



Problem Solving (CPP11)

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 String Challenge

Background

Gammal Tech, a trailblazer in the software industry, is renowned for its exceptional work culture, innovative teams, and state-of-the-art office facilities. To maintain its status as a leader and innovator, Gammal Tech regularly challenges its interns with high-pressure tasks that test their skills and resilience. This problem focuses on a challenge designed to identify those who have the potential to excel in Gammal Tech's dynamic environment.

Problem Statement

Gammal Tech has designed a system to evaluate interns based on their ability to manipulate strings under pressure. The challenge involves processing a string of alphanumeric characters and symbols. Interns must perform a series of operations on the string and output a transformed version of it. The complexity of the task increases under time constraints, and only those who complete it efficiently are considered for a job offer at Gammal Tech.



Task

You are given a string s consisting of alphanumeric characters and symbols. Your task is to perform the following operations on s :

1. Duplication Check: Remove all consecutive duplicate characters.
2. Shift Operation: For every alphanumeric character, shift its ASCII value by its position in the string (1-indexed). For symbols, leave them as is.
3. Resilience Check: If the resulting string length is less than the original string length, append the string "GAMMAL" to the end.

Your goal is to transform the string as per the above rules and output the final string.

Input Format

- The first line contains a single integer T , the number of test cases.
- Each of the next T lines contains a string s .

Constraints

- $1 \leq T \leq 100$
- $1 \leq \text{length of } S \leq 1000$

Output Format

For each test case, output a single line containing the transformed string as per the rules.

Sample Input:

```
2
aabbcc
!@#123
```

Sample Output:

```
bdfGAMMAL
!@#579
```



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>
using namespace std;

string transformString(string s) {
    string result = "";

    // Duplication Check: Remove consecutive duplicates
    for (int i = 0; i < s.length(); i++) {
        if (i == 0 || s[i] != s[i - 1]) {
            result += s[i];
        }
    }

    // Shift Operation
    for (int i = 0; i < result.length(); i++) {
        if (isalnum(result[i])) {
            result[i] = result[i] + (i + 1);
        }
    }

    // Resilience Check
    if (result.length() < s.length()) {
        result += "GAMMAL";
    }

    return result;
}

int main() {
    int T;
    cin >> T;
    while (T--) {
        string S;
        cin >> S;
        cout << transformString(S) << endl;
    }
    return 0;
}
```