LAB 9

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?

Solution 1:

Equivalent Classes:

E1: month<1, day<1, year<1900
E2: month<1, day<1, 1900<=year<=2015
E3: month<1, day<1, year>2015
E4: month<1, 1<=day<=31, year<1900
E5: month<1, 1<=day<=31, 1900<=year<=2015
E6: month<1, 1<=day<=31, year>2015
E7: month<1, day>31, year<1900
E8: month<1, day>31, 1900<=year<=2015
E9: month<1, day>31, year>2015
E10: 1<=month<=12, day<1, year<1900
E11: 1<=month<=12, day<1, 1900<=year<=2015
E12: 1<=month<=12, day<1, year>2015
E13: 1<=month<=12, 1<=day<=31, year<1900
E14: 1<=month<=12, 1<=day<=31, 1900<=year<=2015
E15: 1<=month<=12, 1<=day<=31, year<2015
E16: 1<=month<=12, day>31, year<1900
E17: 1<=month<=12, day>31, 1900<=year<=2015
E18: 1<=month<=12, day>31, year>2015
E19: 12<month, day<1, year<1900
E20: 12<month, day<1, 1900<=year<=2015
E21: 12<month, day<1, year>2015
E22: 12<month, 1<=day<=31, year<1900
E23: 12<month, 1<=day<=31, 1900<=year<=2015
E24: 12<month, 1<=day<=31, year>2015
E25: 12<month, day>31, year<1900
E26: 12<month, day>31, 1900<=year<=2015
E27: 12<month, day>31, year>2015

S. No.	Month	Day	Year	Expected Output	Code Output
1.	0	0	1899	Invalid	Invalid
2.	0	0	1900	Invalid	Invalid
3.	0	0	2016	Invalid	Invalid
4.	0	1	1899	Invalid	Invalid
5.	0	1	1900	Invalid	Invalid
6.	0	1	2016	Invalid	Invalid
7.	0	32	1899	Invalid	Invalid
8.	0	32	1900	Invalid	Invalid
9.	0	32	2016	Invalid	Invalid
10.	1	0	1899	Invalid	Invalid
11.	1	0	1900	Invalid	Invalid
12.	1	0	2016	Invalid	Invalid
13.	1	1	1899	Invalid	Invalid
14.	1	1	1900	Previous Date	Previous Date
15.	1	1	2016	Invalid	Invalid
16.	1	32	1899	Invalid	Invalid
17.	1	32	1900	Invalid	Invalid
18.	1	32	2016	Invalid	Invalid
19.	1	0	1899	Invalid	Invalid
20.	13	0	1900	Invalid	Invalid
21.	13	0	2016	Invalid	Invalid
22.	13	1	1899	Invalid	Invalid
23.	13	1	1900	Invalid	Invalid
24.	13	1	2016	Invalid	Invalid

25.	13	32	1899	Invalid	Invalid
26.	13	32	1900	Invalid	Invalid
27.	13	32	2016	Invalid	Invalid

Code in java ->

```
import java.util.*;

class Main{
   public static void main(String args[])
   {
        int month_date, day_date, year_date;
        int number_of_testcases;
        Scanner s = new Scanner(System.in);
        int valid = 0;
        number_of_testcases = s.nextInt();
        while(number_of_testcases > 0)
        {
                valid = 0;
                day_date = s.nextInt();
                if((day_date >= 1) && (day_date <= 31))
                {
                        valid = 1;
                }
                month_date = s.nextInt();
                if((month_date >= 1) && (month_date <= 12))
                {
                        valid = valid + valid;
                }
                year_date = s.nextInt();
                if((year_date >= 1900) && (year_date <= 2015))
                {
                        valid = valid + valid;
                }
                if(valid == 4)
                {
                        if(day_date == 1)
                        {
                                if(month_date == 1)
                                {
                                        year_date = (year_date - 1);
                                        System.out.println("31.12."+year_date);
                                }
```

```
                                        else
                                        {
                                                month_date = (month_date - 1);

System.out.println("31."+month_date+"."+year_date);
                                        }
                                }
                                else
                                {
                                        day_date = (day_date - 1);

System.out.println(day_date+"."+month_date+"."+year_date);
                                }
                        }
                        else
                        {
                                System.out.println("Invalid Date");
                        }
                        number_of_testcases = (number_of_testcases - 1);
                }
        }
}
```

