



Problem Solving (CPP18)

هذا البرنامج التدريبي مُصاغ بعناية لتمكين المتدربين من تطوير قدراتهم الفكرية على غرار المبرمجين المحترفين، والتعاون بكفاءة ضمن فريق محترف في شركة "جمال تك" أو أي مؤسسة متعددة الجنسيات أخرى. نظرًا لأهمية اللغة الإنجليزية في بيئة العمل العالمية، يتم تقديم المحتوى التدريبي بالإنجليزية. لا يشترط إتقان اللغة بشكل كامل، لكن من الضروري امتلاك القدرة الكافية لفهم المتطلبات وتنفيذها بشكل فعال. يُمكن للمتدربين استخدام مترجم جوجل أو الاستعانة بـ "شات جي بي تي" للتغلب على أية عقبات لغوية، المهم هو الفهم الدقيق للمطلوب وتحقيقه بنجاح.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمارين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق في نهاية الملف.

قد يتضمن الحل كودًا برمجيًا غير مفسر بعد، والغرض من ذلك هو تشجيعك على محاولة فهم الأكواد البرمجية الجديدة التي لم تتعرض لها من قبل. هذه المهارة ضرورية في سوق العمل، حيث تتطور لغات البرمجة باستمرار ويظهر كل يوم لغات جديدة. ستواجه دائمًا أكوادًا لم تدرسها من قبل، ومن المهم أن تكون قادرًا على فهمها بنفسك دون الحاجة إلى دراسة مسبقة. يمكنك الاستعانة بمحرك البحث جوجل، أو استخدام ChatGPT، أو حتى اللجوء لأصدقائك للمساعدة. الهدف الأساسي هو أن تصل إلى فهم معنى كل كود بأي طريقة ممكنة لتتمكن من إيجاد موقعك في سوق العمل.

إن وجود كود برمجي غير مفسر يشكل تحديًا يتوجب عليك إيجاد حل له. هذا النوع من التدريبات يعد جزءًا أساسيًا من تدريبات 'Problem Solving'، التي تهدف إلى تمكينك من أداء عملك بفاعلية بغض النظر عن التحديات والعقبات. هذه القدرة على حل المشكلات هي ما يتمتع به العاملون في 'جمال تك'، ومن الضروري أن تطور في نفسك هذه المهارة لتصبح عضوًا فعالًا في فريق عمل 'جمال تك'.

Gammal Tech's Communication Network

Background

Gammal Tech, a trailblazer in the software industry, has a unique approach to team communication. Recognizing the diverse linguistic backgrounds of its global team, the company embraces this diversity, allowing team members to communicate in any language. They have developed a system that identifies the best communicators across different languages, ensuring seamless collaboration and innovation.

Problem Statement

As a software engineer at Gammal Tech, you are tasked to design a system that maps each team member's preferred language and identifies those who can bridge communication gaps between different language speakers. The goal is to find the best communicators who can effectively interact with the most team members in their non-native languages.

Input Format



- The first line contains an integer N , the number of employees.
- The next N lines contain the name of an employee followed by their preferred language, separated by a space.
- The last line contains a string s , representing the language of an urgent project.

Output Format

- Print the name(s) of the employee(s) who can communicate in language s and with the highest number of other employees in different languages.
- If there are multiple such employees, print their names in alphabetical order, each on a new line.

Constraints

- $1 \leq N \leq 1000$
- Names consist only of alphabets and are unique.
- Languages are represented by unique strings.

Sample Input:

```
5
Alice English
Bob French
Charlie Spanish
Diana Russian
Eve English
English
```

Sample Output:

```
Alice
Eve
```

Explanation: Both Alice and Eve can communicate in the urgent project's language, English. They are the best communicators as they can interact with all other team members who speak French, Spanish, and Russian.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمرين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق



C++ Programming Solution:

```
#include <iostream>
#include <map>
#include <set>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int N;
    cin >> N;
    map<string, set<string>> languageMap; // Maps language to employees
    map<string, string> employeeLanguage; // Maps employee to their
    language
    string name, language;
    for (int i = 0; i < N; ++i) {
        cin >> name >> language;
        languageMap[language].insert(name);
        employeeLanguage[name] = language;
    }

    string projectLanguage;
    cin >> projectLanguage;

    // Find employees who can communicate in the project language
    set<string> potentialCommunicators = languageMap[projectLanguage];

    // Find the best communicators
    map<int, set<string>> communicatorScore; // Maps score to employee
    for (const auto& employee : potentialCommunicators) {
        int score = 0;
        for (const auto& lang : languageMap) {
            if (lang.first != employeeLanguage[employee]) {
                score += lang.second.size();
            }
        }
        communicatorScore[score].insert(employee);
    }

    // Output the best communicators
    if (!communicatorScore.empty()) {
        auto it = communicatorScore.rbegin(); // The highest score
        for (const auto& employee : it->second) {
            cout << employee << endl;
        }
    }

    return 0;
}
```