# CÁCH ĐÁNH GIÁ ĐIỂM THỰC HÀNH HỌC PHẦN: IT3150 – Project 1- 2023.1

# I. Quy định, yêu cầu:

- Tài liệu và nội dung thực hành chấm điểm trên hệ thống: https://lab.soict.hust.edu.vn/
- Bài tập trên lớp chấm điểm tự động (các bài không chấm trên hệ thống làm vào máy tính → làm báo cáo thực hành – Theo mẫu).
- Hạn nộp báo cáo trên Teams (Bài tập trên lớp + Bài tập về nhà): 1 tuần.

# II. Đánh giá điểm thực hành

- 1. Chuyên cần (đúng giờ, nghiêm túc trong giờ học) Điểm danh trên Teams: 10%
- 2. Báo cáo thực hành (bài tập trên lớp + Về nhà) theo mẫu nộp trên Teams: 40%
- 3. Trắc nghiệm Form trên Teams: 10%
- 4. Kiểm tra thực hành: 40%. (Tiết 2,3 buổi thực hành thứ 5).

# Điểm thưởng: $5\% \rightarrow 10\%$ (Cho Mục 1,2 điểm TB từ 9-10).

Tham gia thực hành đúng giờ đầy đủ theo thời khóa biểu (nếu có lý do không đi thực hành đúng kíp được thì gửi mail xin phép thực hành bù trước 1 ngày qua mail <a href="hoalt@soict.hust.edu.vn">hoalt@soict.hust.edu.vn</a>, Tiêu đề: đăng ký học bù – IT3040 – MaLopTH. Các kíp có thể bù:

TT	Thời gian, địa điểm, Tuần học	Mã nhóm	Mã lớp
1			
2			
3			
4			
5			
6			
7			

Nếu nghỉ không có lý do 3 buổi, không thực hành bù thì điểm chuyên cần, báo cáo và BTVN coi như 0 điểm thực hành.

## Mai Minh Hoàng – 20215381

# Contents

Bài 1: Phân tích việc gứi mã cúa một cuộc thi lập trình	3
Bài 2: Phân tích dữ liệu công dân	13
Table of Figure	
Figure 1 b1- Code	
Figure 2 b1- Input 1	
Figure 3 b1- Output 1	
Figure 4 b1- Input 2	8
Figure 5 b1- Output 2	8
Figure 6 b1- Input 3	9
Figure 7 b1- Output 3	9
Figure 8 b1- Input 4	
Figure 9 b1- Output 4	10
Figure 10 b2- Code	
Figure 11 b2- Input 1	
Figure 12 b2- Output 1	
Figure 13 b2- Input 2	18
Figure 14 b2- Output 2	18
Figure 15 b2- Input 3	19
Figure 16 b2- Output 3	20
Figure 17 b2- Input 4	20
Figure 18 b2- Output 4	21

# Báo cáo tuần 8

# Bài 1: Phân tích việc gửi mã của một cuộc thi lập trình

Dữ liệu gửi dự thi lập trình gồm một dãy các dòng, mỗi dòng có các thông tin sau:

<UserID> <ProblemID> <TimePoint> <Status> <Point>

trong đó người dùng <UserID> gửi mã của mình để giải quyết vấn đề <ProblemID> tại thời điểm <TimePoint> và nhận trạng thái <Trạng thái> và điểm <Điểm>

- <UserID>: chuỗi có độ dài từ 3 đến 10
- < ProblemID>: chuỗi có định dạng Pxy trong đó x, y là các chữ số 0,1,...,9 (ví dụ P03, P10)
- <TimePoint>: chuỗi biểu thị thời điểm có định dạng HH:MM:SS (ví dụ: 09:45:20 nghĩa là thời điểm 9 giờ 45 phút 20 giây)
- <Status>: chuỗi có hai trường hợp (ERR, OK)
- <Point>: số nguyên từ {0, 1, 2, ..., 10}

Người dùng có thể gửi mã để giải quyết từng vấn đề nhiều lần. Điểm mà người dùng nhận được cho một bài toán là điểm tối đa trong số các bài gửi cho bài toán đó.

