IT314

Lab7 - Testing
Software Engineering
Winter Semester 2022-23

Name: Pansuriya Viraj Dhirajbhai
Student Id:202001196
Date:13/04/2023

Section-A:

Test-Case:

Day	Month	Year	Expected Outcome
15	6	1900	14 june , 1900
15	6	1901	14 june 1901
15	6	1958	14 june , 1958
15	6	2014	14 june , 2014
15	6	2015	14 june , 2015
1	6	1958	31 may , 1958
2	6	1958	1 june , 1958
30	6	1958	29 june , 1958
31	6	1958	Invalid Date
15	1	1958	14 jan , 1958
15	2	1958	14 feb, 1958
15	11	1958	14 nov, 1958
5	12	1958	14 dec, 1958

Equivalence Partition:

Day:

Day	Output
1 to 28	Valid
< 1	Invalid
> 31	Invalid
=30	Valid
=29	Valid for leap year
=31	Valid for month which contains 3 days

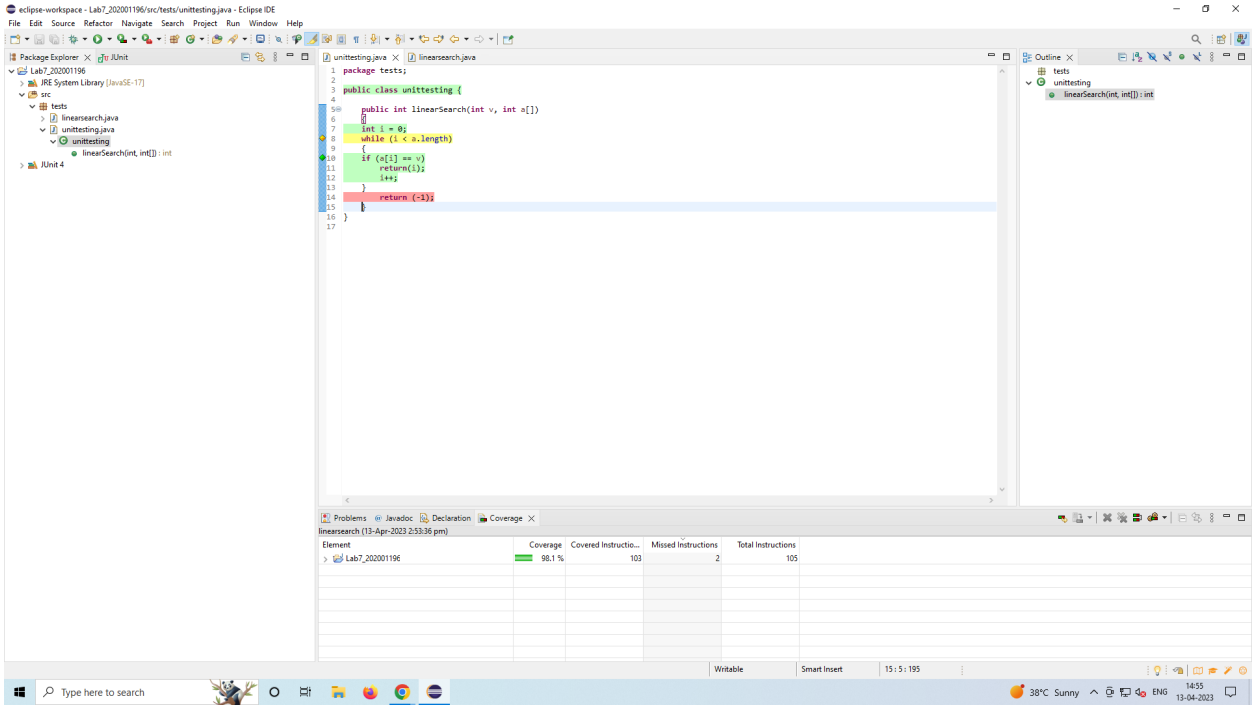
Month:

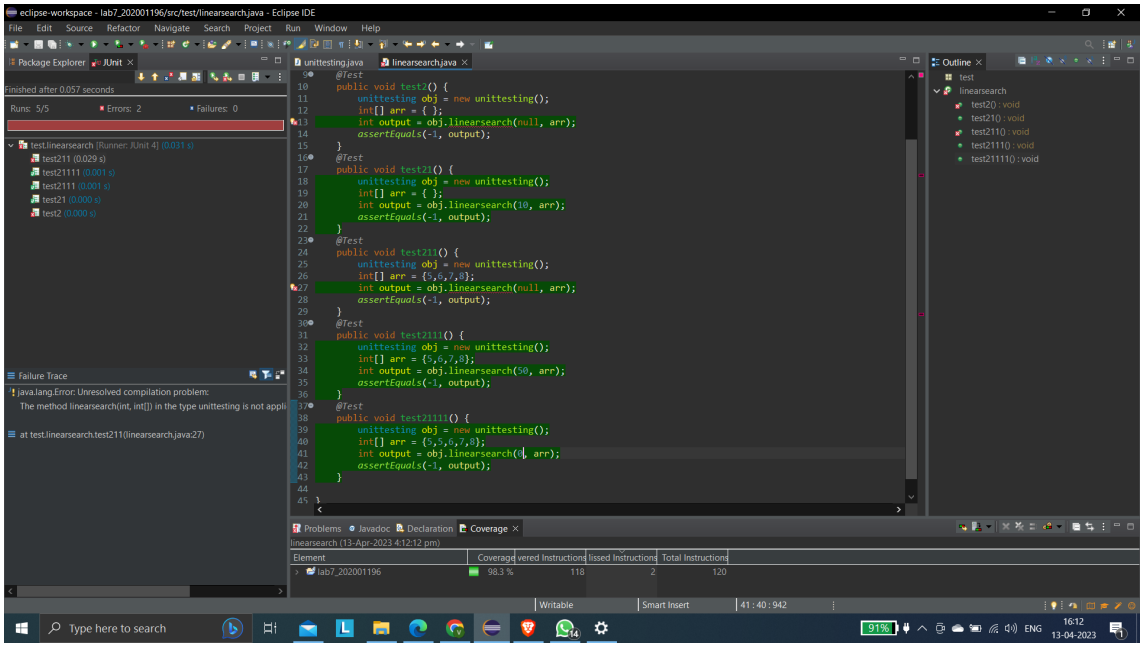
Month	Output
1 to 12	Valid
< 1	Invalid
> 12	Invalid

Year:

Year	Output
1900 to 2015	Valid
< 1900	Invalid
> 2015	Invalid

Problem 1:


eclipse-workspace - Lab7_202001196/src/tests/unittesting.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
Lab7_202001196
JRE System Library [JavaSE-17]
src
tests
linearsearch.java
unittesting.java
unittesting
linearSearch(int, int[]) : int
JUnit 4
unittesting.java
linearsearch.java
package tests;
public class unittesting {
public int linearSearch(int v, int a[])
{
int i = 0;
while (i < a.length)
{
if (a[i] == v)
return(i);
i++;
}
return (-1);
}
}
Outline
tests
unittesting
linearSearch(int, int[]) : int
Problems
Javadoc
Declaration
Coverage
linearsearch (13-Apr-2023 2:53:36 pm)
Element
Coverage
Covered Instructio...
Missed Instructions
Total Instructions
Lab7_202001196
98.1 %
103
2
105
Writable
Smart Insert
15 : 5 : 195
Type here to search
38°C Sunny
ENG
14:55
13-04-2023


eclipse-workspace - lab7_202001196/src/test/linearsearch.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
Finished after 0.057 seconds
Runs: 5/5
Errors: 2
Failures: 0
test.linearsearch [Runner: JUnit 4] (0.031 s)
test211 (0.029 s)
test21111 (0.001 s)
test2111 (0.001 s)
test21 (0.000 s)
test2 (0.000 s)
Failure Trace
java.lang.Error: Unresolved compilation problem:
The method linearsearch(int, int[]) in the type unittesting is not appli
at test.linearsearch.test211(linearsearch.java:27)
unittesting.java
linearsearch.java
@Test
public void test2() {
unittesting obj = new unittesting();
int[] arr = { };
int output = obj.linearsearch(null, arr);
assertEquals(-1, output);
}
@Test
public void test21() {
unittesting obj = new unittesting();
int[] arr = { };
int output = obj.linearsearch(10, arr);
assertEquals(-1, output);
}
@Test
public void test211() {
unittesting obj = new unittesting();
int[] arr = {5,6,7,8};
int output = obj.linearsearch(null, arr);
assertEquals(-1, output);
}
@Test
public void test2111() {
unittesting obj = new unittesting();
int[] arr = {5,6,7,8};
int output = obj.linearsearch(50, arr);
assertEquals(-1, output);
}
@Test
public void test21111() {
unittesting obj = new unittesting();
int[] arr = {5,5,6,7,8};
int output = obj.linearsearch(0, arr);
assertEquals(-1, output);
}
}
Outline
test
linearsearch
test2() : void
test21() : void
test211() : void
test21110 : void
test21111() : void
Problems
Javadoc
Declaration
Coverage
linearsearch (13-Apr-2023 4:12:12 pm)
Element
Coverage
vered Instructions
lissed Instructions
Total Instructions
lab7_202001196
98.3 %
118
2
120
Writable
Smart Insert
41 : 40 : 942
Type here to search
91%
ENG
16:12
13-04-2023

