IT314 - Lab8
Vraj Patel - 202201401

Q.1

Tester action and input data(date,month,year)	Expected output
Equivalence partitioning	
1, 1, 1900	Previous Date: 31/12/1899
1, 2, 1900	Previous Date: 31/01/1900
1, 1, 2015	Previous Date: 31/12/2014
1, 2, 2015	Previous Date: 31/01/2015
1, 3, 2015	Previous Date: 29/02/2015 (leap year)
1, 1, 1899	An Error message
1, 13, 1900	An Error message
Boundary value analysis	
1, 1, 1900	Previous Date: 31/12/1899
1, 31, 2015	Previous Date: 30/11/2015
1, 2, 1900	Previous Date: 31/01/1900
1, 2, 2015	Previous Date: 31/01/2015
1, 0, 2015	An Error message
1, 12, 2015	Previous Date: 30/11/2015
1, 31, 2014	Previous Date: 30/11/2014

Q.2

P1

Tester action and input data(date,month,year)	Expected output
Equivalence partitioning	
Search for 5 in {1, 2, 3, 4, 5}	4 (index of 5)
Search for 1 in {1, 2, 3, 4, 5}	0 (index of 1)
Search for 10 in {1, 2, 3, 4, 5}	-1 (not found)
Search for -1 in {1, -1, 2, 3}	1 (index of -1)
Search for 3 in {}	-1 (empty array)
Boundary value analysis	
Search for the first element in {5}	0 (index of 5)
Search for the last element in {1, 2, 3, 4, 5}	4 (index of 5)
Search for a value less than the minimum in {1, 2, 3}	-1 (not found)
Search for a value greater than the maximum in {1, 2, 3}	-1 (not found)
Search for 2 in {1, 2, 3}	1 (index of 2)
Search in a single element array {3}	0 (index of 3)
Search for a non-existent value in {3}	-1 (not found)

P2

Equivalence Partitioning Test Cases

Tester Action and Input Data	Expected Outcome
Count occurrences of 2 in `{1, 2, 3, 2, 4}`	2
Count occurrences of 1 in `{1, 1, 1, 1}`	4
Count occurrences of 3 in `{4, 5, 6}`	0
Count occurrences of -1 in `{1, -1, -1, 2}`	2
Count occurrences of 5 in `[]`	0

Boundary Value Analysis Test Cases

Tester Action and Input Data	Expected Outcome
Count occurrences of the first element in `{3}`	1
Count occurrences of the last element in `{1, 2, 3}`	1
Count occurrences of a value less than the minimum in `{1, 2, 3}`	0
Count occurrences of a value greater than the maximum in `{1, 2, 3}`	0
Count occurrences of 2 in `{1, 2, 2, 2, 3}`	3
Count occurrences in a single-element array `{2}`	1

Tester Action and Input Data	Expected Outcome
Count occurrences of a non-existent value in `{2}`	0

P3

Equivalence Partitioning Test Cases

Tester Action and Input Data	Expected Outcome
Search for 5 in `{1, 2, 3, 4, 5}`	4
Search for 1 in `{1, 2, 3, 4, 5}`	0
Search for 10 in `{1, 2, 3, 4, 5}`	-1
Search for -1 in `{-2, -1, 0, 1, 2}`	1
Search for 3 in `{1, 2, 3}`	2
Search for 6 in `{1, 2, 3, 4, 5}`	-1

Boundary Value Analysis Test Cases

Tester Action and Input Data	Expected Outcome
Search for the first element in `{3}`	0
Search for the last element in `{1, 2, 3, 4, 5}`	4
Search for a value less than the minimum in `{1, 2, 3}`	-1
Search for a value greater than the maximum in `{1, 2, 3}`	-1
Search for 2 in `{1, 2, 2, 2, 3}`	1 (first occurrence)
Search in a single element array `{2}`	0

Tester Action and Input Data	Expected Outcome
Search for a non-existent value in `{1, 2, 3}`	-1

P4

Equivalence Partitioning Test Cases

Tester Action and Input Data	Expected Outcome
Inputs: `3, 3, 3`	EQUILATERAL (0)
Inputs: `3, 3, 5`	ISOSCELES (1)
Inputs: `3, 4, 5`	SCALENE (2)
Inputs: `1, 2, 3`	INVALID (3)
Inputs: `0, 2, 2`	INVALID (3)
Inputs: `-1, 2, 2`	INVALID (3)

Boundary Value Analysis Test Cases

Tester Action and Input Data	Expected Outcome
Inputs: `1, 1, 1`	EQUILATERAL (0)
Inputs: `1, 1, 2`	ISOSCELES (1)
Inputs: `2, 2, 3`	ISOSCELES (1)
Inputs: `1, 2, 2`	ISOSCELES (1)
Inputs: `1, 2, 3`	INVALID (3)
Inputs: `3, 4, 5`	SCALENE (2)

Tester Action and Input Data	Expected Outcome
Inputs: `3, 5, 9`	INVALID (3)

