第 5 讲

数组程序设计

主要内容

5.1 再识数组

5.2 数组的定义和初始化

5.3 数组的数据处理

5.4 数组程序设计举例

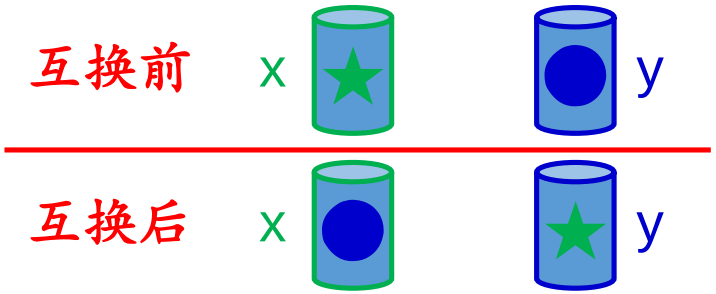
5.1 再识数组

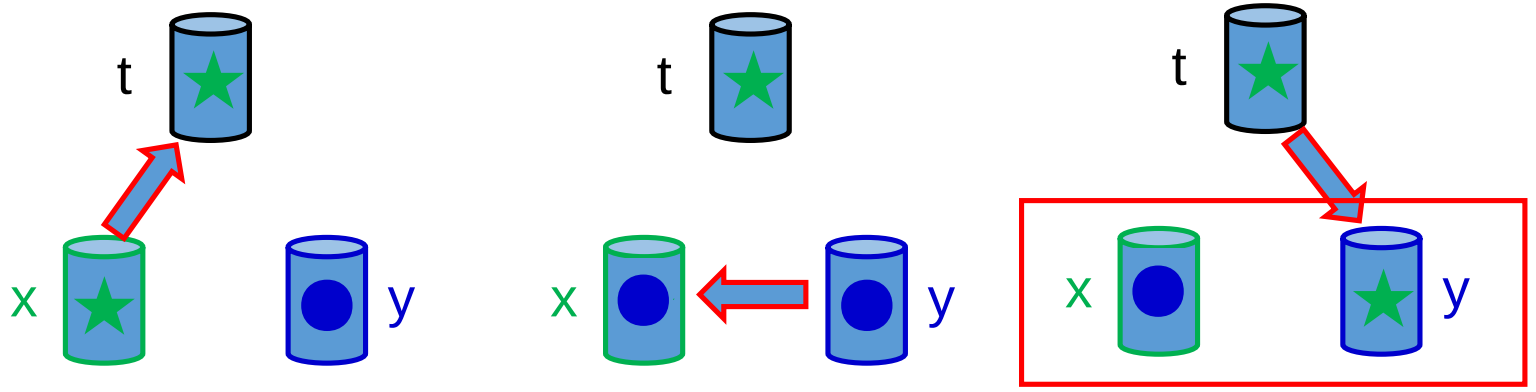
数组的引入

- 一个变量只能存储一个值，只能对其单独进行处理。变量对于批量数据无能为力

例1：对两个数据x、y从大到小进行排序，排序算法为：

- ✓ 进行x和y的比较
- ✓ 如果x<y，交换x和y的值


互换前
x
y
互换后
x
y


t
x
y
t
x
y
t
x
y

例2：对x、y、z三个数据从大到小排序。

排序算法为：

- ✓ 进行x和y的比较，如果x<y，交换x和y的值
- ✓ 进行y和z的比较，如果y<z，交换y和z的值
- ✓ 进行x和y的比较，如果x<y，交换x和y的值

例3：设计一个程序，对全年级170名学生的“计算机程序设计”课程的成绩进行排序。需要

- ✓ 定义170个变量来存储这些成绩
- ✓ 设计代码输入（赋值）这些成绩
- ✓ 设计代码来进行具体的排序过程
- ✓ 设计代码输出排好序的结果 ...

分析可知：

排序过程中每一趟的操作步骤基本相同，只是操作对象不同：使用循环；可使用循环控制变量自身的值或包含循环控制变量的表达式来确定要访问的变量

回顾数列：

- ✓ 数列的元素a_1、a_2、…、a_{100}由名字和下标确定
- ✓ 使用循环来控制访问数列中的所有元素，只要使用循环控制变量自身的值或包含循环控制变量的表达式作为数列元素的下标值即可 → 数组

- 数组将一组数据类型相同的数据按顺序存储在连续的空间中
- 数组提供了一种组织数据的机制，所以属于构造数据类型
- 处理大批量、存储在数组中的数据，比处理分散存放到变量中大批数据更加容易和高效
- 但比普通变量使用复杂，代码要长，调试难

