IT-314
Software Engineering

LAB-8

Dev Vyas- 202201453

Q1. Previous Date Problem

Equivalence Classes:

1. **Valid dates:** Day, Month, and Year within valid ranges.
2. **Invalid dates:**

   Day, Month, or Year out of valid range.

   February 29 on a non-leap year.

Boundary Conditions:

- First day of the month.
- Last day of the month.
- Leap year boundaries.
- Minimum/Maximum year boundaries.

Tester Action & Input	Expected Outcome	Equivalence Class	BVA
2/2/2000	1/2/2000	Valid date	
32/1/2000	"Invalid date"	Invalid day	
1/13/2000	"Invalid date"	Invalid month	
1/1/1899	"Invalid date"	Invalid year	X
29/2/2000	28/2/2000	Leap year valid	
1/1/1900	"Invalid date"	Minimum boundary	X
31/12/2015	30/12/2015	Maximum boundary	X
1/3/2000	29/2/2000	First day boundary	X
31/1/2000	30/1/2000	Last day boundary	X

Q2. Linear Search

Equivalence Classes:

1. **Element found:** Value v exists in array a.
2. **Element not found:** Value v does not exist in array a.

Boundary Conditions:

- Array is empty.
- Single element in the array.
- Value v is at the first or last position.

Tester Action & Input	Expected Outcome	Equivalence Class	BVA
Search 5 in [1, 3, 5, 7, 9]	Returns 2	Element found	
Search 10 in [1, 2, 3]	Returns -1	Element not found	
Search 3 in []	Returns -1	Empty array	X
Search 3 in [3]	Returns 0	Single element	X
Search 1 in [1, 2, 3]	Returns 0	First position	X
Search 3 in [1, 2, 3]	Returns 2	Last position	X

Q3. Count Items

Equivalence Classes:

1. **Multiple occurrences:** Value v appears multiple times.
2. **Single occurrence:** Value v appears exactly once.
3. **No occurrence:** Value v does not appear in array a.

Boundary Conditions:

- Array is empty.
- Single element in array.

Tester Action & Input	Expected Outcome	Equivalence Class	BVA
Count 5 in [1, 5, 5, 5, 9]	3	Multiple occurrences	
Count 1 in [1, 2, 3, 4]	1	Single occurrence	
Count 10 in [1, 2, 3]	0	No occurrence	
Count 3 in []	0	Empty array	X
Count 3 in [3]	1	Single element	X

Q4. Binary Search

Equivalence Classes:

1. **Element found:** Value v exists in the array a.
2. **Element not found:** Value v does not exist in the array a.

Boundary Conditions:

- Array is empty.
- Single element in the array.
- Value v at first or last position.

Tester Action & Input	Expected Outcome	Equivalence Class	BVA
Search 5 in [1, 3, 5, 7, 9]	Returns 2	Element found	
Search 10 in [1, 2, 3, 4]	Returns -1	Element not found	
Search 3 in []	Returns -1	Empty array	X
Search 3 in [3]	Returns 0	Single element	X
Search 1 in [1, 2, 3]	Returns 0	First position	X
Search 3 in [1, 2, 3]	Returns 2	Last position	X

Q5. Triangle Classification

Equivalence Classes:

1. **Equilateral triangle:** All sides are equal.
2. **Isosceles triangle:** Two sides are equal.
3. **Scalene triangle:** No sides are equal.
4. **Invalid triangle:** The sum of two sides is not greater than the third.

Boundary Conditions:

- Minimum length of sides.
- Maximum length of sides.
- The sum of two sides equals the third (invalid).

Tester Action & Input	Expected Outcome	Equivalence Class	BVA
3, 3, 3	Equilateral	Equilateral	
3, 3, 4	Isosceles	Isosceles	
3, 4, 5	Scalene	Scalene	
1, 2, 3	Invalid	Invalid triangle	
0, 3, 4	Invalid	Invalid triangle	X
1, 1, 2	Invalid (sum of sides equals the third)	Invalid triangle	X

Q6

a) Identify the Equivalence Classes for the System

1. **Valid Triangles:**
    - **Equilateral Triangle:** All three sides are equal.
    - **Isosceles Triangle:** Two sides are equal.
    - **Scalene Triangle:** No sides are equal.
    - **Right-Angled Triangle:** Pythagoras' theorem holds ($A^2 + B^2 = C^2$).
2. **Invalid Triangles:**
    - **Non-Triangles:** Sum of two sides is not greater than the third ($A + B \leq C$ or similar cases).
    - **Negative or Zero Sides:** Triangle sides must be positive numbers.