#### **Mai Minh Hoàng – 20215381**

Thực hiện một chuỗi các truy vấn thuộc các loại sau:

- ?total\_number\_submissions: trả về số lượng bài dự thi
- ?number\_error\_submision: trả về số lượng bài gửi có trạng thái ERR
- ?number\_error\_submision\_of\_user <UserID>: trả về số lần gửi có trạng thái ERR của người dùng <UserID>
- ?total\_point\_of\_user <UserID>: trả về tổng điểm của người dùng <UserID>
- ?number\_submission\_ Period <from\_time\_point> <to\_time\_point>: trả về số lần gửi trong khoảng thời gian từ <from\_time\_point> đến <to\_time\_point> (đã bao gồm)

Source code

```
2 //20215381
3 #include <iostream>
4 #include<set>
5 #include<sstream>
6 #include<unordered_map>
7 #include<vector>
9 using namespace std;
10
11 long countSubmit_81 = 0; // Biến đếm tổng số lượng bài nộp
12 long sumtotalError_81 = 0; // Biến đếm tổng số lỗi
13 unordered_map<string, long> totalError_81; // Bản đồ lưu tổng số lỗi theo người dùng
14 unordered_map<string, int > totalUser_Problem_point_81; // Bản đồ lưu tổng điểm theo người dùng và vấn để
15 unordered_map<string, long> total_point_81; // Bản đồ lưu tổng điểm theo người dùng
16 long time_submit_81[86400]; // Mảng lưu thời gian nộp
18 int main()
2θ
            ios_base::sync_with_stdio(θ);//tắt đồng bộ
            while (getline(cin, s)) {//nhập đầu vào
                    if (s == "#") {//gặp # thì dừng
23
25
                    stringstream ss(s); // phân tách mỗi dòng
26
27
                    string UserId;
28
                    string Problem;
29
                    long point;
3Θ
                    string Status;
                    string Time;
                    ss >> UserId >> Problem >> Time >> Status >> point;
33
                    countSubmit_81++; // Tăng biến đếm tổng số lượng bài nộp
                    if (Status == "ERR") {
```

#### Mai Minh Hoàng - 20215381

```
36
                             sumtotalError_81++; // Tăng biến đếm tổng số lỗi
                            totalError_81[UserId]++; // Tăng số lỗi của người dùng
37
38
39
                    totalError_81[Status] += point; // Cập nhật điểm lỗi
40
                    if (point > totalUser_Problem_point_81[UserId + Problem]) {
                            total_point_81[UserId] += point - totalUser_Problem_point_81[UserId + Problem]; // Cập nhật tổng điểm
                            totalUser_Problem_point_81[UserId + Problem] = point; // Cập nhật điểm của người dùng và vấn để
42
43
44
45
                    int h = stoi(Time.substr(0, 2));
46
                    int m = stoi(Time.substr(3, 5));
47
                    int sss = stoi(Time.substr(6));
48
                    time_submit_81[h * 60 * 60 + m * 60 + sss] += 1; // Cập nhật thời gian nập
49
50
            for (int i = 0; i <= 86400; i++) {//mỗi thời điểm sẽ có tổng submit bằng chính nó cộng với tất cả thời gian trước đó
51
                    time_submit_81[i] += time_submit_81[i - 1]; // \hat{cap} nhật thời gian nộp
52
53
            while (getline(cin, s)) {//nhập tiếp
54
                    if (s == "#") {//gặp # thì dừng}
55
                            break:
56
57
                    if (s[8] == 's') {// nếu chuỗi dạng number_submission_period
58
59
                            stringstream ss(s);
60
                            string a, b, c;
61
                            ss >> a >> b >> c;//tách ra a b c trong xâu
62
                            int h = stoi(b.substr(0, 2));//tách ra giờ
63
                            int m = stoi(b.substr(3, 5));//phút
64
                            int sss = stoi(b.substr(6));//giây
65
                            int timee = h * 60 * 60 + m * 60 + sss;//tương tự trên nhưng là thời điểm sau
66
                             int h1 = stoi(c.substr(0, 2));
67
                            int m1 = stoi(c.substr(3, 5));
68
                            int sss1 = stoi(c.substr(6));
69
                            int timee1 = h1 * 60 * 60 + m1 * 60 + sss1;
70
71
                            cout << time_submit_81[timee1] - time_submit_81[timee - 1] << "\n"; // In ra thời gian nộp
```

```
73
75
                             if (s[7] == 'n') {//nếu chuỗi dạng ?total_number_submissions
                                     cout << countSubmit_81 << "\n"; // In ra tổng số lượng bài nộp</pre>
77
78
                             else if (s[8] == 'e') {
79
                                     if (s == "?number_error_submision") {//nếu chuỗi là ?number_error_submision
80
81
                                              cout << sumtotalError_81 << "\n"; // In ra tổng số lỗi
82
83
                                              stringstream ss(s);
                                              string a, b;
85
86
                                              ss >> a >> b;
87
                                              cout << totalError_81[b] << "\n"; // In ra tổng số lỗi theo người dùng</pre>
88
89
90
91
                             else {//nếu chuỗi dạng ?total_point_of_user
92
                                      stringstream ss(s);
93
                                      string a, b;
94
                                      ss >> a >> b;
                                      cout << total_point_81[b] << "\n"; // In ra tổng điểm theo người dùng</pre>
95
96
97
98
```

