Lab 7: Software Testing

Name: Priya Vinda   ID: 202001261

Section A

Consider a program for determining the previous date. Its input is triple of day, month and year with the following ranges 1 <= month <= 12, 1 <= day <= 31, 1900 <= year <= 2015.The possible output dates would be previous date or invalid date. Design the equivalence class test cases?

Equivalence classes:

1. Valid dates: (Dates from (1, 1, 1900) to (31, 12, 2015) such that day, month and year are in the given ranges. ex: (13, 2, 2002), (3, 7, 2010), (21, 12, 1995) etc.)

2. Invalid dates: The input triple (day, month, year) that represents an invalid date, such as (30, 2, 2002) or (29, 2, 1900).

3. Out of range dates: e (Dates from (1, 1, 1900) to (31, 12, 2015) such that day, month and year are not in the given ranges. ex: (33, 6, 2020), (1,16,2033), (15,0,2022) etc.)

Based on these equivalence classes, we can design the following test cases to calculate previous date

Valid Dates:

Input	Expected Output
30-07-2003	29-07-2003
01-01-2011	31-12-2010
29-02-2000	28-02-2000

Invalid Dates:

Input	Expected Output
31-06-2003	Invalid
30-02-2000	Invalid
29-02-2010	Invalid

Out of Range Dates:

Input	Expected Output
32-07-2003	Invalid
01-30-2011	Invalid

Boundary value analysis:

1 Earliest date: 1-1-1900
2. Last possible date: 31-12-2015
3. Leap year: 29-2-2000
4. Invalid leap year date: 29-2-2001

5. Previous of earliest date: 31-12-1899
6. A day after last possible date: 1-1-2016

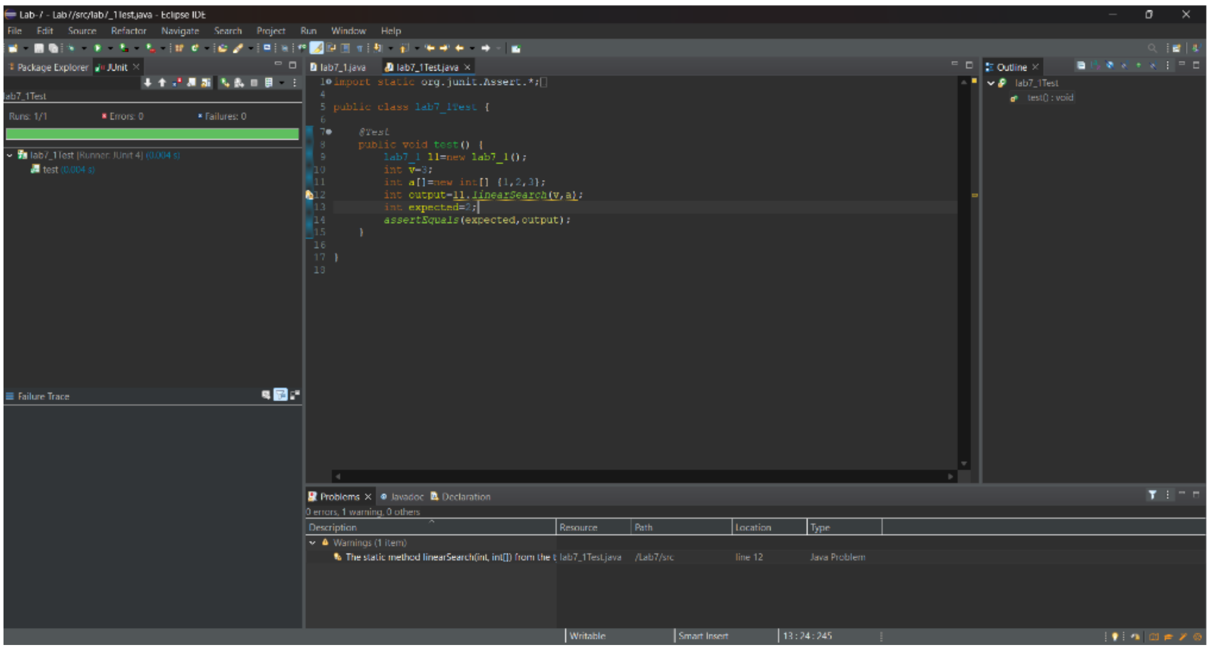
Boundary test cases:

Input	Output
01-01-1900	Invalid
31-12-2015	30-12-2015
29-02-2000	28-02-2000
29-02-2001	Invalid
31-01-2000	30-012000

Program 1: 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.

Text Action and Input Data	Expected Output
Equivalence Partitioning	
a = [2,3,4,6,5], v= 5	4
a = [6,5,4,3], v= 2	-1
a = [], v=2	-1
Boundary Value Analysis	
Minimum length array a=[], v=4	-1
Maximum length array a=[2,3,4,5,6,7,8,9,0], v=22	-1
Minimum value of v a=[0,1,2,3], v=0	0
Maximum value of v a=[1,2,3,4,5,6,7,8,9], v=9	8

Test case passed:



Runs: 1/1   Errors: 0   Failures: 0

lab7_1Test [Runner: JUnit 4] (0.004 s)
test (0.004 s)