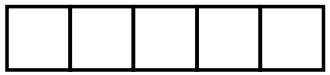
适用场合：

只有在必须将大量的数据集全部同时放入内存中的情况下才使用数组

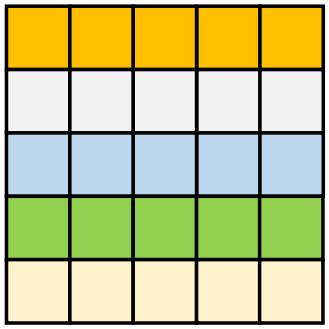
- 区别于变量，数组可以存储一组值。所存储的这组值的个数称为“**数组长度**”(或数组大小)
  - ✓ 因为决定着为数组分配的空间的大小，数组长度必须为常量，是一个非负整数，该常量在程序运行过程中不能更改
  - ✓ 数组长度要恰当，否则会出现不够用或浪费现象
  - ✓ 为方便阅读和修改，一般使用符号常量指定数组长度，当需要修改数组长度时，只需修改符号常量的值即可

- 数组中的存储单元称为“**数组元素**”
  - ✓ 数组元素通常是连续的空间
  - ✓ 可以通过赋值或键盘输入数据方式为数组元素指定值（内容）
  - ✓ 可以使用循环遍历数组，对每个数组元素实施相同的操作
  - ✓ 每个数组元素本质上与普遍变量相同，可以在数学表达式等中使用数组元素参与运算

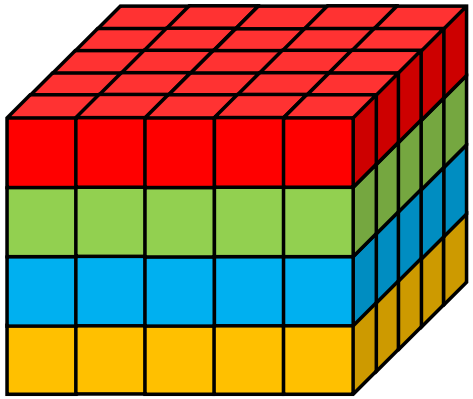
- 每一个数组元素都指定了一个唯一编号：称为“**下标**”
  - ✓ 下标是数组中具体元素的标志
  - ✓ 可以通过下标操作数组元素
  - ✓ 大多数程序设计语言中，下标从**0**开始
  - ✓ 下标可以是具有整数值的变量或表达式
  - ✓ 下标使用不当会造成数组“越界”
  - ✓ 下标可以有多个，为多维数组



1D

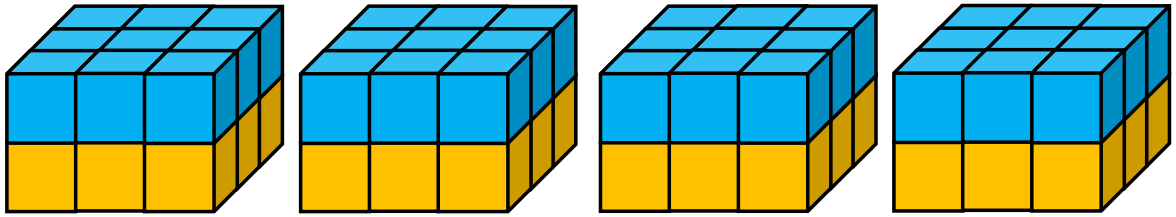


2D

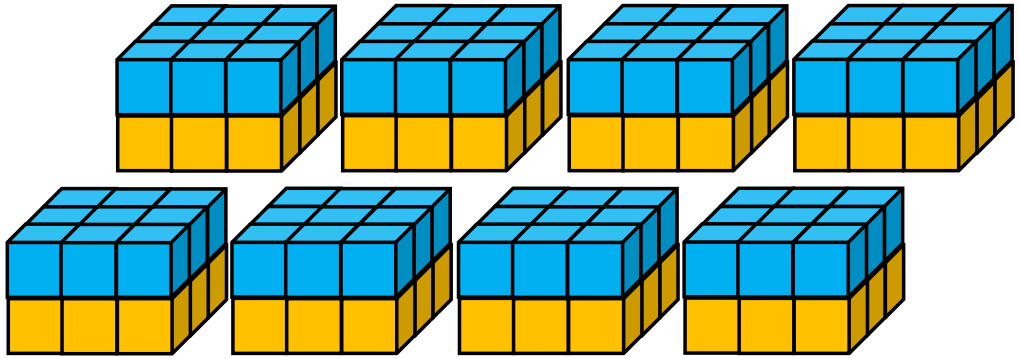


3D

4D



5D



5.2 数组的定义和初始化

- “**数组的定义**”要解决的问题：
  - ✓ 数组名是什么？
  - ✓ 数组有多少个元素？
  - ✓ 数组元素的类型是什么？

- 大多数程序设计语言可以在定义一个数组时给该数组赋初值，称为“**数组的初始化**”
  - ✓ 一系列用于初始化的数值称为初始化列表
  - ✓ 这些数值按照它们在列表中的顺序存储在数组元素中
  - ✓ 数组初始化是在编译阶段进行的，这样可以减少程序的运行时间

C语言中数组的定义及初始化

元素类型 数组名[数组长度];

