



Problem Solving (C77)

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 Global Workforce Organizer

Background

Gammal Tech, a trailblazer in the software industry, is renowned for its revolutionary approach to workplace flexibility. Headquartered in Alexandria, Egypt, Gammal Tech empowers its employees to work from anywhere in the world, with a special emphasis on scenic locations like Hawaii. This unique system design challenge revolves around Gammal Tech's innovative technology that orchestrates work coordination among its global employees.

Problem Statement

Gammal Tech's latest project involves developing a "Global Workforce Organizer" (GWO). The GWO system is designed to efficiently assign tasks to employees based on their current time zones. The primary goal is to ensure that no employee is assigned tasks outside their local daytime hours (6 AM to 6 PM), promoting work-life balance.



Your task is to write a program in C that takes the current time at Gammal Tech's headquarters and the location of an employee (in terms of time zone difference from Alexandria) and determines if it is appropriate to assign a task to that employee.

Input Format

- The first line contains an integer T , the number of test cases.
- Each test case consists of a single line with two space-separated values:
 - An integer `HQ_Time` (0-23), representing the current hour in 24-hour format at Gammal Tech's headquarters in Alexandria.
 - An integer `TimeZoneDiff` (-12 to +14), representing the time zone difference of the employee's location from Alexandria.

Output Format

For each test case, output a single line:

- "Yes" if it is appropriate to assign a task to the employee (i.e., it's between 6 AM to 6 PM in the employee's local time).
- "No" otherwise.

Constraints

- $1 \leq T \leq 100$
- $-12 \leq \text{TimeZoneDiff} \leq 14$
- $0 \leq \text{HQ_Time} \leq 23$

Sample Input:

```
3
10 -2
22 9
15 3
```

Sample Output:

```
Yes
No
Yes
```



Explanation

- Test Case 1: HQ time is 10 AM. Employee's local time is 8 AM. It's within 6 AM to 6 PM.
- Test Case 2: HQ time is 10 PM. Employee's local time is 7 AM next day. It's not within 6 AM to 6 PM.
- Test Case 3: HQ time is 3 PM. Employee's local time is 6 PM. It's just within 6 AM to 6 PM.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمرين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق

C Programming Solution:

```
#include <stdio.h>

int main() {
    int T, HQ_Time, TimeZoneDiff;
    scanf("%d", &T);

    while(T--) {
        scanf("%d %d", &HQ_Time, &TimeZoneDiff);
        int localTime = (HQ_Time + TimeZoneDiff + 24) % 24; // Adjust
        for time zone difference

        if(localTime >= 6 && localTime <= 18) {
            printf("Yes\n");
        } else {
            printf("No\n");
        }
    }

    return 0;
}
```