

# 12 - 数组的综合应用

## C++ 程序设计基础

SOJ 信息学竞赛教练组

2024 年 6 月 5 日

## 1 复习回顾

## 2 一维数组的应用

## 3 二维数组的应用

## 4 总结

# 一维数组

- 一维数组的声明

# 一维数组

- 一维数组的声明
  - 元素类型 数组名 [数组大小];

# 一维数组

- 一维数组的声明
  - 元素类型 数组名 [数组大小];
  - `int a[105];` // 定义了可以存放 105 个整数的数组 a

# 一维数组

- 一维数组的声明
  - 元素类型 数组名 [数组大小];
  - `int a[105];` // 定义了可以存放 105 个整数的数组 a
- 一维数组的元素访问

# 一维数组

- 一维数组的声明
  - 元素类型 数组名 [数组大小];
  - `int a[105];` // 定义了可以存放 105 个整数的数组 a
- 一维数组的元素访问
  - 数组名称 [下标]

# 一维数组

- 一维数组的声明
  - 元素类型 数组名 [数组大小];
  - `int a[105];` // 定义了可以存放 105 个整数的数组 a
- 一维数组的元素访问
  - 数组名称 [下标]
- 一维数组的遍历



# 一维数组

- 一维数组的声明
  - 元素类型 数组名 [数组大小];
  - `int a[105];` // 定义了可以存放 105 个整数的数组 a
- 一维数组的元素访问
  - 数组名称 [下标]
- 一维数组的遍历
  - 使用单重循环遍历一维数组

- 多维数组的声明

# 多维数组

- 多维数组的声明
  - 元素类型 数组名称 [第一维大小][第二维大小]...;

# 多维数组

- 多维数组的声明
  - 元素类型 数组名称 [第一维大小][第二维大小]...;
  - `int a[45][65];` // 定义了 45 行 65 列的二维数组 a

# 多维数组

- 多维数组的声明
  - 元素类型 数组名称 [第一维大小][第二维大小]...;
  - `int a[45][65];` // 定义了 45 行 65 列的二维数组 a
- 多维数组的元素访问

# 多维数组

- 多维数组的声明
  - 元素类型 数组名称 [第一维大小][第二维大小]...;
  - `int a[45][65];` // 定义了 45 行 65 列的二维数组 a
- 多维数组的元素访问
  - 数组名称 [第一维下标][第二维下标]...

# 多维数组

- 多维数组的声明
  - 元素类型 数组名称 [第一维大小][第二维大小]...;
  - `int a[45][65];` // 定义了 45 行 65 列的二维数组 a
- 多维数组的元素访问
  - 数组名称 [第一维下标][第二维下标]...
- 多维数组的遍历

# 多维数组

- 多维数组的声明
  - 元素类型 数组名称 [第一维大小][第二维大小]...;
  - `int a[45][65];` // 定义了 45 行 65 列的二维数组 a
- 多维数组的元素访问
  - 数组名称 [第一维下标][第二维下标]...
- 多维数组的遍历
  - 使用多重循环遍历多维数组



# 目录

1 复习回顾

2 一维数组的应用

3 二维数组的应用

4 总结

## 例 12.1：统计人数 I

### 编程题

- 编写程序，输入一个整数  $n$  ( $1 \leq n \leq 100$ )，表示有  $n$  名学生，从左向右依次输入每个学生的成绩（成绩是 0 到 100 之间的整数）。  
统计每个学生的左边有几个成绩比自己好的同学，输出以空格间隔。
- 样例输入  
5  
98 90 79 100 87
- 样例输出  
0 1 2 0 3

## 例 12.1：统计人数 I

- 样例说明：

5

98 90 79 100 87

0 1 2 0 3

## 例 12.1：统计人数 I

- 样例说明：

5

98 90 79 100 87

0 1 2 0 3

- 第 1 个同学的左侧没有其他同学，输出 0

## 例 12.1：统计人数 I

- 样例说明：

5

98 90 79 100 87

0 1 2 0 3

- 第 1 个同学的左侧没有其他同学，输出 0
- 第 2 个同学的左侧有 1 个同学 98，输出 1

## 例 12.1：统计人数 I

- 样例说明：

5

98 90 79 100 87

0 1 2 0 3

- 第 1 个同学的左侧没有其他同学，输出 0
- 第 2 个同学的左侧有 1 个同学 98，输出 1
- 第 3 个同学的左侧有 2 个同学 98, 90，输出 2

