

1359: 围成面积

题目描述

编程计算由“*”号围成的下列图形的面积。面积计算方法是统计*号所围成的闭合曲线中水平线和垂直线交点的数目。如下图所示，在 10×10 的二维数组中，有“*”围住了15个点，因此面积为15。

```

0 0 0 0 0 0 0 0 0 0
0 0 0 0 * * * 0 0 0
0 0 0 0 * 0 0 * 0 0
0 0 0 0 0 * 0 0 * 0
0 0 * 0 0 0 * 0 * 0
0 * 0 * 0 * 0 0 * 0
0 * 0 0 * * 0 * * 0
0 0 * 0 0 0 0 * 0 0
0 0 0 * * * * 0 0
0 0 0 0 0 0 0 0 0 0

```

输入格式

10×10 的图形。

输出格式

输出面积。

输入样例

```

0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0
0 0 0 0 1 0 0 1 0 0
0 0 0 0 0 1 0 0 1 0
0 0 1 0 0 0 1 0 1 0
0 1 0 1 0 1 0 0 1 0
0 1 0 0 1 1 0 1 1 0
0 0 1 0 0 0 0 1 0 0
0 0 0 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0

```

输出样例

解析

本题的描述不够严谨，需要强调一下，只能存在一个闭合环。参考P1162。

编码

```
#include <bits/stdc++.h>

using namespace std;
//原始数组
int maps[32][32];
//标记数组
int flags[32][32];

//题目不够严谨，必须要求只有一个闭合环

//第一个表示不动，是充数的，后面的四个分别是上下左右四个方向
int dx[4] = {-1, 1, 0, 0};
int dy[4] = {0, 0, -1, 1};
int n = 10, i, j;

void dfs(int p, int q) {
    int i;
    //判断坐标点的有效性
    if (p < 0 || p > n + 1 || q < 0 || q > n + 1 || flags[p][q] != 0)
        return;
    }
    flags[p][q] = 1; //染色
    //向四个方向搜索
    for (i = 0; i < 4; i++) {
        dfs(p + dx[i], q + dy[i]);
    }
}

int main() {
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= n; j++) {
            cin >> maps[i][j];
            //开始执行染色，0标记为0
            if (maps[i][j] == 0) {
                flags[i][j] = 0;
            }
            //1则被标记为2
            else {
                flags[i][j] = 2;
            }
        }
    }
    //搜索 从0, 0开始搜
```

```
dfs(0, 0);
int num = 0;
//打印最终结果
for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j++) {
        //经过染色后，如果该坐标的值依然是0，说明它就被围在中间的点。
        if (flags[i][j] == 0) {
            num++;
        }
    }
}
cout << num;
}
```

逻辑航线培优教育，信息学奥赛培训专家。



