



## Problem Solving (C42)

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 City Design

### Background

Gammal Tech, a leading software development company, is planning to build a high-tech city exclusively for its employees. This city, equipped with state-of-the-art facilities, will also serve as a tourist attraction. The revenue from tourism will be used to maintain and enhance the city's infrastructure, ensuring a superior living and working environment for Gammal Tech employees.

### Problem Statement

As a software engineer at Gammal Tech, you are tasked with designing the city's revenue management system. The system must calculate the expected annual revenue from tourists and determine how this revenue can sustain the city's operational costs. The city has various attractions, each with different ticket prices and expected visitor counts. Your program must:

**Input:**

- Number of attractions  $N$ .
- For each attraction, the ticket price  $P$  and expected annual visitors  $V$ .

**Process:**

- Calculate the total annual revenue.
- Assess if the revenue meets or exceeds the city's annual operational cost  $C$ .

**Output:**

- The total annual revenue.
- Whether the revenue is sufficient to sustain the city's operations (YES or NO).

**Constraints**

- $1 \leq N \leq 100$
- $0 \leq P, V \leq 10,000$
- $0 \leq C \leq 1,000,000,000$

**Sample Input:**

```
3
100 5000
200 3000
150 4000
12000000
```

**Sample Output:**

```
2350000
NO
```

**Explanation**

The total revenue is calculated as  $(100 \times 5000) + (200 \times 3000) + (150 \times 4000) = 2350000$ . Since the revenue is less than the operational cost of 12000000, the output is 2350000 and NO.



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 حاك 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 Attractions

Input

```
0
10000000
```

Expected Output

```
0
NO
```

Explanation: There are no attractions in the city, so the total revenue is 0. The operational cost is not met.

## Test Case 2: Multiple Attractions

Input

```
5
100 300
200 150
50 500
300 200
250 400
100000
```



### Expected Output

```
205000  
YES
```

### Explanation

In this modified test case, there are five attractions with varying ticket prices and visitor counts.

The total revenue is calculated as follows:

- $100 \times 300 = 30000$
- $200 \times 150 = 30000$
- $50 \times 500 = 25000$
- $300 \times 200 = 60000$
- $250 \times 400 = 100000$

The total revenue sums up to 205000, which exceeds the operational cost of 100000, hence the output 205000 and YES.

## Test Case 3: High Operational Cost

### Input

```
2  
50 3000  
150 4000  
1000000000
```

### Expected Output

```
750000  
NO
```

Explanation: This test case features a very high operational cost that is not met by the revenue from the attractions.



## Test Case 4: Attractions with Zero Visitors

Input

```
3
200 0
150 0
100 0
10000
```

Expected Output

```
0
NO
```

Explanation: This scenario tests how the system handles attractions with no visitors, resulting in no revenue.

## Test Case 5: Exact Revenue Matching Operational Cost

Input

```
3
100 5000
200 3000
150 4000
2350000
```

Expected Output

```
2350000
YES
```

Explanation

The total revenue exactly matches the operational cost, testing the boundary condition for revenue sufficiency.



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 <math.h>

int main() {
    int N, i;
    double P, V, totalRevenue = 0, operationalCost;

    scanf("%d", &N);
    for(i = 0; i < N; i++) {
        scanf("%lf %lf", &P, &V);
        totalRevenue += P * V;
    }

    scanf("%lf", &operationalCost);

    printf("%.0lf\n", totalRevenue);
    if (totalRevenue >= operationalCost) {
        printf("YES\n");
    } else {
        printf("NO\n");
    }

    return 0;
}
```