## 例 12.1：统计人数 I

- 样例说明：

5

98 90 79 100 87

0 1 2 0 3

- 第 1 个同学的左侧没有其他同学，输出 0
- 第 2 个同学的左侧有 1 个同学 98，输出 1
- 第 3 个同学的左侧有 2 个同学 98, 90，输出 2
- 第 4 个同学的左侧有 3 个同学 98, 90, 79，输出 0

## 例 12.1：统计人数 I

- 样例说明：

5

98 90 79 100 87

0 1 2 0 3

- 第 1 个同学的左侧没有其他同学，输出 0
- 第 2 个同学的左侧有 1 个同学 98，输出 1
- 第 3 个同学的左侧有 2 个同学 98, 90，输出 2
- 第 4 个同学的左侧有 3 个同学 98, 90, 79，输出 0
- 第 5 个同学的左侧有 4 个同学 98, 90, 79, 100，输出 3



## 例 12.1: 统计人数 I

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    for (int i = 1; i <= n; i++) {
12        int cnt = 0;           // 重新声明 a[i] 的计数器
13        for (int j = 1; j < i; j++) { // 遍历 a[i] 左边的每一个学生
14            if (a[j] > a[i]) cnt++;
15        }
16        cout << cnt << " ";    // 数完 a[i] 后马上输出
17    }
18
19    return 0;
20 }
```

## 例 12.1: 统计人数 I

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    for (int i = 1; i <= n; i++) {
12        int cnt = 0;           // 重新声明 a[i] 的计数器
13        for (int j = 1; j < i; j++) { // 遍历 a[i] 左边的每一个学生
14            if (a[j] > a[i]) cnt++;
15        }
16        cout << cnt << " ";    // 数完 a[i] 后马上输出
17    }
18
19    return 0;
20 }
```

## 例 12.2: 统计人数 II

### 编程题

- 编写程序，输入一个整数  $n$  ( $1 \leq n \leq 100$ )，表示  $A$  班有  $n$  名学生，接下来输入  $A$  班  $n$  名学生的成绩。  
输入一个整数  $m$  ( $1 \leq m \leq 100$ )，表示  $B$  班有  $m$  名学生，接下来输入  $B$  班  $m$  名学生的成绩。(成绩均为 0 100 之间的整数)  
对  $A$  班的每个学生，统计在  $B$  班的  $m$  个人里有几个同学成绩比他好，输出以空格间隔。

- 样例输入

```
5
98 90 79 100 87
7
70 89 99 85 92 60 90
```

- 样例输出

```
1 2 5 0 4
```

## 例 12.2：统计人数 II

- 样例说明：

5

98 90 79 100 87

7

70 89 99 85 92 60 90

1 2 5 0 4

## 例 12.2：统计人数 II

- 样例说明：

5

98 90 79 100 87

7

70 89 99 85 92 60 90

1

2 5 0 4

- 对于 A 班第 1 个同学，B 班成绩比他好的同学有 99，输出 1

## 例 12.2：统计人数 II

- 样例说明：

```
5
98 90 79 100 87
7
70 89 99 85 92 60 90
1 2 5 0 4
```

- 对于 A 班第 1 个同学，B 班成绩比他好的同学有 99，输出 1
- 对于 A 班第 2 个同学，B 班成绩比他好的同学有 99,92，输出 2

## 例 12.2：统计人数 II

- 样例说明：

5

98 90 79 100 87

7

70 89 99 85 92 60 90

1 2 5 0 4

- 对于 A 班第 1 个同学，B 班成绩比他好的同学有 99，输出 1
- 对于 A 班第 2 个同学，B 班成绩比他好的同学有 99,92，输出 2
- 对于 A 班第 3 个同学，B 班成绩比他好的同学有 89,99,85,92,90，输出 5