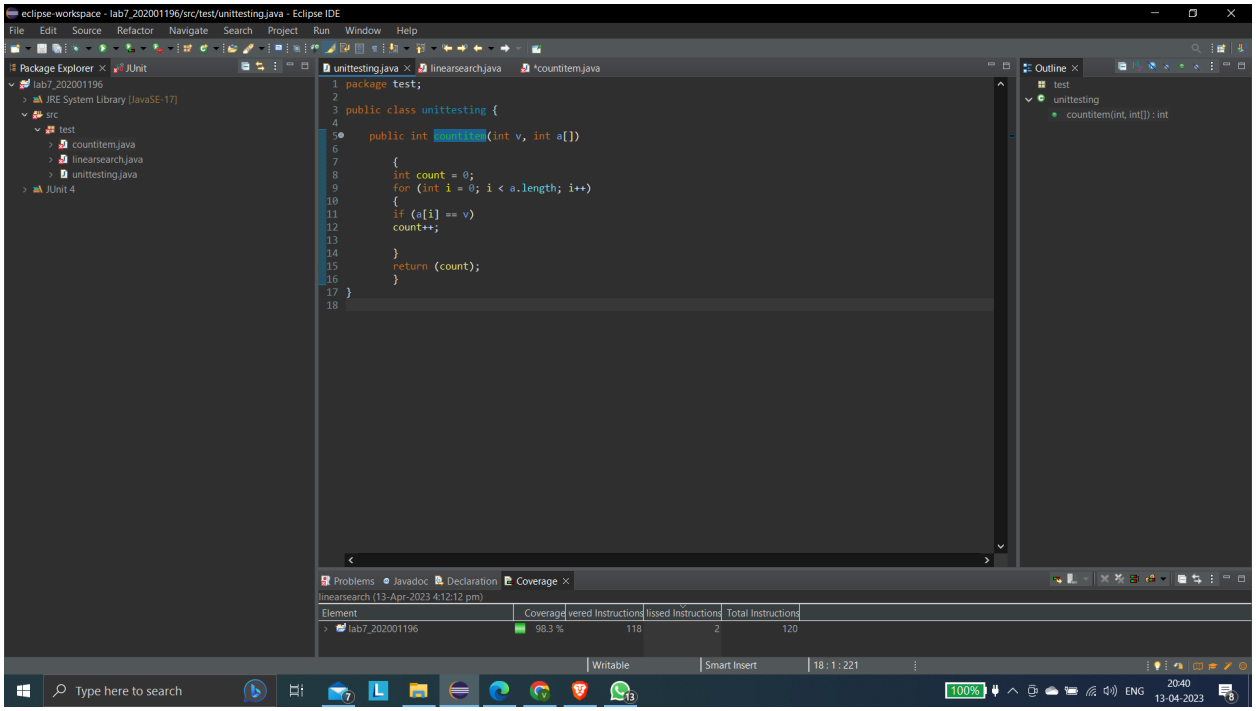
Equivalence Partition

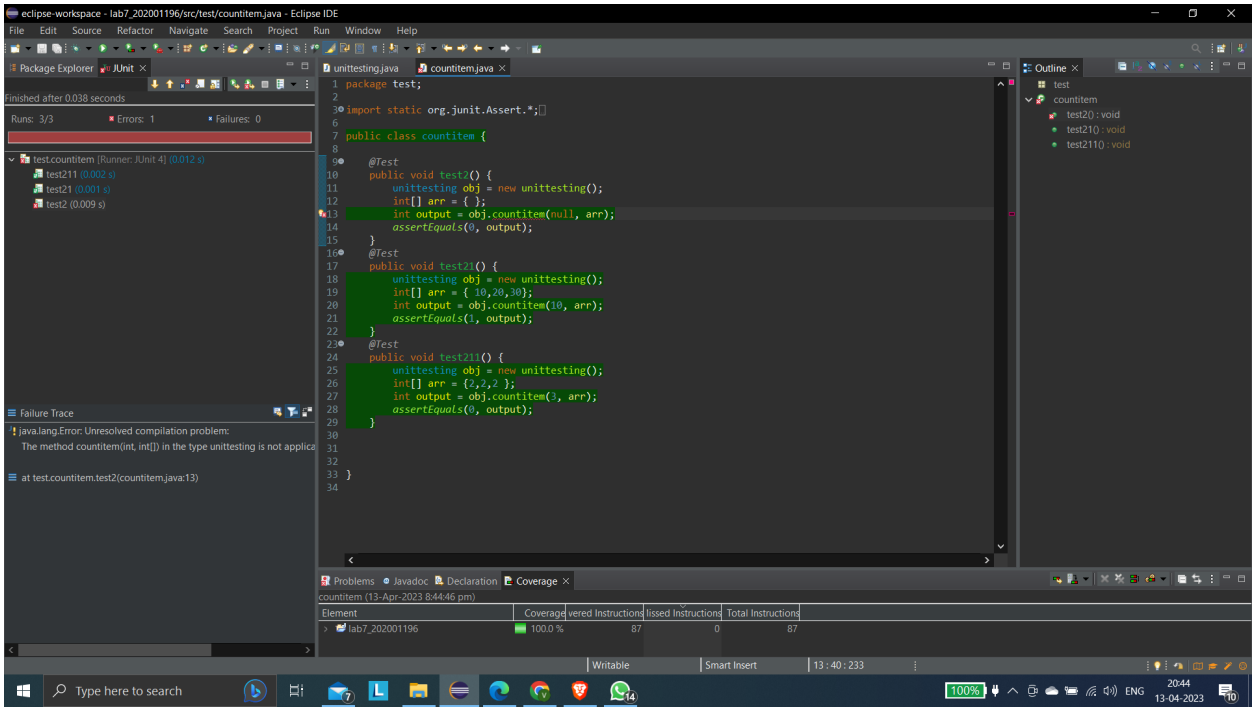
Array a	Variable v	Expected Output	a	v
Empty	Non-Exist in Array	-1	[]	500
Non Empty	Not Exists in array	-1	[5,6,7,8]	50
Non Empty	Exists in array	0	[5,6,7,8]	5

Boundary Value Analysis

Array	Value	Expected Outcome
[]	50	-1
[10]	10	0
[10]	50	-1
[10,50,70,90,100]	10	0
[10,50,70,90,100]	70	2
[10,50,70,90,100]	100	4
[10,50,70,90,100]	500	-1

Problem 2:


eclipse-workspace - lab7_202001196/src/test/unittesting.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit
lab7_202001196
JRE System Library [JavaSE-17]
src
test
countitem.java
linearsearch.java
unittesting.java
JUnit 4
unittesting.java linearsearch.java *countitem.java
1 package test;
2
3 public class unittesting {
4
5 public int countitem(int v, int a[])
6
7 {
8 int count = 0;
9 for (int i = 0; i < a.length; i++)
10 {
11 if (a[i] == v)
12 count++;
13
14 }
15 return (count);
16 }
17 }
18
Outline
test
unittesting
countitem(int, int[]) : int
Problems Javadoc Declaration Coverage
linearsearch (13-Apr-2023 4:12:12 pm)
Element Coverage vered Instructions lissed Instructions Total Instructions
lab7_202001196 98.3 % 118 2 120
Writable Smart Insert 18 : 1 : 221
Type here to search
100%
ENG
20:40
13-04-2023


eclipse-workspace - lab7_202001196/src/test/countitem.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit
Finished after 0.038 seconds
Runs: 3/3 Errors: 1 Failures: 0
test.countitem [Runner: JUnit 4] (0.012 s)
test211 (0.002 s)
test21 (0.001 s)
test2 (0.009 s)
Failure Trace
java.lang.Error: Unresolved compilation problem:
The method countitem(int, int[]) in the type unittesting is not applica
at test.countitem.test2(countitem.java:13)
unittesting.java countitem.java
1 package test;
2
3 import static org.junit.Assert.*;
6
7 public class countitem {
8
9 @Test
10 public void test2() {
11 unittesting obj = new unittesting();
12 int[] arr = { };
13 int output = obj.countitem(null, arr);
14 assertEquals(0, output);
15 }
16 @Test
17 public void test21() {
18 unittesting obj = new unittesting();
19 int[] arr = { 10,20,30};
20 int output = obj.countitem(10, arr);
21 assertEquals(1, output);
22 }
23 @Test
24 public void test211() {
25 unittesting obj = new unittesting();
26 int[] arr = {2,2,2 };
27 int output = obj.countitem(3, arr);
28 assertEquals(0, output);
29 }
30
31
32
33 }
34
Outline
countitem
test2() : void
test21() : void
test211() : void
Problems Javadoc Declaration Coverage
countitem (13-Apr-2023 8:44:46 pm)
Element Coverage vered Instructions lissed Instructions Total Instructions
lab7_202001196 100.0 % 87 0 87
Writable Smart Insert 13 : 40 : 233
Type here to search
100%
ENG
20:44
13-04-2023