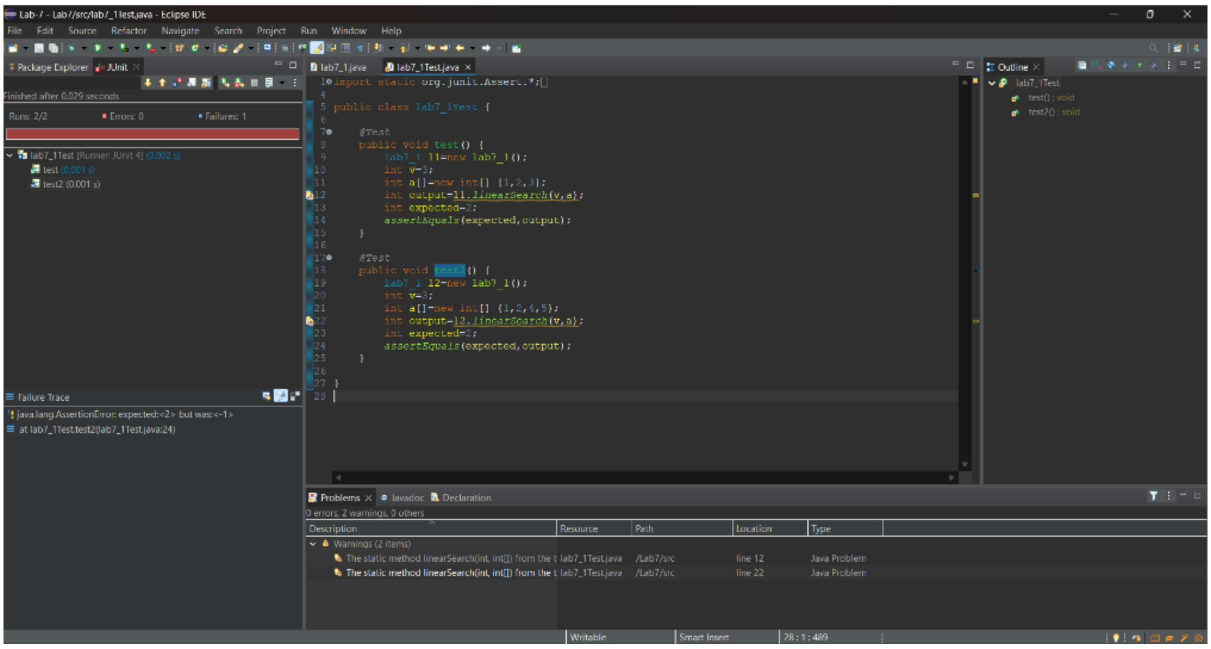
```java
import static org.junit.Assert.*;

public class lab7_1Test {

    @Test
    public void test() {
        lab7_1 l1=new lab7_1();
        int v=3;
        int a[]=new int[] {1,2,3};
        int output=l1.linearSearch(v,a);
        int expected=2;
        assertEquals(expected,output);
    }

}
```

0 errors, 1 warning, 0 others

Description	Resource	Path	Location	Type
Warnings (1 item)				
The static method linearSearch(int, int[]) from the t	lab7_1Test.java	/Lab7/src	line 12	Java Problem

Input: A=[1,2,3] v=3   Output: 2

Test case failed:



Finished after 0.029 seconds

Runs: 2/2   Errors: 0   Failures: 1

lab7_1Test [Runner: JUnit 4] (0.002 s)
test (0.001 s)
test2 (0.001 s)

```
java.lang.AssertionError: expected:<2> but was:<-1>
at lab7_1Test.test2(lab7_1Test.java:24)
```

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

public class lab7_1Test {

    @Test
    public void test() {
        lab7_1 l1=new lab7_1();
        int v=3;
        int a[]=new int[] {1,2,3};
        int output=l1.linearSearch(v,a);
        int expected=2;
        assertEquals(expected,output);
    }

    @Test
    public void test2() {
        lab7_1 l2=new lab7_1();
        int v=3;
        int a[]=new int[] {1,2,4,5};
        int output=l2.linearSearch(v,a);
        int expected=2;
        assertEquals(expected,output);
    }

}
```

0 errors, 2 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (2 items)				
The static method linearSearch(int, int[]) from the t	lab7_1Test.java	/Lab7/src	line 12	Java Problem
The static method linearSearch(int, int[]) from the t	lab7_1Test.java	/Lab7/src	line 22	Java Problem

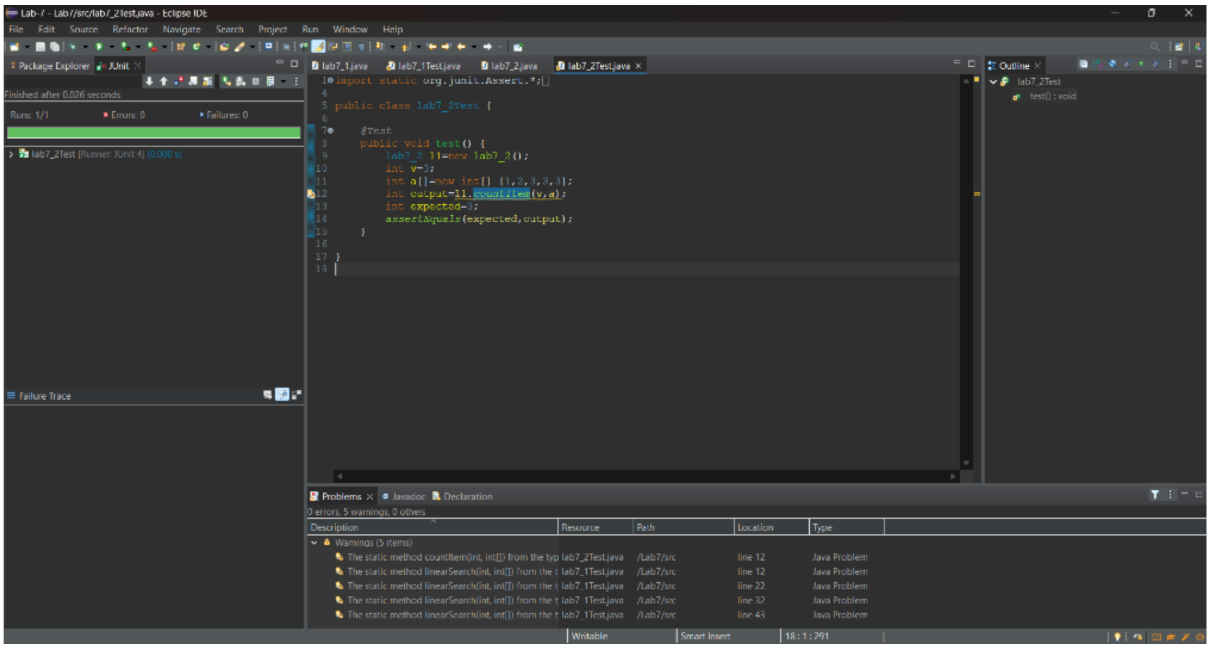
Input: A=[1,2,4,5] v=3   Output: 2

Program 2: The function countItem returns the number of times a value v appears in an array of integers a.

Text Action and Input Data	Expected Output

Equivalence Partitioning	
a = [1,2,3,4,2], v= 2	2
a = [6,5,4,3], v= 2	0
a = [], v=2	0
Boundary Value Analysis	
Minimum length array a =[] v=4	0
Maximum length array a = [2,2,2,2,2,2], v=2	6

Test Case passed:


Lab-7 - Lab7/src/lab7_2Test.java - Eclipse IDE

Finished after 0.026 seconds

Runs: 1/1 Errors: 0 Failures: 0

lab7_2Test [Runner: JUnit 4] (0.000 s)

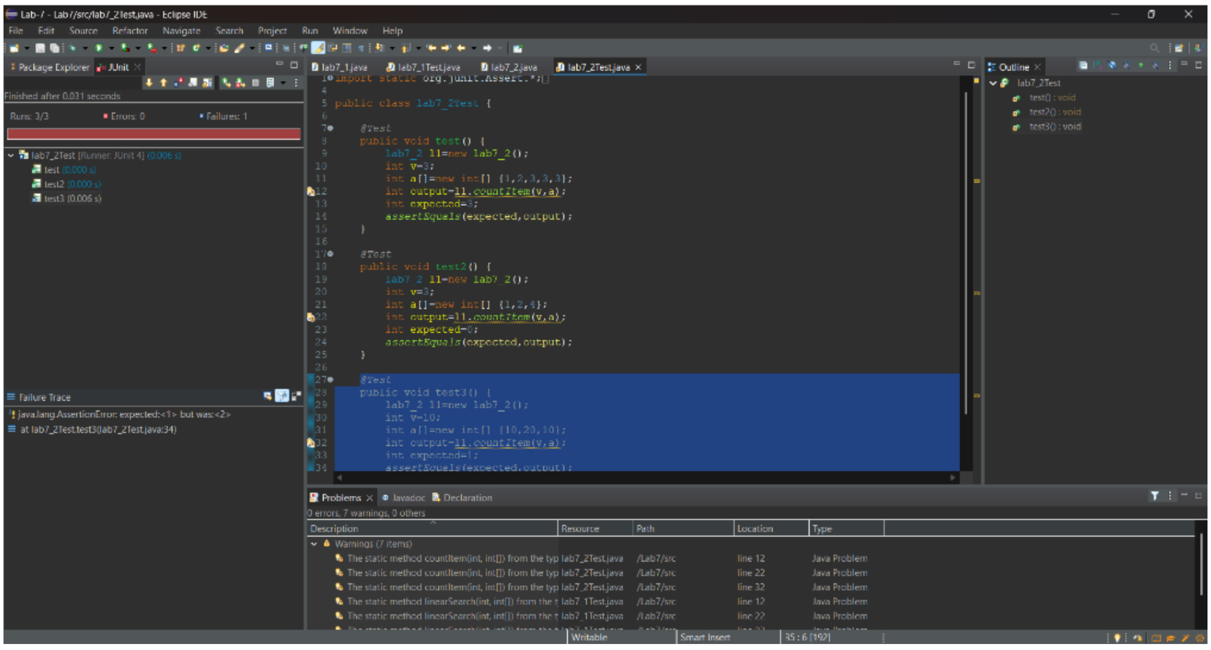
```java
import static org.junit.Assert.*;