Solution 2:

Equivalent Classes:
E1: months with 31 days[Jan(01),Mar(03),May(05),Jul(07),Aug(08),Oct(10),Dec(12)]
E2: months with 30 days[Apr(04),June(06),Sept(09),Nov(11)]
E3: months with 28/29 days[Feb(02)]
E4: month < 1
E5: month > 12
E6: days between 1 to 28
E7: days = 29
E8: days = 30
E9: days = 31
E10: days < 1
E11: days > 31
E12: leap years between 1900 to 2015
E13: non-leap years between 1900 to 2015
E14: year>2015
E15: year<1900

S.No.	Date	Month	Year	Expected output	Code output

1	E6	E1	E12	Previous date	Previous date
2	E7	E1	E12	Previous date	Previous date
3	E8	E1	E12	Previous date	Previous date
4	E9	E1	E12	Previous date	Previous date
5	E10	E1	E12	Invalid	Invalid
6	E11	E1	E12	Invalid	Invalid
7	E6	E2	E12	Previous date	Previous date
8	E7	E2	E12	Previous date	Previous date
9	E8	E2	E12	Previous date	Previous date
10	E9	E2	E12	Invalid	Invalid
11	E10	E2	E12	Invalid	Invalid
12	E11	E2	E12	Invalid	Invalid
13	E6	E3	E12	Previous date	Previous date
14	E7	E3	E12	Previous date	Previous date
15	E8	E3	E12	Invalid	Invalid
16	E9	E3	E12	Invalid	Invalid
17	E10	E3	E12	Invalid	Invalid
18	E11	E3	E12	Invalid	Invalid
19	E6	E4	E12	Invalid	Invalid
20	E7	E4	E12	Invalid	Invalid
21	E8	E4	E12	Invalid	Invalid
22	E9	E4	E12	Invalid	Invalid
23	E10	E4	E12	Invalid	Invalid
24	E11	E4	E12	Invalid	Invalid
25	E6	E5	E12	Invalid	Invalid
26	E7	E5	E12	Invalid	Invalid
27	E8	E5	E12	Invalid	Invalid

28	E9	E5	E12	Invalid	Invalid
29	E10	E5	E12	Invalid	Invalid
30	E11	E5	E12	Invalid	Invalid
31	E6	E1	E13	Previous date	Previous date
32	E7	E1	E13	Previous date	Previous date
33	E8	E1	E13	Previous date	Previous date
34	E9	E1	E13	Previous date	Previous date
35	E10	E1	E13	Invalid	Invalid
36	E11	E1	E13	Invalid	Invalid
37	E6	E2	E13	Previous date	Previous date
38	E7	E2	E13	Previous date	Previous date
39	E8	E2	E13	Previous date	Previous date
40	E9	E2	E13	Invalid	Invalid
41	E10	E2	E13	Invalid	Invalid
42	E11	E2	E13	Invalid	Invalid
43	E6	E3	E13	Previous date	Previous date
44	E7	E3	E13	Invalid	Invalid
45	E8	E3	E13	Invalid	Invalid
46	E9	E3	E13	Invalid	Invalid
47	E10	E3	E13	Invalid	Invalid
48	E11	E3	E13	Invalid	Invalid
49	E6	E4	E13	Invalid	Invalid
50	E7	E4	E13	Invalid	Invalid
51	E8	E4	E13	Invalid	Invalid
52	E9	E4	E13	Invalid	Invalid
53	E10	E4	E13	Invalid	Invalid
54	E11	E4	E13	Invalid	Invalid