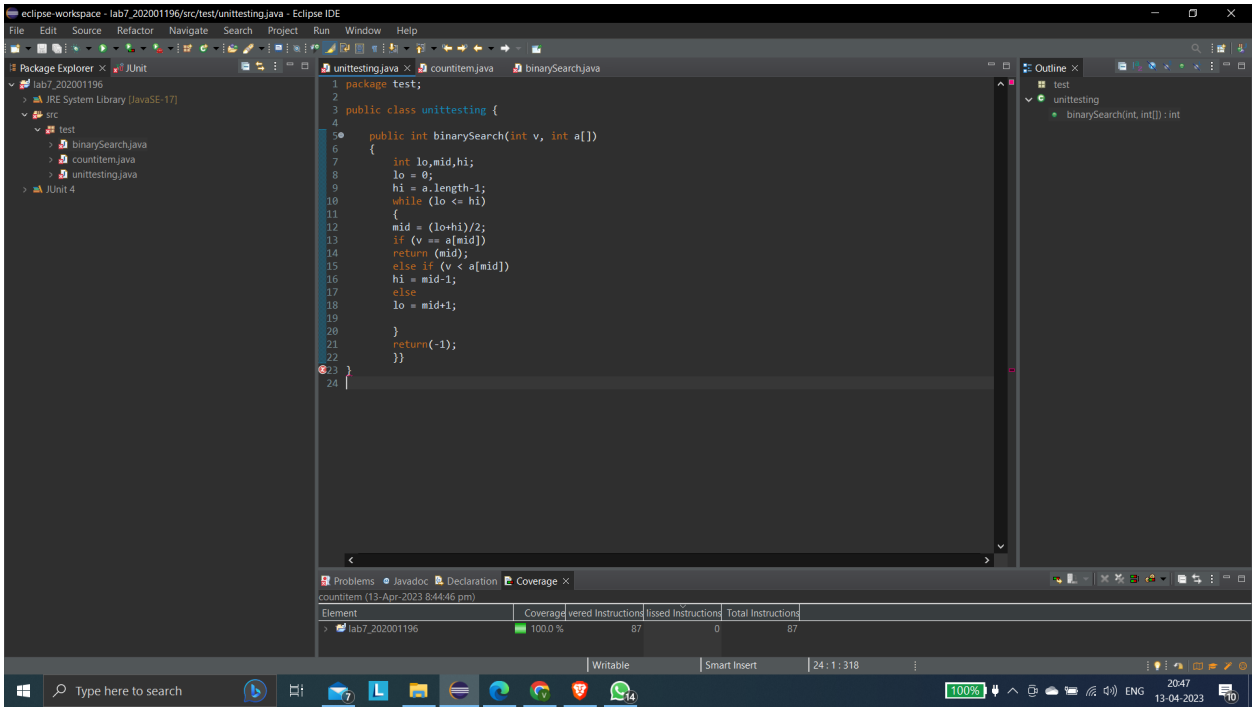
Equivalence Partition

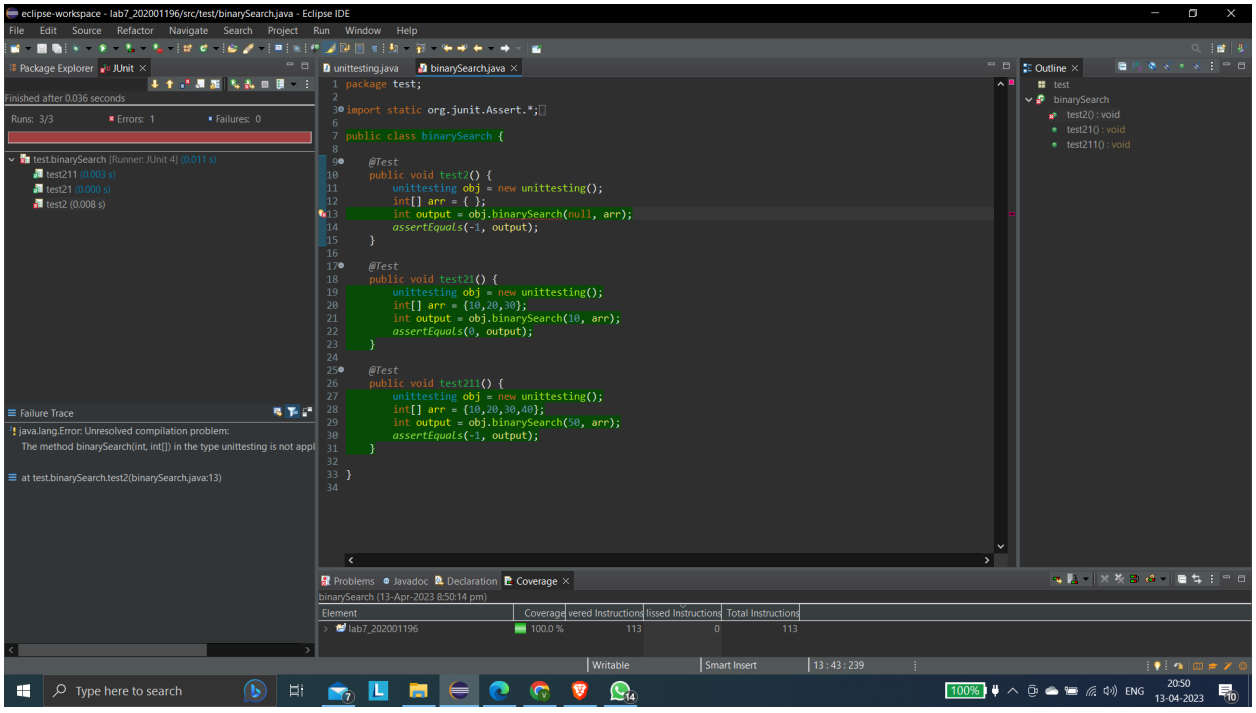
Array a	Variable v	Expected Output	a	v
Empty	Non-Exist in Array	0	[]	500
Non Empty	Not Exists in array	0	[5,6,7,8]	50
Non Empty	Exists in array	1	[5,6,7,8]	5

Boundary Value Analysis

Array	Value	Expected Outcome
[]	50	0
[10]	10	1
[10]	50	0
[10,50,70,90,100]	10	1
[70,70,70,70,70]	70	5
[10,50,100,100,100]	100	3
[10,50,70,90,100]	500	0

Problem 3:


eclipse-workspace - lab7_202001196/src/test/unittesting.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit
lab7_202001196
JRE System Library [JavaSE-17]
src
test
binarySearch.java
countitem.java
unittesting.java
JUnit 4
unittesting.java countitem.java binarySearch.java
1 package test;
2
3 public class unittesting {
4
5 public int binarySearch(int v, int a[])
6 {
7 int lo,mid,hi;
8 lo = 0;
9 hi = a.length-1;
10 while (lo <= hi)
11 {
12 mid = (lo+hi)/2;
13 if (v == a[mid])
14 return (mid);
15 else if (v < a[mid])
16 hi = mid-1;
17 else
18 lo = mid+1;
19
20 }
21 return(-1);
22 }}
23 }
24
Outline
test
unittesting
binarySearch(int, int[]) : int
Problems Javadoc Declaration Coverage
countitem (13-Apr-2023 8:44:46 pm)
Element Coverage vered Instructions lissed Instructions Total Instructions
lab7_202001196 100.0 % 87 0 87
Writable Smart Insert 24 : 1 : 318
Type here to search
100%
ENG
20:47
13-04-2023


