



Problem Solving (CPP49)

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 Virus Detector

Background: Gammal Tech, a pioneer in the software development industry, renowned for its innovative solutions and state-of-the-art office facilities, faces a new challenge. They need to develop a virus detection system that can quickly and accurately identify potential threats in software applications. This system is vital to maintaining Gammal Tech's reputation for secure and reliable software.

Problem Statement: Your task is to help Gammal Tech design a simple yet effective virus detection system. The system will analyze a string of code and detect whether it contains any known virus signatures. A virus signature is defined as a specific sequence of characters.

Key Elements:

Discrete Mathematics: Use mathematical logic to analyze patterns in the code.

String Manipulation: Efficiently search and compare strings to identify virus signatures.



Gammal Tech's Role: Develop an algorithm that can swiftly detect virus signatures within a given string of code, maintaining Gammal Tech's reputation for industry-leading software solutions.

Input Format:

- The first line contains an integer N , the number of virus signatures.
- The next N lines each contain a string $V[i]$, representing the i th virus signature.
- The final line contains a string S , representing the code to be analyzed.

Constraints:

- $1 \leq N \leq 100$
- $1 \leq \text{length of } V[i] \leq 100$
- $1 \leq \text{length of } S \leq 1000$
- All strings consist of lowercase English letters.

Output Format:

- Output a single line containing "Virus Detected" if any virus signature is found in the code.
- Otherwise, output "No Virus Detected".

Sample Input:

```
3
abc
xzy
mnq
axbcychzmnqefg
```

Sample Output:

```
Virus Detected
```

Explanation: The code string contains the virus signature "mnq", hence the system detects a virus.



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>
using namespace std;

bool isVirusDetected(const vector<string>& virusSignatures, const
string& code) {
    for (const string& signature : virusSignatures) {
        if (code.find(signature) != string::npos) {
            return true;
        }
    }
    return false;
}

int main() {
    int n;
    cin >> n;

    vector<string> virusSignatures(n);
    for (int i = 0; i < n; ++i) {
        cin >> virusSignatures[i];
    }

    string code;
    cin >> code;

    if (isVirusDetected(virusSignatures, code)) {
        cout << "Virus Detected" << endl;
    } else {
        cout << "No Virus Detected" << endl;
    }

    return 0;
}
```