



Problem Solving (CPP16)

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 Communication Grid

Background:

Gammal Tech, a top-tier software development company, prides itself on its multicultural, innovative environment, where language is no barrier to success. Their latest project involves developing a revolutionary communication grid that seamlessly integrates different languages, fostering unparalleled collaboration among global teams. Your task is to contribute to this cutting-edge initiative.

Problem Statement:

You are given a 2D grid representing Gammal Tech's new office layout, where each cell can either be a workstation (\mathbb{W}), a translation hub (\mathbb{T}), or a walkway (\cdot). A translation hub is a state-of-the-art device that instantly translates any language spoken near it. The goal is to determine the minimum number of translation hubs needed so that every workstation is adjacent (horizontally or vertically) to at least one hub, ensuring seamless communication across all teams.



Input Format:

- The first line contains two integers, n and m ($1 \leq n, m \leq 1000$), representing the number of rows and columns in the grid.
- n lines follow, each with m characters (W , T , or $.$), representing the office layout.

Output Format:

Print a single integer, the minimum number of translation hubs (T) needed.

Sample Input:

```
4 5
W.W..
.WW..
..W.W
....W
```

Sample Output:

```
3
```

Explanation: You can place three translation hubs at positions (1,2), (2,3), and (4,4) to ensure all workstations are adjacent to at least one hub.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمرين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق



C++ Programming Solution:

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n, m;
    cin >> n >> m;
    vector<string> grid(n);
    for (int i = 0; i < n; i++) {
        cin >> grid[i];
    }

    int hubs = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (grid[i][j] == 'W') {
                bool needsHub = true;
                // Check adjacent cells for a hub
                if (i > 0 && grid[i-1][j] == 'T') needsHub = false;
                if (j > 0 && grid[i][j-1] == 'T') needsHub = false;
                if (i < n-1 && grid[i+1][j] == 'T') needsHub = false;
                if (j < m-1 && grid[i][j+1] == 'T') needsHub = false;

                // Place a hub if needed
                if (needsHub) {
                    hubs++;
                    grid[i][j] = 'T';
                }
            }
        }
    }

    cout << hubs << endl;
    return 0;
}
```