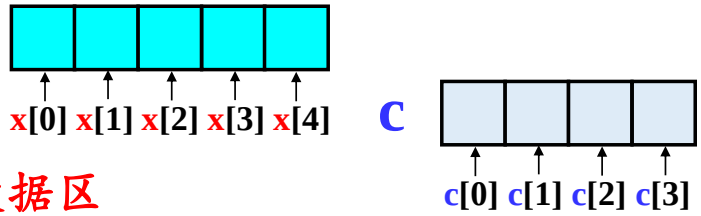
元素类型 数组名[长度1] [长度2];

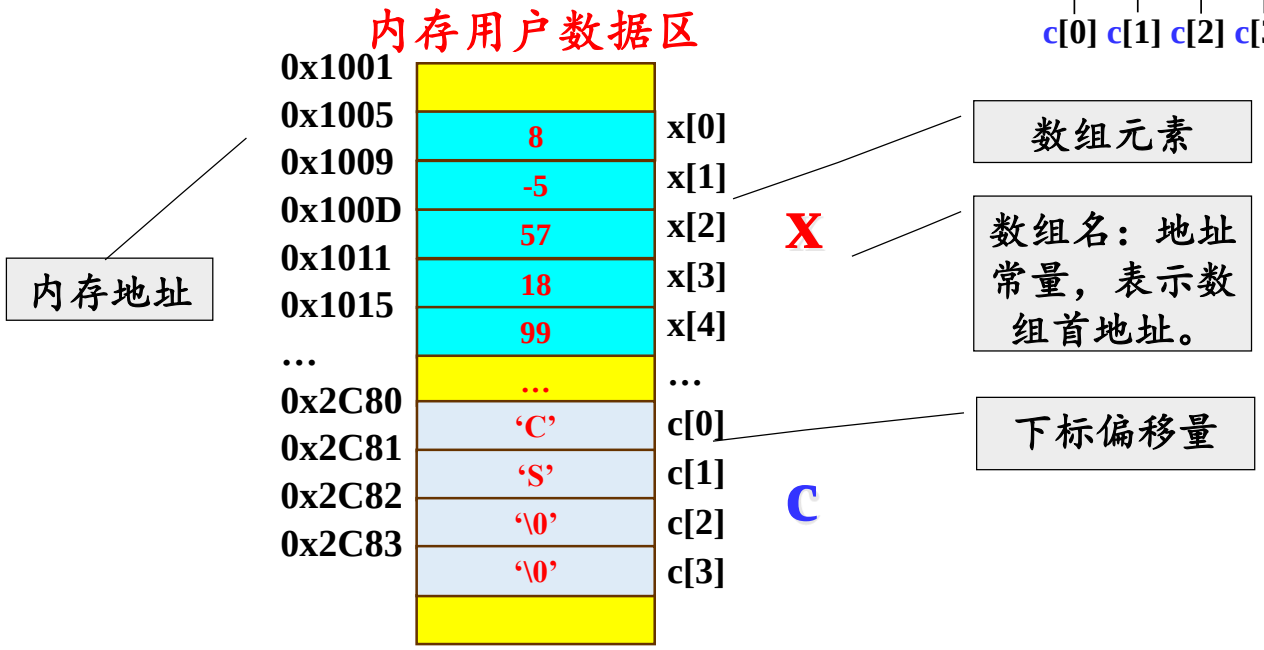
元素类型 数组名[长度1] [长度2][长度3];

例：

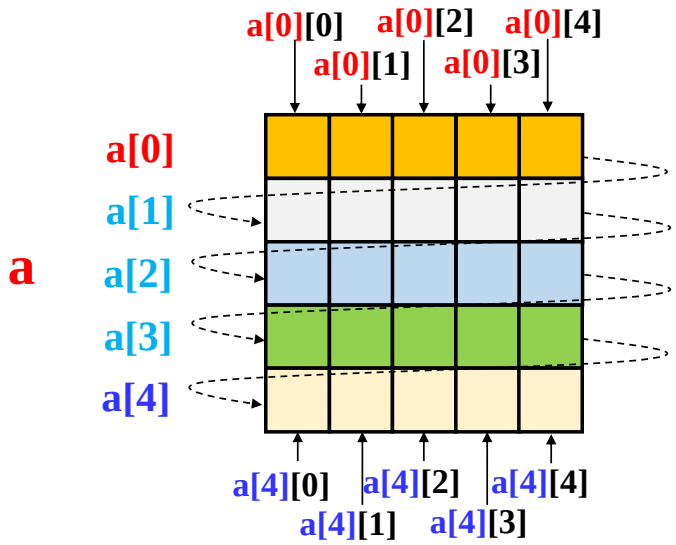
```
int   x[10];      int   a[3][4];      int   u[2][5][8];
float y[7];       float b[4][3];      float v[3][6][9];
char  z[8];       char  c[2][5];      char  w[5][7][10];
```

