

1191: 流感传染

题目描述

有一批易感人群住在网格状的宿舍区内，宿舍区为 $n*n$ 的矩阵，每个格点为一个房间，房间里可能住人，也可能空着。在第一天，有些房间里的人得了流感，以后每天，得流感的人会使其邻居传染上流感，（已经得病的不变），空房间不会传染。请输出第 m 天得流感的人数。

输入

第一行一个数字 n ， n 不超过100，表示有 $n*n$ 的宿舍房间。

接下来的 n 行，每行 n 个字符，'.'表示第一天该房间住着健康的人，'#'表示该房间空着，'@'表示第一天该房间住着得流感的人。

接下来的一行是一个整数 m ， m 不超过100。

输出

输出第 m 天，得流感的人数。

输入样例

```
5
....#
.#.@.
.#@..
#....
.....
4
```

输出样例

```
16
```

解析

本题使用正向推导会容易一些，即计算每一天每个患者的传染路径，最后，再输出全部的患者数量即可。

注意，本题在计算传染时，不能直接将健康者修改为患者，如果这样操作，则会让我们将一些原本不是传染源的数据误认为是传染源，例如下图。

	1	2	3
1			
2		@	
3			

[2,2]的患者会将[1,2],[2,1][3,2][2,3]进行传染,但是这些数据在当天并不会传染别人,因此不能直接将他们进行标记。否则的话,我们很可能把[3,2]以及[2,3]当做新的起始点。

因此,我们需要把那些即将患病的患者用一个新的数组进行存储,以便于再次计算。

编码

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

using namespace std;
const int maxN = 101;

int n; // 矩阵边界
int m; // 经过的天数

int countNum; // 实际感染人数
char today[maxN][maxN]; // 今日感冒信息
char tomorrow[maxN][maxN]; // 次日感冒信息

// 统计总的患病人数和标记被感染的患者
void Record(int i, int j) {
    // 必须是健康人,才能被传染
    if (tomorrow[i][j] == '.') {
        // 把健康人标记成患者
        tomorrow[i][j] = '@';
        // 统计患病人数
        countNum++;
    }
}

void Infect(int i, int j) {
    // 超出边界
    if (i < 0 || j < 0 || i >= n || j >= n) {
        return;
    }
    // 将健康人传染
    if (today[i][j] == '.') {
        // 传染
        Record(i, j);
    }
}

// 向四个方向传染
void InfectCheck(int i, int j) {
    if (today[i][j] == '@') {
        Infect(i, j - 1);
        Infect(i, j + 1);
        Infect(i - 1, j);
        Infect(i + 1, j);
    }
}
```

```

int main(int argc, char **argv) {
    scanf("%d", &n);
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            cin >> today[i][j];
            tomorrow[i][j] = today[i][j];
            //记录最初患病两个人
            if (today[i][j] == '@') {
                countNum++;
            }
        }
    }

    scanf("%d", &m);

    //首日不发生传染，因此从第二天开始模拟
    for (int k = 2; k <= m; k++) {
        //开始进行传染模拟
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < n; ++j) {
                InfectCheck(i, j);
            }
        }
        //将已经被标记患病的数据还原至code中
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < n; ++j) {
                if (tomorrow[i][j] == '@')
                    today[i][j] = tomorrow[i][j];
            }
        }
    }
    printf("%d", countNum);
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

