Group: 31

Lab: 9

Hemang Nakarani :	201801158
Dharmin Hirapara :	201801135
Shah Yash Pareshkumar :	201801131
Sudarshan Kundnani :	201801140
Shantanu Tyagi :	201801015
Srinivas Talnikar :	201801406
Sudhanshu Mishra :	201801114
Harshil Joshi :	201801022
Urvish Pandya :	201801170
Kiran Ravuri :	201801421

Question 1:

Equivalence Class:

Equivalent Classes	Day	Month	Year	Expected Output
Day > 31	67	6	2012	Invalid
Day < 1	-2	4	2000	Invalid
Within Constraints	21	4	2014	A date within constraints would be 20-4-2014
Month > 12	38	14	2013	Invalid
Month < 1	5	-3	2012	Invalid
Year > 2015	17	8	2021	Invalid
Leap Year	29	2	2004	The valid date, in this case, would be 28 -02-2004
Days exceeding	31	2	2006	Invalid

the actual number of days in a month				
Year < 1900	25	12	1789	Invalid
None Leap Year	29	2	2009	Invalid

Boundary Value:

Test Case #	Day	Month	Year	Expected Output
1	0	Any	Any	Invalid
2	1	Any	Any	Valid
3	22	Any	Any	Valid
4	30	Any(Except 2)	Any	Valid
5	30	2	Any	Invalid
6	31	Any(Except 2,4,6,9,11)	Any	Valid
7	31	2,4,6,9,11	Any	Invalid
8	32	Any	Any	Invalid
9	Any	0	Any	Invalid
10	Any	1	Any	Valid
11	Any	7	Any	Valid
12	Any(Except 31)	11	Any	Valid
13	31	11	Any	Invalid
14	Any	12	Any	Valid
15	Any	13	Any	Invalid
16	Any	Any	1900	Valid
17	Any	Any	1899	Invalid

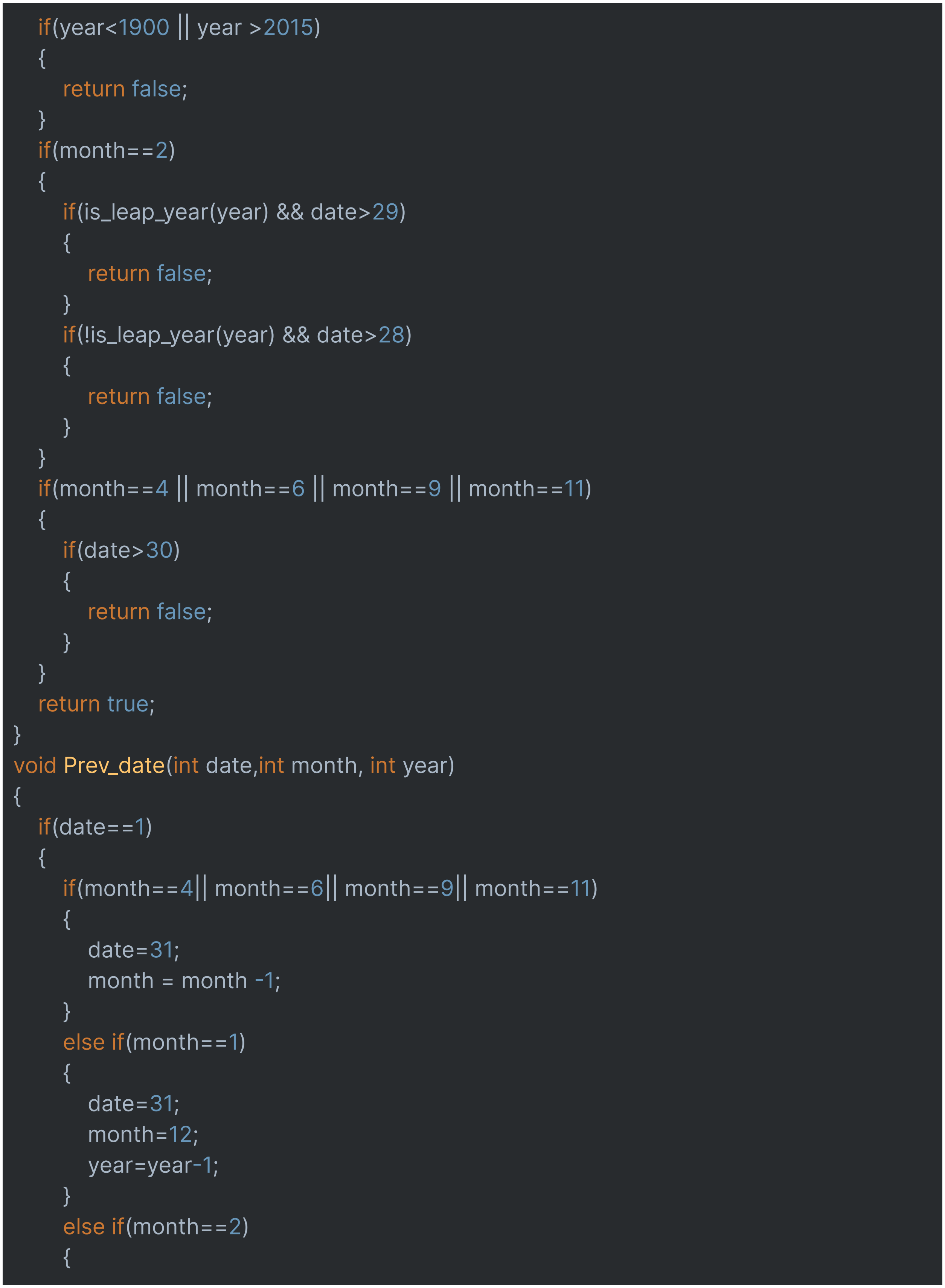
18	Any	Any	1985	Valid
19	Any	Any	2014	Valid
20	Any	Any	2015	Valid
21	Any	Any	2016	Invalid

