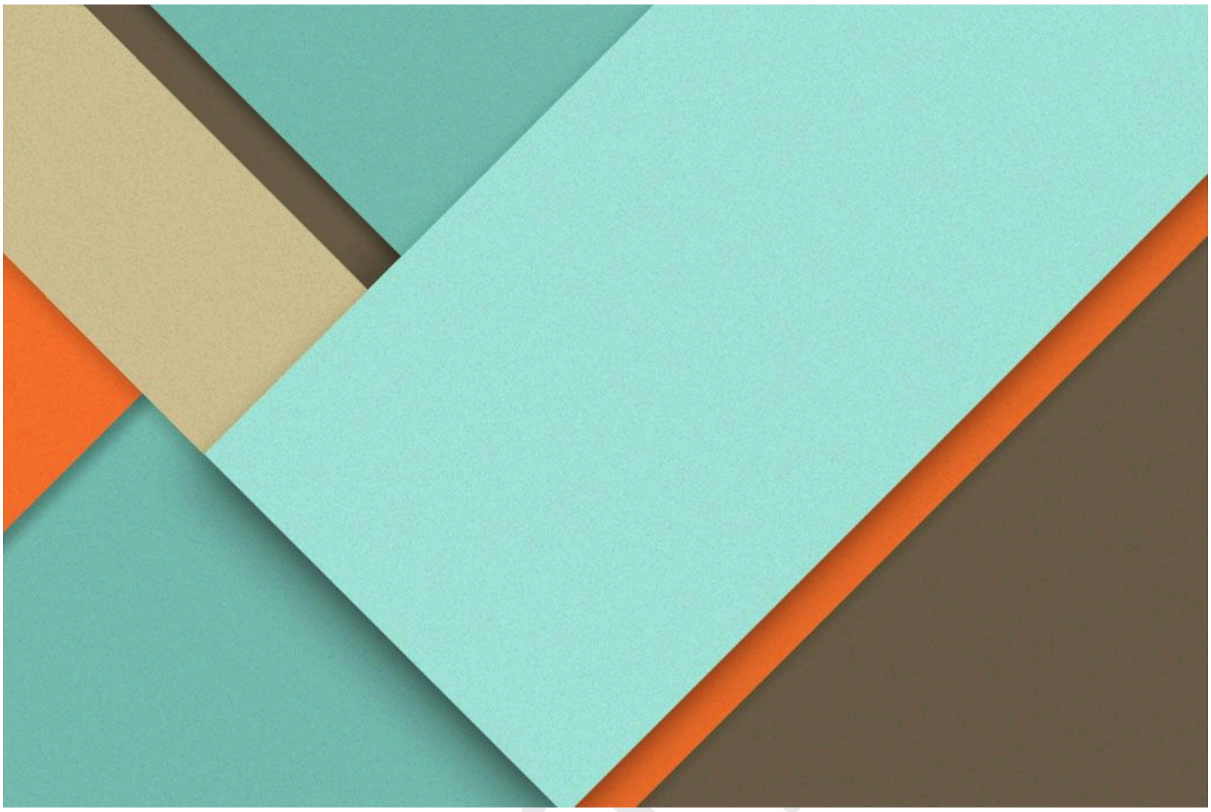

IT314 : Software Engineering

LAB - 08 : Functional Testing (Black-Box)

Name : Savani Vedant

ID : 202201178

Question 1 :

In this Question we are having 3 inputs, date, month and year. According to given problem statement, we can have following equivalence classes:

Equivalence Class :

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

The Program will give output either a previous date or an error message indicating an invalid date according to the test case.

Test Cases :

The below given test cases cover all the test cases for equivalence class partitioning and Boundary Value Analysis (BVA).

Input Values (date, month, year)	Equivalence Classes Covered	Result (date, month, year)
(15, 8, 2012)	E1, E4, E7	(14, 8, 2012)
(35, 8, 2012)	E1, E6, E7	Invalid Date
(-3, 8, 2012)	E1, E5, E7	Invalid Date
(14, 13, 1995)	E3, E4, E7	Invalid Month
(43, -8, 1940)	E2, E6, E7	Invalid Date and Month
(1, 1, 1900)	E1, E4, E7	(31, 12, 1899)
(31, 12, 2015)	E1, E4, E7	(30, 12, 2015)
(29, 2, 2000)	E1, E4, E7	(28, 2, 2000)
(29, 2, 2011)	E1, E4, E7	Invalid Date
(1, 3, 2012)	E1, E4, E7	(29, 2, 2012)
(1, 5, 2010)	E1, E4, E7	(30, 4, 2010)
(31, 6, 2014)	E1, E6, E7	Invalid Date
(0, 7, 2005)	E1, E5, E7	Invalid Date
(15, 0, 2011)	E2, E4, E7	Invalid Month
(15, 8, 1850)	E1, E4, E8	Invalid Year
(15, 8, 2020)	E1, E4, E9	Invalid Year
(31, 4, 2015)	E1, E6, E7	Invalid Date
(1, 1, 1899)	E1, E4, E8	Invalid Year

Question 2 :

P1: Linear Search

Classes for Equivalence Partitioning :

No.	Value
E1	v belongs to array a
E2	v does not belongs to array a
E3	array a is empty
E4	array contains duplicates

Classes for Boundary Value Analysis:

No.	Value
B1	v is the first element of array a
B2	v is the last element of array a
B3	array a contains only one element

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 = []	-1
v = 0, a = [1, 2, 3,3]	2
Boundary Value Analysis	
v = 1, a = [1]	0
v = 3, a = [1, 2, 3]	2
v = 3, a = [1, 2, 3]	2

P2: Count Item

Classes for Equivalence Partitioning :