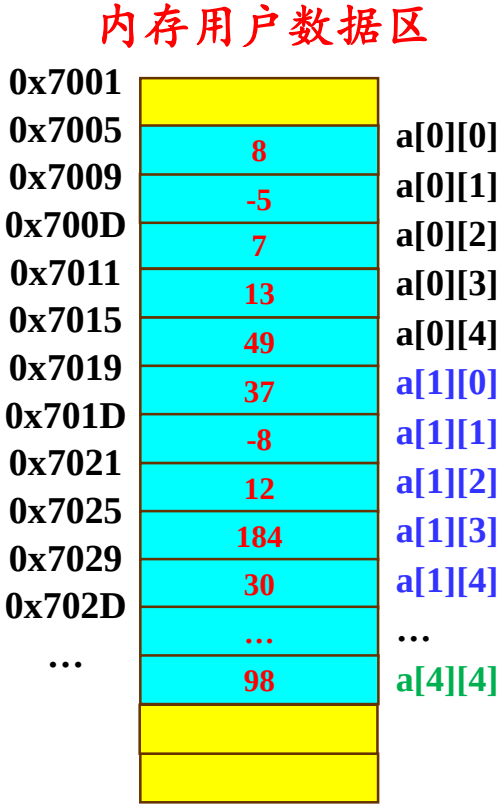
例：int x[5];
char c[4];





例：int a[5][5];


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


内存用户数据区
0x7001
0x7005
0x7009
0x700D
0x7011
0x7015
0x7019
0x701D
0x7021
0x7025
0x7029
0x702D
…
8
-5
7
13
49
37
-8
12
184
30
…
98
a[0][0]
a[0][1]
a[0][2]
a[0][3]
a[0][4]
a[1][0]
a[1][1]
a[1][2]
a[1][3]
a[1][4]
…
a[4][4]

元素类型 数组名[数组长度]={初值0,初值1,初值2,…};

例：
```
int    x[10]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
float y[7]={3, 1.2, -5.6, 0, 7.8, 3.14, -21};
char z[8]={‘E’, ‘C’, ‘U’, ‘S’, ‘T’, ‘ ’, ‘C’, ‘S’};
```

也可：
```
int    x[ ]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
int  t[5]={0};
float y[10]={3, 1.2, -5.6, 0, 7.8, 3.14, -21};
char z[8]={‘E’, ‘C’, ‘U’, ‘S’, ‘T’};
```

```
int    a[3][4]={{1,2,3,4},{5,6,7,8},{10,11,12,13}};
float b[4][3]={{1.1,2.2,3.3},{0,0,0},{2.1,3.2,4.3},{1,1,1}};
char c[2][5]={{'C','H','I','N','A'},{'E','C','U','S','T'}};
```

```
int    a[3][4]={1,2,3,4,5,6,7,8,10,11,12,13};
float b[4][3]={1.1,2.2,3.3,0,0,0,2.1,3.2,4.3,1,1,1};
char c[2][5]={'C','H','I','N','A','E','C','U','S','T'};
```

```
int    a[ ][4]={{1,2,3,4},{5,6,7,8},{10,11,12,13}};
float b[ ][3]={{1.1,2.2,3.3},{0,0,0},{2.1,3.2,4.3},{1,1,1}};
char c[ ][5]={{'C','H','I','N','A'},{'E','C','U','S','T'}};
```

```
int    a[ ][4]={{1,2,3},{5,6},{10}};
float b[4][3]={{1.1,2.2,3.3},{0},{2.1,3.2,4.3},{1,1,1}};
char  c[ ][5]={'C','H','I','N','A','E','C','U','S','T'};
```

```
// 提倡如下写法：
int    a[3][4]={{1,2,3,4},
                {5,6,7,8},
                {10,11,12,13}};
float b[4][3]={{1.1,2.2,3.3},
                {0,0,0},
                {2.1,3.2,4.3},
                {1,1,1}};
char c[2][5]={{‘C’,‘H’,‘I’,‘N’,‘A’},
                {‘E’,‘C’,‘U’,‘S’,‘T’}};
```

注意：

（1）数组长度应为整型常量表达式，不允许用变量进行动态定义；

（2）同一函数中，数组名不允许与变量名相同；

（3）数组名是常量，代表数组在内存中的首地址

（4）初始化时初值个数应小于等于数组长度；

（5）不允许对数组进行整体赋值

```
int   x[10]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, y[10];
y=x;                                          ×
```

（6）对于字符数组，可以用字符常量进行初始化：

char z[8]={‘E’, ‘C’, ‘U’, ‘S’, ‘T’, ‘ ’, ‘C’, ‘S’};

也可以用字符串常量进行初始化：

char z1[9]={“ECUST CS”};

或 char z2[9]=“ECUST CS”;

或 char z3[ ]=“ECUST CS”;

// 可测量数组所占内存大小 sizeof(z3) 9

字符串结束标志：‘\0’

// ASCII为0，空操作符，仅用于标志字符串的结束

char str[3][10]={“Beijing”, “Shanghai”, “Nanjing”};

5.3 数组的数据处理

- 对数组的任何操作，即“**数组操作**”都是通过元素实现的
  - ✓ 赋值、运算
  - ✓ 输入、输出
  - ✓ 查找、排序
  - ✓ 字符串处理

数组名[下标]

数组名[下标][下标]

数组名[下标][下标]...[下标]