55	E6	E5	E13	Invalid	Invalid
56	E7	E5	E13	Invalid	Invalid
57	E8	E5	E13	Invalid	Invalid
58	E9	E5	E13	Invalid	Invalid
59	E10	E5	E13	Invalid	Invalid
60	E11	E5	E13	Invalid	Invalid
61	E6	E1	E14	Invalid	Invalid
62	E7	E1	E14	Invalid	Invalid
63	E8	E1	E14	Invalid	Invalid
64	E9	E1	E14	Invalid	Invalid
65	E10	E1	E14	Invalid	Invalid
66	E11	E1	E14	Invalid	Invalid
67	E6	E2	E14	Invalid	Invalid
68	E7	E2	E14	Invalid	Invalid
69	E8	E2	E14	Invalid	Invalid
70	E9	E2	E14	Invalid	Invalid
71	E10	E2	E14	Invalid	Invalid
72	E11	E2	E14	Invalid	Invalid
73	E6	E3	E14	Invalid	Invalid
74	E7	E3	E14	Invalid	Invalid
75	E8	E3	E14	Invalid	Invalid
76	E9	E3	E14	Invalid	Invalid
77	E10	E3	E14	Invalid	Invalid
78	E11	E3	E14	Invalid	Invalid
79	E6	E4	E14	Invalid	Invalid
80	E7	E4	E14	Invalid	Invalid
81	E8	E4	E14	Invalid	Invalid

82	E9	E4	E14	Invalid	Invalid
83	E10	E4	E14	Invalid	Invalid
84	E11	E4	E14	Invalid	Invalid
85	E6	E5	E14	Invalid	Invalid
86	E7	E5	E14	Invalid	Invalid
87	E8	E5	E14	Invalid	Invalid
88	E9	E5	E14	Invalid	Invalid
89	E10	E5	E14	Invalid	Invalid
90	E11	E5	E14	Invalid	Invalid
91	E6	E1	E15	Invalid	Invalid
92	E7	E1	E15	Invalid	Invalid
93	E8	E1	E15	Invalid	Invalid
94	E9	E1	E15	Invalid	Invalid
95	E10	E1	E15	Invalid	Invalid
96	E11	E1	E15	Invalid	Invalid
97	E6	E2	E15	Invalid	Invalid
98	E7	E2	E15	Invalid	Invalid
99	E8	E2	E15	Invalid	Invalid
100	E9	E2	E15	Invalid	Invalid
101	E10	E2	E15	Invalid	Invalid
102	E11	E2	E15	Invalid	Invalid
103	E6	E3	E15	Invalid	Invalid
104	E7	E3	E15	Invalid	Invalid
105	E8	E3	E15	Invalid	Invalid
106	E9	E3	E15	Invalid	Invalid
107	E10	E3	E15	Invalid	Invalid
108	E11	E3	E15	Invalid	Invalid

109	E6	E4	E15	Invalid	Invalid
110	E7	E4	E15	Invalid	Invalid
111	E8	E4	E15	Invalid	Invalid
112	E9	E4	E15	Invalid	Invalid
113	E10	E4	E15	Invalid	Invalid
114	E11	E4	E15	Invalid	Invalid
115	E6	E5	E15	Invalid	Invalid
116	E7	E5	E15	Invalid	Invalid
117	E8	E5	E15	Invalid	Invalid
118	E9	E5	E15	Invalid	Invalid
119	E10	E5	E15	Invalid	Invalid
120	E11	E5	E15	Invalid	Invalid