## 例 12.2：统计人数 II

- 样例说明：

5

98 90 79 100 87

7

70 89 99 85 92 60 90

1 2 5 0 4

- 对于 A 班第 1 个同学，B 班成绩比他好的同学有 99，输出 1
- 对于 A 班第 2 个同学，B 班成绩比他好的同学有 99,92，输出 2
- 对于 A 班第 3 个同学，B 班成绩比他好的同学有 89,99,85,92,90，输出 5
- 对于 A 班第 4 个同学，B 班没有成绩比他好的同学，输出 0



## 例 12.2：统计人数 II

- 样例说明：

5

98 90 79 100 87

7

70 89 99 85 92 60 90

1 2 5 0 4

- 对于 A 班第 1 个同学，B 班成绩比他好的同学有 99，输出 1
- 对于 A 班第 2 个同学，B 班成绩比他好的同学有 99, 92，输出 2
- 对于 A 班第 3 个同学，B 班成绩比他好的同学有 89, 99, 85, 92, 90，输出 5
- 对于 A 班第 4 个同学，B 班没有成绩比他好的同学，输出 0
- 对于 A 班第 5 个同学，B 班成绩比他好的同学有 89, 99, 92, 90，输出 4

## 例 12.2: 统计人数 II

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105], b[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    int m;
12    cin >> m;
13    for (int i = 1; i <= m; i++) cin >> b[i];
14    for (int i = 1; i <= n; i++) {
15        int cnt = 0; // 重新声明 a[i] 的计数器
16        for (int j = 1; j <= m; j++) { // 遍历 b 班的每一个学生 b[j]
17            if (b[j] > a[i]) cnt++;
18        }
19        cout << cnt << " "; // 数完 a[i] 后马上输出
20    }
21
22    return 0;
23 }
```

## 例 12.2: 统计人数 II

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105], b[105];
6
7 int main() {
8     int n;
9     cin >> n;
10    for (int i = 1; i <= n; i++) cin >> a[i];
11    int m;
12    cin >> m;
13    for (int i = 1; i <= m; i++) cin >> b[i];
14    for (int i = 1; i <= n; i++) {
15        int cnt = 0; // 重新声明 a[i] 的计数器
16        for (int j = 1; j <= m; j++) { // 遍历 b 班的每一个学生 b[j]
17            if (b[j] > a[i]) cnt++;
18        }
19        cout << cnt << " "; // 数完 a[i] 后马上输出
20    }
21
22    return 0;
23 }
```

# 目录

1 复习回顾

2 一维数组的应用

**3 二维数组的应用**

4 总结

## 例 12.3: 计算两个矩阵的和

### 编程题

- 编写程序，输入两个整数  $n$  和  $m$  ( $1 \leq n, m \leq 100$ )，接下来输入两个  $n$  行  $m$  列、整数类型的二维数组  $a$  和  $b$ ，求这两个二维数组的和。

- 样例输入

```
2 3
6 3 1
3 2 2
1 6 4
3 2 3
```

- 样例输出

```
7 9 5
6 4 5
```

## 例 12.3：计算两个矩阵的和

- 两个  $n$  行  $m$  列的二维数组的和也是一个  $n$  行  $m$  列的二维数组
- 结果中第  $i$  行第  $j$  列的元素等于原来两个数组中第  $i$  行第  $j$  列的元素之和。

A	+	B	=	C
6 3 1		1 6 4		7 9 5
3 2 2		3 2 3		6 4 5

## 例 12.3: 计算两个矩阵的和

- 两个  $n$  行  $m$  列的二维数组的和也是一个  $n$  行  $m$  列的二维数组
- 结果中第  $i$  行第  $j$  列的元素等于原来两个数组中第  $i$  行第  $j$  列的元素之和。

$$\begin{array}{c} \text{A} \\ \boxed{6} \ 3 \ 1 \\ 3 \ 2 \ 2 \end{array} + \begin{array}{c} \text{B} \\ \boxed{1} \ 6 \ 4 \\ 3 \ 2 \ 3 \end{array} = \begin{array}{c} \text{C} \\ \boxed{7} \ 9 \ 5 \\ 6 \ 4 \ 5 \end{array}$$