C语言中的数组操作

- 赋值、运算、输出

```
#define N 20
 void main( )
{
  int i, num[N]={1,1};
  for(i=2; i<N; i++)
     num[i]=num[i-1]+num[i-2];
  for(i=0; i<N; i++)
     printf("%5d", num[i]);      //斐波那契数列
 }
```

```
    1    1    2    3    5    8   13   21   34   55   89  144  233  377  610  987 1597 2584 4181 6765
```

- 输入、查找

```
#include<stdio.h>
#define N 10                          //从键盘上输入10个数，输出其
 void main( ) {                       //中的最大值、最小值和平均值
   int a[N], i, max, min;
  float sum=0;
  for(i=0; i<N; i++)   scanf("%d", &a[i]);
  sum=max=min=a[0];
  for(i=1; i<N; i++)
  {
     if(a[i]>max)  max=a[i];
     else if(a[i]<min) min=a[i];
     sum+=a[i];
  }
 printf("Max=%d, Min=%d, Average=%.2f\n", max, min, sum/N);
}
```

```
45 89 96 100 -345 999 0 123 456 -234
 Max=999, Min=-345, Average=132.90
```

- 插入x到第i个位置
  - 思路：将第i个到最后的数组元素向后移一位，将新数据元素插入到第i个位置；后移操作必须从最后一个数组元素开始

1.循环j从n-1到i, step:-1
  a[j+1]=a[j]
2. a[i]=x
3. n++

j指向需要移位的数组元素。
思考：如果j指向移位后的存放位置，如何修改算法

```
for(j=n-1; j>=i; j--)   a[j+1]=a[j];
a[i]=x;
n++;
```

- 删除第i个数组元素
  - 思路：将第i+1个到第n-1个数组元素前移一位

循环j从i+1 到n-1
  a[j-1]=a[j]
n=n-1

```
for(j=i+1; j<n; j++)
    a[j-1]=a[j];
n--;
```

- 字符串的处理

（1）字符/字符串的输入/出

功能	输入函数	输出函数
输入/出一个字符	getchar( )	putchar( )
格式化输入/出函数	scanf( )	printf( )
非格式化输入/出函数	gets( )	puts( )
基于文件的输入/出函数	fgets( )	fputs( )

- 说明：

```
char str[10];
scanf(“%s”, str);   // 该函数不会以整行形式读入，
        // 而是一个一个地从缓冲区中读取字符，但
        // 不会读取空白字符（空格、制表符等）
printf(“%s ”, str); // 逐个输出str中的字符，直
        // 到遇到‘\0’；若无‘\0’，则会继续向后将内
        // 存中的字符输出，直到遇到‘\0’
```

```
gets(str);   // 该函数将空白当作合法字符一起读入，
             // 直到遇到一个换行符为止。函数执行
             // 成功，将返回一个字符数组首地址
puts(str);  // 输出str中的字符串，并在其后添加一
            // 个换行符。函数执行成功，返回一个
            // 非负值（0），否则返回EOF（-1）
```

（2）字符串处理函数

一般形式	功能说明	返回值
strlen(字符串)	求字符串的长度	有效字符的个数
strcpy(字符数组1, 字符串2)	将字符串2复制到字符数组1中	字符数组1的首地址
strcmp(字符串1, 字符串2)	比较两个字符串	字符串1= =字符串2，返回0 字符串1>字符串2，返回正整数 字符串1<字符串2，返回负整数
strcat(字符数组1,字符数组2)	连接两个字符串	字符数组1的首地址
strchr(字符串, 字符)	在字符串中查找字符	找到，返回字符第1次出现的位置；未找到，返回空地址
strupr(字符串)/strlwr(字符串)	转换为大/小写	不是标准C函数，只能在VC中用