public class lab7_2Test {

    @Test
    public void test() {
        lab7_2 l1=new lab7_2();
        int v=3;
        int a[]=new int[] {1,2,3,3,3};
        int output=l1.countItem(v,a);
        int expected=3;
        assertEquals(expected,output);
    }

}
```

```
0 errors, 5 warnings, 0 others
Description                                              Resource          Path        Location   Type
Warnings (5 items)
The static method countItem(int, int[]) from the typ     lab7_2Test.java   /Lab7/src   line 12    Java Problem
The static method linearSearch(int, int[]) from the t    lab7_1Test.java   /Lab7/src   line 12    Java Problem
The static method linearSearch(int, int[]) from the t    lab7_1Test.java   /Lab7/src   line 22    Java Problem
The static method linearSearch(int, int[]) from the t    lab7_1Test.java   /Lab7/src   line 32    Java Problem
The static method linearSearch(int, int[]) from the t    lab7_1Test.java   /Lab7/src   line 43    Java Problem
```

Writable | Smart Insert | 18 : 1 : 291

A=[1,2,3,3,3] v=3 Output: 3

Test Case failed:



Lab-7 - Lab7/src/lab7_2Test.java - Eclipse IDE

Finished after 0.031 seconds

Runs: 3/3   Errors: 0   Failures: 1

```
lab7_2Test [Runner: JUnit 4] (0.006 s)
    test (0.000 s)
    test2 (0.000 s)
    test3 (0.006 s)
```

Failure Trace

```
java.lang.AssertionError: expected:<1> but was:<2>
at lab7_2Test.test3(lab7_2Test.java:34)
```

```java
1  import static org.junit.Assert.*;
4
5  public class lab7_2Test {
6
7      @Test
8      public void test() {
9          lab7_2 l1=new lab7_2();
10         int v=3;
11         int a[]=new int[] {1,2,3,3,3};
12         int output=l1.countItem(v,a);
13         int expected=3;
14         assertEquals(expected,output);
15     }
16
17     @Test
18     public void test2() {
19         lab7_2 l1=new lab7_2();
20         int v=3;
21         int a[]=new int[] {1,2,4};
22         int output=l1.countItem(v,a);
23         int expected=0;
24         assertEquals(expected,output);
25     }
26
27     @Test
28     public void test3() {
29         lab7_2 l1=new lab7_2();
30         int v=10;
31         int a[]=new int[] {10,20,10};
32         int output=l1.countItem(v,a);
33         int expected=1;
34         assertEquals(expected,output);
```

0 errors, 7 warnings, 0 others

