IT314-Software Engineering

Lab8-Functional Testing (Black-Box)

202201526-Elvis Kotadiya

Question 1: Consider a program for determining the previous date. Its input is triple of day, month and year with the following ranges 1 <= month <= 12, 1 <= day <= 31, 1900 <= year <= 2015.The possible output dates would be previous date or invalid date. Design the equivalence class test cases?

- Equivalence Partitioning:

  Equivalence Classes:

  - Valid Day: 1 ≤ day ≤ 31
  - Invalid Day: day < 1 or day > 31
  - Valid Month: 1 ≤ month ≤ 12
  - Invalid Month: month < 1 or month > 12
  - Valid Year: 1900 ≤ year ≤ 2015
  - Invalid Year: year < 1900 or year > 2015

  Test Cases:

Input Data	**Expected Outcome**
day=0, month=5, year=2000	Invalid date
day=15, month=13, year=2000	Invalid date
day=32, month=5, year=2000	Invalid date
day=15, month=5, year=2016	Invalid date
day=15, month=5, year=2000	Previous date
day=1, month=1, year=1900	Invalid date

- Boundary Value Analysis:

  Boundary Conditions:

  - Minimum/Maximum valid day, month, year.
  - Previous and next days to check transition.
  - Transition of months and years for boundary conditions.

  Test Cases:

Input Data	**Expected Outcome**
day=1, month=1, year=1900	Invalid date
day=31, month=12, year=2015	Previous date
day=30, month=2, year=2012	Invalid date
day=29, month=2, year=2012	Previous date
day=1, month=3, year=2012	Previous date

Question 2:

Programs:

P1. The function linearSearch searches for a value v in an array of integers a. If v appears in the array a, then the function returns the first index i, such that a[i] == v; otherwise, -1 is returned.

- Equivalence Partitioning:

  Equivalence Classes:
  - Array contains value v
  - Array does not contain value v
  - Empty array

  Test Cases:

Input Data	**Expected Outcome**
v=5, a=[1, 2, 3, 5, 9]	3
v=10, a=[1, 2, 3, 5, 9]	-1
v=5, a=[]	-1

- Boundary Value Analysis:

  Boundary Conditions:

  - Array contains multiple instances of v
  - Array contains no instances of v
  - Array contains a single instance of v

Test Cases:

Input Data	Expected Outcome
v=5, a=[5]	0
v=3, a=[1, 2, 3]	2
v=1, a=[1]	0

P2. The function countItem returns the number of times a value v appears in an array of integers a.

- Equivalence Partitioning:

  Equivalence Classes:
  - Array contains value v
  - Array does not contain value v
  - Empty array

  Test Cases:

Input Data	Expected Outcome
v=5, a=[1, 2, 3, 5, 9]	1
v=5, a=[5, 5, 5]	3
v=10, a=[1, 2, 3, 5, 9]	0

- Boundary Value Analysis:

  Boundary Conditions:

  - Empty array
  - Single-element arrays with and without the value

  Test Cases:

Input Data	**Expected Outcome**
v=5, a=[5]	1
v=3, a=[1, 2, 3]	1
v=1, a=[1]	1

P3. The function binarySearch searches for a value v in an ordered array of integers a. If v appears in the array a, then the function returns an index i, such that a[i] == v; otherwise, -1 is returned.

- Equivalence Partitioning:

  Equivalence Classes:

  - Value exists in array
  - Value does not exist in array

Test Cases:

Input Data	**Expected Outcome**
v=5, a=[1, 2, 3, 5, 9]	3
v=7, a=[1, 2, 3, 5, 9]	-1

- Boundary Value Analysis:

  Boundary Conditions:

  - Minimum, maximum, and mid elements of the array
  - Single-element array

  Test Cases:

Input Data	**Expected Outcome**
v=1, a=[1, 2, 3, 5, 9]	0
v=9, a=[1, 2, 3, 5, 9]	4
v=5, a=[5]	0

P4. The following problem has been adapted from The Art of Software Testing, by G. Myers (1979). The function triangle takes three integer parameters that are interpreted as the lengths of the sides
of a triangle. It returns whether the triangle is equilateral (three lengths equal), isosceles (two lengths equal), scalene (no lengths equal), or invalid (impossible lengths).

- Equivalence Partitioning:

  Equivalence Classes:

  - Equilateral triangle (all sides equal)
  - Isosceles triangle (two sides equal)
  - Scalene triangle (no sides equal)
  - Invalid triangle (sides do not form a valid triangle)

  Test Cases:

Input Data	**Expected Outcome**
a=3, b=3, c=3	Equilateral
a=3, b=3, c=4	Isosceles
a=3, b=4, c=5	Scalene
a=1, b=1, c=2	invalid

- Boundary Value Analysis:

  Boundary Conditions:

  - Testing for conditions where two sides are equal (Isosceles)
  - Testing near the limits where the triangle may be invalid

Test Cases:

Input Data	Expected Outcome
a=2, b=2, c=3	Isosceles
a=1, b=1, c=2	invalid
a=3, b=4, c=5	Scalene

P5. The function prefix (String s1, String s2) returns whether or not the string s1 is a prefix of string s2 (you may assume that neither s1 nor s2 is null).

- Equivalence Partitioning:

  Equivalence Classes:
  - s1 is a prefix of s2
  - s1 is not a prefix of s2

  Test Cases:

Input Data	Expected Outcome
s1="test", s2="testing"	true
s1="best", s2="testing"	flase

- Boundary Value Analysis:

  Boundary Conditions:

  - Single-character string as s1
  - s1 and s2 having the same length but different content

  Test Cases:

Input Data	**Expected Outcome**
s1="t", s2="test"	true
s1="testing", s2="testing"	true
s1="test", s2="testing"	true

P6: Consider again the triangle classification program (P4) with a slightly different specification: The program reads floating values from the standard input. The three values A, B, and C are interpreted as representing the lengths of the sides of a triangle. The program then prints a message to the standard output that states whether the triangle, if it can be formed, is scalene, isosceles, equilateral, or right angled. Determine the following for the above program:

a) Identify the equivalence classes for the system

Equivalence Classes:

- Equilateral triangle
- Isosceles triangle
- Scalene triangle
- Right-angled triangle
- Invalid triangle

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)

Test case:

Input Data	**Expected Outcome**
A=3, B=3, C=3	Equilateral
A=3, B=3, C=4	Isosceles
A=3, B=4, C=6	Scalene
A=3, B=4, C=5	Right-angled
A=1, B=1, C=2	invalid

Boundary Conditions:

- Right-angled triangle boundary where $A^2 + B^2 = C^2$
- Equilateral triangle boundary $A = B = C$
- Isosceles triangle boundary $A = B$, $A = C$, or $B = C$
- Scalene has condition only default where $C < B+A$

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

Input Data	**Expected Outcome**
A=3, B=4, C=5	Scalene

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

Input Data	**Expected Outcome**
A=5, B=3, C=5	Isosceles

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

Input Data	**Expected Outcome**
A=5, B=5, C=5	Equilateral

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

Input Data	**Expected Outcome**
A=3, B=4, C=5	Right-angled

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

Input Data	**Expected Outcome**
A=3, B=4, C=6	invalid

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

Input Data	**Expected Outcome**
A=3, B=4, C=-6	invalid