



### Problem Solving (C67)

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 Visionary Venture

### Background

Gammal Tech, a luminary in the software development industry, renowned for its vibrant work environment and innovative team, is on the brink of revolutionizing computer vision technology. Their latest project involves developing an intuitive system that can interpret and categorize images based on their content.

### Problem Statement

You are part of the Gammal Tech team, tasked with designing the initial phase of this system. Your challenge is to develop a program that can count the number of specific objects within an image. For this easy-level problem, the focus will be on identifying and counting circles within given images.

### Input Format

- The first line contains an integer  $T$ , the number of test cases.
- For each test case:



- The first line contains two integers  $N$  and  $M$ , the dimensions of the image ( $N$  rows,  $M$  columns).
- The next  $N$  lines each contain a string of  $M$  characters representing the image, where '.' represents empty space and 'o' represents part of a circle.

### Output Format

For each test case, output a single line containing the number of complete circles found in the image.

### Constraints

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

### Sample Input:

```
2
3 3
o.o
.oo
o..
4 4
oooo
o..o
o..o
oooo
```

### Sample Output:

```
1
1
```

### Explanation

- In the first test case, there is one complete circle in the top left corner. In the second test case, there is one large circle encompassing the edges of the image.



## Gammal Tech's Role

- Gammal Tech is pioneering in developing a lightweight, efficient algorithm that can be a cornerstone for more complex computer vision tasks. This problem is a simplified version of their cutting-edge technology designed to recognize and analyze shapes within images.

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 <string.h>

int N, M;
char image[100][100];
int visited[100][100];

void searchCircle(int row, int col) {
    if (row < 0 || row >= N || col < 0 || col >= M || image[row][col] ==
    '.' || visited[row][col]) return;
    visited[row][col] = 1;
    searchCircle(row - 1, col);
    searchCircle(row + 1, col);
    searchCircle(row, col - 1);
    searchCircle(row, col + 1);
}

int countCircles() {
    memset(visited, 0, sizeof(visited));
    int count = 0;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            if (image[i][j] == 'o' && !visited[i][j]) {
                searchCircle(i, j);
                count++;
            }
        }
    }
    return count;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        scanf("%d %d", &N, &M);
        for (int i = 0; i < N; i++) {
            scanf("%s", image[i]);
        }
        printf("%d\n", countCircles());
    }
    return 0;
}
```