```
Description                                                Resource           Path        Location   Type
Warnings (7 items)
  The static method countItem(int, int[]) from the typ     lab7_2Test.java    /Lab7/src   line 12    Java Problem
  The static method countItem(int, int[]) from the typ     lab7_2Test.java    /Lab7/src   line 22    Java Problem
  The static method countItem(int, int[]) from the typ     lab7_2Test.java    /Lab7/src   line 32    Java Problem
  The static method linearSearch(int, int[]) from the t    lab7_1Test.java    /Lab7/src   line 12    Java Problem
  The static method linearSearch(int, int[]) from the t    lab7_1Test.java    /Lab7/src   line 22    Java Problem
```

Writable   Smart Insert   35 : 6 [192]

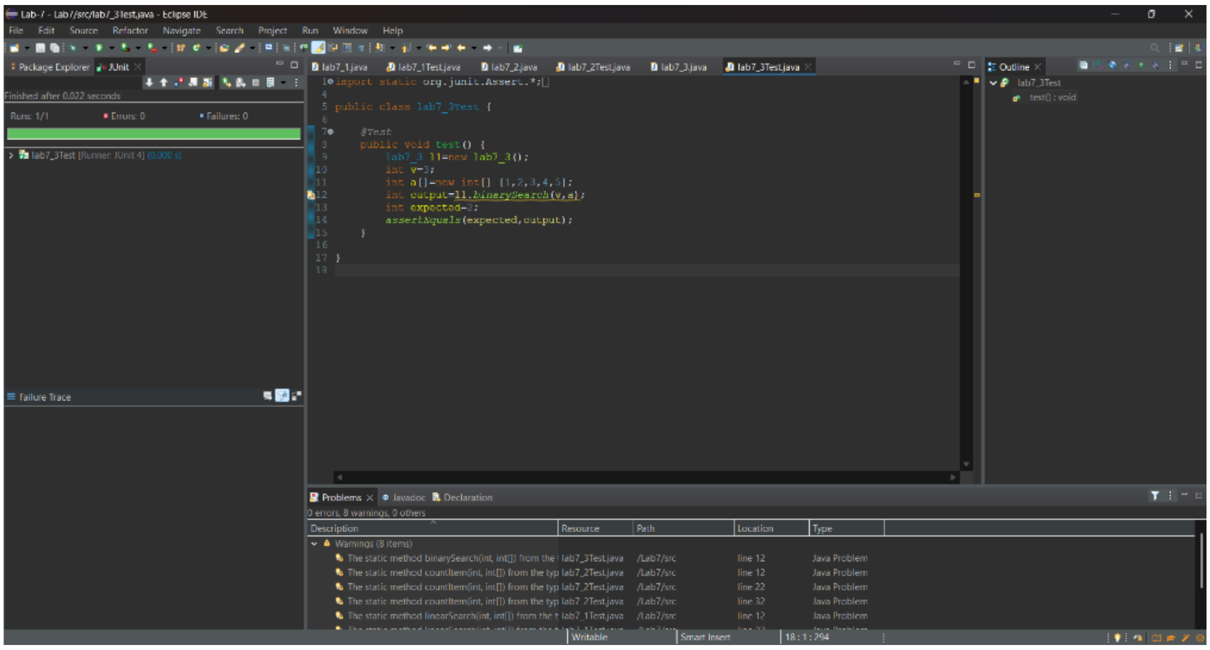
A=[10,20,10] v=10   Output: 1

Program 3: 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.

Text Action and Input Data	Expected Output
Equivalence Partitioning	
a = [1,2,3,4,5], v= 2	1
a = [3,3,6], v= 3	0
a = [], v=2	-1
a = [1,2,5,9], v=3	-1
Boundary Value Analysis	
Minimum length of array a = [] , v = 5	-1
Maximum length array a = [2,2,2,2,2,2], v=2	0
Minimum value of v a = [1,2,3] , v = 1	0

Maximum value of v a = [2,6,9], v = 9	2

Test Case passed:



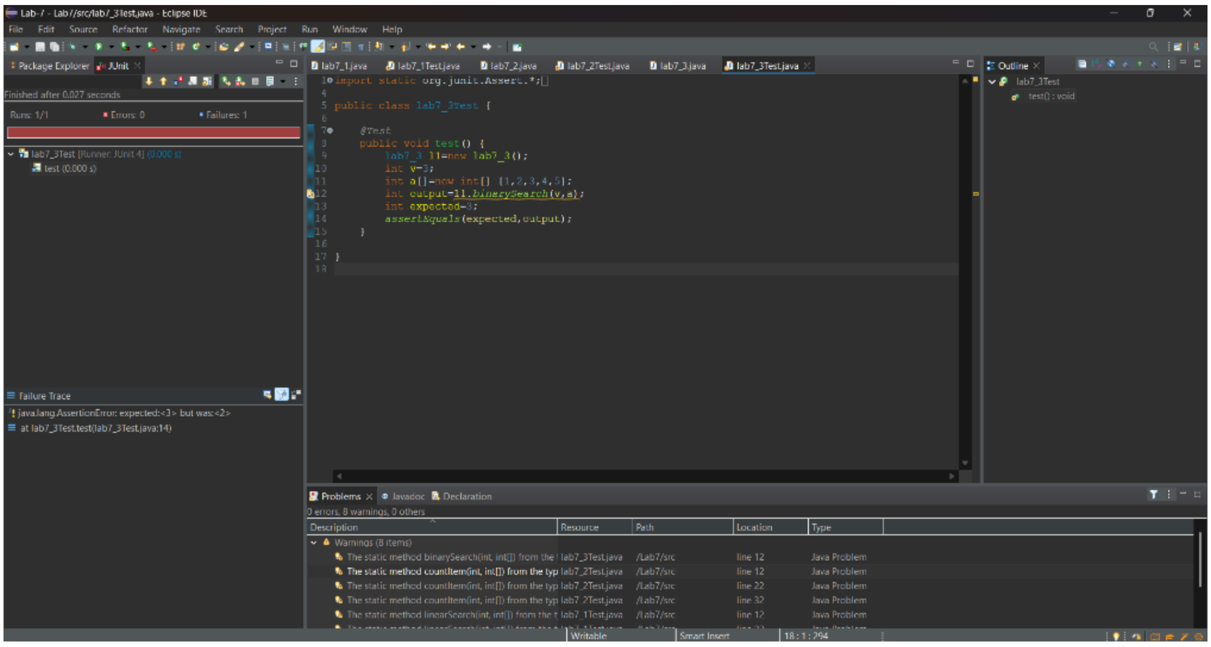
```
Lab-7 - Lab7/src/lab7_3Test.java - Eclipse IDE

Finished after 0.022 seconds
Runs: 1/1    Errors: 0    Failures: 0
> lab7_3Test [Runner: JUnit 4] (0.000 s)
```

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

public class lab7_3Test {

    @Test
    public void test() {
        lab7_3 l1=new lab7_3();
        int v=3;
        int a[]=new int[] {1,2,3,4,5};
        int output=l1.binarySearch(v,a);
        int expected=2;
        assertEquals(expected,output);
    }

}
```

```
0 errors, 8 warnings, 0 others
Description                                           Resource          Path         Location   Type
v Warnings (8 items)
    The static method binarySearch(int, int[]) from the  lab7_3Test.java   /Lab7/src    line 12    Java Problem
    The static method countItem(int, int[]) from the typ lab7_2Test.java   /Lab7/src    line 12    Java Problem
    The static method countItem(int, int[]) from the typ lab7_2Test.java   /Lab7/src    line 22    Java Problem
    The static method countItem(int, int[]) from the typ lab7_2Test.java   /Lab7/src    line 32    Java Problem
    The static method linearSearch(int, int[]) from the t lab7_1Test.java  /Lab7/src    line 12    Java Problem
Writable    Smart Insert    18 : 1 : 294
```