5.4 数组程序设计举例

例4： 输入10个学生的成绩，由高到底输出排好序的成绩。

- **解题思路：**
  - 排序方法：选择排序、冒泡排序、希尔排序
  - 排序规律：“升序”——从小到大
    “降序”——从大到小

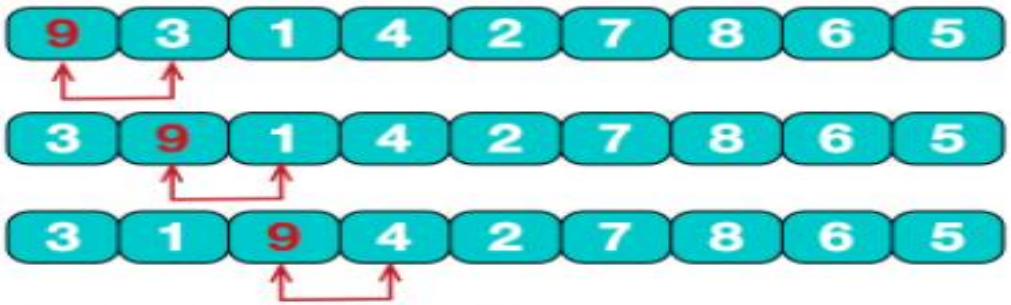
- 选择排序法：基本思想：每一轮从待排序的数据元素中选择最大（或最小）的一个元素作为首元素，直到所有元素排完为止
  - ✓ 第一轮在数组中找到最大（小）值，并将其与数组中的第一个元素互换位置；
  - ✓ 第二轮在剩下的元素中找到次大（小）值，并将其与数组中的第二个元素互换位置
  - ✓ …
- 程序用两重循环实现，外循环用*i*控制轮数，内循环用*j*控制该轮中的第*j*次比较

```
#include<stdio.h>
#define N 10
void main( )
{
   int s[N], i, j, tmp;
  printf(" Input %d scores: ", N);
  for(i=0; i<N; i++)
     scanf("%d", &s[i]);
  for(i=0; i<N-1; i++)        // 第i轮
     for(j=i+1; j<N; j++)    // 第i轮中第j次比较
        if(s[j]>s[i])  { tmp=s[i];  s[i]=s[j]; s[j]=tmp; }
  printf(" The sorted scores:\n");
  for(i=0; i<N; i++)
     printf("%4d", s[i]);
  printf("\n");
}
```

```
Input 10 scores: 78 65 89 84 100 45 89 95 23 99
The sorted scores:
100  99  95  89  89  84  78  65  45  23
```

- **冒泡排序法：基本思想**：对相邻的元素进行两两比较，不符合要求则进行交换，这样，每一趟会将最小或最大的元素“浮”到顶端，最终达到完全有序
- 程序用两重循环实现，外循环用*i*控制轮数，内循环用*j*控制该轮中第几次比较