---

b) Identify Test Cases to Cover the Identified Equivalence Classes

Equivalence Class	Tester Action & Input (A, B, C)	Expected Output	Test Case Description
Equilateral Triangle	5.0, 5.0, 5.0	Equilateral	All sides equal
Isosceles Triangle	5.0, 5.0, 8.0	Isosceles	Two sides equal
Scalene Triangle	5.0, 6.0, 7.0	Scalene	No sides equal
Right-Angled Triangle	3.0, 4.0, 5.0	Right-angled	Pythagoras holds ($A^2 + B^2 = C^2$)
Non-Triangle (Invalid)	1.0, 2.0, 3.0	Invalid Triangle	A + B = C case
Non-Triangle (Invalid)	1.0, 10.0, 2.0	Invalid Triangle	A + B < C case
Negative Side (Invalid)	-3.0, 4.0, 5.0	Invalid Triangle	Negative side length
Zero Side (Invalid)	0.0, 4.0, 5.0	Invalid Triangle	Zero side length

---

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

For the **scalene triangle**, one key boundary condition is that the sum of two sides must be greater than the third side. Test cases to verify this condition are:

Test Case Description	Input (A, B, C)	Expected Output
A + B = C boundary	1.0, 2.0, 3.0	Invalid Triangle
A + B slightly greater than C	3.0, 4.0, 5.0	Scalene
A + B much greater than C	6.0, 8.0, 5.0	Scalene

---

d) Boundary Condition: A = C (Isosceles Triangle)

For the **isosceles triangle**, one boundary condition is when two sides are equal, particularly A = C. Test cases to verify this condition are:

Test Case Description	Input (A, B, C)	Expected Output
A = C (isosceles)	5.0, 8.0, 5.0	Isosceles
A = C with other side larger	5.0, 9.0, 5.0	Isosceles
A = C with other side smaller	5.0, 3.0, 5.0	Isosceles

---

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

For the **equilateral triangle**, all three sides must be equal. The boundary condition for this is when A = B = C. Test cases are:

Test Case Description	Input (A, B, C)	Expected Output
A = B = C	3.0, 3.0, 3.0	Equilateral
A = B = C (larger value)	10.0, 10.0, 10.0	Equilateral

---

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

For the **right-angled triangle**, the condition $A^2 + B^2 = C^2$ must hold. Test cases to verify this are:

Test Case Description	Input (A, B, C)	Expected Output
Pythagorean Triple (3, 4, 5)	3.0, 4.0, 5.0	Right-angled
Larger Pythagorean Triple	6.0, 8.0, 10.0	Right-angled

Close to Pythagorean but invalid	3.0, 4.0, 6.0	Scalene

---

g) Non-Triangle Case: Exploring the Boundary

For the **non-triangle case**, the sum of two sides must be greater than the third side. The boundary where A + B = C or A + B < C is invalid. Test cases are:

Test Case Description	Input (A, B, C)	Expected Output
A + B = C	1.0, 2.0, 3.0	Invalid Triangle
A + B < C	1.0, 1.0, 3.0	Invalid Triangle

---

h) Non-Positive Input

Non-positive inputs (i.e., zero or negative side lengths) are invalid for triangle classification. Test cases to explore this boundary are:

Test Case Description	Input (A, B, C)	Expected Output
Negative side length (A < 0)	-1.0, 2.0, 2.0	Invalid Triangle
Negative side length (B < 0)	1.0, -2.0, 2.0	Invalid Triangle
Zero side length (A = 0)	0.0, 4.0, 5.0	Invalid Triangle

Final Python Program for Testing:

```
def classify_triangle(A, B, C):
    # Check for invalid triangle (Non-positive side)
    if A <= 0 or B <= 0 or C <= 0:
        return "Invalid Triangle"

    # Check if it's an invalid triangle (triangle inequality)
    if A + B <= C or A + C <= B or B + C <= A:
        return "Invalid Triangle"
```

```
    # Check for equilateral triangle
    if A == B == C:
        return "Equilateral"

    # Check for isosceles triangle
    if A == B or B == C or A == C:
        return "Isosceles"

    # Check for right-angled triangle (Pythagoras theorem)
    if round(A**2 + B**2, 5) == round(C**2, 5) or round(A**2 + C**2,
5) == round(B**2, 5) or round(B**2 + C**2, 5) == round(A**2, 5):
        return "Right-angled"

    # Otherwise, it's a scalene triangle
    return "Scalene"

# Test cases based on equivalence classes and boundary conditions
test_cases = [
    (5.0, 5.0, 5.0),   # Equilateral
    (5.0, 5.0, 8.0),   # Isosceles
    (5.0, 6.0, 7.0),   # Scalene
    (3.0, 4.0, 5.0),   # Right-angled
    (1.0, 2.0, 3.0),   # Non-triangle (A + B = C)
    (-3.0, 4.0, 5.0),  # Invalid (negative side)
    (0.0, 4.0, 5.0),   # Invalid (zero side)
]

for A, B, C in test_cases:
    print(f"Input: {A}, {B}, {C} -> Output: {classify_triangle(A, B,
C)}")
```

This program and test suite cover all the required equivalence classes and boundary conditions for floating-point triangle classification.