



Problem Solving (C78)

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'.

Optimal Office Rotation at Gammal Tech

Background

Gammal Tech, a pioneer in software development and an exemplar of innovation, has launched a unique program called 'Work from Hawaii'. This program allows employees to work from various exciting global offices, fostering happiness and productivity. The company has offices in the most vibrant locations worldwide and rotates employees between offices based on their preferences. Some employees thrive by changing offices weekly, others monthly, and some prefer not to travel. The challenge is to design a system that optimally schedules these rotations to maximize employee satisfaction and performance.

Problem Statement

Your task is to develop a system that determines the optimal office rotation schedule for Gammal Tech employees. The system should consider individual employee preferences for rotation frequency and allocate them to different offices accordingly.



The goal is to maximize the overall happiness score, which is calculated based on how closely the rotation matches each employee's preference.

Input Format

- The first line contains two integers N and M , where N is the number of employees and M is the number of different office locations.
- The next N lines each contain an employee's preference: 'W' for weekly rotation, 'M' for monthly rotation, and 'S' for staying in the same office.
- Each employee's preference line is followed by a line with M integers, representing the happiness score (between 1 and 100) for each office location.

Output Format

- For each employee, output the office locations (as integers from 1 to M) they will be working at for the next N periods (weeks or months, based on their preference).
- If there are multiple optimal solutions, any one of them will be considered correct.

Constraints

- $1 \leq N, M \leq 100$

Sample Input:

```
3 4
W
80 90 75 85
M
70 85 90 65
S
60 75 80 70
```

Sample Output:

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



Explanation

- Employee 1 prefers weekly rotations and gets their highest happiness scores in the order of offices 1, 2, 3, and back to 1.
- Employee 2 prefers a monthly rotation and gets the highest score at office 3.
- Employee 3 prefers to stay at one office and is most content at office 4.

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>

#define MAX_EMPLOYEES 100
#define MAX_OFFICES 100

int main() {
    int n, m, i, j;
    char preference;
    int happiness[MAX_EMPLOYEES][MAX_OFFICES];

    // Reading the number of employees and offices
    scanf("%d %d", &n, &m);

    for (i = 0; i < n; i++) {
        int maxIdx = 0;
        scanf(" %c", &preference);

        for (j = 0; j < m; j++) {
            scanf("%d", &happiness[i][j]);
            if (happiness[i][maxIdx] < happiness[i][j]) {
                maxIdx = j;
            }
        }

        if (preference == 'W') {
            for (j = 0; j < m; j++) {
                printf("%d ", (maxIdx + j) % m + 1);
            }
            printf("\n");
        } else if (preference == 'M') {
            printf("%d\n", maxIdx + 1);
        } else {
            printf("%d\n", maxIdx + 1);
        }
    }

    return 0;
}
```