Software Enginnering (IT314)

LAB 8

Kishan Pansuriya (202201504)

Q-1

1. Equivalence Classes

Equivalence Class Partitioning (ECP):

1. Valid Dates
   - E1: Valid date (e.g., 1st Jan 2000)
   - E2: Valid leap year date (e.g., 29th Feb 2012)
   - E3: Valid date at month boundaries (e.g., 1st March)
   - E4: Valid last day of a month (e.g., 31st Jan)
2. Invalid Dates
   - E5: Invalid day (e.g., 32nd Jan)
   - E6: Invalid month (e.g., 13th month)
   - E7: Invalid year (e.g., 1899)
   - E8: Invalid leap year date (e.g., 29th Feb 2013)

Boundary Value Analysis (BVA):

1. Valid Boundaries
   - E1: 1st Jan (edge case)
   - E2: 31st Dec (edge case)
   - E3: 1st Feb (valid)
   - E4: 29th Feb on a leap year (valid)
   - E5: 28th Feb on a non-leap year (valid)
2. Invalid Boundaries
   - E6: 0th Jan (invalid)
   - E7: 32nd Jan (invalid)
   - E8: 1st month (invalid month)
   - E9: 29th Feb on a non-leap year (invalid)
   - E10: 1900 (boundary year)

Test Cases for ECP

Input(d,m,y)	Expected Outcome
1, 1, 2000	31, 12, 1999
29, 2, 2012	28, 2, 2012
1, 3, 2000	29, 2, 2000
31, 1, 2000	30, 1, 2000
32, 1, 2000	"Invalid Date"
15, 13, 2000	"Invalid Date"
15, 1, 1899	"Invalid Date"
29, 2, 2013	"Invalid Date"

Test Cases for BVA:

Input (d,m,y)	Expected Outcome
1, 1, 1900	31, 12, 1899
1, 1, 2000	31, 12, 1999
31, 12, 2015	30, 12, 2015
29, 2, 2016	28, 2, 2016
28, 2, 2015	27, 2, 2015
0, 1, 2000	"Invalid Date"
32, 1, 2000	"Invalid Date"
1, 0, 2000	"Invalid Date"
29, 2, 2015	"Invalid Date"
1, 1, 2015	31, 12, 2014

<u>Function to Determine the Previous Date</u>

```
public class PrevDateCalc {

    public static String prevDate(int d, int m, int y) {

        if (m < 1 || m > 12 || y < 1900 || y > 2015) {

            return "Invalid Date";

        }



        int[] daysInMonth = {31, (isLeapYear(y) ? 29 : 28), 31, 30,
31, 30, 31, 31, 30, 31, 30, 31};



        if (d < 1 || d > daysInMonth[m - 1]) {

            return "Invalid Date";

        }



        if (d > 1) {

            return (d - 1) + ", " + m + ", " + y;

        } else {

            if (m == 1) {

                return 31 + ", " + 12 + ", " + (y - 1);

            } else {

                return daysInMonth[m - 2] + ", " + (m - 1) + ", " + y;

            }

        }

    }
    private static boolean isLeapYear(int year) {

        return (year % 4 == 0 && year % 100 != 0) || (year % 400 ==
0);

    }



    public static void main(String[] args) {

        System.out.println(prevDate(1, 1, 2000)); // Expected: 31, 12,
1999

        System.out.println(prevDate(29, 2, 2012)); // Expected: 28, 2,
2012

        System.out.println(prevDate(1, 3, 2000)); // Expected: 29, 2,
2000

        System.out.println(prevDate(31, 1, 2000)); // Expected: 30, 1,
2000

        System.out.println(prevDate(32, 1, 2000)); // Expected:
Invalid Date

        System.out.println(prevDate(15, 13, 2000)); // Expected:
Invalid Date

        System.out.println(prevDate(15, 1, 1899)); // Expected:
Invalid Date

        System.out.println(prevDate(29, 2, 2013)); // Expected:
Invalid Date

    }

}
```

Q-2

Code-1

Equivalence Classes:

1. **Valid Searches**
    - E1: Value exists in the array.
    - E2: Value does not exist in the array.
2. **Empty Array**
    - E3: Array is empty

Test Cases for ECP

Input (i , v)	**Expected Outcome**
5, [1, 2, 3, 4, 5]	4
10, [1, 2, 3, 4, 5]	-1
2, [2, 2, 2, 2, 2]	0
1, [1, 3, 5, 7]	0
0, [1, 2, 3]	-1
-1, [-1, 0, 1]	0
3, []	-1

Test Cases for BVA:

Input (i , v)	**Expected Outcome**
1, [1]	0
2, [1]	-1
3, [3, 1, 2]	0
1, []	-1
0, [0]	0

Code-2

Equivalence Classes:

1. **Counts of Value**
    - E1: Value appears multiple times.
    - E2: Value appears once.
    - E3: Value does not appear.
2. **Empty Array**
    - E4: Array is empty.