eclipse-workspace - lab7_202001196/src/test/binarySearch.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit
Finished after 0.036 seconds
Runs: 3/3 Errors: 1 Failures: 0
test.binarySearch [Runner: JUnit 4] (0.011 s)
test211 (0.003 s)
test21 (0.000 s)
test2 (0.008 s)
Failure Trace
java.lang.Error: Unresolved compilation problem:
The method binarySearch(int, int[]) in the type unittesting is not appl
at test.binarySearch.test2(binarySearch.java:13)
unittesting.java binarySearch.java
1 package test;
2
3 import static org.junit.Assert.*;
6
7 public class binarySearch {
8
9 @Test
10 public void test2() {
11 unittesting obj = new unittesting();
12 int[] arr = { };
13 int output = obj.binarySearch(null, arr);
14 assertEquals(-1, output);
15 }
16
17 @Test
18 public void test21() {
19 unittesting obj = new unittesting();
20 int[] arr = {10,20,30};
21 int output = obj.binarySearch(10, arr);
22 assertEquals(0, output);
23 }
24
25 @Test
26 public void test211() {
27 unittesting obj = new unittesting();
28 int[] arr = {10,20,30,40};
29 int output = obj.binarySearch(50, arr);
30 assertEquals(-1, output);
31 }
32
33 }
34
Outline
test
binarySearch
test2() : void
test21() : void
test211() : void
Problems Javadoc Declaration Coverage
binarySearch (13-Apr-2023 8:50:14 pm)
Element Coverage vered Instructions lissed Instructions Total Instructions
lab7_202001196 100.0 % 113 0 113
Writable Smart Insert 13 : 43 : 239
Type here to search
100%
ENG
20:50
13-04-2023

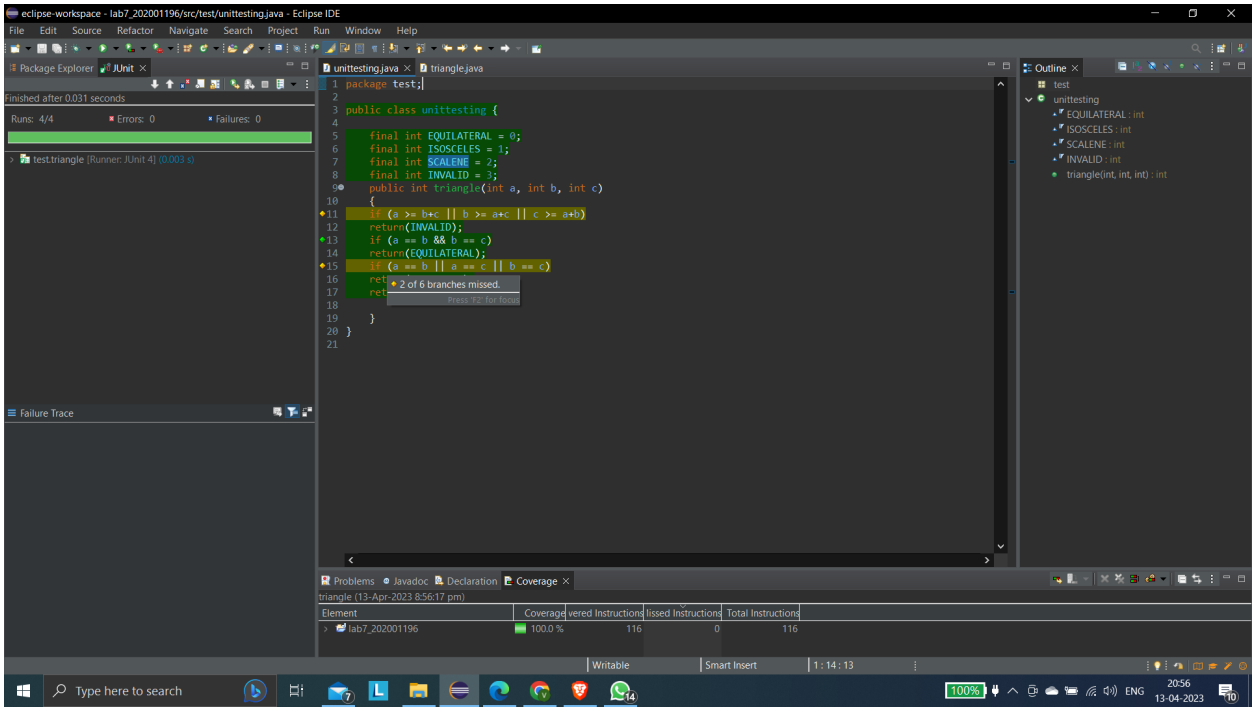
Equivalence Partition

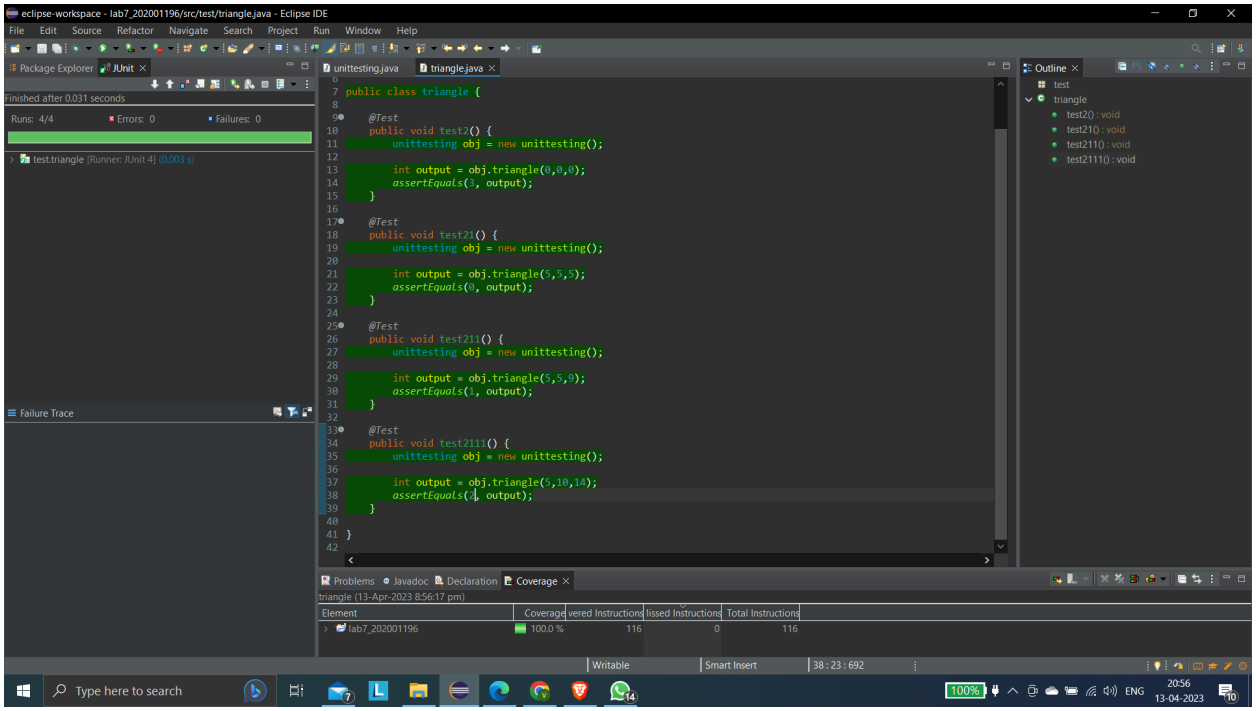
Array	Value	Expected Outcome
[]	50	-1
[1,5,10]	5	1
[1,5,10]	1	0
[1,5,10]	10	2
[1,5,10,50]	6	-1

