



实验舱  
青少年编程  
走近科学 走进名校

# 挑战信息学奥林匹克

C++程序设计  
一维数组 (1)

# 问题讨论：

编一个程序，要求：

读入 $n$ 个整数 ( $n \leq 10000$ )以后，将这 $n$ 个数反向输出。

【输入样例】

5

23 43 87 99 57

【输出样例】

57 99 87 43 23

---

# 解决方法

- 借助数学概念，引入下标变量：

$a_0, a_1, a_2, a_3, a_4, \dots a_i \dots a_n$

- 在C++中表示为：

$a[0], a[1], a[2], a[3], a[4], \dots a[i] \dots a[n]$

- 先将数据存储在下标变量中，然后再逆序输出。
-

## 问题讨论:

编一个程序，要求：

读入 $n$ 个整数 ( $n \leq 10000$ )以后，将这 $n$ 个数反向输出。

【输入样例】

5

23 43 87 99 57

【输出样例】

57 99 87 43 23

```
/*数组读取数据*/
```

```
for ( int i = 1; i <= n; i++ )  
{  
    cin >> a[i];  
}
```

```
/*逆序输出数组*/
```

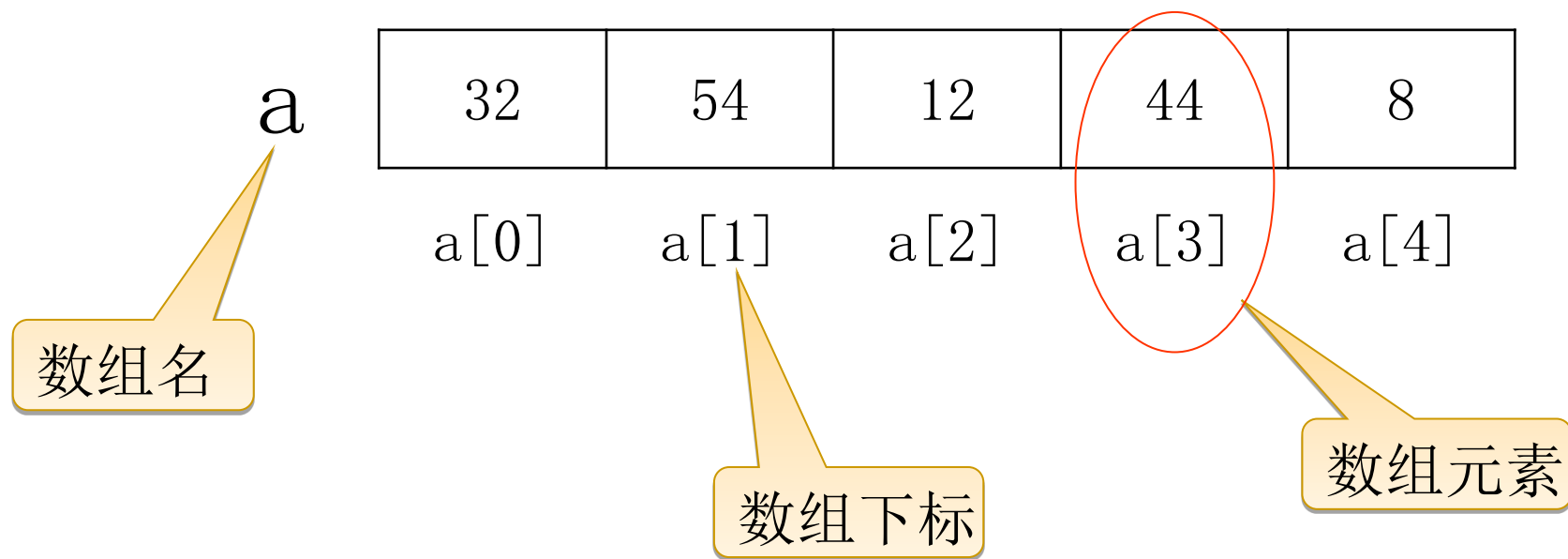
```
for ( int i = n; i >= 1; i-- )  
{  
    cout << a[i] << " ";  
}
```

## 程序设计需要解决的一个重要问题：

- 在程序执行时，如何存储大量的数据？
  - 存储的方式要便于计算机自动地进行操作
  - 解决方法之一：数组（顺序存储）
-

# 数组概念

- 一组连续下标的变量
- 同一数组的所有元素在内存中占用一片连续的存储单元



# 一维数组定义

类型名 数组名[元素个数]

例如:

常数

`int a[100];`

`float b[10];`

`char c[20];`

`int d[n];` //d数组定义非法

说明:

- 下标值必须从0开始
- `int a[100];`
  - a数组有100个元素
  - 下标值0~99
  - `a[100]`不合法
- 在使用数组的过程中注意不能越界

# 一维数组的引用

数组名[下标]

