

# 二维数组入门

---

## 人员

---

温郝冬、罗启宸、吴青汉、王奕皓、周苇杰、陶汇笙、张昱霖、马瑞昕、董浩桢、康佳、李沛都、郭栩睿 到课

崔宸赫 请假

## 作业检查

---

温郝冬 已完成

李沛都 未打卡

罗启宸 已完成

吴青汉 已完成

王奕皓 已完成

周苇杰 已完成

陶汇笙 未打卡

张昱霖 已完成

康佳 未打卡

郭栩睿 未打卡

崔宸赫 未打卡

马瑞昕 未打卡

董浩桢 已完成

## 作业

---

必做: noi 1.8 03: 计算矩阵边缘元素之和

选做: noi 1.8 02: 同行列对角线的格子

下周默写: noi 1.8 02: 同行列对角线的格子

## 课堂表现

---

有几位同学课上说话比较严重，希望下节课能够自觉改善

## 课堂内容

---

### 1326 - 需要安排几位师傅加工零件

1. 特判是否要输出NO
2. 排序
3. 从左往右，一直加
4. 直到总量超过要求的数时，输出*i*，然后break

```

#include <iostream>
using namespace std;

int a[105];

int main() {
    int n, m;
    cin >> n >> m;
    int sum = 0;
    for (int i = 1; i <= m; i++) {
        cin >> a[i];
        sum += a[i];
    }
    if (sum < n) {
        cout << "NO" << endl;
    }
    else {
        for (int i = 1; i <= m; i++) {
            for (int j = i+1; j <= m; j++) {
                if (a[i] < a[j]) {
                    int t = a[i];
                    a[i] = a[j];
                    a[j] = t;
                }
            }
        }

        int cnt = 0;
        for (int i = 1; i <= m; i++) {
            cnt += a[i];
            if (cnt >= n) {
                cout << i << endl;
                break;
            }
        }
    }
    return 0;
}

```

## 1497 - 分数线的划定

1. 从大到小排序
2. 取第 x 个人的成绩即可

```

#include <bits/stdc++.h>

using namespace std;

int a[1005];

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> a[i];
    }
    int x; cin >> x;
    for (int i = 1; i <= n; ++i) {
        for (int j = i+1; j <= n; ++j) {
            if (a[i] < a[j]) {
                int t = a[i];
                a[i] = a[j];
                a[j] = t;
            }
        }
    }
    cout << a[x] << endl;
    return 0;
}

```

## 二维数组

```
// 二维数组输入输出

#include <iostream>
using namespace std;

int a[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++) {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

## noi 1.8 01:矩阵交换行

```
a[m][1]    ---    a[n][1]
a[m][2]    ---    a[n][2]
...
a[m][5]    ---    a[n][5]

for (int i = 1; i <= 5; i++) {
    // 交换 a[m][i] 和 a[n][i]
    int t = a[m][i];
    a[m][i] = a[n][i];
    a[n][i] = t;
}
```

```

#include <iostream>
using namespace std;

int a[10][10];

int main() {
    for (int i=1;i<=5;i++) {
        for (int j=1;j<=5;j++) {
            cin >> a[i][j];
        }
    }
    int m, n;
    cin >> m >> n;

    for (int i = 1; i <= 5; i++) {
        int t = a[m][i];
        a[m][i] = a[n][i];
        a[n][i] = t;
    }

    for (int i = 1; i <= 5; i++) {
        for (int j = 1; j <= 5; j++) {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}

```

### noi 1.8 03:计算矩阵边缘元素之和

1. 输入二维数组
2. 遍历二维数组，判断一个数在不在边缘。如果在边缘，就进行累加
3. 如何判断一个数在不在边缘？ 行==1 || 行==n || 列==1 || 列==m

此题目不提供具体代码，要求同学们课下独立思考实现