Boundary Value Analysis

Array	Value	Expected Outcome
[]	50	-1
[10]	10	0
[10]	20	-1
[10,20,30,40,50]	10	0
[10,20,30,40,50]	30	2
[10,20,30,40,50]	50	4
[10,20,30,40,50]	5	-1
[10,20,30,40,50]	100	-1

Problem 4:


eclipse-workspace - lab7_202001196/src/test/unittesting.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit
Finished after 0.031 seconds
Runs: 4/4 Errors: 0 Failures: 0
test.triangle [Runner: JUnit 4] (0.003 s)
Failure Trace
unittesting.java triangle.java
package test;
public class unittesting {
final int EQUILATERAL = 0;
final int ISOSCELES = 1;
final int SCALENE = 2;
final int INVALID = 3;
public 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)
2 of 6 branches missed.
Press 'F2' for focus
}
}
Outline
test
unittesting
EQUILATERAL : int
ISOSCELES : int
SCALENE : int
INVALID : int
triangle(int, int, int) : int
Problems Javadoc Declaration Coverage
triangle (13-Apr-2023 8:56:17 pm)
Element Coverage vered Instructions lissed Instructions Total Instructions
lab7_202001196 100.0 % 116 0 116
Writable Smart Insert 1 : 14 : 13
Type here to search
100% ENG 20:56 13-04-2023


eclipse-workspace - lab7_202001196/src/test/triangle.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit
Finished after 0.031 seconds
Runs: 4/4 Errors: 0 Failures: 0
test.triangle [Runner: JUnit 4] (0.003 s)
Failure Trace
unittesting.java triangle.java
public class triangle {
@Test
public void test2() {
unittesting obj = new unittesting();
int output = obj.triangle(0,0,0);
assertEquals(3, output);
}
@Test
public void test21() {
unittesting obj = new unittesting();
int output = obj.triangle(5,5,5);
assertEquals(0, output);
}
@Test
public void test211() {
unittesting obj = new unittesting();
int output = obj.triangle(5,5,9);
assertEquals(1, output);
}
@Test
public void test2111() {
unittesting obj = new unittesting();
int output = obj.triangle(5,10,14);
assertEquals(2, output);
}
}
Outline
test
triangle
test2() : void
test21() : void
test211() : void
test2111() : void
Problems Javadoc Declaration Coverage
triangle (13-Apr-2023 8:56:17 pm)
Element Coverage vered Instructions lissed Instructions Total Instructions
lab7_202001196 100.0 % 116 0 116
Writable Smart Insert 38 : 23 : 692
Type here to search
100% ENG 20:56 13-04-2023

