence class.
(Hint: you must need to be ensure that the identified set of test cases cover all identified equivalence classes)

Test cases:
TC1 (EC1): A=3, B=4, C=5 (right-angled triangle)
TC2 (EC1): A=5, B=5, C=5 (equilateral triangle)
TC3 (EC1): A=5, B=6, C=7 (scalene triangle)
TC4 (EC1): A=5, B=5, C=7 (isosceles triangle)

TC5 (EC2): A=-2, B=4, C=5 (invalid input)
TC6 (EC2): A=0, B=4, C=5 (invalid input)

c) For the boundary condition A + B > C case (scalene triangle), identify test cases to verify the boundary.

Test cases for the boundary condition A + B > C:
TC7 (EC4): A=2, B=3, C=6 (sum of A and B is equal to C)

d) For the boundary condition A = C case (isosceles triangle), identify test cases to verify the boundary.

Test cases for the boundary condition A = C:
TC8 (EC4): A=5, B=6, C=5 (A equals to C)

e) For the boundary condition A = B = C case (equilateral triangle), identify test cases to verify the boundary.

Test cases for the boundary condition A = B = C:
TC9 (EC4): A=1, B=1, C=1 (all sides are equal)

f) For the boundary condition A2 + B2 = C2 case (right-angle triangle), identify test cases to verify the boundary.

Test cases for the boundary condition A^2 + B^2 = C^2:
TC10 (EC4): A=3, B=4, C=5 (right-angled triangle)

g) For the non-triangle case, identify test cases to explore the boundary.

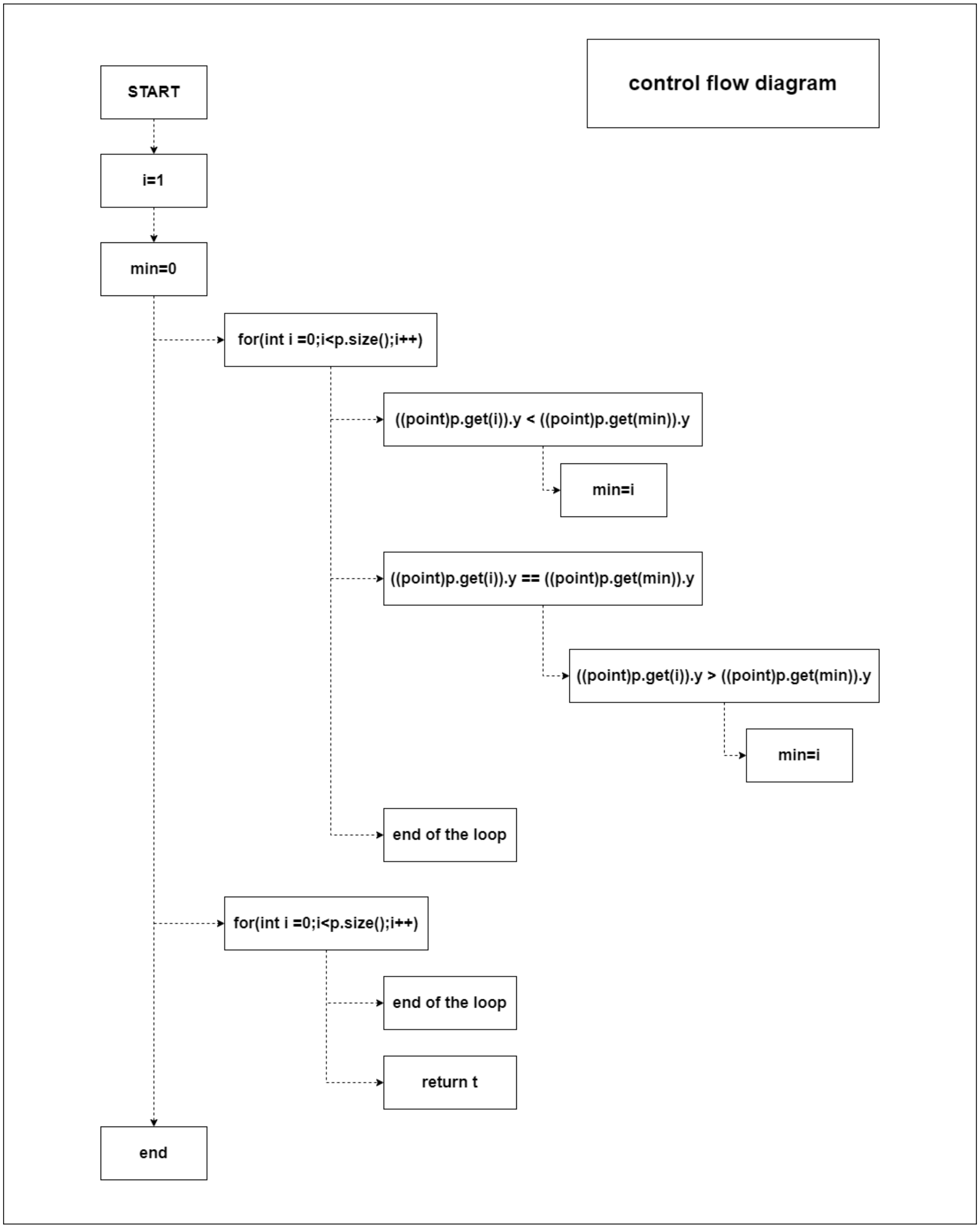
Test cases for the non-triangle case:
TC11 (EC3): A=2, B=2, C=4 (sum of A and B is less than C)