P5

Equivalence Partitioning Test Cases

Tester Action and Input Data	Expected Outcome
Inputs: `“abc”, “abcdef”`	true
Inputs: `“abc”, “abc”`	true
Inputs: `“abc”, “xyzabc”`	false
Inputs: `“abc”, “a”`	false
Inputs: `“”, “abc”`	true
Inputs: `“abc”, “”`	false
Inputs: `“abcd”, “abcde”`	true
Inputs: `“abcdef”, “abc”`	false

Boundary Value Analysis Test Cases

Tester Action and Input Data	Expected Outcome
Inputs: `“”, “”`	true
Inputs: `“a”, “a”`	true
Inputs: `“a”, “ab”`	true

Tester Action and Input Data	**Expected Outcome**
Inputs: `“ab”, “a”`	false
Inputs: `“a”, “abc”`	true
Inputs: `“abc”, “abcd”`	true
Inputs: `“abcd”, “ab”`	false

P6: Triangle Program (Floating-Point Inputs)

a) Equivalence Classes

The following are the identified equivalence classes for the triangle program:

1. **Equilateral Triangle**: All sides are equal (A = B = C).
2. **Isosceles Triangle**: Two sides are equal (A = B, B = C, or A = C).
3. **Scalene Triangle**: No sides are equal (A ≠ B ≠ C).
4. **Right-angled Triangle**: Follows the property A2+B2=C2A^2 + B^2 = C^2A2+B2=C2.
5. **Invalid Triangle**: The sum of two sides is less than or equal to the third.

b) Test Cases to Cover the Identified Equivalence Classes

Input (A, B, C)	**Expected Outcome**	**Equivalence Class**
3.0, 3.0, 3.0	Equilateral triangle	Equilateral Triangle
5.0, 5.0, 3.0	Isosceles triangle	Isosceles Triangle
6.0, 7.0, 8.0	Scalene triangle	Scalene Triangle
2.0, 5.0, 10.0	Invalid triangle	Invalid Triangle
3.0, 4.0, 5.0	Scalene (right-angled)	Right-angled Triangle

c) Boundary Condition for Scalene Triangle: A + B > C

For the boundary condition A + B > C, the following test cases are used to verify the boundary:

Input (A, B, C)	**Expected Outcome**	**Boundary Condition**
3.0, 4.0, 7.0	Invalid triangle	A + B = C (boundary case)
3.0, 4.0, 6.9	Scalene triangle	A + B > C (valid case)
3.0, 4.0, 7.1	Scalene triangle	A + B < C (valid case)

d) Boundary Condition for Isosceles Triangle: A = C

For the boundary condition A = C, the following test cases verify the boundary:

Input (A, B, C)	Expected Outcome	Boundary Condition
5.0, 7.0, 5.0	Isosceles triangle	A = C (valid case)
5.0, 7.0, 4.9	Scalene triangle	A ≠ C (near-boundary case)
5.0, 7.0, 5.1	Scalene triangle	A ≠ C (near-boundary case)

e) Boundary Condition for Equilateral Triangle: A = B = C

For the boundary condition A = B = C, the following test cases verify the boundary:

Input (A, B, C)	Expected Outcome	Boundary Condition
3.0, 3.0, 3.0	Equilateral triangle	A = B = C (valid case)
3.0, 3.0, 2.9	Isosceles triangle	A ≠ B ≠ C (near-boundary)
3.0, 3.0, 3.1	Isosceles triangle	A ≠ B ≠ C (near-boundary)

f) Boundary Condition for Right-Angled Triangle: $A^2 + B^2 = C^2$

For the boundary condition $A^2 + B^2 = C^2$ (right-angle), the following test cases verify the boundary:

Input (A, B, C)	Expected Outcome	Boundary Condition
3.0, 4.0, 5.0	Scalene (right-angled)	$A^2 + B^2 = C^2$ (valid case)
3.0, 4.0, 5.1	Scalene triangle	$A^2 + B^2 < C^2$ (near-boundary)
3.0, 4.0, 4.9	Scalene triangle	$A^2 + B^2 > C^2$ (near-boundary)

g) Non-Triangle Case (Invalid Triangle)

For non-triangle cases, the following test cases explore the boundary conditions where the sum of two sides is less than or equal to the third:

Input (A, B, C)	Expected Outcome	Boundary Condition
1.0, 2.0, 3.0	Invalid triangle	A + B = C (boundary case)
1.0, 2.0, 3.1	Scalene triangle	A + B < C (valid case)
1.0, 2.0, 2.9	Scalene triangle	A + B > C (valid case)

h) Non-Positive Input (Invalid Case)

For non-positive input values, the following test cases ensure invalid inputs are handled correctly:

Input (A, B, C)	Expected Outcome	Boundary Condition
0.0, 3.0, 3.0	Invalid triangle	Non-positive value
-1.0, 2.0, 3.0	Invalid triangle	Negative value
3.0, 0.0, 4.0	Invalid triangle	Non-positive value
3.0, -2.0, 4.0	Invalid triangle	Negative value