## 例 12.3: 计算两个矩阵的和

- 两个  $n$  行  $m$  列的二维数组的和也是一个  $n$  行  $m$  列的二维数组
- 结果中第  $i$  行第  $j$  列的元素等于原来两个数组中第  $i$  行第  $j$  列的元素之和。

$$\begin{array}{c} \text{A} \\ 6 \quad \boxed{3} \quad 1 \\ 3 \quad 2 \quad 2 \end{array} + \begin{array}{c} \text{B} \\ 1 \quad \boxed{6} \quad 4 \\ 3 \quad 2 \quad 3 \end{array} = \begin{array}{c} \text{C} \\ 7 \quad \boxed{9} \quad 5 \\ 6 \quad 4 \quad 5 \end{array}$$



## 例 12.3：计算两个矩阵的和

- 两个  $n$  行  $m$  列的二维数组的和也是一个  $n$  行  $m$  列的二维数组
- 结果中第  $i$  行第  $j$  列的元素等于原来两个数组中第  $i$  行第  $j$  列的元素之和。

$$\begin{array}{ccc} \text{A} & + & \text{B} & = & \text{C} \\ \begin{array}{ccc} 6 & 3 & \boxed{1} \\ 3 & 2 & 2 \end{array} & & \begin{array}{ccc} 1 & 6 & \boxed{4} \\ 3 & 2 & 3 \end{array} & & \begin{array}{ccc} 7 & 9 & \boxed{5} \\ 6 & 4 & 5 \end{array} \end{array}$$

## 例 12.3: 计算两个矩阵的和

- 两个  $n$  行  $m$  列的二维数组的和也是一个  $n$  行  $m$  列的二维数组
- 结果中第  $i$  行第  $j$  列的元素等于原来两个数组中第  $i$  行第  $j$  列的元素之和。

A	+	B	=	C
6 3 1		1 6 4		7 9 5
<b>3</b> 2 2		<b>3</b> 2 3		<b>6</b> 4 5

## 例 12.3: 计算两个矩阵的和

- 两个  $n$  行  $m$  列的二维数组的和也是一个  $n$  行  $m$  列的二维数组
- 结果中第  $i$  行第  $j$  列的元素等于原来两个数组中第  $i$  行第  $j$  列的元素之和。

$$\begin{array}{ccc} \text{A} & + & \text{B} & = & \text{C} \\ \begin{array}{ccc} 6 & 3 & 1 \\ 3 & \boxed{2} & 2 \end{array} & & \begin{array}{ccc} 1 & 6 & 4 \\ 3 & \boxed{2} & 3 \end{array} & & \begin{array}{ccc} 7 & 9 & 5 \\ 6 & \boxed{4} & 5 \end{array} \end{array}$$

## 例 12.3：计算两个矩阵的和

- 两个  $n$  行  $m$  列的二维数组的和也是一个  $n$  行  $m$  列的二维数组
- 结果中第  $i$  行第  $j$  列的元素等于原来两个数组中第  $i$  行第  $j$  列的元素之和。

$$\begin{array}{ccc} \text{A} & + & \text{B} & = & \text{C} \\ \begin{array}{ccc} 6 & 3 & 1 \\ 3 & 2 & \boxed{2} \end{array} & & \begin{array}{ccc} 1 & 6 & 4 \\ 3 & 2 & \boxed{3} \end{array} & & \begin{array}{ccc} 7 & 9 & 5 \\ 6 & 4 & \boxed{5} \end{array} \end{array}$$

## 例 12.3：计算两个矩阵的和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105][105], b[105][105], c[105][105];
6
7 int main() {
8     int n, m;
9     cin >> n >> m;
10    // 输入 a 数组
11    for (int i = 1; i <= n; i++) {
12        for (int j = 1; j <= m; j++) {
13            cin >> a[i][j];
14        }
15    }
16    // 输入 b 数组
17    for (int i = 1; i <= n; i++) {
18        for (int j = 1; j <= m; j++) {
19            cin >> b[i][j];
20        }
21    }
```

## 例 12.3: 计算两个矩阵的和