h) For non-positive input, identify test points.

Test points for non-positive input:
TP1 (EC2): A=0, B=4, C=5 (invalid input)
TP2 (EC2): A=-2, B=4, C=5 (invalid input)
Note: Test cases TC1 to TC10 covers all identified equivalence classes.

Section B :

1. **Convert the Java code comprising the beginning of the doGraham method into a control flow graph (CFG).**


control flow diagram
START
i=1
min=0
for(int i =0;i<p.size();i++)
((point)p.get(i)).y < ((point)p.get(min)).y
min=i
((point)p.get(i)).y == ((point)p.get(min)).y
((point)p.get(i)).y > ((point)p.get(min)).y
min=i
end of the loop
for(int i =0;i<p.size();i++)
end of the loop
return t
end

2. Test sets:

a) Statement coverage test sets: To achieve statement coverage, we need to make sure that every statement in the code is executed at least once.

Test 1: p = empty vector
Test 2: p = vector with one point
Test 3: p = vector with two points with the same y component
Test 4: p = vector with two points with different y components
Test 5: p = vector with three or more points with different y components
Test 6: p = vector with three or more points with the same y component

b) Branch coverage test sets: To achieve branch coverage, we need to make sure that every possible branch in the code is taken at least once

Test 1: p = empty vector
Test 2: p = vector with one point
Test 3: p = vector with two points with the same y component
Test 4: p = vector with two points with different y components
Test 5: p = vector with three or more points with different y components, and none of them have the same x component
Test 6: p = vector with three or more points with the same y component, and some of them have the same x component
Test 7: p = vector with three or more points with the same y component, and all of them have the same x component

c) Basic condition coverage test sets: To achieve basic condition coverage, we need to make sure that every basic condition in the

code (i.e., every Boolean subexpression) is evaluated as both true and false at least once

Test 1: p = empty vector
Test 2: p = vector with one point
Test 3: p = vector with two points with the same y component, and the first point has a smaller x component
Test 4: p = vector with two points with the same y component, and the second point has a smaller x component
Test 5: p = vector with two points with different y components
Test 6: p = vector with three or more points with different y components, and none of them have the same x component
Test 7: p = vector with three or more points with the same y component, and some of them have the same x component
Test 8: p = vector with three or more points with the same y component, and all of them have the same x component.