A=[1,2,3,4,5] v=3 Output: 2

Test case failed:



```
Lab-7 - Lab7/src/lab7_3Test.java - Eclipse IDE

Finished after 0.027 seconds
Runs: 1/1    Errors: 0    Failures: 1
v lab7_3Test [Runner: JUnit 4] (0.000 s)
    test (0.000 s)

Failure Trace
java.lang.AssertionError: expected:<3> but was:<2>
at lab7_3Test.test(lab7_3Test.java:14)
```

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

public class lab7_3Test {

    @Test
    public void test() {
        lab7_3 l1=new lab7_3();
        int v=3;
        int a[]=new int[] {1,2,3,4,5};
        int output=l1.binarySearch(v,a);
        int expected=3;
        assertEquals(expected,output);
    }

}
```

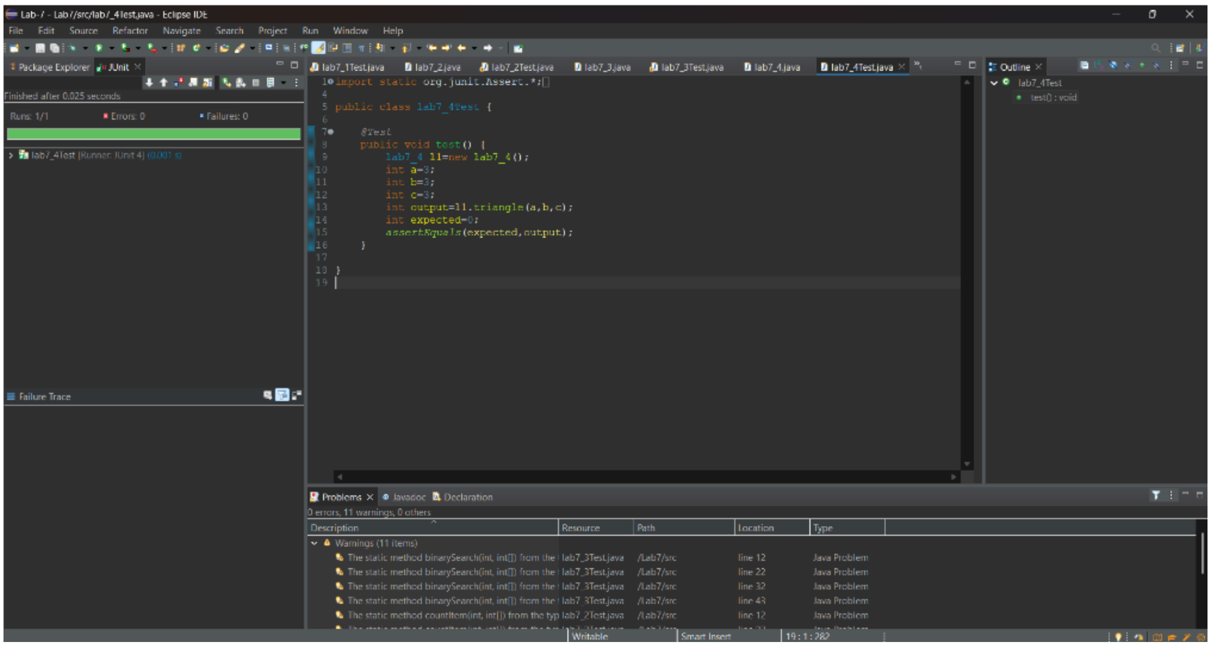
```
0 errors, 8 warnings, 0 others
Description                                           Resource          Path         Location   Type
v Warnings (8 items)
    The static method binarySearch(int, int[]) from the  lab7_3Test.java   /Lab7/src    line 12    Java Problem
    The static method countItem(int, int[]) from the typ lab7_2Test.java   /Lab7/src    line 12    Java Problem
    The static method countItem(int, int[]) from the typ lab7_2Test.java   /Lab7/src    line 22    Java Problem
    The static method countItem(int, int[]) from the typ lab7_2Test.java   /Lab7/src    line 32    Java Problem
    The static method linearSearch(int, int[]) from the t lab7_1Test.java  /Lab7/src    line 12    Java Problem
Writable    Smart Insert    18 : 1 : 294
```

A=[1,2,3,4,5] v=3 Output: 3

Program 4: 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).

Text Action and Input Data	Expected Output
Equivalence Partitioning	
1,1,1	0
4,4,6	1
3,4,5	2
Boundary Value Analysis	
1,5,6	3
0,0,0	3
0,9,-2	3
1,200000000000,9999999999999	3

Test Case passed:



Runs: 1/1 Errors: 0 Failures: 0

Finished after 0.025 seconds

lab7_4Test [Runner: JUnit 4] (0.001 s)

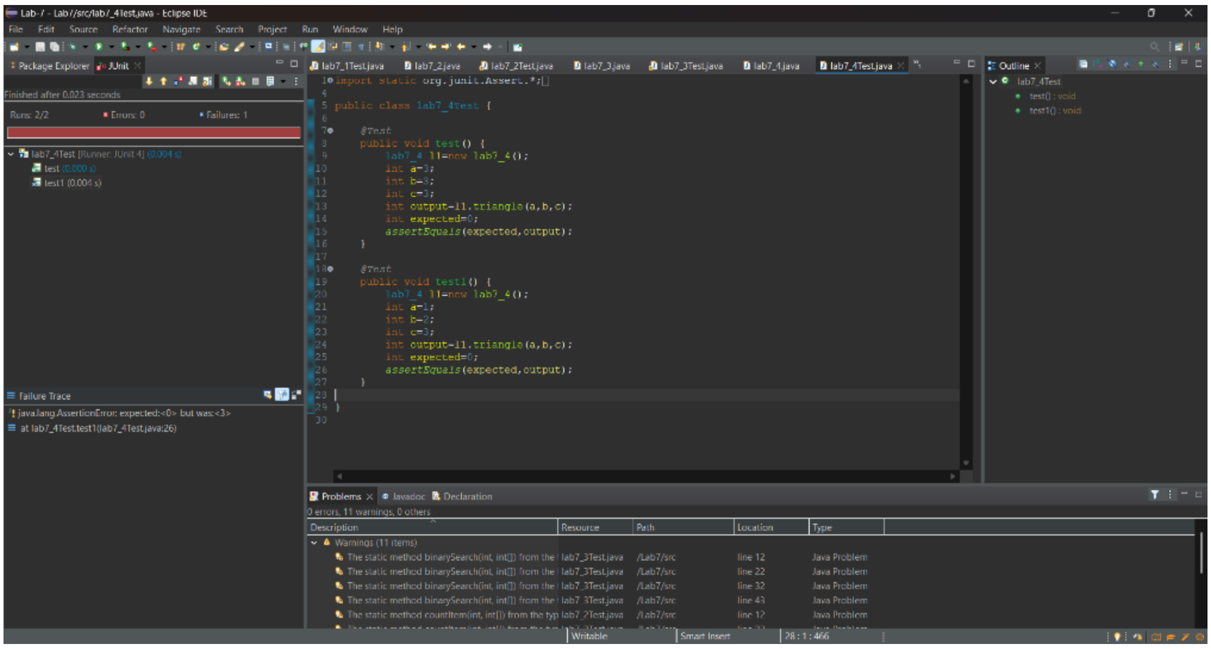
```java
import static org.junit.Assert.*;

public class lab7_4Test {

    @Test
    public void test() {
        lab7_4 l1=new lab7_4();
        int a=3;
        int b=3;
        int c=3;
        int output=l1.triangle(a,b,c);
        int expected=0;
        assertEquals(expected,output);
    }