```
22  // 计算两个矩阵的和
23  for (int i = 1; i <= n; i++) {
24      for (int j = 1; j <= m; j++) {
25          c[i][j] = a[i][j] + b[i][j];
26      }
27  }
28  // 输出结果
29  for (int i = 1; i <= n; i++) {
30      for (int j = 1; j <= m; j++) {
31          cout >> c[i][j] << " ";
32      }
33      cout << endl;
34  }
35
36  return 0;
37 }
```

## 例 12.4：二维数组的旋转

### 编程题

- 首先输入两个整数  $n, m$  ( $1 \leq n, m \leq 100$ ), 表示接下来需要输入  $n$  行  $m$  列的正整数 (正整数均小于 1000), 这些整数组成了一个  $n$  行  $m$  列的矩阵, 请将矩阵顺时针翻转 90 度后输出。
- 样例输入  
2 3  
1 3 7  
8 4 6
- 样例输出  
8 1  
4 3  
6 7

## 例 12.4：二维数组的旋转

- 当前数组：

1(1,1) 3(1,2) 7(1,3)

8(2,1) 4(2,2) 6(2,3)

- 当前输出：

- 输出旋转后的数组时，行坐标  $i$  和列坐标  $j$  如何变化？



## 例 12.4：二维数组的旋转

- 当前数组：

1(1,1)	3(1,2)	7(1,3)
8(2,1)	4(2,2)	6(2,3)

第1列

- 当前输出：

8(2,1) 1(1,1)

- 输出旋转后的数组时，行坐标  $i$  和列坐标  $j$  如何变化？

## 例 12.4：二维数组的旋转

- 当前数组：

1(1,1)	3(1,2)	7(1,3)
8(2,1)	4(2,2)	6(2,3)

第2列

- 当前输出：

8(2,1)	1(1,1)
4(2,2)	3(1,2)

- 输出旋转后的数组时，行坐标  $i$  和列坐标  $j$  如何变化？

## 例 12.4：二维数组的旋转

- 当前数组：

1(1,1)	3(1,2)	7(1,3)
8(2,1)	4(2,2)	6(2,3)

第3列

- 当前输出：

8(2,1)	1(1,1)
4(2,2)	3(1,2)
6(2,3)	7(1,3)

- 输出旋转后的数组时，行坐标  $i$  和列坐标  $j$  如何变化？

## 例 12.4：二维数组的旋转

```
1 #include <iostream>
2
3 using namespace std;
4
5 int a[105][105];
6
7 int main() {
8     int n, m;
9     cin >> n >> m;
10    for (int i = 1; i <= n; i++) {
11        for (int j = 1; j <= m; j++) {
12            cin >> a[i][j];
13        }
14    }
15    for (int j = 1; j <= m; j++) { // 输出第 j 列的所有元素
16        for (int i = n; i >= 1; i--) {
17            cout << a[i][j] << " ";
18        }
19        cout << endl;
20    }
21
22    return 0;
23 }
```

## 例 12.5: 计算小矩阵的和

### 编程题

- 首先输入两个整数  $n, m$  ( $1 \leq n, m \leq 100$ ), 表示接下来需要输入  $n$  行  $m$  列的正整数 (正整数均小于 1000), 最后再输入四个整数  $x, y, r, c$  ( $1 \leq x, x + r - 1 \leq n$ ), ( $1 \leq y, y + c - 1 \leq m$ ), 请输出原矩阵中以  $(x, y)$  为左上角的  $r$  行  $c$  列的小矩阵元素的总和。
- 样例输入
- 样例输出

