IT314
SOFTWARE ENGINEERING

Lab8-Functional Testing (Black-Box)

Name: Akabari Kartavya
ID: 202201213

Q.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? Write a set of test cases (i.e., test suite)—a specific set of data—to properly test the programs. Your test suite should include both correct and incorrect inputs.

1. Enlist which set of test cases have been identified using Equivalence Partitioning and Boundary Value Analysis separately.

Equivalence Class	Description	Valid or Invalid
E1	1 <= Day <=31	Valid
E2	Day<1	Invalid
E3	Day>31	Invalid
E4	1<=Month<=12	Valid
E5	Month<1	Invalid
E6	Month>12	Invalid
E7	1900<=Year<=2015	Valid
E8	Year<1900	Invalid
E9	Year>2015	Invalid

Equivalence Class Test Cases:

Number	Day	Month	Year	Covered equivalence class	Valid/Invalid
1.	8	3	1901	E1, E4, E7	Valid

					Output - 8/3/1901
2.	-6	3	2000	E2	Invalid
3.	32	3	2000	E3	Invalid
4.	12	-1	2004	E5	Invalid
5.	28	2	2001	E1,E4,E7	Valid Output - 28/2/2001
6.	23	11	2004	E1,E4,E7	Valid Output - 23/11/2004
7.	12	13	2002	E6	Invalid
8.	12	12	1899	E8	Invalid
9.	10	1	2016	E9	Invalid

Boundary Analysis Test cases:

Number	Day	Month	Year	Covered equivalence class	Valid/Invalid
1.	1	1	1900	E1, E4, E7	Valid Output - 1/1/1900
2.	0	1	2002	E2	Invalid
3.	31	3	2015	E1,E4,E7	Valid Output - 31/3/2015
4.	32	4	2002	E3	Invalid
5.	31	12	1900	E1,E4,E7	Valid Output - 31/12/1900

6.	1	1	2015	E1,E4,E7	Valid Output - 1/1/2015
7.	13	0	2010	E5	Invalid
8.	15	13	2011	E6	Invalid
9.	18	4	1900	E1,E4,E7	Valid Output - 18/4/1900
10.	12	9	2015	E1,E4,E7	Valid Output - 12/09/2015
11.	13	6	1899	E8	Invalid
12.	17	5	2016	E9	Invalid

2. Modify your programs such that they run, and then execute your test suites on the program. While executing your input data in a program, check whether the identified expected outcome (mentioned by you) is correct or not.

```cpp
#include <iostream>
#include <tuple>
using namespace std;
string prev_date(int d, int m, int y) {
    if (m < 1 || m > 12 || y < 1900 || y > 2015 || d < 1 || d > 31) {
        return "Invalid";
    }
    return "Valid";
}
```

Q.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 Class Descriptions:

- **E1:** The array has no elements.
- **E2:** The value v exists within the array.
- **E3:** The value v is absent from the array.
- **E4:** The array has a single element that is equal to v.
- **E5:** The array has a single element that is not equal to v.

Test Cases:

Number	Input Data(Array a, Value v)	Expected Outcome	Covered Equivalence Class
1.	a = [ ], v = 5	-1	E1
2.	a = [1,2,3,4,5], v = 5	4	E2
3.	a = [1,2,3,4,6], v = 5	-1	E3
4.	a = [5], v = 5	0	E4
5.	a = [4], v = 5	-1	E5

Boundary Value Analysis:

- The array size is at its minimum size, either empty or contains just one element.
- The value is at the start or end of the array.
- The value is not present, but close to elements in the array.

Test Case	Input Data (Array a,Value v)	Expected Outcome	Boundary Condition
1.	a = [1], v = 1	0	Single element array, value present
2.	a = [1], v = 2	-1	Single element array, value absent
3.	a = [1,2,3,4,5], v = 2	1	Value is at the index 1 in the array
4.	a = [1,2,3,4,5], v = 5	4	Value is at the end of the array (4th index)
5.	a = [1,2,3,4,5], v = 6	-1	Value absent

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

● By Equivalence Class:-

- E1: The array contains multiple occurrences of the value.
- E2: The array does not contain the value.
- E3: The array is empty.
- E4: The array has a single element, and it equals to the value.
- E5: The array has a single element, but it does not equal the value.

Test Cases:

Number	Input Data(Array a, Value v)	Expected Outcome	Covered Equivalence Class
1.	v=2, a =[1,2,1,2]	2	E1
2.	v=5, a =[2,3,4]	0	E2
3.	v=1, a =[]	0	E3
4.	v=3, a =[2]	0	E5
5.	v=3, a = [3]	1	E4

● Boundary Value Analysis :

Number	Input Data(Array a, Value v)	Expected Outcome	Covered Equivalence Class
1.	v=1, a=[1,2,3]	1	E1
2.	v=4, a =[2,3,1]	0	E2
3.	v=1, a =[ ]	0	E3
4.	v=2, a =[2]	1	E4
5.	v=2, a = [3]	0	E5

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.

Assumption: the elements in the array a are sorted in non-decreasing order.

Equivalence Class Description:

- **E1:** The array has no elements.
- **E2**: The value v is found in the array.
- **E3**: The value v is absent from the array.
- **E4**: The array contains one element, and it is equal to v.
- **E5:** The array contains one element, but it is not equal to v.

Test Case	Input Data (Array a,Value v)	Expected Outcome	Equivalence Class
1.	a = [ ], v = 5	-1	E1
2.	a = [1,2,3,4,5], v = 5	4	E2
3.	a = [1,2,3,4,6], v = 5	-1	E3
4.	a = [5], v = 5	0	E4
5.	a = [4], v = 5	-1	E5

Boundary Conditions:

- The array size is at its minimum size, either empty or contains just one element.
- The value is at the start, middle, or end of the array.
- The value is not present but close to elements in the array.

