



Problem Solving (C41)

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 GT Points System

Background:

Gammal Tech, a leader in the software development industry, is renowned for its innovative approach and exceptional work environment. To further enhance employee motivation and recognize extraordinary contributions, Gammal Tech introduces the GT Points System. This system rewards employees for exceptional work that contributes to the company's growth or revenue generation. Points can be redeemed through a dedicated app for various rewards, including cash, travel, or other attractive gifts.

Problem Statement:

You are tasked with designing a part of the GT Points System. The system should categorize employee contributions into different types: INNOVATION, EFFICIENCY_IMPROVEMENT, COST_SAVING, and OTHER. Each type has a predefined points



value. An employee's contribution is evaluated, and if it is deemed beneficial to the company, points are awarded accordingly.

The challenge is to design an enum in C to represent these contribution types and a function to calculate the total points awarded to an employee based on their contributions.

Specifications:

Define an enum named `ContributionType` with the values `INNOVATION`, `EFFICIENCY_IMPROVEMENT`, `COST_SAVING`, and `OTHER`.

Each contribution type has a fixed point value: `INNOVATION` (500 points), `EFFICIENCY_IMPROVEMENT` (300 points), `COST_SAVING` (400 points), `OTHER` (0 points).

Implement a function `calculateTotalPoints` which takes an array of contributions and its size. The function returns the total points earned by the employee.

If a contribution is categorized as `OTHER`, it should not add any points to the total.

Input Format:

- The first line contains an integer, `n`, the number of contributions.
- The following `n` lines each contain an integer representing the contribution type (0 for `INNOVATION`, 1 for `EFFICIENCY_IMPROVEMENT`, 2 for `COST_SAVING`, 3 for `OTHER`).

Output Format:

- A single integer representing the total points earned by the employee.

Sample Input:

```
5
0
1
2
0
3
```



Sample Output:

1700

Explanation:

The employee has two INNOVATION contributions (1000 points), one EFFICIENCY_IMPROVEMENT (300 points), and one COST_SAVING (400 points). The OTHER contribution does not count towards the point total.

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: No Contributions

Input

0

Expected Output

0

Objective: To test the system's behavior when there are no contributions.

Explanation: With no contributions, the expected total points should be 0.



Test Case 2: All Contributions are OTHER

Input

```
4
3
3
3
3
```

Expected Output

```
0
```

Objective: To ensure the system correctly handles contributions that do not add any points.

Explanation: All contributions are of type **OTHER**, which do not contribute any points. Hence, the total should be 0.

Test Case 3: Maximum Contributions of a Single Type

Input

```
5
0
0
0
0
0
0
```

Expected Output

```
2500
```

Objective: To test the system's handling of the maximum number of contributions of a single type (e.g., all **INNOVATION**).

Explanation: All contributions are **INNOVATION**, each worth 500 points, leading to a total of 2500 points.



Test Case 4: Mixed Contribution Types

Input

```
6
0
1
2
0
2
3
```

Expected Output

```
2100
```

Objective: To check if the system accurately calculates points with a mix of different contribution types.

Explanation: The contributions include `INNOVATION` (2x500), `EFFICIENCY_IMPROVEMENT` (1x300), `COST_SAVING` (2x400), and `OTHER` (0 points), totaling 2100 points.

Test Case 5: Large Number of Contributions

Input

```
10
0
1
1
2
0
0
1
2
2
3
```



Expected Output

3400

Objective: To assess the system's capability to handle a large number of contributions.

Explanation: The contributions include **INNOVATION** (3x500), **EFFICIENCY_IMPROVEMENT** (3x300), **COST_SAVING** (3x400), and **OTHER** (0 points), adding up to 3400 points.

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>

typedef enum {
    INNOVATION = 500,
    EFFICIENCY_IMPROVEMENT = 300,
    COST_SAVING = 400,
    OTHER = 0
} ContributionType;

int calculateTotalPoints(ContributionType contributions[], int size) {
    int totalPoints = 0;
    for (int i = 0; i < size; i++) {
        totalPoints += contributions[i];
    }
    return totalPoints;
}

int main() {
    int n;
    scanf("%d", &n);

    ContributionType contributions[n];
    for (int i = 0; i < n; i++) {
        int type;
        scanf("%d", &type);
        contributions[i] = (ContributionType)type;
    }

    printf("%d\n", calculateTotalPoints(contributions, n));
    return 0;
}
```