Dhirubhai Ambani
Institute of Information and Communication Technology

Subject: Software Engineering

Subject code: IT314

Name: Darpan Lunagariya

ID: 2022 01462

Lab 8 - Functional Testing (Black Box)

Question 1:

We have 3 inputs, date, month and year. According to given problem statement, we have following equivalence classes:

Equivalence Classes:

No.	Value	Valid / Invalid
E1	1<= month <=12	Valid
E2	month < 1	Invalid
E3	month > 12	Invalid
E4	Valid Date (According to month)	Valid
E5	Date < 1	Invalid
E6	Date > 31	Invalid
E7	1900 <= Year <= 2015	Valid
E8	Year < 1900	Invalid
E9	Year > 2015	Invalid

Test Cases:

No.	Input Values (date, month, year)	Equivalence Classes Covered	Result (date, month, year)
1	(15, 8, 2012)	E1, E4, E7	(14, 8, 2012)
2	(35, 8, 2012)	E1, E6, E7	Invalid Date
3	(-3, 8, 2012)	E1, E5, E7	Invalid Date
4	(14, 13, 1995)	E3, E4, E7	Invalid Month
5	(43, -8, 1940)	E2, E6, E7	Invalid Date and Month
6	(1, 1, 1900)	E1, E4, E7	(31, 12, 1899)
7	(31, 12, 2015)	E1, E4, E7	(30, 12, 2015)
8	(29, 2, 2000)	E1, E4, E7	(28, 2, 2000)
9	(29, 2, 2011)	E1, E4, E7	Invalid Date
10	(1, 3, 2012)	E1, E4, E7	(29, 2, 2012)
11	(1, 5, 2010)	E1, E4, E7	(30, 4, 2010)
12	(31, 6, 2014)	E1, E6, E7	Invalid Date
13	(0, 7, 2005)	E1, E5, E7	Invalid Date
14	(15, 0, 2011)	E2, E4, E7	Invalid Month
15	(15, 8, 1850)	E1, E4, E8	Invalid Year
16	(15, 8, 2020)	E1, E4, E9	Invalid Year
17	(31, 4, 2015)	E1, E6, E7	Invalid Date
18	(1, 1, 1899)	E1, E4, E8	Invalid Year
19	(30, 9, 1900)	E1, E4, E7	(29, 9, 1900)
20	(1, 1, 2016)	E1, E4, E9	Invalid Year

Question 2:

Problem 1: Linear Search

Equivalence Classes:

No.	Value
E1	v belongs to array a
E2	v does not belongs to array a

Test Cases:

Tester Action and Input Data	Expected Outcome
Equivalence Partitioning	
v = 5, a = [1, 3, 5, 7, 9]	2
v = 10, a = [2, 4, 6, 8]	-1
v = 3, a = [3]	0
v = 0, a = [1, 2, 3]	-1
Boundary Value Analysis	
v = 1, a = [1]	0
v = 1, a = [1, 2, 3]	0
v = 3, a = [1, 2, 3]	2
v = 4, a = [1, 2, 3, 4]	3
v = 5, a = [1, 2, 3, 4, 5]	4

Problem 2: Count Number of Occurrence

Tester Action and Input Data	Expected Outcome
Equivalence Partitioning	
v = 5, a = [1, 3, 5, 7, 9]	1
v = 10, a = [2, 4, 6, 8]	0
v = 3, a = [3, 3, 3]	3
v = 0, a = []	0
Boundary Value Analysis	
v = 1, a = [1]	1
v = 1, a = [1, 1, 1]	3
v = 3, a = [1, 2, 3, 3, 3]	3
v = 0, a = [1, 2, 3]	0
v = 4, a = [1, 2, 3, 4, 4, 4]	3
v = 5, a = [1, 2, 3, 4, 5, 5, 5, 5]	4

Generating test cases using boundary value analysis is not possible. Because, element can either exist in an array or not, there cannot be a boundary value possible.

Problem 3: Binary Search

Equivalence Classes:

Since in assumption, it is mentioned that an array is assumed to be sorted in non-decreasing order, it is not required to make an equivalence class that represents if an array is sorted or not.

Therefore, resulting equivalence classes are mentioned below.

No.	Value
E1	v belongs to array a
E2	v does not belongs to array a

Test Cases:

Tester Action and Input Data	Expected Outcome
Equivalence Partitioning	
v = 5, a = [1, 3, 5, 7, 9]	2
v = 10, a = [2, 4, 6, 8, 10]	-1
v = 3, a = [1, 3, 5, 7, 9]	1
v = 0, a = [1, 3, 5, 7, 9]	-1
Boundary Value Analysis	
v = 1, a = [1]	0
v = 1, a = [1, 3, 5, 7, 9]	0
v = 9, a = [1, 3, 5, 7, 9]	4
v = 0, a = [1, 3, 5, 7, 9]	-1
v = 5, a = [1, 3, 5, 7, 9]	2
v = 10, a = [1, 3, 5, 7, 9, 10]	5

Problem 4: Type of Triangle

Equivalence Classes:

Since in assumption, it is mentioned that an array is assumed to be sorted in non-decreasing order, it is not required to make an equivalence class that represents if an array is sorted or not.