}
```

0 errors, 11 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (11 items)				
The static method binarySearch(int, int[]) from the	lab7_3Test.java	/Lab7/src	line 12	Java Problem
The static method binarySearch(int, int[]) from the	lab7_3Test.java	/Lab7/src	line 22	Java Problem
The static method binarySearch(int, int[]) from the	lab7_3Test.java	/Lab7/src	line 32	Java Problem
The static method binarySearch(int, int[]) from the	lab7_3Test.java	/Lab7/src	line 43	Java Problem
The static method countItem(int, int[]) from the typ	lab7_2Test.java	/Lab7/src	line 12	Java Problem

A=3, B=3, C=3 Output: 0 Equilateral

Test Case failed:



```
Finished after 0.023 seconds
Runs: 2/2    Errors: 0    Failures: 1
lab7_4Test [Runner: JUnit 4] (0.004 s)
    test (0.000 s)
    test1 (0.004 s)

Failure Trace
java.lang.AssertionError: expected:<0> but was:<3>
at lab7_4Test.test1(lab7_4Test.java:26)
```

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

public class lab7_4Test {

    @Test
    public void test() {
        lab7_4 l1=new lab7_4();
        int a=3;
        int b=3;
        int c=3;
        int output=l1.triangle(a,b,c);
        int expected=0;
        assertEquals(expected,output);
    }

    @Test
    public void test1() {
        lab7_4 l1=new lab7_4();
        int a=1;
        int b=2;
        int c=3;
        int output=l1.triangle(a,b,c);
        int expected=0;
        assertEquals(expected,output);
    }

}
```

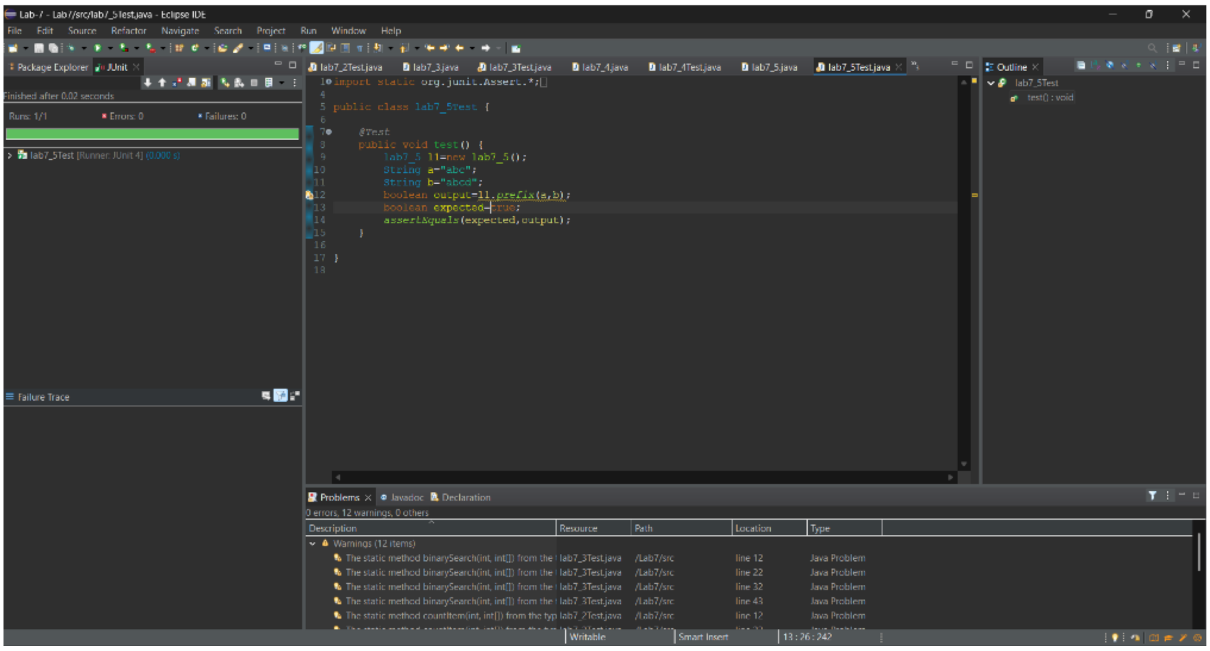
```
0 errors, 11 warnings, 0 others
Description                                          Resource          Path        Location   Type
Warnings (11 items)
The static method binarySearch(int, int[]) from the  lab7_3Test.java   /Lab7/src   line 12    Java Problem
The static method binarySearch(int, int[]) from the  lab7_3Test.java   /Lab7/src   line 22    Java Problem
The static method binarySearch(int, int[]) from the  lab7_3Test.java   /Lab7/src   line 32    Java Problem
The static method binarySearch(int, int[]) from the  lab7_3Test.java   /Lab7/src   line 43    Java Problem
The static method countItem(int, int[]) from the typ lab7_2Test.java   /Lab7/src   line 12    Java Problem
```

A=1, B=2, C=3   Output: 0

Program 5 The function prefix (String s1, String s2) returns whether or not the string s1 is a prefix of string s2 (you may assume that neither s1 nor s2 is null).