相邻元素两两比较，反序则交换



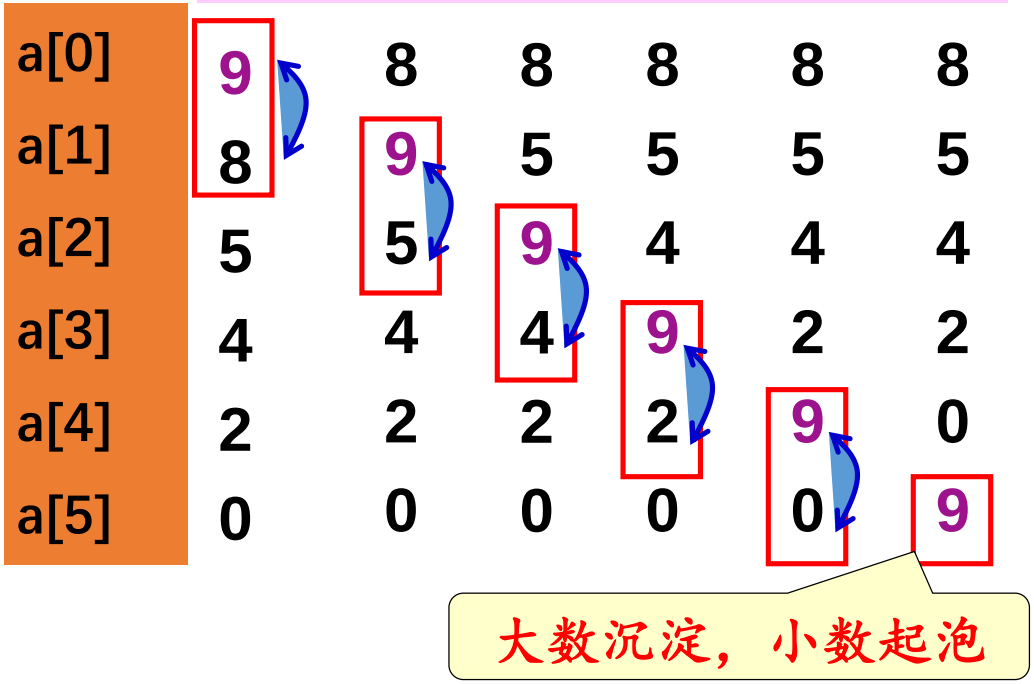
第一轮完毕，将最大元素9浮到数组顶端

同理,第二轮将第二大元素8浮到数组顶端

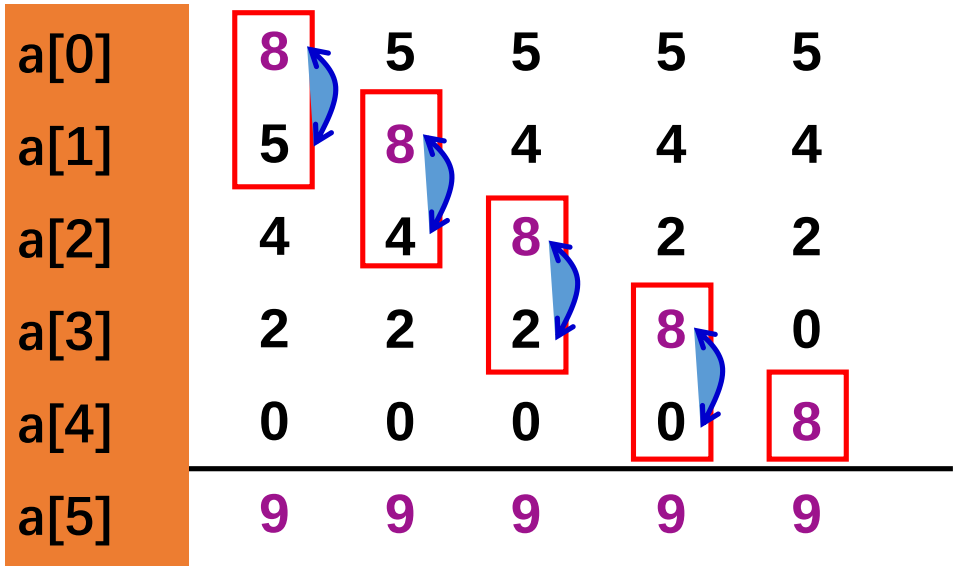
排序完成

示例：

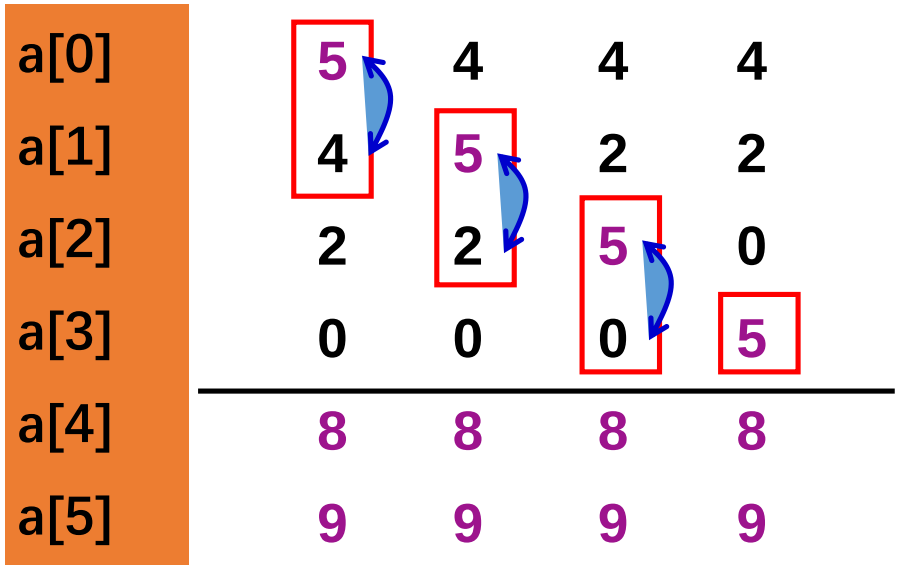
```
for(j=0;j<5;j++)
    if (a[j]>a[j+1])
    { t=a[j];a[j]=a[j+1];a[j+1]=t; }
```



```
for(j=0;j<4;j++)
    if (a[j]>a[j+1])
    { t=a[j];a[j]=a[j+1];a[j+1]=t; }
```

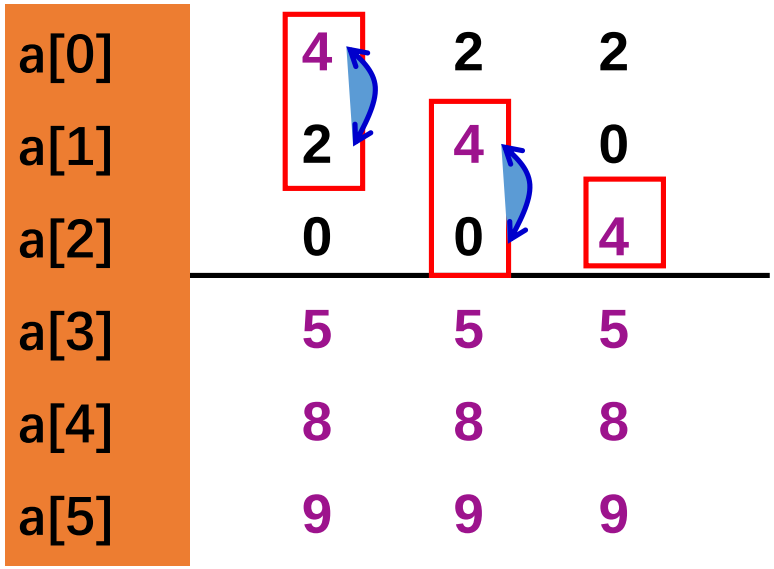


```
for(j=0;j<3;j++)
    if (a[j]>a[j+1])
    { t=a[j];a[j]=a[j+1];a[j+1]=t; }
```



