Jutge.org

The Virtual Learning Environment for Computer Programming

Priority clients X90063_en

Information about clients of a store is available. For each one, whe know the client identifier, the birth date and the amount of the purchases she has made. Examining the information, a customer is classified as a priority client when she has made at least three purchases of amount greater than or equal to one thousand euros.

Complete the code shown below, **without modifying anything already defined**, to obtain the customer list ordered according to the following criteria. Priority customers should appear first and then the non-priority ones. In each of the two groups, those who have made more total expenditure in the store will appear first. Then, if two clients belong to the same group and have made the same total expense, the older one will be listed first. Finally, the possible ties that still persist are resolved with the customer identifier using the usual order of the strings.

Note that the `main()` function is complete.

Exam score: 2.50 **Automatic part:** 40.00%

Input

A non-negative integer *n* followed by the information of *n* customers. For each customer, first it appears the client identifier string, then three integers *d m y* representing her birth date (day, month and year, respectively) and, finally, a non-negative integer *m* followed by a sequence of *m* integers that represent the amounts of the *m* customer purchases.

Output

The list of clients ordered according to the criteria defined above. Each line of the output shows one customer data: whether she is a priority client, the total amount of purchases, the birth date and the client identifier as shown in the example.

Sample input

```
10

JuanPalomo
4 7 1958
5
123 345 563 231 777

MaryAnt
23 5 1993
5
5000 250 250 250 250

CarmenLozano
5 12 2000
9
100 100 100 100 100 100 100 100 100

LaraMay
22 10 1981
3
300 300 300

JohnLong
12 3 1978
3
1234 1000 2000

AlexFandos
24 9 2001
6
234 652 10 23 500 500

PedroRoque
3 7 1970
5
1000 1000 1000 1000 1000

AlbaMas
3 7 1970
6
1000 1000 1000 1000 600 400
```

```
HarryBig
20 2 1970
2
5000 2000

CarlesFont
3 5 1987
5
10 12 8 10 10
```

Sample output

```
1 5000 03-07-1970 AlbaMas
1 5000 03-07-1970 PedroRoque
1 4234 12-03-1978 JohnLong
0 7000 20-02-1970 HarryBig
0 6000 23-05-1993 MaryAnt
0 2039 04-07-1958 JuanPalomo
0 1919 24-09-2001 AlexFandos
0 900 22-10-1981 LaraMay
0 900 05-12-2000 CarmenLozano
0 50 03-05-1987 CarlesFont
```

Observation

Complete the following code.

```
/////////////////////////////////////////////////////////////////
/////////////some lines of code are needed here/////////////////
/////////////////////////////////////////////////////////////////

struct Date {
    int day;
    int month;
    int year;
};

struct Client {
    string client_id;
    Date birth_date;
    bool priority;
    int total_amount;
};

//returns true when client a is older than b. Returns false otherwise
bool is_older(const Client& a, const Client& b) {
    Date x = a.birth_date;
    Date y = b.birth_date;
    if (x.year != y.year) return x.year < y.year;
    if (x.month != y.month) return x.month < y.month;
    return x.day < y.day;
}


/////////////////////////////////////////////////////////////////
/////////////some lines of code are needed here/////////////////
/////////////////////////////////////////////////////////////////


//gets client info vector v
void get_client_vector(vector<Client>& v) {
    int n = v.size();
    for (int i = 0; i < n; ++i) get_client(v[i]);
}

//prints date info in format dd-mm-yyyy
void print_date(Date date) {
    if (date.day < 10) cout << 0;
    cout << date.day << ’-’;
    if (date.month < 10) cout << 0;
    cout << date.month << ’-’;
    cout << date.year;
}

//writes v on output channel
void write_client_vector(const vector<Client>& v) {
    int n = v.size();
    for (int i = 0; i < n; ++i) {
        cout << v[i].priority << ’ ’ << v[i].total_amount << ’ ’;
        print_date(v[i].birth_date);
        cout << ’ ’ << v[i].client_id << endl;
    }
}

int main() {
    int n;
    cin >> n;
    vector<Client> v(n);
    get_client_vector(v);
    sort(v.begin(),v.end(),cmp);
    write_client_vector(v);
}
```

Problem information

Author : Pro1
Generation : 2021-01-12 16:51:24