Therefore, resulting equivalence classes are mentioned below.

No.	Value
E1	Length of side 1 >= 1
E2	Length of side 1 < 1
E3	Length of side 2 >= 1
E4	Length of side 2 < 1
E5	Length of side 3 >= 1
E6	Length of side 3 < 1

Test Cases:

Tester Action and Input Data	Expected Outcome
Equivalence Partitioning	
a = 3, b = 3, c = 3	EQUILATERAL
a = 3, b = 3, c = 4	ISOSCELES
a = 3, b = 4, c = 5	SCALENE
a = 1, b = 2, c = 3	INVALID
a = 10, b = 2, c = 3	INVALID
a = 3, b = 10, c = 3	INVALID
Boundary Value Analysis	
a = 1, b = 1, c = 1	EQUILATERAL
a = 1, b = 1, c = 2	ISOSCELES
a = 2, b = 2, c = 3	ISOSCELES
a = 3, b = 3, c = 3	EQUILATERAL
a = 3, b = 4, c = 5	SCALENE
a = 5, b = 5, c = 10	INVALID

Problem 5: Check whether str1 is prefix of str2

Test Cases:

Tester Action and Input Data	Expected Outcome
Equivalence Partitioning	
s1 = "abc", s2 = "abcdef"	true
s1 = "abc", s2 = "def"	false
s1 = "", s2 = "abcdef"	true
s1 = "abc", s2 = "abc"	true
Boundary Value Analysis	
s1 = "a", s2 = "a"	true
s1 = "a", s2 = "abc"	true
s1 = "abc", s2 = "abc"	true
s1 = "abcdef", s2 = "abc"	false
s1 = "a", s2 = ""	false
s1 = "", s2 = ""	true

Problem 6: Type of Triangle

Test Cases:

a) Equivalence Classes:

Equivalence Class	Description
EC1	Scalene triangle (all sides are different lengths)
EC2	Isosceles triangle (two sides are equal)
EC3	Equilateral triangle (all sides are equal)
EC4	Right-angled triangle (satisfies Pythagorean theorem)
EC5	Non-triangle (sum of any two sides is less than the third side)
EC6	Non-positive input

b) Test Cases for Equivalence Classes:

Test Case	Equivalence Class
TC1: A = 3.0, B = 4.0, C = 5.0	EC1
TC2: A = 3.0, B = 3.0, C = 4.0	EC2
TC3: A = 3.0, B = 3.0, C = 3.0	EC3
TC4: A = 3.0, B = 4.0, C = 5.0 (where A^2 + B^2 = C^2)	EC4
TC5: A = 1.0, B = 2.0, C = 3.0	EC5
TC6: A = -1.0, B = 2.0, C = 3.0	EC6

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

Test Case	Description
TC7: A = 1.0, B = 2.0, C = 3.0	Minimum possible scalene triangle
TC8: A = 100.0, B = 100.1, C = 200.0	Maximum possible scalene triangle

d) Boundary Cases for Isosceles Triangle (A = C):

Test Case	Description
TC9: A = 3.0, B = 4.0, C = 3.0	Minimum possible isosceles triangle
TC10: A = 100.0, B = 200.0, C = 100.0	Maximum possible isosceles triangle

e) Boundary Cases for Equilateral Triangle (A = B = C):

Test Case	Description
TC11: A = 1.0, B = 1.0, C = 1.0	Minimum possible equilateral triangle
TC12: A = 100.0, B = 100.0, C = 100.0	Maximum possible equilateral triangle

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

Test Case	Description
TC13: A = 3.0, B = 4.0, C = 5.0	Minimum possible right-angled triangle
TC14: A = 100.0, B = 100.0, C = 141.42 (rounded to 2 decimal places)	Maximum possible right-angled triangle

g) Boundary Cases for Non-Triangle (A + B <= C):

Test Case	Description
TC15: A = 1.0, B = 2.0, C = 3.0	Minimum non-triangle case
TC16: A = 100.0, B = 100.0, C = 200.0	Maximum non-triangle case

h) Test Cases for Non-Positive Input:

Test Case	Description
TC17: A = 0.0, B = 2.0, C = 3.0	Zero input
TC18: A = -1.0, B = 2.0, C = 3.0	Negative input