#### Test:



Figure 3 b1- Output 1



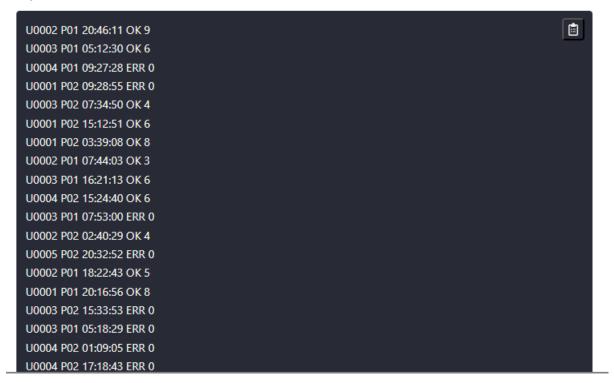
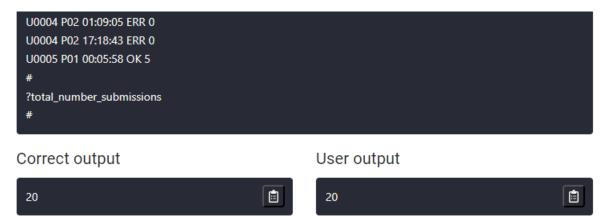


Figure 5 b1- Output 2



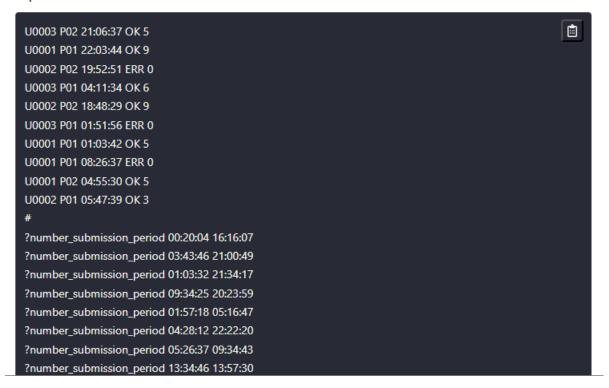
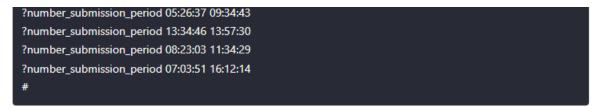


Figure 7 b1- Output 3



#### Correct output



#### User output



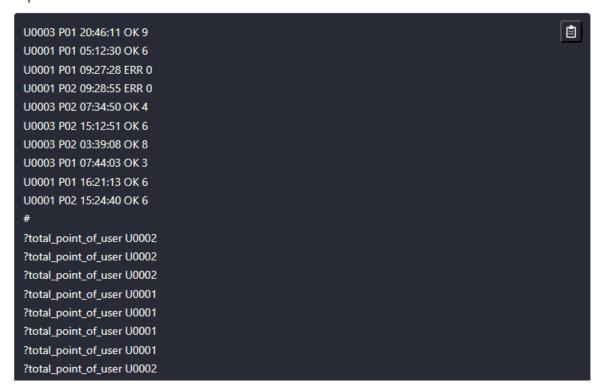
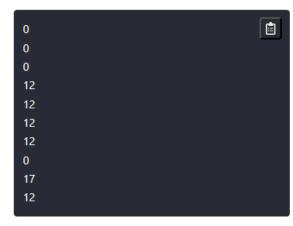


