



Problem Solving (CPP13)

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 cutting-edge software development company renowned for its innovative approach and exceptional working environment, is on the lookout for responsible and proactive interns. To assess the interns' ability to take initiative and solve problems effectively, a unique challenge has been designed. This challenge not only tests their coding skills but also their resourcefulness and dedication to finding solutions, reflecting the company's ethos of excellence and innovation.

Problem Statement:

You are an intern at Gammal Tech. You have been given a task that involves integer division, but there's a twist: a crucial piece of information is missing from your assignment. Your goal is to determine this missing information using any available resources and complete the task successfully. Your approach to solving this problem will determine whether you receive a job offer from Gammal Tech.

Task:

You are given an array of integers and a target value. Your task is to find two numbers in



the array that, when divided, give the target value. However, the division method you must use is not specified. You must determine whether to use floor division or standard division to get the exact target value. If more than one pair of numbers can produce the target value, select the pair with the smallest first element.

Input Format:

- The first line contains an integer N , the size of the array.
- The second line contains N space-separated integers, the elements of the array.
- The third line contains an integer, the target value.

Output Format:

- Output two integers: the pair of numbers from the array that meet the criteria. If no such pair exists, print `No solution`.

Constraints:

- $2 \leq N \leq 10^5$
- $1 \leq \text{Array elements} \leq 10^6$
- $1 \leq \text{Target value} \leq 10^6$

Sample Input:

```
5
2 4 8 16 32
2
```

Sample Output:

```
4 8
```

Explanation: Here, the pair `(4, 8)` from the array divided equals the target value `2`. The division method used is standard division.



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 <unordered_map>
#include <vector>
using namespace std;

pair<int, int> findPair(const vector<int>& nums, int target) {
    unordered_map<int, int> hash;
    for (int num : nums) {
        if (num != 0 && target % num == 0) {
            if (hash.find(target / num) != hash.end()) {
                return {min(num, target / num), max(num, target / num)};
            }
        }
        hash[num] = 1;
    }
    return {-1, -1}; // No solution found
}

int main() {
    int n, target;
    cin >> n;
    vector<int> nums(n);
    for (int i = 0; i < n; i++) {
        cin >> nums[i];
    }
    cin >> target;

    pair<int, int> result = findPair(nums, target);
    if (result.first == -1) {
        cout << "No solution" << endl;
    } else {
        cout << result.first << " " << result.second << endl;
    }
    return 0;
}
```