***Any: Stands for the range within the given constraints.**

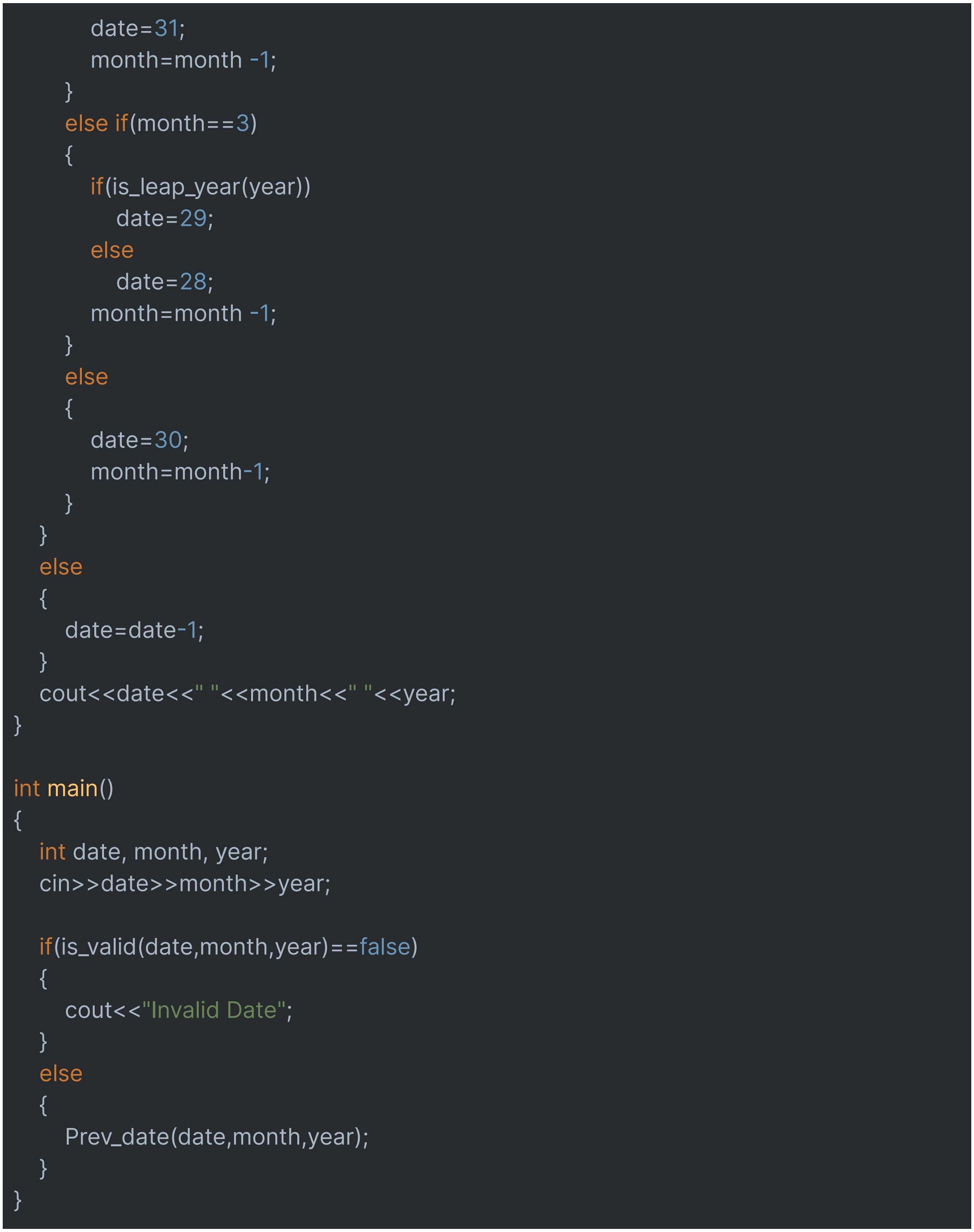
Code:

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

bool is_leap_year(int year)
{
    if(year%4==0)
    {
        if(year%100==0)
        {
            if (year%400==0)
                return true;
            else
                return false;
        }
        else
            return true;
    }
    else
        return false;
}
bool is_valid(int date,int month,int year)
{
    if(date<1 || date>31)
    {
        return false;
    }
    if(month<1 || month>12)
    {
        return false;
    }
```



```
    if(year<1900 || year >2015)
    {
        return false;
    }
    if(month==2)
    {
        if(is_leap_year(year) && date>29)
        {
            return false;
        }
        if(!is_leap_year(year) && date>28)
        {
            return false;
        }
    }
    if(month==4 || month==6 || month==9 || month==11)
    {
        if(date>30)
        {
            return false;
        }
    }
    return true;
}
void Prev_date(int date,int month, int year)
{
    if(date==1)
    {
        if(month==4|| month==6|| month==9|| month==11)
        {
            date=31;
            month = month -1;
        }
        else if(month==1)
        {
            date=31;
            month=12;
            year=year-1;
        }
        else if(month==2)
        {
```



```
            date=31;
            month=month -1;
        }
        else if(month==3)
        {
            if(is_leap_year(year))
                date=29;
            else
                date=28;
            month=month -1;
        }
        else
        {
            date=30;
            month=month-1;
        }
    }
    else
    {
        date=date-1;
    }
    cout<<date<<" "<<month<<" "<<year;
}

int main()
{
    int date, month, year;
    cin>>date>>month>>year;

    if(is_valid(date,month,year)==false)
    {
        cout<<"Invalid Date";
    }
    else
    {
        Prev_date(date,month,year);
    }
}
```

Question 2:

Input Constraints

ID: 00000-999990
Quantity 1-99
Max cart total ≤ 999.99

Equivalence classes:

ID:
E1: 00000 ≤ ID ≤ 99999
E2: ID < 00000
E3: ID > 99999

Quantity(Q):
E4: Q=0
E5: 0 < Q ≤ 99, 0 ≤ cart total ≤ 999.99
E7: 0 < Q ≤ 99, cart total > 999.99
E8: Q < 0
E9: Q > 99

Final Equivalence Classes

E1: Q=0, ID Existing in Cart
E2: Q =0, ID Not-Existing in Cart
E3: 00000 ≤ ID ≤ 99999, 0 < Q ≤ 99, 0 ≤ cart total ≤ 999.99
E4: 00000 ≤ ID ≤ 99999, 0 < Q ≤ 99, cart total > 999.99
E5: 00000 ≤ ID ≤ 99999, Q < 0
E6: 00000 ≤ ID ≤ 99999, Q > 99,
E7: ID < 00000
E8: ID > 99999

Boundary Value Analysis:

No.	Test Case	Expected Output
1	Q=0, ID Existing in cart	Item Removed From Cart
2	Q=0, ID Not-Existing in Cart	Invalid Input
3	ID=-1, Q=any, CartTotal=any	Invalid ID
4	ID=100000, Q=any, CartTotal=any	Invalid ID
5	ID=1, Q=1, CartTotal=999.99	Cart Total as Expected (Valid Input)
6	ID=1, Q=1, CartTotal=999.98	Cart Total as Expected (Valid Input)
7	ID=1, Q=1, CartTotal=1000	Invalid CartTotal
8	ID=99998, Q=1,CartTotal= 999.99	Cart Total as Expected (Valid Input)
9	ID=99998, Q=1,CartTotal= 999.98	Cart Total as Expected (Valid Input)
10	ID=99998, Q=1,CartTotal= 1000	Invalid CartTotal
11	ID=1, Q = -1, CartTotal=any	Invalid Quantity
12	ID=1, Q = 2, CartTotal=999.99	Cart Total as Expected (Valid Input)
13	ID=1, Q = 98, CartTotal=999.99	Cart Total as Expected (Valid Input)
14	ID=1, Q = 100, CartTotal=any	Invalid Quantity