

# 二维数组练习1

---

## 人员

---

韩承睿、牟茗、刘嘉航、辛帅辰、高健桓、崔吉诺、刘祺、方俊喆、秦显森、黄家瑞 到课

李翰如、崔宸赫 线上

徐浩然、刘峰烁 未到

## 作业

---

必做: noi 1.8 14: 扫雷游戏地雷数计算

选做: noi 1.8 02: 同行列对角线的格子

## 课堂表现

---

牟茗同学上课思路比较快，写代码也很快，提出表扬！！

有些同学在老师讲题时不认真听，老师写代码时老师写一行，比着抄一行，以后需要注意。要自己先听懂思路，然后自己去实现代码。

## 课堂内容

---

### noi 1.8 03 计算矩阵边缘元素之和

可以定义一个sum变量用来计算边缘元素的和

如何判断一个元素是否在数组边缘？

-> 设这个元素为  $a[i][j]$ ，即第  $i$  行，第  $j$  列

当元素在 第一行 或 最后一行 或 第一列 或 最后一列 时，说明在变样

即  $i==1 \ || \ i==m \ || \ j==1 \ || \ j==n$  时

```

#include<bits/stdc++.h>
using namespace std;
int w[105][105];
int main(){
    int m, n; cin >> m >> n;
    for (int i = 1; i <= m; ++i) {
        for (int j = 1; j <= n; ++j) {
            cin >> w[i][j];
        }
    }

    int sum = 0;
    for (int i = 1; i <= m; ++i) {
        for (int j = 1; j <= n; ++j) {
            if (i==1 || i==m || j==1 || j==n) {
                sum += w[i][j];
            }
        }
    }
    cout << sum << endl;
    return 0;
}

```

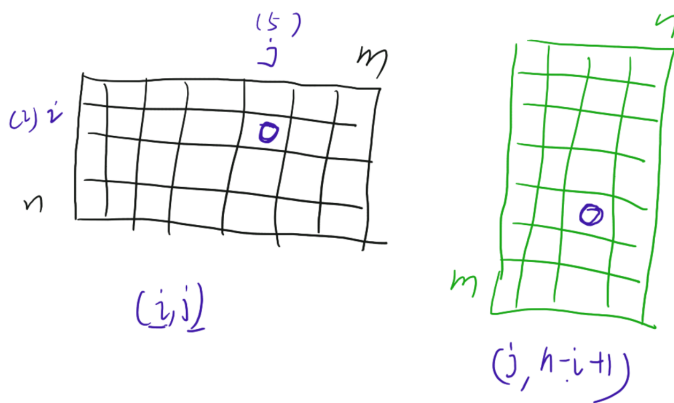
noi 1.8 08 矩阵加法

```

#include<bits/stdc++.h>
using namespace std;
int main(){
    int i[105][105],s,h,l;
    cin>>h>>l;
    for(int a=0;a<h;a++)
    {
        for(int b=0;b<l;b++)
        {
            cin>>i[a][b];
        }
    }
    for(int a=0;a<h;a++)
    {
        for(int b=0;b<l;b++)
        {
            cin>>s;
            i[a][b]+=s;
        }
    }
    for(int a=0;a<h;a++)
    {
        for(int b=0;b<l;b++)
        {
            cout<<i[a][b]<<" ";
        }
        cout<<endl;
    }
    return 0;
}

```

## noi 1.8 11 图像旋转



画出上面这个图，找到 旋转前 和 旋转后 的对应关系，这个题就没问题了

设旋转前的数组为数组a，旋转后为数组b

那么应该是  $a[i][j]$  对应于  $b[j][n-i+1]$

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

using namespace std;

int a[105][105], b[105][105];

int main()
{
    int n,m;
    cin >> n >> m;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            cin >> a[i][j];
        }
    }

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            b[j][n-i+1] = a[i][j];
        }
    }

    for (int i = 1; i <= m; i++) {
        for (int j = 1; j <= n; j++) {
            cout << b[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

noi 1.8 14 扫雷游戏地雷数计算

以下是点(i,j)的8个相邻位置

i-1 j-1

i-1 j

i-1 j+1

i j-1

i j+1

i+1 j-1

i+1 j

i+1 j+1

```
int dx[8] = {-1, -1, -1, 0, 0, 1, 1, 1};
```

```
int dy[8] = {-1, 0, 1, -1, 1, -1, 0, 1};
```

n行m列

```
for (int k = 0; k < 8; k++) {  
    int ni = i+dx[k], nj = j+dy[k];  
    如果 a[ni][nj] 这个点在范围内, 并且是*  
    合法 -> 行>=1 && 行<=n && 列>=1 && 列<=m  
}
```

点 (i,j)

```
for (int k = 0; k < 8; k++) {  
    int ni = i+dx[k], nj = j+dy[k];  
    //          i+dx[k],j+dy[k]  
    // k==0 -> i-1,    j-1  
    // k==1 -> i-1,    j  
    // k==2 -> i-1,    j+1  
    // k==3 -> i,      j-1  
    // k==4 -> i,      j+1  
    // k==5 -> i+1     j-1  
    // k==6 -> i+1     j  
    // k==7 -> i+1     j+1  
}
```

// 字符串的输入输出

```
char s[100];
```

```
cin>>s; // 从s[0]开始输入
```

```
cin>>(s+1); // 从s[1]开始输入
```

```
cout << s; // 从s[0]开始输出
```

```
cout << (s+1); // 从s[1]开始输出
```

```
// 输入二维字符数组
char s[105][105];
for (int i = 1; i <= n; i++) {
    cin >> (s[i]+1);
}
```

此题课上分析的很透彻了，不提供具体代码，要求同学们课下思考实现该题目