例如：

`int a[10];`

`a[5]`

`a[i]`

`a[i+j] //i+j<10`

`a[i++]`

说明：

- 数组引用时，下标可以是整型的表达式
  - 整数
  - 变量
  - 表达式
  - 函数
- 数组元素只能一个一个引用
- 数组元素可以像普通变量一样操作



# 一维数组的输入与输出（下标：1~n）

## ■ 一维数组输入

```
for ( int i = 1; i <= n; i++ ) {  
    cin >> a[i];  
}
```

## ■ 一维数组输出（顺序输出）

```
for ( int i = 1; i <= n; i++ ) {  
    cout << a[i] << ' ';  
}
```

---

# 例1：数组逆序输出

## 题目描述

顺序读入  $n$  个整数，先顺序输出，然后逆序输出。

## 【输入格式】

两行，第一行  $n$ ， $n \leq 30000$ ，第二行有  $n$  个整数。

## 【输出格式】

两行，一行是原来顺序的  $n$  个整数，每个整数后面一个空格。

## 【输入样例】

```
5
32 54 12 44 8
```

## 【输出样例】

```
32 54 12 44 8
8 44 12 54 32
```

## 例1：数组逆序输出

```
int a[30010];
int n;
cin >> n;
/*数组读取数据*/
for ( int i = 1; i <= n; i++ )
{
    cin >> a[i];
}
```

```
/*顺序输出数组*/
for ( int i = 1; i <= n; i++ )
{
    cout << a[i] << ' ';
}
cout << endl;
/*逆序输出数组*/
for ( int i = n; i >= 1; i-- )
{
    cout << a[i] << " ";
}
cout<<endl;
```

## 例2：查找指定数

### 【描述】

输出一个整数序列中与指定数字相同的数的个数。

### 【输入】

输入包含三行：

第一行为  $N$ ，表示整数序列的长度(  $N \leq 100$  )；

第二行为  $N$  个整数，整数之间以一个空格分开,每个数不超过 1000 ；

第三行包含一个整数，为指定的整数  $m$  。

### 【输出】

输出为  $N$  个数中与  $m$  相同的数的个数。

### 【样例输入】

```
3
2 3 2
2
```

### 【样例输出】

```
2
```

## 查找指定数参考代码

循环读取n个数据

枚举n个数据，统计  
**m**出现的个数

```
int a[101];
int n, m, t = 0;
cin >> n;
for (int i = 1; i <= n; i++)
{
    cin >> a[i];
}
cin >> m;
for (int i = 1; i <= n; i++)
{
    if ( a[i] == m ) t++;
}
cout << t << endl;
```

## 例3：数字交换

### 题目描述

有一个数组，编一程序交换这组数中任意指定的两个位置的数字。

### 输入

第一行，一个数N（不超过100个）。

第二行，N个数用空格隔开。

第三行，一个整数m（不超过100）。

下面m行，每行两个整数，交换数字的位置。

### 输出

交换后的一行数（中间用空格隔开）

### 样例输入

16

3 6 11 45 23 70 67 34 26 89 90 15 56 50 20 10

2

3 5

13 15

### 样例输出

3 6 23 45 11 70 67 34 26 89 90 15 20 50 56 10

# 算法分析

- 根据题目给定的数据范围，确定数组至少有100个元素。
- 根据样例，数组下标1~n更为方便。
- 算法过程
  1. 读取数组数据 ( $a[1] \sim a[n]$ )
  2. 读取交换位置信息，并且交换数据。
  3. 输出数组

2

3 5

13 15

3	6	11	45	23	70	67	34	26	89	90	15	56	50	20	10
3	6	23	45	11	70	67	34	26	89	90	15	20	50	56	10
$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$	$a[8]$	$a[9]$	$a[10]$	$a[11]$	$a[12]$	$a[13]$	$a[14]$	$a[15]$	$a[16]$

# 算法分析

- 读取数组交换的位置，进行数据交换。

```
cin >> m;  
for ( int i = 0; i < m; i++ )  
{  
    cin >> x >> y;  
    swap(a[x], a[y]);  
}
```





## 例4: 计算书费

### 【描述】

下面是一个图书的单价表:

计算概论 28.9 元/本

数据结构与算法 32.7 元/本

数字逻辑 45.6元/本

C++程序设计教程 78 元/本

人工智能 35 元/本

计算机体系结构 86.2 元/本

编译原理 27.8元/本

操作系统 43 元/本

计算机网络 56 元/本

JAVA程序设计 65 元/本

给定每种图书购买的数量，编程计算应付的总费用。

---

## 例4: 计算书费

### 【输入】

输入一行，包含10个整数(大于等于0，小于等于100)，分别表示购买的《计算概论》、《数据结构与算法》、《数字逻辑》、《C++程序设计教程》、《人工智能》、《计算机体系结构》、《编译原理》、《操作系统》、《计算机网络》、《JAVA程序设计》的数量（以本为单位）。每两个整数用一个空格分开。