Test Cases for ECP

Input (i, v)	**Expected Outcome**
2, [1, 2, 2, 3, 2]	3
5, [1, 2, 3, 4]	0
3, [3, 3, 3, 3]	4
1, [1, 1, 2, 1, 1]	4
2, [1, 2, 3, 2]	2
0, [-1, 0, 1, 0]	2
5, []	0

Test Cases for BVA:

Input (i , v)	**Expected Outcome**
1, [1]	1
2, [2]	1
3, [1, 2, 3]	1
4, [4, 4, 4]	3
1, []	0

Code-3

Equivalence Classes:

1. **Valid Searches**
   - E1: Value exists in the array.
   - E2: Value does not exist in the array.
2. **Empty Array**
   - E3: Array is empty.
3. **Single Element Array**
   - E4: Array has one element.

Test Cases for ECP

Input (i , v)	**Expected Outcome**
3, [1, 2, 3, 4, 5]	2
6, [1, 2, 3, 4, 5]	-1
1, [1, 2, 3]	0
4, [1, 2, 3, 4, 5]	3
0, [-1, 0, 1]	1
-1, [-1, 0, 1]	0
5, []	-1

Test Cases for BVA:

Input (i, v)	**Expected Outcome**
1, [1]	0
2, [1]	-1
1, [1, 2]	0
2, [1, 2]	1
3, []	-1

Code-4

Equivalence Classes:

1. **Valid Triangles**
   - E1: Equilateral triangle.
   - E2: Isosceles triangle.
   - E3: Scalene triangle.
2. **Invalid Triangles**
   - E4: Triangle inequality violated.
3. **Zero or Negative Values**
   - E5: One or more sides are zero or negative.

Test Cases for ECP

Input (a,b,c)	**Expected Outcome**
3, 3, 3	0
3, 3, 4	1
3, 4, 5	2
1, 1, 2	3
0, 1, 1	3
5, 5, 10	3
1, 1, 1	0

Test Cases for BVA:

Input (a,b,c)	**Expected Outcome**
1, 1, 1	0
2, 2, 2	0
2, 2, 5	3
1, 2, 3	3
0, 1, 2	3

Code-5

Equivalence Classes:

1. **s1 is a prefix of s2**
   - E1: s1 is equal to s2.
   - E2: s1 is a proper prefix of s2 (shorter than s2).
2. **s1 is not a prefix of s2**
   - E3: s1 is longer than s2.
   - E4: s1 and s2 have different starting characters.
   - E5: s1 is an empty string (prefix of any string).
   - E6: s2 is an empty string (only if s1 is also empty).

Test Cases for ECP

Input (s1,s2)	**Expected Outcome**
"hello", "hello"	true
"hell", "hello"	true
"hello", "hell"	false
"hello", "world"	false
"", "anything"	true
"anything", ""	false

Test Cases for BVA

Input (s1,s2)	**Expected Outcome**
"abc", "abc"	true
"abcd", "abc"	false
"abc", "abcd"	true
"", ""	true
"abc", ""	false

Code-6

Equivalence Classes:

1. **Valid Triangles**
   - Class 1: Equilateral triangle.
   - Class 2: Isosceles triangle.
   - Class 3: Scalene triangle.
   - Class 4: Right-angled triangle.
2. **Invalid Triangles**
   - Class 5: Triangle inequality violated.
3. **Non-triangles**
   - Class 6: One or more sides are non-positive.

Test Cases for ECP

Input (a,b,c)	**Expected Outcome**
3, 3, 3	"Equilateral"
3, 3, 5	"Isosceles"
3, 4, 5	"Scalene"
3, 4, 4	"Isosceles"
3, 4, 6	"Scalene"
3, 4, 8	"Invalid"
1, 2, 3	"Invalid"
0, 1, 1	"Invalid"

Test Cases for BVA

Input (a,b,c)	**Expected Outcome**
3, 4, 5	"Scalene"
2, 2, 4	"Invalid"
1, 1, 2	"Invalid"
2, 3, 2	"Isosceles"
5, 7, 5	"Isosceles"
2, 2, 2	"Equilateral"
5, 5, 5	"Equilateral"
3, 4, 5	"Right-angled"
5, 12, 13	"Right-angled"
1, 2, 3	"Invalid"
3, 3, 7	"Invalid"
0, 1, 1	"Invalid"
-1, 1, 1	"Invalid"