Text Action and Input Data	Expected Output
Equivalence Partitioning	
“”, “”	true
“”, “lab”	true
“lab”, ”lab”	true
“lab”, “lag”	false
“labs”, “lab”	false
Boundary Value Analysis	
“”, “”	true
“”, “lab”	true
“lab”, “lag”	false
“labs”, “lab”	false

Test Case passed:



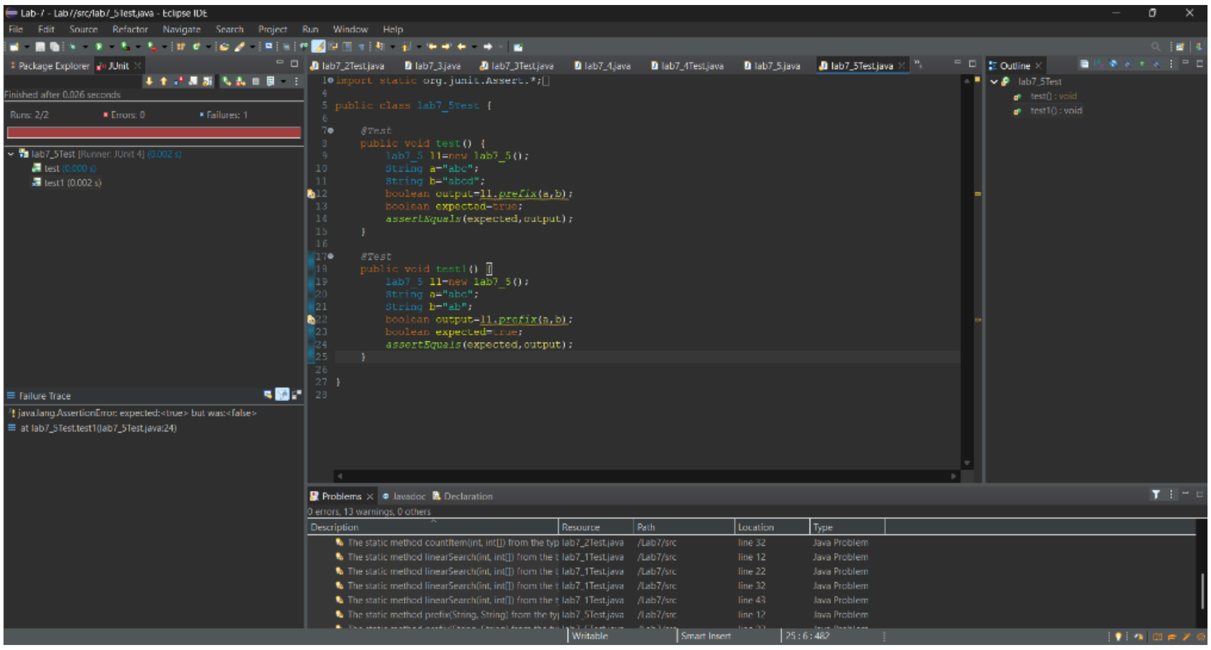
```
Lab-7 - Lab7/src/lab7_5Test.java - Eclipse IDE

Finished after 0.02 seconds
Runs: 1/1    Errors: 0    Failures: 0
lab7_5Test [Runner: JUnit 4] (0.000 s)
```

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

public class lab7_5Test {

    @Test
    public void test() {
        lab7_5 l1=new lab7_5();
        String a="abc";
        String b="abcd";
        boolean output=l1.prefix(a,b);
        boolean expected=true;
        assertEquals(expected,output);
    }

}
```

```
0 errors, 12 warnings, 0 others
Description                                               Resource          Path        Location  Type
Warnings (12 items)
The static method binarySearch(int, int[]) from the t    lab7_3Test.java   /Lab7/src   line 12   Java Problem
The static method binarySearch(int, int[]) from the t    lab7_3Test.java   /Lab7/src   line 22   Java Problem
The static method binarySearch(int, int[]) from the t    lab7_3Test.java   /Lab7/src   line 32   Java Problem
The static method binarySearch(int, int[]) from the t    lab7_3Test.java   /Lab7/src   line 43   Java Problem
The static method countItem(int, int[]) from the typ     lab7_2Test.java   /Lab7/src   line 12   Java Problem
Writable    Smart Insert    13 : 26 : 242
```

S1= abc s2 = abcd   Output: true

Test case failed:



```
Lab-7 - Lab7/src/lab7_5Test.java - Eclipse IDE

Finished after 0.026 seconds
Runs: 2/2    Errors: 0    Failures: 1
lab7_5Test [Runner: JUnit 4] (0.002 s)
    test (0.000 s)
    test1 (0.002 s)

Failure Trace
java.lang.AssertionError: expected:<true> but was:<false>
at lab7_5Test.test1(lab7_5Test.java:24)
```

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

public class lab7_5Test {

    @Test
    public void test() {
        lab7_5 l1=new lab7_5();
        String a="abc";
        String b="abcd";
        boolean output=l1.prefix(a,b);
        boolean expected=true;
        assertEquals(expected,output);
    }

    @Test
    public void test1() {
        lab7_5 l1=new lab7_5();
        String a="abc";
        String b="ab";
        boolean output=l1.prefix(a,b);
        boolean expected=true;
        assertEquals(expected,output);
    }

}
```

```
0 errors, 13 warnings, 0 others
Description                                               Resource          Path        Location  Type
The static method countItem(int, int[]) from the typ     lab7_2Test.java   /Lab7/src   line 32   Java Problem
The static method linearSearch(int, int[]) from the t    lab7_1Test.java   /Lab7/src   line 12   Java Problem
The static method linearSearch(int, int[]) from the t    lab7_1Test.java   /Lab7/src   line 22   Java Problem
The static method linearSearch(int, int[]) from the t    lab7_1Test.java   /Lab7/src   line 32   Java Problem
The static method linearSearch(int, int[]) from the t    lab7_1Test.java   /Lab7/src   line 43   Java Problem
The static method prefix(String, String) from the ty     lab7_5Test.java   /Lab7/src   line 12   Java Problem
Writable    Smart Insert    25 : 6 : 482
```

S1 = abc s2 = ab   Output: false

Program6: 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 stating whether the triangle can be formed, is scalene, isosceles, equilateral, or right-angled. Determine the following for the above program:

a) Equivalence Classes for the system are:

EC1: The lengths of all sides are positive real numbers

EC2: The length of at-least one side is 0 or negative.