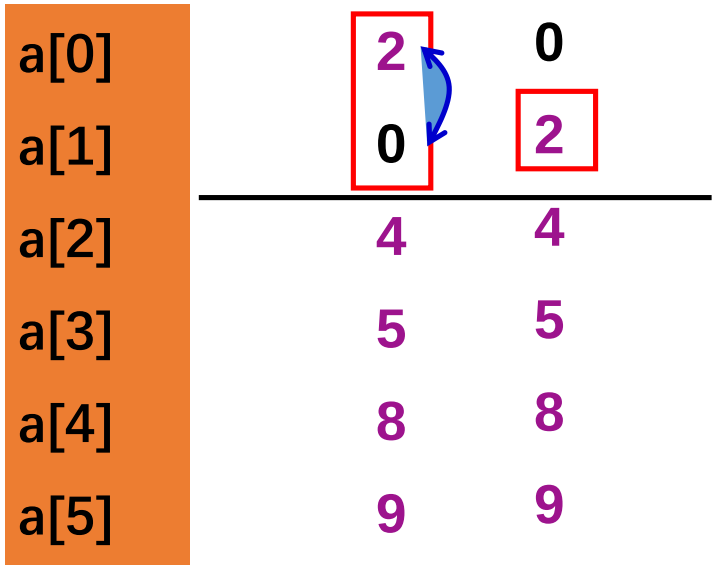
41

```
for(j=0;j<2;j++)
    if (a[j]>a[j+1])
    { t=a[j];a[j]=a[j+1];a[j+1]=t; }
```



42

```
for(j=0;j<1;j++)
    if (a[j]>a[j+1])
    { t=a[j];a[j]=a[j+1];a[j+1]=t; }
```



```
for(j=0;j<5;j++)
    if (a[j]>a[j+1])
    { ……}
```

```
for(j=0;j<4;j++)
    if (a[j]>a[j+1])
    { ……}
```

……

```
for(j=0;j<1;j++)
    if (a[j]>a[j+1])
    { ……}
```

```
for(i=0;i<5;i++)
    for(j=0;j<5-i;j++)
        if (a[j]>a[j+1])
        { ……}
```

```
#include<stdio.h>
#define N 10
void main( )
{
   int s[N], i, j, tmp;
  printf(" Input %d scores: ", N);
  for(i=0; i<N; i++)
     scanf("%d", &s[i]);
  for(i=0; i<N-1; i++)        // 第i轮
     for(j=0; j<N-1-i; j++)   // 第i轮中第j次比较
        if(s[j+1]>s[j])  { tmp=s[j]; s[j]=s[j+1]; s[j+1]=tmp; }
  printf(" The sorted scores:\n");
  for(i=0; i<N; i++)
     printf("%4d", s[i]);
  printf("\n");
}
```

```
Input 10 scores: 78 65 89 84 100 45 89 95 23 99
The sorted scores:
100  99  95  89  89  84  78  65  45  23
```

例5： 求解矩阵$C_{m\times m}=A_{m\times n}\times B_{n\times m}$。

- **解题思路：**
  - 当A的列数等于B的行数时，A与B可以相乘
  - C的行数等于A的行数，列数等于B的列数
  - C的第m行第n列的元素等于A的第m行的元素与B的第n列对应元素乘积之和

$$\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix}\begin{pmatrix}1 & 4\\ 2 & 5\\ 3 & 6\end{pmatrix}=\begin{pmatrix}1\times1+2\times2+3\times3 & 1\times4+2\times5+3\times6\\ 4\times1+5\times2+6\times3 & 4\times4+5\times5+6\times6\end{pmatrix}=\begin{pmatrix}14 & 32\\ 32 & 77\end{pmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \begin{bmatrix} 12 & 11 & 10 \\ 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

```
60      50      40
180     154     128
300     258     216
```

```
#include<stdio.h>
#define M 3
#define N 4
void main( ) {
  int i, j, k;
  int A[M][N]={{1,2,3,4},{5,6,7,8},{9,10,11,12}};
  int B[N][M]={{12,11,10},{9,8,7},{6,5,4},{3,2,1}};
  int C[M][M]={0};
  for(i=0; i<M; i++)
    for(j=0; j<M; j++)
      for(k=0; k<N; k++)
        C[i][j]+=A[i][k]*B[k][j];
  for(i=0; i<M; i++)
  {
    for(j=0; j<M; j++)
      printf(" %-5d",  C[i][j]);
    printf("\n");
  }
}
```

例6： ACM-ICPC共有10支队伍参赛，需要有一支队伍发言，经决定由按字典序排在最前面的队伍发言，请编程实现。

- **解题思路：**
  - 使用二维数组存储10支队伍的英文名称
  - 使用一维数组存储发言队伍名称
  - 使用系统函数进行字符串的处理

```
#include<stdio.h>
#include<string.h>
void main()
{
  int i;
  char team[10][20], Firteam[20];

  for(i=0; i<10; i++)
      gets(team[i]);
  strcpy(Firteam, team[0]);
  for(i=1; i<10; i++)
      if(strcmp(Firteam, team[i])>0)
         strcpy(Firteam, team[i]);

  printf(" The give speech team is %s\n", Firteam);
}
```

```
EcustCS
Redbull
Flybird
Dongfeng
Bluecoat
NorthEarth
EcnuSoft
CrossRed
T-3d
Greengun
The give speech team is  Bluecoat
```

Homework

- **实践**

  实验3 数组程序设计

- **作业**

  《教材》：P168-169　3、5、9、10