

Question 1

Equivalence Partitioning

Equivalence Classes

a. Month

- i. M1 : Month < 1
- ii. M2 : Month > 12
- iii. M3 : Months with 31 days [Jan(01),Mar(03),May(05),Jul(07),Aug(08),Oct(10),Dec(12)]
- iv. M4 : Months with 30 days [Apr(04),Jun(06),Sept(09),Nov(11)]
- v. M5 : Months with 28/29 days [Feb(02)]

b. Days

- i. D1 : Day < 1
- ii. D2 : Day > 31
- iii. D3 : 1 to 28
- iv. D4 : 29
- v. D5 : 30
- vi. D6 : 31

c. Year

- i. Y1: > 2015
- ii. Y2: < 1900
- iii. Y3: Leap years from 1900 to 2015
- iv. Y4: Non-Leap years from 1900 to 2015

Test Cases :

ID	Days	Months	Years	Output
1	D1	ANY	ANY	Invalid
2	D2	ANY	ANY	Invalid
3	ANY	M1	ANY	Invalid
4	ANY	M2	ANY	Invalid
5	ANY	ANY	Y1	Invalid
6	ANY	ANY	Y2	Invalid
7	D3	M3	Y3	Previous date
8	D4	M3	Y3	Previous date
9	D5	M3	Y3	Previous date

10	D6	M3	Y3	Previous date
11	D3	M3	Y4	Previous date
12	D4	M3	Y4	Previous date
13	D5	M3	Y4	Previous date
14	D6	M3	Y4	Previous date
15	D3	M4	Y3	Previous date
16	D4	M4	Y3	Previous date
17	D5	M4	Y3	Previous date
18	D6	M4	ANY	Invalid
19	D3	M4	Y4	Previous date
20	D4	M4	Y4	Previous date
21	D5	M4	Y4	Previous date
22	D3	M5	Y3	Previous date
23	D4	M5	Y3	Previous date
24	D5	M5	ANY	Invalid
25	D6	M5	ANY	Invalid
26	D3	M5	Y4	Previous date
27	D4	M5	Y4	Invalid

Boundary Value:

Equivalence Class	Month	Day	Year	Output
Day = 1	$1 < x \leq 12$	1	$1900 < X < 2015$	Previous date
Day = 29	2	29	$1900 < X < 2015$ and $X \% 4 == 0$	Previous date
Day = 29	2	29	$1900 < X < 2015$ and $X \% 4$	Previous date

			!=0	
Day = 31	1, 3, 5, 7, 8 10, 12	31	1900 < x ≤2015	Previous date
Day = 31	2, 4, 6, 9, 11	31	1900 < x ≤2015	Invalid
Month = 1	1	1 < x ≤31	1900 < x ≤2015	Previous date
Month = 1	1	1	1900 < x ≤2015	Previous date
Month = 12	12	1 < x ≤31	1900 < x ≤2015	Invalid
Year = 1900	1 < x ≤12	1 < x ≤31	1899 or less	Invalid
Year = 2015	1 < x ≤12	1 < x ≤31	2016 or more	Invalid

The steps to determine the previous date:

- Decrement the day by 1 if the day in the entered date is not 1.
- If day is 1, set the day to 31 and decrement the month by 1.
- If the entered date is 1 January (1/1/YYYY) decrement the year by 1
 - Set the month to 12 and the day to 31
- For rest of the test cases output that the input is invalid

Q1_2)

```
#include<bits/stdc++.h>
using namespace std;

bool checkYear(int year)
{
    if (year % 400 == 0)
        return true;

    if (year % 100 == 0)
        return false;

    if (year % 4 == 0)
        return true;
    return false;
}
```

```

int main(){
    int day,month,year;
    int t;
    cin>>t;
    while(t--){
        cin>>day>>month>>year;
        if(month==2 && year%4!=0 && day==29)
        {
            cout<<"Invalid Test"<<endl;
        }
        else if(day>=1 && day<=31 && month>=1 && month<=12 && year>=1900 &&
year<=2015){
            // cout<<day<<"-"<<month<<"-"<<year<<endl;
            if(month==1 && day==1)
            {
                month=12;
                day=31;
                year--;
            }
            else if(day==1)
            {
                month--;
            }

if((month==1)|| (month==3)|| (month==5)|| (month==7)|| (month==8)|| (month==10)|| (month==12
))
        {
            day=31;
            //cout<<31<<"-"<<month<<"-"<<year<<endl;
        }
        else if((month==4)|| (month==6)|| (month==9)|| (month==11))
        {
            day=30;
        }
        else if(month==2)
        {
            if(checkYear(year))
            {
                day=29;
            }
            else
            {
                day=28;
            }
        }
    }
}

```

```

    }
    else
    {
        day--;

    }
    cout<<"Previous Date "<<day<<"-"<<month<<"-"<<year<<endl;
}
else{
    cout<<"Invalid Test"<<endl;
}

}
}

```

```

1 1 2001
Previous Date 31-12-2000
1 3 2004
Previous Date 29-2-2004
29 2 2005
Invalid Test
25 10 1995
Previous Date 24-10-1995
32 2 2010
Invalid Test
2 13 1990
Invalid Test
5 12 1890
Invalid Test
4 5 2001
Previous Date 3-5-2001
1 9 2010
Previous Date 31-8-2010
12 12 1985
Previous Date 11-12-1985

```

Question 2

Given constraints:

ID : 00000 - 99999

Quantity : 1 - 99

Maximum cart total : less than or equal to \$999.99 i.e $\leq \$999.99$

Equivalence classes:**ID**

1) ID between 00000 - 99999 (both inclusive) i.e. $00000 \leq ID \leq 99999$

2) ID less than 00000 (00000 excluded) i.e. $ID < 00000$

3) ID greater than 99999 (99999 excluded) i.e. $ID > 99999$

Quantity

4) Quantity between 0-99 (both inclusive) i.e. $0 \leq \text{Quantity} \leq 99$

5) Quantity less than 0 (0 excluded) i.e. $\text{Quantity} < 0$

6) Quantity greater than 99 (99 excluded) i.e. $\text{Quantity} > 99$

Cart total (in dollars)

7) Cart total between 0-999.99 (both inclusive) i.e. $0 \leq \text{Cart Total} \leq 999.99$

8) Cart total greater than 999.99 (999.99 excluded) i.e. $\text{Cart Total} > 999.99$

Let us assume that cart total is \$200 (for some selected items) and the price of an item with ID 12489 is \$100.

Test Case	Input Data	Expected Outcome
ID<00000	-12489	Error
ID>99999	100099	Error
Quantity < 0	-5	Error
Quantity > 99	101	Error
Valid ID	ID = 12489	Item Price = 100\$
Valid Cart Total	ID = 12489 Quantity = 6	Cart Total = 800\$
Invalid Cart Total	ID=12489 quantity=10	Cart total=\$1200(error since > 999.99)
Quantity = 0	ID = 12489	Item with ID 12489 removed from shopping cart [if Item with ID 12489 was purchased previously]

Quantity = 0	ID = 12489	Error [if Item with ID 12489 was NOT purchased previously]
--------------	------------	---