```
4 5
1 3 3 4 0
8 4 3 5 6
3 9 2 1 5
4 2 1 6 5
1 3 3 2
```

18

## 例 12.5: 计算小矩阵的和

### 编程题

- 首先输入两个整数  $n, m$  ( $1 \leq n, m \leq 100$ ), 表示接下来需要输入  $n$  行  $m$  列的正整数 (正整数均小于 1000), 最后再输入四个整数  $x, y, r, c$  ( $1 \leq x, x + r - 1 \leq n$ ), ( $1 \leq y, y + c - 1 \leq m$ ), 请输出原矩阵中以  $(x, y)$  为左上角的  $r$  行  $c$  列的小矩阵元素的总和。
- 样例输入
- 样例输出

4	5			
1	3	3	4	0
8	4	3	5	6
3	9	2	1	5
4	2	1	6	5
1	3	3	2	

第一行, 第三列

18

起点: (1,3)

## 例 12.5: 计算小矩阵的和

### 编程题

- 首先输入两个整数  $n, m$  ( $1 \leq n, m \leq 100$ ), 表示接下来需要输入  $n$  行  $m$  列的正整数 (正整数均小于 1000), 最后再输入四个整数  $x, y, r, c$  ( $1 \leq x, x + r - 1 \leq n$ ), ( $1 \leq y, y + c - 1 \leq m$ ), 请输出原矩阵中以  $(x, y)$  为左上角的小矩阵元素的总和。
- 样例输入
- 样例输出

4	5			
1	3	3	4	0
8	4	3	5	6
3	9	2	1	5
4	2	1	6	5
1	3	3	2	

第三行, 第五列

小矩阵: 3 行 2 列

18

## 例 12.5: 计算小矩阵的和

- 当前输入:

4	5	
1	3	3 4 0
8	4	3 5 6
3	9	2 1 5
4	2	1 6 5
1	3	3 2

第 x 行, 第 y 列

起点: (x,y)



## 例 12.5: 计算小矩阵的和

- 当前输入:

4 5

1 3 3 4 0 第 x 行, 第 y 列

8 4 3 5 6 第 ? 行, 第 ? 列

3 9 2 1 5 第 ? 行, 第 ? 列

4 2 1 6 5

1 3 3 2 小矩阵: r 行 c 列

起点: (x,y)

## 例 12.5: 计算小矩阵的和

- 当前输入:

4	5			
1	3	3	4	0
8	4	3	5	6
3	9	2	1	5
4	2	1	6	5

第  $x$  行, 第  $y$  列

第 ? 行, 第 ? 列

小矩阵:  $r$  行  $c$  列

起点:  $(x, y)$

- 遍历小矩阵时, 小矩阵的终点行、终点列应当如何计算?

## 例 12.5: 计算小矩阵的和

```
1 int a[105][105];
2
3 int main() {
4     int n, m;
5     cin >> n >> m;
6     for (int i = 1; i <= n; i++) {
7         for (int j = 1; j <= m; j++) {
8             cin >> a[i][j];
9         }
10    }
11    int x, y, r, c;
12    cin >> x >> y >> r >> c;
13    int sum = 0;
14    for (int i = x; i < x + r; i++) {
15        for (int j = y; j < y + c; j++) {
16            sum += a[i][j];
17        }
18    }
19    cout << sum << endl;
20
21    return 0;
22 }
```

## 例 12.6：出现最频繁的点

### 编程题

- 给定  $n$  ( $1 \leq n \leq 100$ ) 个二维空间的点，每个点的坐标  $(x, y)$  都是整数 ( $0 \leq x, y \leq 99$ )，求其中出现最频繁的出现次数。
- 样例输入  
4  
1 2  
2 3  
2 3  
5 6
- 样例输出  
2

## 例 12.6: 出现最频繁的点

```
1 int cnt[105][105]; // cnt[i][j]:记录第 i 行第 j 列出现该点的次数
2
3 int main() {
4     int n;
5     cin >> n;
6     int x, y;
7     for (int i = 1; i <= n; i++) {
8         cin >> x >> y;
9         cnt[x][y]++;
10    }
11    int mx = 0; // 打擂台找最大值, 行和列范围都是 0~99
12    for (int i = 0; i <= 99; i++) {
13        for (int j = 0; j <= 99; j++) {
14            if (cnt[i][j] > mx) {
15                mx = cnt[i][j];
16            }
17        }
18    }
19    cout << mx << endl;
20
21    return 0;
22 }
```

# 目录

1 复习回顾

2 一维数组的应用

3 二维数组的应用

4 总结

- 一维数组的应用

- 一维数组的应用
  - 一维数组部分求和



- 一维数组的应用
  - 一维数组部分求和
  - 一维数组遍历比较

- 一维数组的应用
  - 一维数组部分求和
  - 一维数组遍历比较
- 二维数组的应用

- 一维数组的应用
  - 一维数组部分求和
  - 一维数组遍历比较
- 二维数组的应用
  - 二维数组的旋转

- 一维数组的应用
  - 一维数组部分求和
  - 一维数组遍历比较
- 二维数组的应用
  - 二维数组的旋转
  - 计算小矩阵的和

# 多维数组

- 一维数组的应用
  - 一维数组部分求和
  - 一维数组遍历比较
- 二维数组的应用
  - 二维数组的旋转
  - 计算小矩阵的和
  - 二维数组的计数应用

# Thank you!