Figure 9 b1- Output 4



### Correct output



## User output



#### Code:

```
//Mai Minh Hoàng
//20215381
#include <iostream>
#include<set>
#include<sstream>
#include<unordered_map>
#include<vector>
using namespace std;
long countSubmit_81 = 0; // Biến đếm tổng số lượng bài nộp
long sumtotalError_81 = 0; // Biến đếm tổng số lỗi
unordered_map<string, long> totalError_81; // Bản đồ lưu tổng số lỗi theo người dùng
unordered_map<string, int > totalUser_Problem_point_81; // Ban đồ lưu tổng điểm theo
người dùng và vấn đề
unordered_map<string, long> total_point_81; // Bản đồ lưu tổng điểm theo người dùng
long time_submit_81[86400]; // Mang lưu thời gian nôp
int main()
      ios_base::sync_with_stdio(0);//tắt đồng bô
      string s;
      while (getline(cin, s)) {//nhập đầu vào
             if (s == "#") {//gặp # thì dừng
                   break;
             stringstream ss(s); // phân tách mỗi dòng
             string UserId;
             string Problem;
             long point;
             string Status;
             string Time;
             ss >> UserId >> Problem >> Time >> Status >> point;
             countSubmit_81++; // Tăng biến đếm tổng số lượng bài nộp
             if (Status == "ERR") {
                   sumtotalError_81++; // Tăng biến đếm tổng số lỗi
                   totalError_81[UserId]++; // Tăng số lỗi của người dùng
             totalError_81[Status] += point; // Câp nhât điểm lỗi
             if (point > totalUser_Problem_point_81[UserId + Problem]) {
                   total_point_81[UserId] += point -
totalUser_Problem_point_81[UserId + Problem]; // Cập nhật tổng điểm
                   totalUser_Problem_point_81[UserId + Problem] = point; // Câp
nhật điểm của người dùng và vấn để
             //tách giờ phút giây từ Time
             int h = stoi(Time.substr(0, 2));
             int m = stoi(Time.substr(3, 5));
             int sss = stoi(Time.substr(6));
             time_submit_81[h * 60 * 60 + m * 60 + sss] += 1; // \hat{C}_{ap} nhật thời gian
nộp
      for (int i = 0; i <= 86400; i++) {//moi thời điểm sẽ có tổng submit bằng
chính nó cộng với tất cả thời gian trước đó
             time_submit_81[i] += time_submit_81[i - 1]; // Cập nhật thời gian nộp
      }
```

```
Mai Minh Hoàng - 20215381
      while (getline(cin, s)) {//nhập tiếp
             if (s == "#") {//gap # thi dùng}
                   break;
             if (s[8] == 's') {// neu chuôi dang number_submission_period
                    stringstream ss(s);
                    string a, b, c;
                    ss >> a >> b >> c;//tách ra a b c trong xâu
                    int h = stoi(b.substr(0, 2));//tách ra giờ
                    int m = stoi(b.substr(3, 5));//phút
                    int sss = stoi(b.substr(6));//giây
                    int timee = h * 60 * 60 + m * 60 + sss; //tuơng tự trên nhưng là
thời điểm sau
                    int h1 = stoi(c.substr(0, 2));
                    int m1 = stoi(c.substr(3, 5));
                    int sss1 = stoi(c.substr(6));
                    int timee1 = h1 * 60 * 60 + m1 * 60 + sss1;
                    cout << time_submit_81[timee1] - time_submit_81[timee - 1] <</pre>
"\n"; // In ra thời gian nộp
             }
             else
                    if (s[7] == 'n') {//néu chuỗi dang ?total_number_submissions
                          cout << countSubmit_81 << "\n"; // In ra tổng số lượng bài</pre>
nôp
                    }
                    else if (s[8] == 'e') {
                          if (s == "?number_error_submision") {//néu chuỗi là
?number_error_submision
                                 cout << sumtotalError_81 << "\n"; // In ra tổng số
lõi
                          else {//nếu chuỗi dạng ?number_error_submision_of_user
                                 stringstream ss(s);
                                 string a, b;
                                 ss >> a >> b;
                                 cout << totalError_81[b] << "\n"; // In ra tổng số
lỗi theo người dùng
                          }
                    else {//néu chuỗi dang ?total_point_of_user
                          stringstream ss(s);
                          string a, b;
                          ss >> a >> b;
                          cout << total_point_81[b] << "\n"; // In ra tổng điểm theo</pre>
người dùng
                    }
```

}

}

# Bài 2: Phân tích dữ liêu công dân

Cho một DataBase về công dân, thực hiện các truy vấn trên DataBase này.

#### Đầu vào

Đầu vào bao gồm hai khối: khối đầu tiên là DataBase và khối thứ hai là danh sách các truy vấn. Hai khối cách nhau bằng một dòng chứa ký tự \*.

