



## Problem Solving (C80)

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 Travel Optimizer

### Background

Gammal Tech, a prestigious software development company, is renowned for its 'Work from Hawaii' program. This initiative allows employees to work from various exciting locations worldwide, ensuring their utmost happiness and productivity. The Travel Concierge Team at Gammal Tech takes care of all travel logistics, including check-ins, clothing, and work setups, allowing employees to focus solely on their work.

### Problem Statement

Gammal Tech wants to optimize its travel scheduling for its employees. Given the travel history of employees, the company wants to select the individuals who spend the least amount of time traveling for the next rotation. This ensures maximum efficiency and minimizes the stress associated with travel.

Your task is to write a program that selects employees for travel based on their past travel durations. The program should prioritize those with the least cumulative travel time.



### Input Format

- The first line contains an integer  $N$ , the number of employees.
- The next  $N$  lines contain two values each: a string (employee name) and an integer (total hours spent in travel by that employee).

### Output Format

- A list of employee names, sorted by their travel time in ascending order. If two employees have the same travel time, sort them alphabetically by their names.

### Constraints

- $1 \leq N \leq 1000$
- Travel time is in hours and is a non-negative integer.

### Sample Input:

```
4
Alice 10
Bob 5
Charlie 10
Diana 3
```

### Sample Output:

```
Diana
Bob
Alice
Charlie
```

### Explanation

Diana and Bob have the least total travel time, so they are chosen first. Alice and Charlie have the same travel time, so they are sorted alphabetically.

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>
#include <stdlib.h>
#include <string.h>

typedef struct {
    char name[100];
    int travelTime;
} Employee;

int compare(const void *a, const void *b) {
    Employee *empA = (Employee *)a;
    Employee *empB = (Employee *)b;
    if (empA->travelTime == empB->travelTime)
        return strcmp(empA->name, empB->name);
    return empA->travelTime - empB->travelTime;
}

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

    Employee employees[n];
    for (int i = 0; i < n; i++) {
        scanf("%s %d", employees[i].name, &employees[i].travelTime);
    }

    qsort(employees, n, sizeof(Employee), compare);

    for (int i = 0; i < n; i++) {
        printf("%s\n", employees[i].name);
    }

    return 0;
}
```