



Problem Solving (CPP5)

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 Global Team Formation

Background: Gammal Tech, a trailblazer in the software development industry, renowned for its innovative approach and diverse teams, faces a unique challenge. To promote global collaboration and diversity, the company mandates that each of its project teams must consist of exactly three members, with no more than one member from the same country. This policy highlights Gammal Tech's commitment to creating a multicultural work environment and leveraging the strength of its international talent pool.

Problem Statement: You are tasked with developing a system to automate the team formation process at Gammal Tech. The system should ensure that each team is composed of three members, and each member should come from a different country. Given a list of software developers along with their respective countries, your program should output all possible valid team combinations following Gammal Tech's guidelines.

Input Format:



- The first line of input contains an integer N ($3 \leq N \leq 100$), the number of software developers.
- The following N lines each contain a string s representing the name of the developer, followed by a string c representing the country of the developer. s and c are separated by a space.

Output Format:

- For each valid team combination, output a line containing the names of the three team members separated by commas.
- If there are no valid combinations, print "No valid team formations possible."

Constraints:

- Names and country strings consist only of alphanumeric characters and are case-sensitive.
- The length of each string is at most 20 characters.

Sample Input:

```
6
Alice USA
Bob UK
Charlie Germany
Diana USA
Eve UK
Frank Germany
```

Sample Output:

```
Alice,Bob,Charlie
Alice,Bob,Frank
Alice,Charlie,Eve
Alice,Eve,Frank
Bob,Charlie,Diana
Bob,Diana,Frank
Charlie,Diana,Eve
Diana,Eve,Frank
```



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 <vector>
#include <unordered_map>
#include <string>
using namespace std;

struct Developer {
    string name;
    string country;
};

vector<Developer> parseInput(int n) {
    vector<Developer> developers;
    string name, country;
    for (int i = 0; i < n; ++i) {
        cin >> name >> country;
        developers.push_back({name, country});
    }
    return developers;
}

bool isValidTeam(const Developer &a, const Developer &b, const Developer &c) {
    return (a.country != b.country) && (a.country != c.country) &&
    (b.country != c.country);
}

void printValidTeams(const vector<Developer>& developers) {
    bool found = false;
    int n = developers.size();
    for (int i = 0; i < n; ++i) {
        for (int j = i + 1; j < n; ++j) {
            for (int k = j + 1; k < n; ++k) {
                if (isValidTeam(developers[i], developers[j],
                developers[k])) {
                    cout << developers[i].name << "," <<
                    developers[j].name << "," << developers[k].name << endl;
                    found = true;
                }
            }
        }
    }
    if (!found) {
        cout << "No valid team formations possible." << endl;
    }
}
```



```
int main() {  
    int n;  
    cin >> n;  
    vector<Developer> developers = parseInput(n);  
    printValidTeams(developers);  
    return 0;  
}
```