Test Case	Input Data (Array a,Value v)	Expected Outcome	Equivalence Class
1.	a = [5], v = 5	0	E4
2.	a = [5], v = 6	-1	E5
3.	a = [1,2,3,4,5], v = 1	0	E2
4.	a = [], v = 3	-1	E1
5.	a = [1,2,3,4,5], v = 5	4	E2
6.	A = [1,2,3,4,5], v = 6	-1	E3

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).

● By Equivalence Class:

- E1: Valid equilateral triangle.
- E2: Valid isosceles triangle.
- E3: Valid scalene triangle.
- E4: Invalid triangle (the sum of any two sides is not greater than the third).
- E5: Triangle with negative side lengths.
- E6: Triangle with zero side lengths.

Test Case	Input Data (Array a,Value v)	Expected Outcome	Equivalence Class
1.	a=3 ,b=3,c=3	Equilateral	E1
2.	a=4,b=4,c=5	Isosceles	E2
3.	a=3,b=4,c=5	Scalene	E3
4.	a=1,b=2,c=3	Invalid	E4
5.	a=-1,b=3,c=-4	Invalid	E5
6.	a=0,b=3,c=4	Invalid	E6

● Boundary Value Analysis :

Test Case	Test-Case	Expected Outcomes	Equivalence Class
1.	a = 2, b = 2, c = 6	Invalid	E4
2.	a = 1, b = 1, c = 2	isosceles	E2
3.	a = -1,b = 1, c = 1	Invalid	E5
4.	a = 0, b = 1, c = 1	Invalid	E6
5.	a = 1, b = 1, c = 1	Equilateral	E1

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 Class Description:

● E1: s1 is longer than s2 (impossible to be a prefix).
● E2: s1 is a valid prefix of s2.

● E3: s1 is not a prefix of s2.
● E4: s1 is an empty string (edge case).
● E5: s2 is an empty string (edge case).

Test Case	Input Data (s1, s2)	Expected Outcome	Covered Equivalence Class
1.	"abcdef", "abc"	false	E1
2.	"abc", "abcdef"	true	E2
3.	"xyz", "abcdef"	false	E3
4.	" ", "abcdef"	true	E4
5.	"abc", ""	false	E5

Boundary Conditions:

● Length of s1 is greater than the length of s2.
● s1 is an empty string, s2 is a non-empty string.
● s2 is an empty string, s1 is non-empty.
● s1 equals s2.

Test Case	Input Data (s1, s2)	Expected Outcome	Covered Equivalence Class
1.	“a”,” ”	false	E5
2.	"abcdef", "abcde"	true	E1,E2
3.	"abc", "abc"	true	E2
4.	" ", " "	true	E4,E2

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
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)
c) For the boundary condition A + B > C case (scalene triangle), identify test cases to verify the boundary.
d) For the boundary condition A = C case (isosceles triangle), identify test cases to verify the boundary.
e) For the boundary condition A = B = C case (equilateral triangle), identify test cases to verify the boundary.
f) For the boundary condition A2 + B2 = C2 case (right-angle triangle), identify test cases to verify the boundary.
g) For the non-triangle case, identify test cases to explore the boundary.
h) For non-positive input, identify test points.

● By Equivalence Class:

- E1: Valid equilateral triangle: All sides are of equal length.
- E2: Valid isosceles triangle: Exactly two sides have the same length.
- E3: Valid scalene triangle: All sides are of different lengths.
- E4: Valid right-angled triangle: Sides satisfy the Pythagorean theorem.
- E5: Invalid triangle: Sides do not meet the triangle inequality rule.
- E6: Invalid input: One or more sides have non-positive values.

Test Case	Output	Equivalence Class
A = 3, B = 3, C = 3	Equilateral	E1
A = 4, B = 4, C = 5	Isosceles	E2
A = 3, B = 4, C = 5	Scalene	E3
A = 3, B = 4, C = 8	Invalid	E5
A = -1, B = 2, C = 3	Invalid	E6
A = 5, B = 12, C = 13	Right-Angled	E4

c) Boundary conditions for A + B > C (Scalene Triangle) (Class E3):

Test Case	Output
A = 1, B = 3, C = 2	Scalene
A = 8, B = 5, C = 6	Scalene

d) Boundary Conditions for A=C (Isosceles Triangle) (Class E2):

Test Case	Output
A = 3, B = 3, C = 4	Isosceles
A = 2, B = 2, C = 3	Isosceles
A = 2, B = 2, C = 1	Isosceles

e) Boundary Conditions for A = B = C (Equilateral Triangle) (Class E1):

Test-Case	Output
A = 2, B = 2, C = 2	Equilateral
A = 5, B = 5, C = 5	Equilateral

f) Boundary Conditions for A2 + B2 = C2 (Right-Angle Triangle) (Class E4):

Test-Case	**Output**
A = 3, B = 4, C = 5	Right-angled
A = 5, B = 12, C = 13	Right-angled

g) Test Cases for Non-Triangle Case (Class E5):

Test-Case	**Output**
A = 1, B = 2, C = 3	Invalid
A = 1, B = 2, C = 4	Invalid
A = 1, B = 1, C = 3	Invalid

h) Test Cases for Non-Positive Input (Class E6):

Test-Case	**Output**
A = 0, B = 2, C = 3	Invalid
A = -1, B = -2, C = 3	Invalid