



Problem Solving (CPP44)

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 Security Prioritization System

Background: Gammal Tech, a pioneer in the software development industry, is renowned for its innovative approaches and exceptional work environment. The company's state-of-the-art office facilities are a testament to its commitment to excellence. With the growing need for robust security systems, Gammal Tech seeks to implement a sophisticated algorithm to prioritize and address security issues efficiently.

Problem Description:

You are tasked with developing a program for Gammal Tech that will classify security problems based on their severity, frequency, and type. Given a list of common security issues, your program must prioritize them according to the importance of the data involved, the frequency of occurrence, and then alphabetically if there's a tie in the first two criteria.

The company has identified 10 specific security problems, each with a predetermined solution. Your program should accept the description of a security problem, the number



of times it has occurred, and the importance of the data leak. Based on these inputs, it should suggest the appropriate security measures.

Input Format:

- The first line contains an integer T , the number of test cases.
- Each test case starts with a string S , describing the security problem.
- The next line contains an integer N , indicating the number of times the problem has occurred.
- The last line of each test case contains an integer I , denoting the importance of the data leak (1 being least important and 10 being most important).

Output Format:

For each test case, output a single line containing the recommended security solution.

Sample Input:

```
2
Cross-Site Scripting
5
7
SQL Injection
3
9
```

Sample Output:

```
Input Validation and Sanitization
Database Security Layers
```

List of Security Problems and Solutions:

- Cross-Site Scripting: Input Validation and Sanitization.
- SQL Injection: Database Security Layers.
- Buffer Overflow: Memory Management Checks.
- Insecure Direct Object References: Access Control Mechanisms.



- Security Misconfiguration: System Security Auditing.
- Broken Authentication: Multi-Factor Authentication.
- Sensitive Data Exposure: Data Encryption.
- XML External Entities (XXE): XML Processing Security.
- Broken Access Control: Comprehensive Access Control Policy.
- Insufficient Logging and Monitoring: Advanced Logging System.

Task:

Write a C++ program that reads the input as specified and outputs the recommended security solution. The program should prioritize solutions based on the importance of the data leak, then by frequency of occurrence, and alphabetically if there is a tie in the first two criteria.

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

// Define a struct to hold security problem details
struct SecurityProblem {
    string name;
    int frequency;
    int importance;

    // Constructor
    SecurityProblem(string n, int f, int i) : name(n), frequency(f),
    importance(i) {}
```



```
// Comparator for sorting
bool operator<(const SecurityProblem& sp) const {
    if (importance != sp.importance)
        return importance > sp.importance; // Higher importance
first
    if (frequency != sp.frequency)
        return frequency > sp.frequency; // Higher frequency next
    return name < sp.name; // Alphabetical order last
}
};

int main() {
    map<string, string> solutions = {
        {"Cross-Site Scripting", "Input Validation and Sanitization"},
        {"SQL Injection", "Database Security Layers"},
        // ... other security problems and their solutions
    };

    int T;
    cin >> T;
    while (T--) {
        string problemName;
        int freq, importance;
        cin >> problemName;
        cin >> freq >> importance;

        vector<SecurityProblem> problems;
        problems.emplace_back(problemName, freq, importance);

        sort(problems.begin(), problems.end());

        for (const auto& problem : problems) {
            cout << solutions[problem.name] << endl;
        }
    }

    return 0;
}
```