Problem Solving (C44)

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 فعّال. 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 Macro Optimizer

## Background

Gammal Tech, a trailblazer in the software development industry, renowned for its exceptional work environment and innovative team, is on the brink of revolutionizing system design. Their latest project focuses on optimizing system performance using a unique macro-based approach. This approach allows for unparalleled efficiency in processing and executing tasks.

## Problem Statement

As a software engineer at Gammal Tech, you are tasked with developing a part of the Macro Optimizer. This Optimizer uses a series of macros to enhance system performance. Your task is to write a program that can read a set of instructions, each represented by a macro, and calculate the total time taken to execute these instructions.

The system operates on a simple principle: each macro represents a specific instruction that takes a predefined amount of time to execute. You are given a list of macros and

their corresponding execution times. Your program must read a sequence of these macros and compute the total execution time.

### Input Format

- The first line contains two integers `N` and `M` (`1 ≤ N, M ≤ 1000`), where `N` is the number of different macros, and `M` is the number of macros in the sequence you need to process.
- The following `N` lines each contain a macro identifier (a string of up to 20 characters) and its execution time in milliseconds (an integer).
- The next `M` lines each contain a macro identifier from the list provided earlier.

### Output Format

Your program should output a single integer, the total execution time of the sequence of macros.

### Sample Input:

```
3 5
LOAD 10
SAVE 20
PROCESS 30
LOAD
PROCESS
SAVE
LOAD
PROCESS
```

### Sample Output:

```
100
```

Explanation

The sequence of macros (LOAD, PROCESS, SAVE, LOAD, PROCESS) takes a total of 10 + 30 + 20 + 10 + 30 = 100 milliseconds to execute.

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: Minimum Input

Input

```
1 1
TASK 5
TASK
```

Expected Output

```
5
```

Purpose: To test the program with the smallest possible input.

Explanation: The sequence contains only one macro with its execution time, testing the program's ability to handle minimal input.

## Test Case 2: Alternating Macros

Input

```
4 8
LOAD 10
SAVE 15
COMPUTE 20
RENDER 25
LOAD
SAVE
COMPUTE
RENDER
LOAD
SAVE
COMPUTE
RENDER
```

Expected Output

```
140
```

Purpose: To test the program's handling of a sequence where macros alternate in a predictable pattern.

Explanation: This test case checks the program's ability to handle a sequence where macros appear in an alternating order. It's simpler than the maximum input scenario but still tests the program's ability to correctly calculate the total time for a moderately long sequence with different macros.

## Test Case 3: Repeated Macros in Sequence

Input

```
3 6
LOAD 10
SAVE 20
PROCESS 30
LOAD
PROCESS
PROCESS
SAVE
LOAD
LOAD
```

Expected Output

```
100
```

Purpose: To check how the program handles repeated macros in the sequence.

Explanation: Tests the program's ability to correctly calculate total time with repeated macros.

## Test Case 4: Non-Existent Macro in Sequence

Input

```
2 3
INIT 15
END 25
INIT
START
END
```

Expected Output

```
40
```

Purpose: To test the program's behavior with a macro in the sequence that is not defined in the macro list.

Explanation: This scenario checks if the program can handle a sequence that includes a macro not listed in the initial macro definitions. Ideally, the program should ignore or skip unrecognized macros.

## Test Case 5: Long Macro Names

Input

```
2 2
LONGMACRONAMEHEREA 10
ANOTHERLONGMACRONAM 20
LONGMACRONAMEHEREA
ANOTHERLONGMACRONAM
```

Expected Output

```
30
```

Purpose: To test the program's handling of macro names that are close to the maximum length.

Explanation: This test checks if the program can correctly handle macro names that are at the upper limit of the allowed character length.

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:

```c
#include <stdio.h>
#include <string.h>

#define MAX_MACROS 1000

int main() {
    int N, M;
    char macros[MAX_MACROS][21];
    int times[MAX_MACROS];
    char sequence[21];
    int totalTime = 0;

    scanf("%d %d", &N, &M);
    for (int i = 0; i < N; i++) {
        scanf("%s %d", macros[i], &times[i]);
    }

    for (int i = 0; i < M; i++) {
        scanf("%s", sequence);
        for (int j = 0; j < N; j++) {
            if (strcmp(sequence, macros[j]) == 0) {
                totalTime += times[j];
                break;
            }
        }
    }

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