

需要打开多少监控器题目描述：

某长方形停车场，每个车位上方都有对应监控器，当且仅当在当前车位或者前后左右四个方向任意一个车位范围停车时，监控器才需要打开；

给出某一时刻停车场的停车分布，请统计最少需要打开多少个监控器；

输入描述：

第一行输入 m, n 表示长宽，满足 $1 < m, n \leq 20$ ；后面输入 m 行，每行有 n 个 0 或 1 的整数，整数间使用一个空格隔开，表示该行已停车情况，其中 0 表示空位， 1 表示已停；

输出描述：

最少需要打开监控器的数量；

补充说明：

示例 1

输入：

```
3 3
0 0 0
0 1 0
0 0 0
```

输出：

```
5
```

说明：

```
#include <iostream>
#include <vector>
using namespace std;
bool checkOpen(int x, int y, vector<vector<int>> & flags)//是否需开启灯
{
    if((y < 0) || (y >= flags.size())) return false;
    if((x < 0) || (x >= flags[0].size())) return false;
    if(flags[y][x]) return false;
    flags[y][x] = 1;
    return true;
}
int main() {
    int h = 0, w = 0, a = 0;
    cin >> h >> w;
    vector<vector<int>> datas;
    vector<int> rowDatas;
    vector<pair<int,int>> ones;
```

```

//获取输入数据
int iCount = 0;//监控数
for(int i =0; i < h; i++)
{
    for(int j=0; j < w;j++)
    {
        cin >> a;
        rowDatas.push_back(a);
        if(a)
        {
            ones.emplace_back(j,i);//x,y
            iCount++;
        }
    }
    datas.push_back(rowDatas);
    rowDatas.clear();
}
int x = 0, y = 0;
for(auto & xy : ones) //左右上下开灯
{
    if(checkOpen(xy.first-1, xy.second, datas)) iCount++;
    if(checkOpen(xy.first+1, xy.second, datas)) iCount++;
    if(checkOpen(xy.first, xy.second-1, datas)) iCount++;
    if(checkOpen(xy.first, xy.second+1, datas)) iCount++;
}
//
cout << iCount;
}
// 64 位输出请用 printf("%lld")

```