

题目描述

一矩形阵列由数字 0 到 9 组成，数字 1 到 9 代表细胞，细胞的定义为沿细胞数字上下左右若还是细胞数字则为同一细胞，求给定矩形阵列的细胞个数。

输入格式

第一行两个整数代表矩阵大小 n 和 m 。

接下来 n 行，每行一个长度为 m 的只含字符 0 到 9 的字符串，代表这个 $n \times m$ 的矩阵。

输出格式

一行一个整数代表细胞个数。

输入输出样例

输入 #1

复制

```
4 10
0234500067
1034560500
2045600671
0000000089
```

输出 #1

复制

```
4
```

说明/提示

数据规模与约定

对于 100% 的数据，保证 $1 \leq n, m \leq 100$ 。

```
#include <iostream>
using namespace std;
int n, m, ans;
int a[105][105];
// 方向数组: 上、左、下、右
int dx[] = {-1, 0, 1, 0};
int dy[] = {0, -1, 0, 1};
// 深度优先搜索, 标记连通区域
```

```
void dfs(int x, int y) {  
    // 越界判断  
    if (x < 0 || x >= n || y < 0 || y >= m)  
        return;  
    // 非目标区域或已访问  
    if (a[x][y] == 0)  
        return;  
    a[x][y] = 0; // 标记为已访问  
    // 遍历四个方向  
    for (int i = 0; i < 4; i++) {  
        dfs(x + dx[i], y + dy[i]);  
    }  
}
```

```
int main() {  
    cin >> n >> m;  
  
    // 读入地图  
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < m; j++) {  
            char c;  
            cin >> c;  
            a[i][j] = c - '0'; // 转换为数字  
        }  
    }  
}
```

// 统计连通分量数量

```
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < m; j++) {  
        if (a[i][j] == 1) {  
            ans++;  
            dfs(i, j);  
        }  
    }  
}
```

```
cout << ans << endl;  
return 0;
```

```
}
```