1. Khối đầu tiên (DataBase về công dân) gồm các dòng (số dòng có thể lên tới 100000), mỗi dòng là thông tin về một người và có dạng:

<code> <dat\_of\_birth> <fathher\_code> <mother\_code> <is\_alive> <region\_code> trong đó:

- <code>: mã của người là chuỗi có độ dài 7
- <date\_of\_birth>: ngày sinh của người đó và có định dạng YYYY-MM-DD (ví dụ 1980-02-birth>), <date of birth> là trước 3000-12-31
- <fathher\_code> và <mother\_code> là mã của cha và mẹ: chúng cũng là chuỗi có độ dài
   Nếu mã là 0000000 thì người hiện tại không có thông tin về bố hoặc mẹ
- <is\_alive>: ký tự có 2 giá trị: 'Y' nghĩa là người đó vẫn còn sống, và 'N' nghĩa là người hiện tại đã chết.
- <region\_code>: mã vùng nơi người đó sinh sống
- 2. Khối thứ hai là danh sách các truy vấn (số lượng truy vấn có thể lên tới 100000) trên DataBase bao gồm các lệnh sau:
  - NUMBER\_PEOPLE: trả về số lượng người (số dòng của DataBase)
  - NUMBER\_PEOPLE\_BORN\_AT <date>: trả về số người có ngày sinh bằng <date>
  - MOST\_ALIVE\_ANCESTOR <code>: tìm tổ tiên cao nhất (xa nhất về khoảng cách thế hệ) của người <code> nhất định. Trả về khoảng cách thế hệ giữa tổ tiên được tìm thấy và người đã cho
  - NUMBER\_PEOPLE\_BORN\_BETWEEN <from\_date> <to\_date>: tính số người có ngày sinh trong khoảng từ <from\_date> đến <to\_date> (<from\_date> và <to\_date> có dạng YYYY-MM-DD, <to\_date> là trước đó 3000-12-31)
  - MAX\_UNRELATED\_PEOPLE: tìm một tập con gồm những người trong đó có hai người bất kỳ trong tập con đó không có cha/mẹ-con và kích thước của tập con đó là lớn nhất. Trả về kích thước của tập hợp con được tìm thấy.

Khối thứ hai được kết thúc bằng một dòng chứa \*\*\*.

#### đầu ra

• Mỗi dòng trình bày kết quả của truy vấn tương ứng (được mô tả ở trên).

Source code

```
È
     #include <list>
     #include <iostream>
    #include<set>
    #include<sstream>
     #include<unordered_map>
     #include<map>
10 using namespace std;
12 long countPeople_81 = 0; // Biến đếm tổng số người
13 long sumtotalPeople_81 = θ; // Biến đếm tổng số người
14 map<string, long> totalPeople_81; // Bản đổ lưu tổng số người theo ngày sinh
15 unordered_map<string, int > notAlone_81; // Bản đổ kiểm tra người không một minh
16 unordered_map<string, int> maxAcestor_81; // Bản đổ lưu số tổ tiên tối đa của mỗi người
17 unordered_map<string, int> peopleBirth_81; // Bản đổ lưu số người theo ngày sinh
18 long countAlone_81 = 0; // Biến đếm số người một mình
19 long countGroup_81 = 0; // Biến đếm số nhóm
20
     int main()
21
23
               ios_base::sync_with_stdio(θ);
               string s;
25
27
               for (int i = 1900; i < 2030; i++) {
                       for (int j = 1; j <= 12; j++) {
                                 for (int k = 1; k <= 31; k++) {
29
30
                                          string year = to_string(i);
                                           string month = (j < 10) ? "0" + to_string(j) : to_string(j);
                                           string day = (k < 10) ? "0" + to_string(k) : to_string(k);
33
                                           string date = year + "-" + month + "-" + day;
34
                                           totalPeople_81[date] = 0; // Khởi tạ
```

