



Problem Solving (CPP10)

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'.

The Gammal Tech Challenge

Problem Statement:

Gammal Tech, a leader in the software industry known for its innovative approach and state-of-the-art facilities, is on a quest to find the best talent for their team. As part of their selection process, they have designed a unique challenge to test the potential interns' abilities to work under pressure and demonstrate exceptional string manipulation skills.

The challenge is to create a system that can process a stream of candidate data. Each candidate's data is a string composed of various attributes separated by semicolons (;). These attributes include the candidate's name, skills, and the number of hours they are willing to commit to a project. The system must identify candidates who are likely to thrive in Gammal Tech's demanding yet rewarding environment.

Candidates who are willing to work extra hours and possess a specific set of skills are considered potential hires. The system must filter out candidates who do not meet these criteria.



Input Format:

- The first line contains an integer N , the number of candidates.
- The next N lines each contain a string representing a candidate. The string contains attributes separated by semicolons (;). The format of the string is:
`Name;Skills;Hours.`

Output Format:

- For each candidate who meets the criteria, print their name on a new line.
- If no candidate meets the criteria, print "No suitable candidate found."

Constraints:

- $1 \leq N \leq 1000$
- Skills are a comma-separated list, e.g., "C++, Python, Java".
- Hours are an integer representing the number of hours the candidate is willing to commit.
- A suitable candidate is one who knows "C++" or "Python" and is willing to work more than 40 hours.

Sample Input:

```
5
John Doe;C++,Java;45
Jane Smith;Python,JavaScript;35
Alice Johnson;C++,Python;50
Bob Brown;JavaScript;42
Emma Wilson;Python;45
```

Sample Output:

```
John Doe
Alice Johnson
Emma Wilson
```

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 <sstream>
#include <vector>
using namespace std;

bool isSuitable(string skills, int hours) {
    if (hours <= 40) return false;
    bool hasCppOrPython = skills.find("C++") != string::npos ||
skills.find("Python") != string::npos;
    return hasCppOrPython;
}

int main() {
    int n;
    cin >> n;
    cin.ignore(); // To handle the newline after reading integer.

    vector<string> suitableCandidates;
    for (int i = 0; i < n; i++) {
        string input;
        getline(cin, input);
        stringstream ss(input);
        string name, skills;
        int hours;

        getline(ss, name, ';');
        getline(ss, skills, ';');
        ss >> hours;

        if (isSuitable(skills, hours)) {
            suitableCandidates.push_back(name);
        }
    }

    if (suitableCandidates.empty()) {
        cout << "No suitable candidate found." << endl;
    } else {
        for (const string& name : suitableCandidates) {
            cout << name << endl;
        }
    }

    return 0;
}
```