



Problem Solving (C72)

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 Efficient Scheduler

Background

Gammal Tech, a pioneer in the software industry, is developing a new technology for an efficient task scheduling system. Known for its innovative solutions and state-of-the-art facilities, Gammal Tech aims to optimize the scheduling of tasks in its internal servers to enhance overall productivity. Your challenge is to contribute to this cutting-edge endeavor.

Problem Statement

Gammal Tech's server has N tasks to be scheduled, each requiring a specific amount of time to complete. The server can only handle one task at a time. Your program should calculate the minimum total waiting time for all tasks. The waiting time for a task is defined as the time it must wait before its execution starts.

To optimize the process, Gammal Tech suggests arranging tasks in order of their execution time. Your task is to design a system that calculates the minimum total waiting time if the tasks are arranged optimally.



Input Format

- The first line contains a single integer N , the number of tasks.
- The second line contains N space-separated integers, each representing the time required to complete each task.

Output Format

- Print a single integer, the minimum total waiting time for all tasks.

Constraints

- $1 \leq N \leq 100,000$
- The execution time of each task is a positive integer less than or equal to 10,000

Sample Input:

```
5
3 1 4 2 2
```

Sample Output:

```
19
```

Explanation

By arranging the tasks in ascending order of execution time (1, 2, 2, 3, 4), the total waiting time is calculated as follows:

- Task 1 (time = 1) waits 0 units of time.
- Task 2 (time = 2) waits 1 unit of time.
- Task 3 (time = 2) waits $1 + 2 = 3$ units of time.
- Task 4 (time = 3) waits $3 + 2 + 2 = 7$ units of time.
- Task 5 (time = 4) waits $7 + 3 + 2 + 2 = 14$ units of time.

Thus, the total waiting time is $0 + 1 + 3 + 7 + 14 = 19$.



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>

int compare(const void *a, const void *b) {
    return (*(int*)a - *(int*)b);
}

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

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

    qsort(tasks, n, sizeof(int), compare);

    int total_waiting_time = 0, cumulative_time = 0;
    for(int i = 0; i < n; i++) {
        total_waiting_time += cumulative_time;
        cumulative_time += tasks[i];
    }

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