No.	Value
E1	v belongs to array a
E2	v does not belongs to array a
E3	array a is empty

Classes for Boundary Value Analysis:

No.	Value
B1	v is the first element of array a
B2	v is the last element of array a
B3	array a contains element v no. of times the size of array

Test Cases:

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 = []	0
Boundary Value Analysis	
v = 1, a = [1,2,3,4]	1
v = 3, a = [1, 2, 3]	1
v = 3, a = [3,3, 3]	3

P3: Binary Search

Classes for Equivalence Partitioning :

No.	Value
E1	v belongs to array a
E2	v does not belongs to array a
E3	array a is empty

Classes for Boundary Value Analysis:

No.	Value
B1	v is the first element of array a
B2	v is the last element of 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 = []	-1
Boundary Value Analysis	
v = 1, a = [1,2,3,4]	1
v = 3, a = [1, 2, 3]	2

P4: Type of Triangle

Classes for Equivalence Partitioning :

No.	Value
E1	All three lengths are equal
E2	Any two sides are equal
E3	Any one side is having length greater than or equal to the sum of other two side
E4	All sides are different but they satisfy triangle inequality

Classes for Boundary Value Analysis:

No.	Value
B1	Any one side is having length equal to the sum of other two sides

Test Cases:

Tester Action and Input Data	Expected Outcome
Equivalence Partitioning	
(2,2,2)	EQUILATERAL (0)
(2,2,3)	ISOSCELES (1)
(10,2,2)	INVALID (3)
(2,3,4)	SCALENE (2)
Boundary Value Analysis	
(5,2,3)	INVALID (3)

P5: Prefix Match

Classes for Equivalence Partitioning :

No.	Value
E1	s1 is prefix of s2
E2	s1 is not prefix of s2
E3	s1 is an empty string and s2 is not
E4	s1 is longer than s2
E5	s1 equal to s2
E6	Both the string are empty

Classes for Boundary Value Analysis:

No.	Value
B1	Single character prefix
B2	String s2 is empty

Test Cases:

Tester Action and Input Data	Expected Outcome
Equivalence Partitioning	
(“pre”, “prefix”)	TRUE
(“not”, “prefix”)	FALSE
(“”, “notempty”)	TRUE
(“longprefix”, “long”)	FALSE
(“prefix”, “prefix”)	TRUE
(“”, “”)	TRUE
Boundary Value Analysis	
(“prefi”, “prefix”)	TRUE
(“prefix”, “”)	FALSE

P6: Type of Triangle

a) Identify the equivalence classes for the system

Valid Triangle Classes:
E1 : Equilateral Triangle : All sides are equal
E2: Isosceles Triangle : Any two sides are equal
E3: Scalene Triangle : All sides are having different length
E4: Right-Angled Triangle : $A^2 + B^2 = C^2$

Invalid Triangle Classes:
E5: Not a Triangle : Does not satisfy triangle inequality
E6: Non-positive Inputs : negative inputs

b) Identify test cases to cover the identified equivalence classes. Also, explicitly mention which test case would cover which equivalence class. (Hint: you must need to be ensure that the identified set of test cases cover all identified equivalence classes)

Input (A, B, C)	Expected Output	Equivalence Class
(3, 3, 3)	"Equilateral"	Equilateral Triangle
(5, 5, 3)	"Isosceles"	Isosceles Triangle
(3, 4, 5)	"Scalene"	Scalene Triangle
(3, 4, 6)	"Scalene"	Scalene Triangle
(1, 1, 2)	"Not a Triangle"	Not a Triangle
(0, 2, 3)	"Not a Triangle"	Non-positive Input
(3, 4, 7)	"Not a Triangle"	Not a Triangle
(0, 0, 0)	"Not a Triangle"	Non-positive Input
(3, 4, 5)	"Right-Angled"	Right-Angled Triangle
(5,12,13)	"Right-Angled"	Right-Angled Triangle

c) For the boundary condition A + B > C case (scalene triangle), identify test cases to verify the boundary.

Input (A, B, C)	Expected Output	Equivalence Class

(2, 3, 4)	"Scalene"	Equilateral Triangle
(2, 3, 5)	"Not a Triangle"	Isosceles Triangle
(2, 2.5, 5)	"Not a Triangle"	Scalene Triangle

d) For the boundary condition A = C case (isosceles triangle), identify test cases to verify the boundary.

Input (A, B, C)	Expected Output
(3, 3, 2)	"Isosceles"
(3, 3, 6)	"Not a Triangle"
(2, 2, 4)	"Not a Triangle"

e) For the boundary condition A = B = C case (equilateral triangle), identify test cases to verify the boundary.

Input (A, B, C)	Expected Output
(5, 5, 5)	"Equilateral"
(0, 0, 0)	"Not a Triangle"

f) For the boundary condition A2 + B2 = C2 case (right-angle triangle), identify test cases to verify the boundary.

Input (A, B, C)	Expected Output
(3, 4, 5)	"Right-Angled"
(1, 1, Math.sqrt(2))	"Right-Angled"
(1, 2, Math.sqrt(5))	"Right-Angled"
(2, 2, Math.sqrt(8))	"Not a Triangle"

g) For the non-triangle case, identify test cases to explore the boundary.

Input (A, B, C)	Expected Output
(1, 2, 3)	"Not a Triangle"

(2, 3, 1)	"Not a Triangle"
(2, 2, 5)	"Not a Triangle"

h) For non-positive input, identify test points.

Input (A, B, C)	Expected Output
(0, 1, 1)	"Not a Triangle"
(-1, 1, 1)	"Not a Triangle"
(1, 0, 1)	"Not a Triangle"
(1, 1, -1)	"Not a Triangle"

THANK YOU