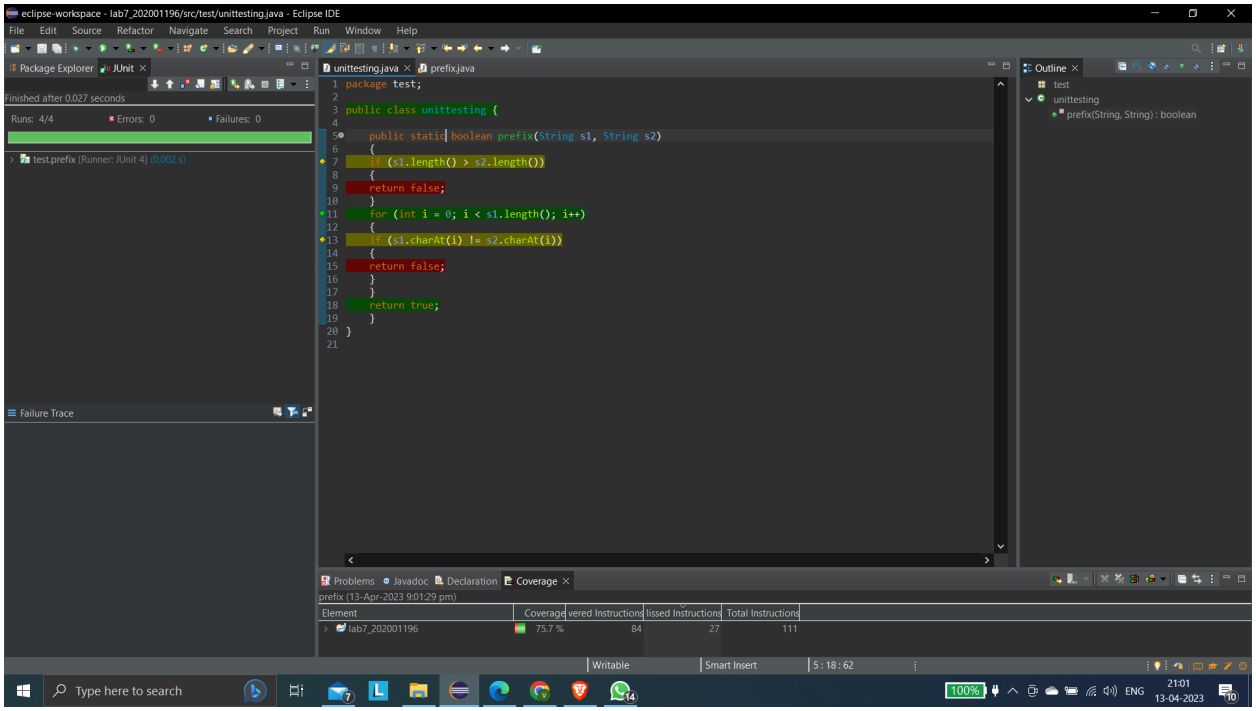
Equivalence Partition

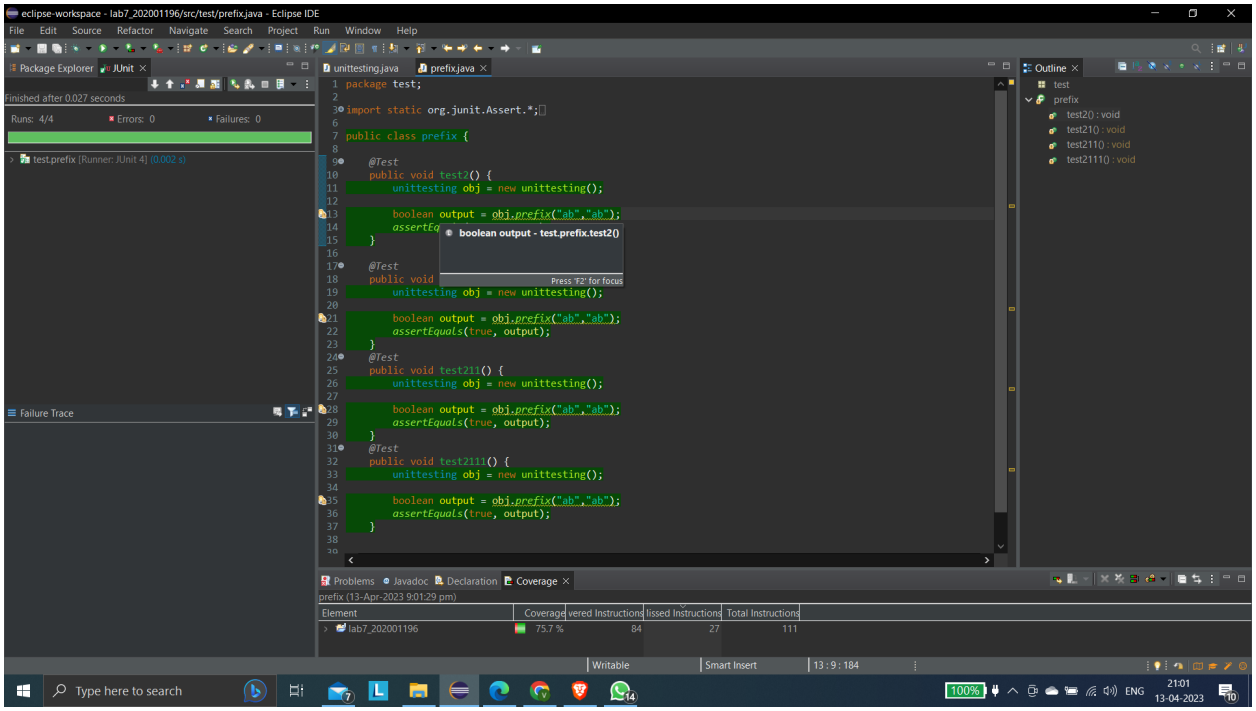
Input	Expected Value
a=0 OR b = 0 OR c = 0	INVALID
a=b=c , where all are positive	EQUILATERAL
a=b (a, b are positive) and c = 0	INVALID
a=b > 0 , c > 0 (c should be less than b)	ISOSCELES
a<b+c , b<a+c , c<a+b	SCALENE
b>a+c (same for a , c)	INVALID

Boundary Value Analysis

Input	**Expected Value**
a=0 , b=0 , c=0	INVALID
a=0,b=1,c=1	INVALID
a=1,b=0,c=1	INVALID
a=1,b=1,c=0	INVALID
a=5,b=1,c=3	INVALID
a=2,b=5,c=3	INVALID
a=b=c=10	EQUILATERAL
a=b=5,c=10	INVALID
a=b=5,c=9	ISOSCELES
a=10,b=c=4	INVALID
a=3,b=4,c=5	SCALENE

Problem 5:


eclipse-workspace - lab7_202001196/src/test/unittesting.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit
Finished after 0.027 seconds
Runs: 4/4 Errors: 0 Failures: 0
test.prefix [Runner: JUnit 4] (0.002 s)
Failure Trace
unittesting.java prefix.java
1 package test;
2
3 public class unittesting {
4
5 public static boolean prefix(String s1, String s2)
6 {
7 if (s1.length() > s2.length())
8 {
9 return false;
10 }
11 for (int i = 0; i < s1.length(); i++)
12 {
13 if (s1.charAt(i) != s2.charAt(i))
14 {
15 return false;
16 }
17 }
18 return true;
19 }
20 }
21
Outline
test
unittesting
prefix(String, String) : boolean
Problems Javadoc Declaration Coverage
prefix (13-Apr-2023 9:01:29 pm)
Element Coverage vered Instructions lissed Instructions Total Instructions
lab7_202001196 75.7 % 84 27 111
Writable Smart Insert 5 : 18 : 62
Type here to search
100% ENG 21:01 13-04-2023


eclipse-workspace - lab7_202001196/src/test/prefix.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit
Finished after 0.027 seconds
Runs: 4/4 Errors: 0 Failures: 0
test.prefix [Runner: JUnit 4] (0.002 s)
Failure Trace
unittesting.java prefix.java
1 package test;
2
3 import static org.junit.Assert.*;
6
7 public class prefix {
8
9 @Test
10 public void test2() {
11 unittesting obj = new unittesting();
12
13 boolean output = obj.prefix("ab","ab");
14 assertEq
15 }
16
17 @Test
18 public void
19 unittesting obj = new unittesting();
20
21 boolean output = obj.prefix("ab","ab");
22 assertEquals(true, output);
23 }
24 @Test
25 public void test211() {
26 unittesting obj = new unittesting();
27
28 boolean output = obj.prefix("ab","ab");
29 assertEquals(true, output);
30 }
31 @Test
32 public void test2111() {
33 unittesting obj = new unittesting();
34
35 boolean output = obj.prefix("ab","ab");
36 assertEquals(true, output);
37 }
38
boolean output - test.prefix.test2()
Press 'F2' for focus
Outline
test
prefix
test2() : void
test21() : void
test211() : void
test2111() : void
Problems Javadoc Declaration Coverage
prefix (13-Apr-2023 9:01:29 pm)
Element Coverage vered Instructions lissed Instructions Total Instructions
lab7_202001196 75.7 % 84 27 111
Writable Smart Insert 13 : 9 : 184
Type here to search
100% ENG 21:01 13-04-2023