#### Mai Minh Hoàng - 20215381

```
37
38
39
40
41
              while (getline(cin, s)) {
                      if (s == "*") {//gap * thi divng}
43
                              break;
44
45
                      stringstream ss(s);//phân tách đầu vào
                      string UserId;
                      string Birth;
47
                      string fCode;
49
                      string mCode;
50
                      string isAlive;
                      string rCode;
                      ss >> UserId >> Birth >> fCode >> mCode >> isAlive >> rCode;
54
                      countPeople_81++; // Tăng biến đếm tổng số người
                      if ((fCode != "0000000") && (mCode != "0000000")) {
                              maxAcestor_81[UserId] = (maxAcestor_81[fCode] > maxAcestor_81[mCode] ? maxAcestor_81[fCode] + 1 : maxAcesto
56
57
                              countGroup_81++; // Tăng biến đếm số nhóm
58
                              notAlone_81[UserId] = 1; // Đánh dấu người này không một mình
                              notAlone_81[fCode] = 1; // Đánh dấu cha/mẹ của người này không một mình
notAlone_81[mCode] = 1; // Đánh dấu cha/mẹ của người này không một mình
59
60
62
                      else {
                              notAlone_81[UserId] = 0; // Đánh dấu người này một mình
64
66
                      peopleBirth_81[Birth] += 1; // Tăng số người sinh vào ngày Birth
67
                      totalPeople_81[Birth] += 1; // Tăng tổng số người sinh vào ngày Birth
68
69
              int previousValue = 0;
              for (auto& p : totalPeople_81) {//cộng dồn ngày sau bằng tổng tất cả người sinh các ngày trước đó
70
                      int currentValue = p.second;
                      p.second = previousValue + currentValue; // Cập nhật tổng số người sinh đến ngày hiện tại
72
```

#### Mai Minh Hoàng - 20215381

```
previousValue = p.second;//gán lại để tới vòng lặp tiếp
75
             for (auto& p : notAlone_81) {//duyệt qua notAlone_81
                     if (p.second == 0) {
78
                             countAlone_81++; // Đếm số người một minh
79
80
81
             while (getline(cin, s)) {
82
                     if (s == "***") {//gặp *** thi dừng
83
84
                     if (s[7] == 'P') {//chuỗi là NUMBER_PEOPLE_BORN_BETWEEN <from_date> <to_date>
85
                             if (s.length() > 34) {
86
87
                                     stringstream ss(s);
88
                                     string a, b, c;
89
                                     ss >> a >> b >> c;
98
91
                                     cout << totalPeople_81[c] - totalPeople_81[b] + peopleBirth_81[b] << "\n"; // Trå về số người sinh
92
                             else if (s.length() == 13) {//chuỗi là NUMBER_PEOPLE
                                     cout << countPeople_81 << "\n"; // Trå về tổng số người
94
95
96
                             else {//chuỗi là NUMBER_PEOPLE_BORN_AT <date>
                                     stringstream ss(s);
                                     string a, b;
98
99
                                    ss >> a >> b;
                                     cout << peopleBirth_81[b] << "\n"; // Trả về số người sinh vào ngày <b>
188
101
102
103
104
                             if (s[0] == 'M' && s[1] == 'O') {//chuỗi là MOST_ALIVE_ANCESTOR <code>
105
106
                                     stringstream ss(s);
107
                                     string a, b;
108
                                     ss >> a >> b;
                     else
                             if (s[0] == 'M' && s[1] == 'O') {//chuỗi là MOST_ALIVE_ANCESTOR <code>
105
106
                                     stringstream ss(s);
107
                                     string a, b;
108
                                     ss >> a >> b;
```

```
184 else

185 if (s[0] == 'M' && s[1] == '0') {//chuỗi là MOST_ALIVE_ANCESTOR < code>

186 stringstream ss(s);

187 string a, b;

188 ss >> a >> b;

189 cout << maxAcestor_81[b] << "\n"; // Trả về số tổ tiên tối đa củo người b

110 }

111 else {//chuỗi là MAX_UNRELATED_PEOPLE

112 // Trả về tổng số người một minh và số nhóm chính là kết quả để 2 người bất kỳ trong nhóm không phả

113 cout << countAlone_81 + countGroup_81 << "\n";

114 }

115

116

117 }

118 }
```

### Test:

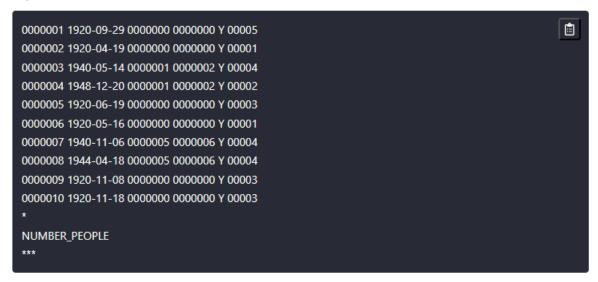
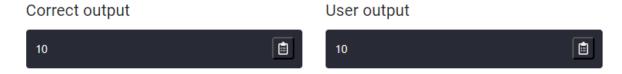