EC3: The sum of the lengths of any two sides is less than the length of the third side

EC4: The sum of the lengths of any two sides is greater than the length of the third side

b) Test cases to cover the equivalence class:

TC1: A=3, B=4, C=5 : right-angled triangle

TC2: A=3, B=3, C=3 : equilateral triangle

TC3: A=3, B=3, C=4 : isosceles triangle

TC4: A=6, B=7, C=8 : scalene triangle

TC5: A=0, B=2, C=3 : invalid

TC6: A=-1, B=9, C=9 : invalid

c) Test cases for boundary condition A + B > C case (scalene triangle):

TC7: A=3, B=6, C=10

d) Test case for boundary condition A=C case (isosceles triangle):

TC8: A=5, B=6, C=5

e) Test case for the boundary condition A = B = C case (equilateral triangle)

TC9: A=5, B=5, C=5

f) Test case for the boundary condition A2 + B2 = C2 case (right-angle triangle)

TC10: A=3, B=4, C=5

g) Test case for the non-triangle case

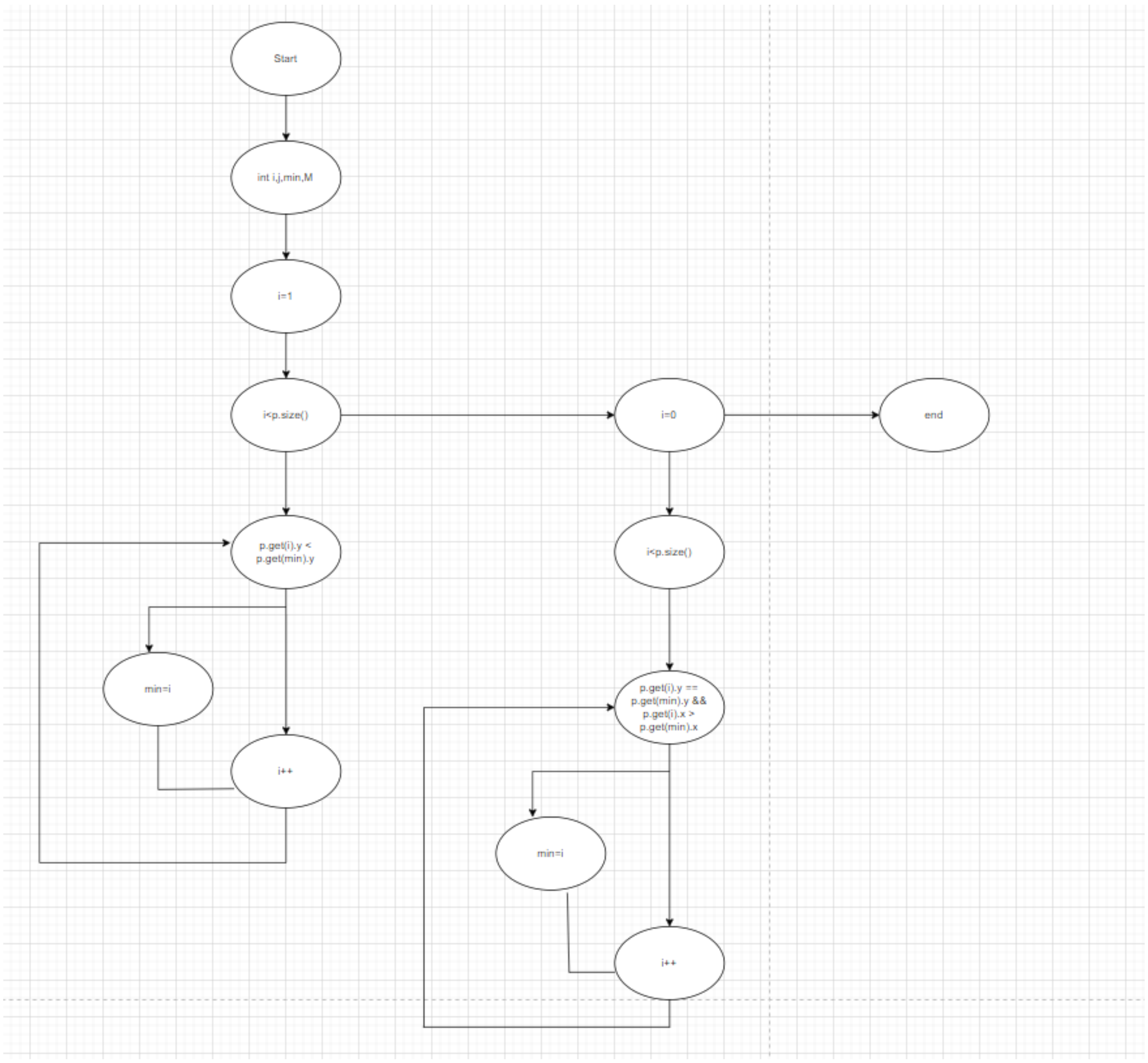
TC11: A=1, B=2, C=3

h) Test case for non-positive input

TC12: A=-1, B=2, C=3

Section B:

The code below is part of a method in the ConvexHull class in the VMAP system. The following is a small fragment of a method in the ConvexHull class. For the purposes of this exercise you do not need to know the intended function of the method. The parameter p is a Vector of Point objects, p.size() is the size of the vector p, (p.get(i)).x is the x component of the ith point appearing in p, similarly for (p.get(i)).y. This exercise is concerned with structural testing of code and so the focus is on creating test sets that satisfy some particular coverage criterion.


Start
int i,j,min,M
i=1
i<p.size()
i=0
end
p.get(i).y < p.get(min).y
i<p.size()
min=i
p.get(i).y == p.get(min).y && p.get(i).x > p.get(min).x
i++
min=i
i++

Statement coverage test cases:

TC1: p = empty vector

TC2: p = vector with one point

TC3: p = vector with two points with the same y component

TC4: p = vector with two points with different y components

TC5: p = vector with three or more points with different y components

TC6: p = vector with three or more points with the same y component

Branch coverage test sets:

TC1: p = empty vector

TC2: p = vector with one point

TC3: p = vector with two points with the same y component

TC4: p = vector with two points with different y components

TC5: p = vector with three or more points with different y components, and none of them have the same x component

TC6: p = vector with three or more points with the same y component, and some of them have the same x component

TC7: p = vector with three or more points with the same y component, and all of them have the same x component

Basic condition coverage test sets:

TC1: p = empty vector

TC2: p = vector with one point

TC3: p = vector with two points with the same y component, and the first point has a smaller x component

TC4: p = vector with two points with the same y component, and the second point has a smaller x component

TC5: p = vector with two points with different y components

TC6: p = vector with three or more points with different y components, and none of them have the same x component

TC7: p = vector with three or more points with the same y component, and some of them have the same x component

TC8: p = vector with three or more points with the same y component, and all of them have the same x component.

Example Test Cases:

1) p=[(x=2,y=3),(x=2,y=2),(x=1,y=5),(x=1,y=4)]
2) p=[(x=5,y=6),(x=3,y=2),(x=3,y=4),(x=1,y=2)]
3) p=[(x=5,y=6),(x=3,y=5),(x=1,y=5),(x=4,y=5),(x=2,y=7)]
4) p=[(x=7,y=8)]
5) p=[]