Equivalence Partition

S1	S2	Expected Outcome
“”	“”	True
“”	“ab”	True
“a”	“ab”	True
“b”	“ab”	False

Boundary Value Analysis

S1	S2	Expected Outcome
“”	“abc”	True
“b”	“abc”	False
“abc”	“abc”	True
“abd”	“abc”	False
“abcd”	“abc”	False
“a”	“”	False
“a”	“a”	True
“a”	“b”	False
“xxxxx”	“xxyxxxxx”	False

Problem 6:

a) Equivalence Class for the System:

Class1: Invalid (0 length OR negative length)
Class2: Non-Triangle (sum of two side is not greater than remain side)
Class3: Equilateral Triangle (All Side are Equal)
Class4: Scalene (All Side are different)
Class5: Isosceles (Two sides are equal)
Class6: Right-Angle Triangle (according to Pythagoras Theorem)

b) Test Cases:

Class 1 : a=0,b=0,c=0 ; a=1,b=0,c=0 ; a=-5,b=4,c=4
Class 2 : a=10,b=4,c=5
Class 3 : a=b=c=50
Class 4: a=3,b=4,c=5
Class 5: a=b=5 , c= 9
Class 6: a=9,b=16,c=25

c) A + B > C

a=5,b=6,c=10

Here sum of two side is greater than longest side that means it is Scalene Triangle

a=3,b=5,c=9

Here the sum of the first two sides is less than the longest side , that means it will not make a triangle.

d) A = C

a= 10 , b = 19, c = 10

Here a and c both are equal, which means it is an Isosceles Triangle.

a=10 , b = 20, c = 10

Here a and c are equal but one catch is that (b>=a+c) which says that it is not a valid triangle.

e) A = B = C

a=b=c= INT_MAX / DOUBLE_MAX

f) A^2 + B^2 = C^2

a=INT_MAX , b=DOUBLE_MAX , c = INT_MAX;
Here a and c are INT_MAX , so the square of it will be out of bound.

a=3,b=4,c=5

g) For the non-triangle case

a=1, b=5 , c= 10

a=45,b=46 , c = 100

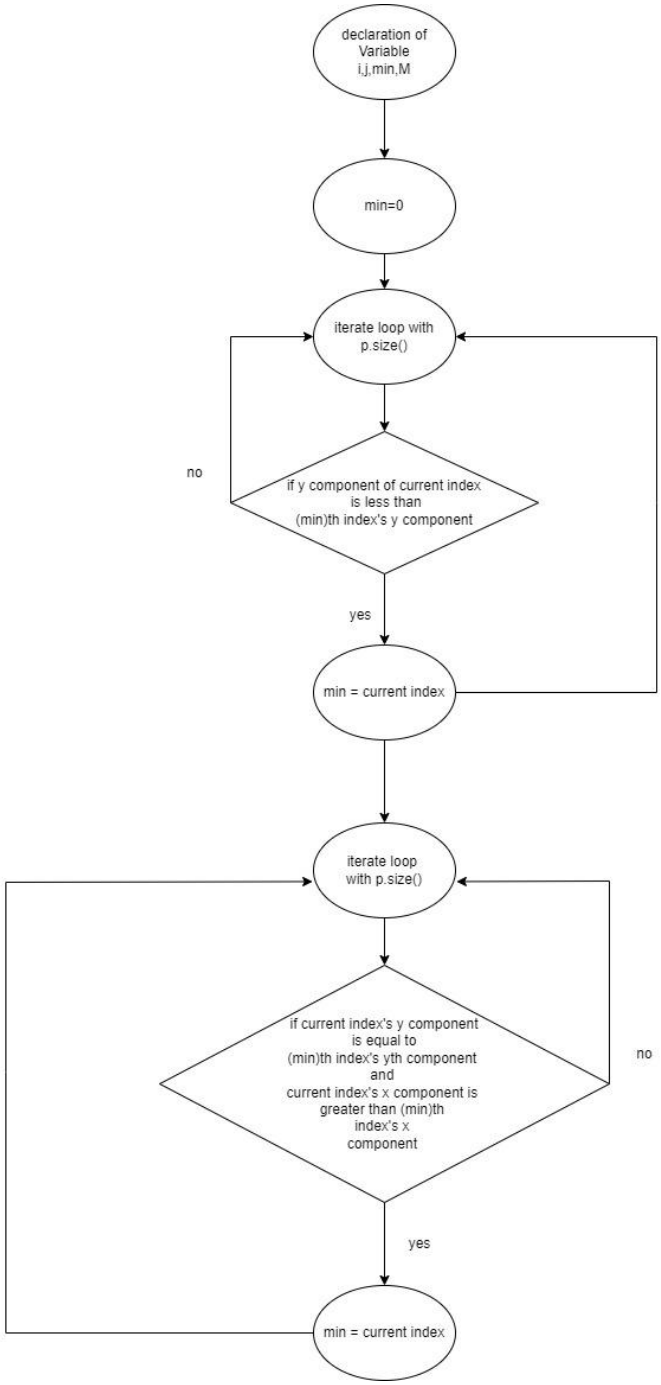
h) For non-positive input,

a=-10,b=40,c=-45

a=-4,b=5,c=10

Section - B

1) Control Flow Graph:



2) Construct test sets for your flow graph that are adequate for the following criteria:

a) Statement Courage:

Here the motive is that : test cases should cover as possible statements of code.

Test cases:
P : (0,0) , (4,0) , (1,1)
P : (0,5) , (1,5) , (2,5) , (0,4) ,(2,4) , (0,0) , (1,0) , (2,0) , (3,0)
Generally , y with the same x should be on a large scale such that the number of statements will increase.

b) Branch Coverage:

Here , our motive is to design a test case such that it should pass from as many branches.

Test Cases:
P : (0,0) , (4,0) , (1,1)
P : (0,5) , (1,5) , (2,5) , (0,4) ,(2,4) , (0,0) , (1,0) , (2,0) , (3,0)
P : (4,10) , (4,9) , (4,8) , (3,8) , (2,8) , (5,8) , (6,8) , (7,8) , (10,8) , (12,8)

c) Basic Condition Coverage:

Here , our motive is to design a test case such that it should pass from as possible boolean Condition.

Test Case:

P: (0,0) , (10,10) , (20,10)
P : (0,0) , (2,0) , (3,0)
P : (4,50) , (5,50) , (6,50) , (7,50) , (8,50) , (9,50) , (10,50) , (11,50) , (40,50)

Generally , for minimum y , cover as much possible x , such that in the second loop it checks both boolean operations.