Figure 12 b2- Output 1



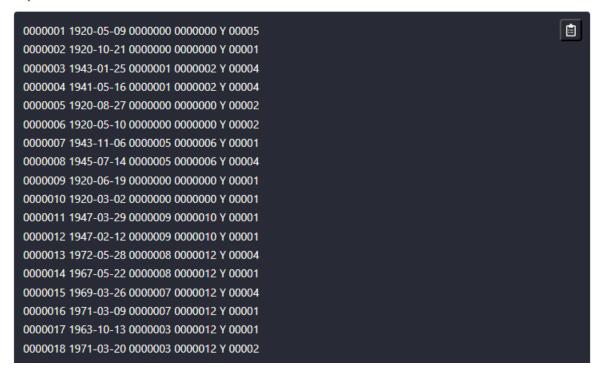
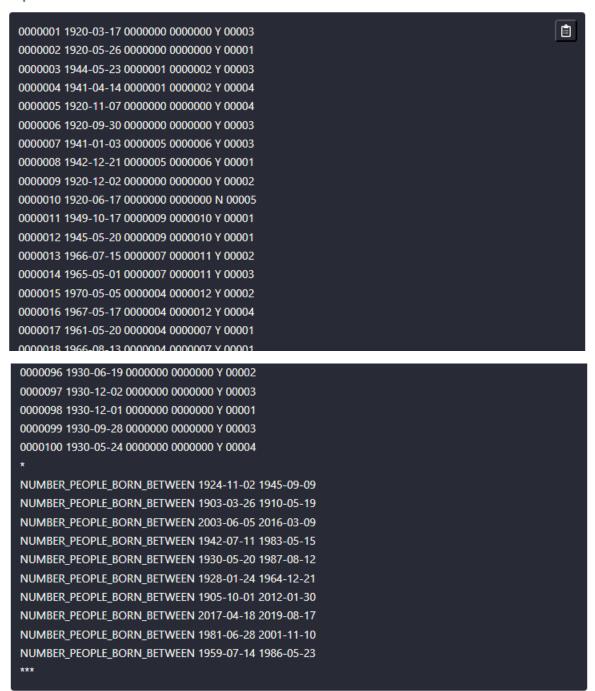


Figure 14 b2- Output 2





## Correct output



## User output

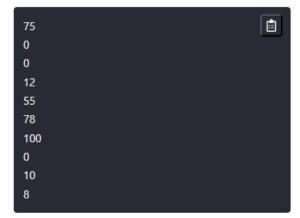
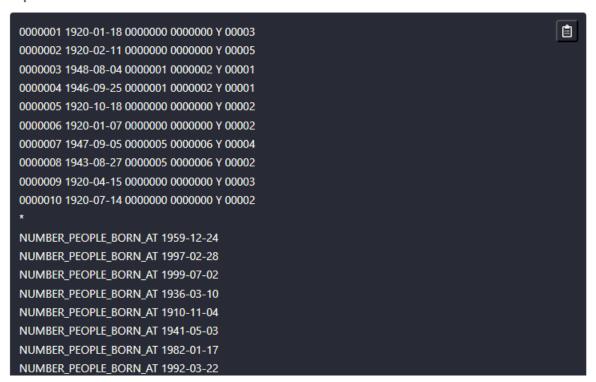
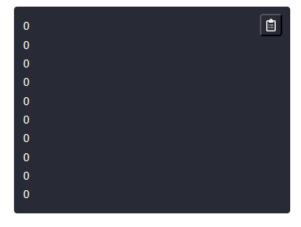


Figure 17 b2- Input 4



```
NUMBER_PEOPLE_BORN_AT 1956-03-03
NUMBER_PEOPLE_BORN_AT 1998-10-02
***
```

