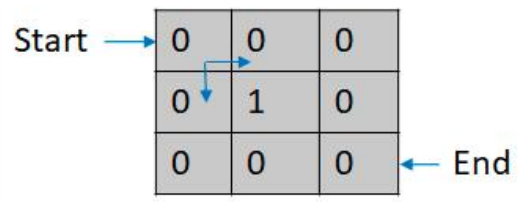


题目描述：

园区某部门举办了 Family Day，邀请员工及其家属参加；将公司园区视为一个矩形，起始园区设置在左上角，终点园区设置在右下角；家属参观园区时，只能向右和向下园区前进；求从起始园区到终点园区会有多少条不同的参观路径；



输入描述：第一行为园区长和宽；后面每一行表示该园区是否可以参观，0表示可以参观，1表示不能参观

输出描述：输出为不同的路径数量

补充说明：

1 <= 园区长 <= 100

1 <= 园区宽 <= 100

示例 1

输入：

3 3

0 0 0

0 1 0

0 0 0

输出：

2

说明：

```
#include <iostream>
#include <vector>
using namespace std;
int countPaths(int m,int n,vector<vector<int>>& grid){
    if(grid[0][0] == 1 || grid[m-1][n-1] == 1){
        return 0;
    }
    vector<vector<int>> dp(m,vector<int>(n,0));
    dp[0][0] = (grid[0][0] == 0) ? 1 : 0;
    for(int i = 1;i<m;i++){
        dp[i][0] = (grid[i][0] == 0 && dp[i-1][0] == 1) ? 1 : 0;
    }
    for(int j = 1 ;j<n;j++){
```

```

        dp[0][j] = (grid[0][j] == 0 && dp[0][j-1] == 1) ? 1 : 0;
    }
    for(int i=1;i<m;i++){
        for(int j=1;j<n;j++){
            if(grid[i][j] == 0){
                dp[i][j] = dp[i-1][j] + dp[i][j-1];
            }
            else {
                dp[i][j] = 0;
            }
        }
    }
    return dp[m-1][n-1];
}

int main() {
    int m,n;
    cin>>m>>n;
    vector<vector<int>> grid(m,vector<int>(n,0));
    for(int i =0 ;i<m;i++){
        for(int j=0;j<n;j++){
            cin>>grid[i][j];
        }
    }
    int paths = countPaths(m,n,grid);
    cout<<paths;
    return 0;
}

```