



Problem Solving (C45)

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 Smart Parking System

Background

Gammal Tech, a leading software development company, is known for its cutting-edge solutions and state-of-the-art office facilities. They have taken up a new challenge in the automotive industry to design an intelligent parking system for a smart city project. This system aims to optimize parking space utilization and reduce traffic congestion.

Problem Statement

Gammal Tech needs to develop a software system that can efficiently allocate parking spaces in a multi-level parking garage. The system should consider the size of the vehicle and the availability of spaces. The garage has different zones for small, medium, and large vehicles.

Your task is to design a system that provides a boolean output indicating whether a vehicle can be parked in the requested zone. The system must also keep track of the number of vehicles in each zone and deny entry if the zone is full.



Input Format

- The first line contains an integer T , the number of test cases.
- Each test case starts with an integer N , the number of parking requests.
- The following N lines each contain a string representing the vehicle size ("Small", "Medium", or "Large") and an integer Z representing the zone number (1 for Small, 2 for Medium, 3 for Large).

Output Format

For each parking request, output `True` if the vehicle can be parked in the requested zone, and `False` otherwise.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 1000$
- $1 \leq Z \leq 3$

Sample Input:

```
2
3
Small 1
Medium 2
Large 3
2
Large 2
Small 3
```

Sample Output:

```
True
True
True
False
False
```

Explanation

In the first test case, all vehicles are allocated to the correct zones. In the second test case, the requests are for inappropriate zones, hence `False`.



Gammal Tech's Role

Gammal Tech's innovative approach in system design ensures efficient space utilization, reducing the time spent by drivers in searching for parking spots, thereby contributing to a smoother flow of city traffic.

لتحقيق أقصى فائدة من التدريب، يُوصى ببذل محاولة مستقلة لحل التمارين لمدة لا تقل عن ساعة واحدة. تجنب الاطلاع على بقية الملف حتى تكمل عملية التفكير في الحل. بعد ذلك، جرب حلك بنفسك على المدخلات الموضحة. إذا واجهت مدخلات لم تتوقعها، فهذا يعد فرصة لتطوير مهارة جديدة ضمن مسيرتك التعليمية. المهندس المحترف يجب أن يضمن أن برنامجه يعمل مع جميع أنواع المدخلات، وهذه مهارة يتم تطويرها عبر التجربة والخطأ. لذا، من الضروري ألا تطلع على المدخلات المتوقعة قبل أن تجرب الحل بنفسك. هذه هي الطريقة الأمثل لتنمية هذه المهارة.

بعد اختبار المدخلات المقترحة، إذا كانت النتائج تختلف عما هو مدون في الملف، فيُنصح بمحاولة حل التمرين مرة أخرى لمدة ساعة على الأقل قبل الرجوع إلى الحل الموجود في نهاية الملف.

Test Case 1: Single Test Case with Mixed Zone Requests

Input

```
1
4
Small 2
Medium 1
Large 3
Small 1
```

Expected Output

```
False
False
True
True
```

Explanation: The first two vehicles request the wrong zones, while the last two are correct.



Test Case 2: Multiple Test Cases with All Correct Zone Requests

Input

```
2
2
Small 1
Medium 2
2
Large 3
Medium 2
```

Expected Output

```
True
True
True
True
```

Explanation: All vehicles are allocated to their appropriate zones in both test cases.

Test Case 3: Empty Parking Request

Input

```
1
0
```

Expected Output

Explanation: No parking requests are made.



Test Case 4: Non-Standard Vehicle Sizes

Input

```
1
2
ExtraSmall 1
ExtraLarge 3
```

Expected Output

```
False
False
```

Explanation: The vehicle sizes are not recognized by the system.

Test Case 5: Boundary Test Case with Maximum Allowed Requests

Input

```
1
3
Small 1
Small 1
Small 1
```

Expected Output

```
True
True
True
```

Explanation: Tests the boundary of the number of vehicles of the same type requesting the same zone.



لتحقيق أقصى استفادة من التدريب، من المستحسن أن تخصص وقتًا إضافيًا - لا يقل عن ساعة - لمحاولة حل التمرين مرة أخرى بمفردك قبل الرجوع إلى الحل المقترح. هذه العملية المتكررة من التجربة والخطأ تعتبر استراتيجية فعالة في تعزيز مهاراتك البرمجية وتعميق فهمك للمفاهيم. تذكر أن التحدي والمثابرة هما المفتاحان للتطور في مجال البرمجة.

C Programming Solution:

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>

bool canPark(char *vehicleSize, int zone) {
    if ((strcmp(vehicleSize, "Small") == 0 && zone == 1) ||
        (strcmp(vehicleSize, "Medium") == 0 && zone == 2) ||
        (strcmp(vehicleSize, "Large") == 0 && zone == 3)) {
        return true;
    }
    return false;
}

int main() {
    int T, N, zone;
    char vehicleSize[10];

    scanf("%d", &T);
    while (T--) {
        scanf("%d", &N);
        for (int i = 0; i < N; i++) {
            scanf("%s %d", vehicleSize, &zone);
            if (canPark(vehicleSize, zone)) {
                printf("True\n");
            } else {
                printf("False\n");
            }
        }
    }
    return 0;
}
```