#### Correct output



#### User output



#### Code:

```
//Mai Minh Hoàng
//20215381
#include <list>
#include <iostream>
#include<set>
#include<sstream>
#include<unordered_map>
#include<vector>
#include<map>
using namespace std;
long countPeople_81 = 0; // Biến đếm tổng số người
long sumtotalPeople_81 = 0; // Biến đếm tổng số người
map<string, long> totalPeople_81; // Bản đồ lưu tổng số người theo ngày sinh
unordered_map<string, int > notAlone_81; // Bản đồ kiểm tra người khống một mình
unordered_map<string, int> maxAcestor_81; // Bản đồ lưu số tố tiên tối đa của mỗi
người
unordered_map<string, int> peopleBirth_81; // Bản đồ lưu số người theo ngày sinh
long countAlone_81 = 0; // Biến đếm số người một mình
long countGroup_81 = 0; // Biến đếm số nhóm
int main()
{
      ios_base::sync_with_stdio(0);
      string s;
      // Tạo bản đồ cho tất cả các ngày từ năm 1900 đến 2030
      for (int i = 1900; i < 2030; i++) {</pre>
             for (int j = 1; j <= 12; j++) {
                   for (int k = 1; k <= 31; k++) {
                          string year = to_string(i);
```

```
Mai Minh Hoàng - 20215381
                         string month = (j < 10) ? "0" + to_string(j) :
to_string(j);
                         string day = (k < 10) ? "0" + to_string(k) : to_string(k);
                         string date = year + "-" + month + "-" + day;
                         totalPeople_81[date] = 0; // Khởi tạo số người sinh vào
ngày đó là 0
                  }
            }
      }
      // Đoc dữ liêu từ đầu vào
      while (getline(cin, s)) {
            if (s == "*") {//gặp * thì dừng
                  break;
            }
            stringstream ss(s);//phân tách đầu vào
            string UserId;
            string Birth;
            string fCode;
            string mCode;
            string isAlive;
            string rCode;
            ss >> UserId >> Birth >> fCode >> mCode >> isAlive >> rCode;
            countPeople_81++; // Tăng biến đếm tổng số người
            if ((fCode != "0000000") && (mCode != "0000000")) {
                  maxAcestor_81[UserId] = (maxAcestor_81[fCode] >
notAlone_81[UserId] = 1; // Đánh dấu người này không một mình
                  notAlone_81[fCode] = 1; // Đánh dấu cha/me của người này không
môt mình
                  notAlone_81[mCode] = 1; // Đánh dấu cha/me của người này không
môt mình
            }
            else {
                  notAlone_81[UserId] = 0; // Đánh dấu người này một mình
            peopleBirth_81[Birth] += 1; // Tăng số người sinh vào ngày Birth
            totalPeople_81[Birth] += 1; // Tăng tổng số người sinh vào ngày Birth
      int previousValue = 0;
      for (auto& p : totalPeople_81) {//công dồn ngày sau bằng tổng tất cả người
sinh các ngày trước đó
            int currentValue = p.second;
            p.second = previousValue + currentValue; // Câp nhất tổng số người sinh
đến ngày hiện tại
            previousValue = p.second;//gán lại để tới vòng lặp tiếp
      for (auto& p : notAlone_81) {//duyêt qua notAlone_81
            if (p.second == 0) {
                  countAlone_81++; // Đếm số người một mình
      while (getline(cin, s)) {
            if (s == "***") {//gặp *** thì dừng
                  break:
```

#### Mai Minh Hoàng - 20215381

```
if (s[7] == 'P') {//chuỗi là NUMBER_PEOPLE_BORN_BETWEEN <from_date>
<to_date>
                    if (s.length() > 34) {
                           stringstream ss(s);
                           string a, b, c;
                           ss >> a >> b >> c;
                           cout << totalPeople_81[c] - totalPeople_81[b] +</pre>
peopleBirth_81[b] << "\n"; // Trả về số người sinh trong khoảng thời gian
                    else if (s.length() == 13) {//chuổi là NUMBER_PEOPLE
                           cout << countPeople_81 << "\n"; // Trả về tống số người</pre>
                    }
                    else {//chuỗi là NUMBER_PEOPLE_BORN_AT <date>
                           stringstream ss(s);
                           string a, b;
                           ss >> a >> b;
                           cout << peopleBirth_81[b] << "\n"; // Trả về số người sinh
vào ngày <b>
                    }
             }
             else
                    if (s[0] == 'M' && s[1] == 'O') {//chuỗi là MOST_ALIVE_ANCESTOR
<code>
                           stringstream ss(s);
                           string a, b;
                           ss >> a >> b;
                           cout << maxAcestor_81[b] << "\n"; // Trả về số tổ tiên tối</pre>
đa của người b
                    else {//chuỗi là MAX_UNRELATED_PEOPLE
                          // Trả về tổng số người một mình và số nhóm chính là kết
quả đế 2 người bất kỳ trong nhóm không phải vợ chồng/ anh chị em
                           cout << countAlone_81 + countGroup_81 << "\n";</pre>
                    }
      }
}
```