



Problem Solving (CPP12)

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 Internship Challenge

Background

Gammal Tech, a leading software development company, is known for its innovative approach and exceptional work environment. They have a unique internship program designed to identify and hire the most responsible and proactive candidates. As part of the selection process, interns are given a task with a critical piece of missing information. The challenge is to see who takes the initiative to seek out the necessary information and complete the task effectively.

Problem Description

You are an intern at Gammal Tech, and you have been given a crucial task in system design that involves integer division. The system you are working on needs to handle large datasets, where each data element is an integer. Your task is to design a module that efficiently processes these integers based on a specific division operation.

However, you've noticed that the division factor, which is crucial for the operation, is missing from the task description. Your responsibility is to figure out the appropriate



division factor and implement the module. You can ask for more information, but remember, showing initiative and problem-solving skills is key to securing a job offer.

Task

Your task is to write a program that reads a series of integer pairs. For each pair (A, B) , you must determine the division factor D such that when A is divided by D , the result is closest to B . You need to output this division factor for each pair.

Input Format

- The first line contains an integer N , the number of integer pairs.
- The next N lines each contain two space-separated integers, A and B .

Output Format

- For each integer pair (A, B) , output a line with the division factor D that makes A / D closest to B .

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq A, B \leq 10^9$

Sample Input:

```
3
20 3
15 7
100 20
```

Sample Output:

```
7
2
5
```



Explanation

- For the first pair (20, 3), dividing 20 by 3 gives approximately 6.67, which is closest to 7.
- For the second pair (15, 2), dividing 15 by 2 gives 7.5, which is closest to 7.
- For the third pair (100, 5), dividing 100 by 5 gives exactly 20.

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>
using namespace std;

int main() {
    int N, A, B;
    cin >> N;

    for (int i = 0; i < N; i++) {
        cin >> A >> B;
        int D = A / B; // Initial guess for division factor

        // Check if increasing or decreasing D gets a closer result
        if (abs(A - D * B) > abs(A - (D + 1) * B)) {
            D++;
        } else if (D != 0 && abs(A - D * B) > abs(A - (D - 1) * B)) {
            D--;
        }

        cout << D << endl;
    }

    return 0;
}
```