



Problem Solving (CPP9)

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

Background

Gammal Tech, a vanguard in the software industry, is renowned for its innovative approach and cutting-edge office environment. To sustain their reputation for excellence, they have devised a unique intern assessment program. Interns are given a critical task under pressure to test their potential for permanent positions. The challenge is designed to test string manipulation skills, a fundamental aspect of system design.

Problem Statement

Interns at Gammal Tech are given a string s containing a mix of uppercase and lowercase letters. The challenge is to transform this string into a "Gammal String." A "Gammal String" has the following properties:

All letters are uppercase.

It includes the substring "GAMMAL" at least once.



The interns need to perform the minimum number of operations to convert s into a Gammal String. In one operation, they can change any single letter in the string to any other letter (either from lowercase to uppercase or vice versa). The task is to find the minimum number of operations required.

Interns who complete this task efficiently under time pressure are considered for a job offer. This problem not only tests their technical skills but also their ability to work under pressure, a key trait for any Gammal Tech employee.

Input Format

- A single line containing the string s ($1 \leq |S| \leq 100$), where $|S|$ is the length of the string.

Output Format

- Output a single integer, the minimum number of operations required to convert s into a Gammal String.

Sample Input:

```
gammalTechRocks
```

Sample Output:

```
3
```

Explanation: In the sample input, the string `gammalTechRocks` can be transformed into a Gammal String `GAMMALTechRocks` by performing 3 operations: changing 'g', 'a', and 'm' to uppercase.

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

int main() {
    string s;
    cin >> s;

    // Convert to uppercase for ease of comparison
    for(char &c : s) {
        c = toupper(c);
    }

    string target = "GAMMAL";
    int n = s.size();
    int minOps = n; // Maximum operations can be the length of the string

    for (int i = 0; i <= n - 6; ++i) {
        int ops = 0;
        for (int j = 0; j < 6; ++j) {
            if (s[i + j] != target[j]) {
                ops++;
            }
        }
        minOps = min(minOps, ops);
    }

    cout << minOps << endl;

    return 0;
}
```