### 【输出】

输出一行，包含一个浮点数f，表示应付的总费用。精确到小数点后一位。

### 【样例输入】

1 5 8 10 5 1 1 2 3 4

### 【样例输出】

2140.2

---

# 问题分析

- 10种图书价格（已知），10种图书购书数量。
- 用数组预先存放图书价格。

28.9	32.7	45.6	78	35	86.2	27.8	43	56	65
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

## 例4: 计算书费

```
const int maxn = 10;
int main()
{
    double a[maxn] = {28.9, 32.7, 45.6, 78, 35, 86.2, 27.8, 43, 56, 65};
    int m;
    double s = 0;
    for (int i = 0; i < maxn; i++)
    {
        cin >> m;
        s = s + m * a[i];
    }

    printf("%.11f\n", s);
    return 0;
}
```

---

# 一维数组的初始化

```
int a[5]={1,2,3,4,5};
```

a	1	2	3	4	5
	a[0]	a[1]	a[2]	a[3]	a[4]

```
int b[10]={0,1,2,3,4};
```

b	0	1	2	3	4	0	0	0	0	0	0
	b[0]	b[1]	b[2]	b[3]	b[4]	b[5]	b[6]	b[7]	b[8]	b[9]	b[10]

```
int c[100]={};
```

0	0	0	0	0	0	0	0	.....
c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	.....

---

## 例5：数组元素的移动

题目描述

数组有 $n$ 个元素，要求把第 $x$ 个位置的元素移到数组元素的移动组的最后。

输入

有3行

第一行有一个整数 $n$  ( $n \leq 100$ )

第二行有 $n$ 个整数

第三行有一个整数 $x$

输出

移动后的数组

样例输入

10

8 7 4 5 12 15 1 16 11 5

3

样例输出

8 7 5 12 15 1 16 11 5 4



8 7 4 5 12 15 1 16 11 5 4

# 算法过程

1. 读取数据到a数组中(a[1]~a[n])
2. 移动数组元素
  - ◆ 将m位的元素移到数组最后
  - ◆ 从m位开始，数组元素向前移动一位
3. 输出数组

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

思考：

- 能否不移动数组元素，直接按照要求输出。





# 参考代码

## ■ 数组元素移动

```
cin >> m;  
for ( int i = 1; i < m; i++ ) cout <<a[i]<<' ';  
for ( int i = m+1; i <= n; i++ ) cout <<a[i]<<' ';  
cout << a[m];
```

---

## 举一反三

假如将前 $m$ 个数组元素移动到数组后面，你会怎么做？



The diagram shows an array of numbers: 8 7 4 5 12 15 1 16 11 5. The first four numbers (8, 7, 4, 5) are enclosed in a red oval. A red arrow originates from the right side of this oval and points to the last element of the array (5), illustrating the movement of the first  $m$  elements to the end of the array.

8 7 4 5 12 15 1 16 11 5

## 例6：去掉最大数和最小数

描述

输入 $n$ 个大于0的整数，去掉里面所有最大值和最小值，剩下的数求平均值。

输入

二行。

第一行，一个整数 $n$  ( $5 \leq n \leq 100$ )。

第二行， $n$ 个正整数，这些数均不大于1000000。

输出

一个浮点数，保留2位小数。

输入样例

10

98 83 83 80 85 71 76 98 82 88

输出样例

82.43

思考：

(1) 数组定义多大？

(2) 用什么算法？

## 例6：去掉最大数和最小数

描述

输入n个大于0的整数，去掉里面所有最大值和最小值，剩下的数求平均值。

输入

二行。

第一行，一个整数n（ $5 \leq n \leq 100$ ）。

第二行，n个正整数，这些数均不大于1000000。

输出

一个浮点数，保留2位小数。

输入样例

10

98 83 83 80 85 71 76 98 82 88

输出样例

82.43

思考：

(1) 数组定义多大？

(2) 用什么算法？

- 查找最大值
- 查找最小值
- 求和
- 计数
- 计算平均值

# 算法分析

1. 定义数组a
  2. 读取数据到数组a中
  3. 枚举a数组，查找最大值和最小值
  4. 枚举a数组，对不等于最大值和最小值的数求和，并且计数
  5. 求平均值并计数
-

# 查找最大值最小值

- 枚举a数组，用擂台法求最大值和最小值

```
int mmax = a[0], mmin = a[0];  
for ( int i = 0; i < n; i++ )  
{  
    if ( a[i] > mmax ) mmax = a[i];  
    if ( a[i] < mmin ) mmin = a[i];  
}
```

# 求和、计数

- 枚举a数组，对不等于最大值和最小值的数求和，并且计数

```
for ( int i = 0; i < n; i++ )
{
    if ( a[i] != mmax && a[i] != mmin)
    {
        t++;
        s += a[i];
    }
}
printf("%.2lf\n", s/t);
```