Code:

```
bool isLeap(int year)
{
    if (year % 400 == 0)
        return true;
    if (year % 100 == 0)
        return false;
    if (year % 4 == 0)
        return true;
    return false;
}
int solve()
{
    int date, month, year;
    cin >> date >> month >> year;
    if (date < 1 || date > 31)
    {
        cout << "Invalid Date\n";
        return 0;
    }
    if (month < 1 || month > 12)
    {
        cout << "Invalid Date\n";
```

```
        return 0;
    }
    if (year > 2015 || year < 1900)
    {
        cout << "Invalid Date\n";
        return 0;
    }
    if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month
== 12)
        {
            if (date == 1)
            {
                if (month == 1)
                {
                    cout << "31/12/" << year - 1 << "\n";
                    return 0;
                }
                if (month == 3)
                {
                    if (isLeap(year))
                    {
                        cout << "29/2/" << year << "\n";
                        return 0;
                    }
                    else
                    {
                        cout << "28/2/" << year << "\n";
                        return 0;
                    }
                }
                if (month == 8)
                {
                    cout << "31/7/" << year << "\n";
                    return 0;
                }
                cout << "30/" << month - 1 << year << "\n";
                return 0;
            }
            else
            {
                cout << date - 1 << "/" << month << "/" << year << "\n";
                return 0;
            }
        }
```

```
else if (month == 2)
{
    if (isLeap(year))
    {
        if (date > 29)
        {
            cout << "Invalid Date\n";
            return 0;
        }
    }
    else
    {
        if (date > 28)
        {
            cout << "Invalid Date\n";
            return 0;
        }
    }
    if (date == 1)
    {
        cout << "31/" << month - 1 << "/" << year << "\n";
        return 0;
    }
    else
    {
        cout << date - 1 << "/" << month << "/" << year << "\n";
        return 0;
    }
}
else
{
    if (date == 31)
    {
        cout << "Invalid Date\n";
        return 0;
    }
    else
    {
        if (date == 1)
        {
            cout << "31/" << month - 1 << "/" << year << "\n";
            return 0;
        }
        else
```

```
            {
                cout << date - 1 << "/" << month << "/" << year << "\n";
                return 0;
            }
        }
    }
    return 0;
}
```

Q2 The system accepts a five-digit numeric item ID number from 00000 to 99999. The system accepts a quantity to be ordered, from 1 to 99. If the user enters a previously ordered item ID and a 0 quantity to be ordered, that item is removed from the shopping cart. Based on these inputs, the system retrieves the item price, calculates the item total (quantity times item price), and adds the item total to the cart total. Due to limits on credit card orders that can be processed, the maximum cart total is $999.99

Equivalence Class:

ID:
I1: 00000 to 99999
I2: <00000
I3: >99999

Quantity:
Q1: < 0
Q2: 1 to 99
Q3 : > 99
Q4: =0

Cart Total:
T1: <= $999.99
T2: >$999.99

Test Cases:

S. No.	ID	Quantity	Total Amount	Expected Output
1	I2	ANY	ANY	Invalid
2	I3	ANY	ANY	Invalid
3	ANY	Q1	ANY	Invalid
4	ANY	Q3	ANY	Invalid
5	I1	Q2	T1	Valid
6	I1	Q2	T2	Invalid
7	I1	Q4	ANY	Discard Item

S. No.	ID	Quantity	Expected Output
1	-000001	0	Invalid
2	-000001	1	Invalid
3	-000001	100	Invalid
4	-000001	-1	Invalid
5	000000	0	Discard
6	000000	1	Add to total
7	000000	100	Invalid
8	000000	-1	Invalid
9	1000000	0	Invalid
10	1000000	1	Invalid
11	1000000	100	Invalid
12	1000000	-1	Invalid

S. No.	Total Amount	Expected Output
1	0	Do not checkout
2	999.99	Valid
3	1000	Invalid

QUESTION	CONTRIBUTORS
Q-1 Solution-1	Shivam(201801111), Smit(201801239), Vatsal(201801162)
Q-1 Solution-2	Vyom(201801062), Shubham(201801213), Vanshika(201801104), Parthraj(201801208)
Q-2	Astha(201801169), Dhruvil(201801056), Preet (201801051)