Question 1 :
As no specification is given we have considered that every year is a normal year i.e no leap year and every month contains 31 days.

Equivalence classes:-

Day:-

1. Day value less than 1 is considered to be invalid.
2. Day value greater than 31 is considered to be invalid.
3. Day value is between 1 and 31(inclusive) will be a valid one.

Month

1. if month value is in between 1 and 12 inclusive than it is a valid
2. if the month value is less than 1 then it is invalid.
3. If the month value is greater than 12 then it is invalid.

Year:-

1. Year value less than 1900 is considered to be invalid.
2. Year value more than 2015 is considered to be invalid.
3. Year value is between 1900 and 2015(both inclusive) will be a valid one.

Set of test cases:-

Test-cases using equivalence partitioning and boundary value analysis :- There are 3 equivalence classes and each has 3 possible values so the total combination will be 3*3*3 = 27.

Test Case	Expected outcome
111 (0/2/1800)	Invalid Input
112 (0/2/2020)	Invalid Input
113 (0/2/2000)	Invalid Input
121 (0/0/1800)	Invalid Input
122 (0/0/2020)	Invalid Input
123 (0/0/2000)	Invalid Input
131 (0/13/1800)	Invalid Input
132 (-1/13/2020)	Invalid Input
133 (0/13/2015)	Invalid Input
211(32/2/1800)	Invalid Input
212(32/2/2020)	Invalid Input

213(32/2/2015)	Invalid Input
221(32/0/1800)	Invalid Input
222 (32/0/2020)	Invalid Input
223 (40/-5/2013)	Invalid Input
231(32/13/2000)	Invalid Input
232 (34/13/2030)	Invalid Input
233(32/13/2000)	Invalid Input
311(2/2/1800)	Invalid Input
312 (25/4/2020)	Invalid Input
313(13/2/2000)	**Valid Input (12/2/2000)**
321(13/-10/1800)	Invalid Input
322 (12/0/2020)	Invalid Input
323 (12/0/2000)	Invalid Input
331 (12/13/1800)	Invalid Input
332 (12/13/2020)	Invalid Input
333 (12/13/2000)	Invalid Input
Boundary (1/12/2000)	31/11/2000
Boundary (1/1/2000)	31/12/1999
Boundary (1/1/1900)	31/12/1899

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

typedef long long int ll;

bool check_leap(ll year)
{

    if (year % 400 == 0)
        return true;

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

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

int main()
```

```
{
   ll day,month,year;
   cin>>day>>month>>year;
   if(day<1 || day>31 || month<1 || month > 12 || year<1900 || year>2015)
   goto here;
   else
   {
      if(month==2 || month==4 || month==6 || month==7 || month==9 || month==11)
      {
         if(day==31)
         goto here;
      }
      if(day==29 || day == 30)
      {
         if(month==2 && !check_leap(year))
         goto here;
      }
      if(month==3 && day==1)
      {
         if(check_leap(year))
         cout<<"29 2 "<< year << endl;
         else
         cout<<"28 2 "<< year << endl;

         goto there;
      }
      if(day==1)
      {
         if(month!=1)
         {
            if(month==2 || month==4 || month==6 || month==8 || month==9 || month==11)
               cout<<"31 "<<month-1<<' '<<year;

            else
               cout<<"30 "<<month-1<<' '<<year;

         }
         else
            cout<<"31 12 "<<year-1;

         goto there;
      }
      cout<<day-1<<' '<<month<<' '<<year<<endl;
      goto there;
   }
   here:
   cout<<"Invalid Date"<<endl;
   there:
   return 0;
}
```

Code 2 with assumptions made as above

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.*;

public class Solution {
   public static boolean check_leap(int year) {
```

```
        if (year % 400 == 0)
            return true;

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

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

    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        int day = Integer.parseInt(br.readLine());
        int month = Integer.parseInt(br.readLine());
        int year = Integer.parseInt(br.readLine());

        if(day<1 || day>31 || month<1 || month > 12 || year<1900 ||
year>2015){
            System.out.println("Invalid input");
        }
        else{
            if(day == 1){
                if(month == 1){
                    System.out.println( "31  12 " + (year -1));
                }
                else{
                    System.out.println("31 " + (month -1) + " " + (year));
                }
            }
            else{
                System.out.println((day-1) + " " + month + " " + year);
            }
        }
    }
}
```

Question2:-

Equivalence classes:-

ID:-

1. ID value less than 00000 is considered to be invalid.
2. ID value greater than 99999 is considered to be invalid.
3. ID value between 00000 and 99999(inclusive) is considered to be valid.

Quantity:-

1. Quantity value less than 0 is considered to be invalid.

2. Quantity value greater than 99 is considered to be invalid.
3. Quantity value between 0 and 99(inclusive) is considered to be valid.

Total cart amount in dollars:-

1. Total cart amount greater than 999.99$ is considered to be invalid.
2. Total cart amount between 0$ and 999.99$(inclusive) is considered to be valid.

Test-cases using equivalence partitioning and boundary value analysis :- There are 3 equivalence classes and first has 3, second has 3 and third has 2 so the total combination will be (3*3)*2 = 18.

Test Case	Expected Outcome
111 (-11111,-1,1111)	Invalid
112(-11111,-1,50)	Invalid
121 (-98654,145,1000)	Invalid
122 (-11111,100,20)	Invalid
131 (-11111,50,1111)	Invalid
132 (-245765,88,123)	Invalid
211 (1000000,-1,1111)	Invalid
212 (420300,-10,2376)	Invalid
221(1000000,100,1111)	Invalid
222 (1111111,111,11)	Invalid
231 (2525252,86,2500)	Invalid
232 (1411414,53,2376)	Invalid
311 (1000,-1,2000)	Invalid
312 (1000,-1,50)	Invalid
321 (1000,200,2000)	Invalid
322 (1000,200,50)	Invalid
331 (1000,50,2000)	Invalid
332 (1000,50,50)	**2500$**

Test Case	Input values	Expected Outcome
ID<00000	-11111	Error

ID > 99999	111111	Error
Quantity < 0	-1	Error
Quantity > 99	190	Error
Invalid cart total	ID = 00012 Quantity = 5	Total = $4000 error value > 999.99
Valid total	ID = 00012 Quantity = 5	Total = 100