第七章 数组

课程名称：C语言程序设计

课型与教法：讲授，经过程序扩展，进行对比学习

课时：2课时

讲课题目：

第7章 数组

基本教材：C语言程序设计（高等教育出版社）廖雷主编

教学目的与要求：经过本章的学习使学生了解数组的意义和基本概念，掌握数组的定义和元素的引用，掌握数组在实际问题处理中的应用。

教学难点：二维数组概念的了解

教学要点：数组的定义和元素的引用、数据的排序

第七章　数组

数组概念

一维数组

二维数组及多维数组

字符数组和字符串

数组应用举例

数组概念

- 构造数据类型之一
- 数组:同类型数据的有序集合，即数组由若干数组元素构成，用数组名标识
- 元素:属同一数据类型，先后顺序拟定，用数组名和下标标识

例
```
double a[10];
char b[3][4];
int c[3][3][4];
 a[2]=12.3;
 b[2][1]='A';
 c[1][1][1]=78;
 printf("%lf\n",a[2]);
```

一维数组

★ 一维数组的阐明

❖ 一般形式：

<类型阐明符> <数组名>[<常量体现式>];

或

<存储类型> <类型阐明符> <数组名>[<常量体现式>];

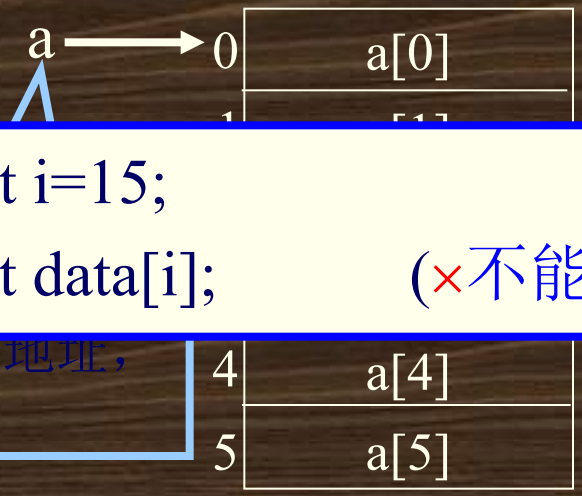
❖ C语言中不允许动态定义数组，即阐明数组时数组长度体现式不能具有变量。

例 int a[10];

[ ] :数组运算符
单目运算符
优先级(15)
左结合
不能用( )

正当标识符

指明数组的大小,也称数组的长度.
是整型值



例 int i=15;
int data[i]; (×不能用变量阐明元素个数)

数组名表达内存首地址，是地址常量

sizeof(元素数据类型)

一维数组

★ 一维数组的引用

- ❖ 数组必须先阐明，后使用
- ❖ 一般形式:

  <数组名>[<下标体现式>]

  其中：1、下标体现式表达元素在数组中的顺序号

  2、必须是整型常量、整型变量或整型体现式

  3、元素下标总是从0开始

  4、下标体现式的有效范围是从0到元素个数-1

- ❖经过对数组元素的引用，数组元素能够像一般变量一样进行操
- ❖C语言

例 double a[10]; int i=2;

例 int da
data[5

例 输出数组中的每一种元素

```
int a[10];
printf("%d",a);                    (×)
for(j=0;j<10;j++)
    printf("%d\t",a[j]);           (√)
```

必须

一维数组

★ 一维数组的初始化

❖ 初始化方式

例： int a[5]={1,2,3,4,5};
等价于：a[0]=1; a[1]=2; a[2]=3; a[3]=4; a[4]=5;

❖ 阐明：

- 对自动(auto)数组不初始化，其元素值为随机数
- 对static和外部数组不初始化，系统会自动赋以0值
- 能够只给部分数组元素赋初值

例： static int a[5];
等价于：a[0]=0; a[1]=0; a[2]=0; a[3]=0; a[4]=0;

int a[]={1,2,3,4,5,6};
编译系统根据初值个数拟定数组元素个数

如 int a[4]={0,2,,5}; (×)

例：求数组元素中的最大和最小值

```
/*ch7_3.c*/
#include <stdio.h>
main()
{  int term[10];
   int i,max,min;
   printf("请输入10个整数：");
   for(i=0;i<10;i++)
      scanf("%d",&term[i]);
   max=term[0];
   min=term[0];
   for(i=1;i<10;i++)
    {  if(term[i]<min)
          min=term[i];
       if(term[i]>max)
          max=term[i];
    }
   printf("最大数为%d",max);
   printf("最小数为%d",min);
}
```

数组阐明

数组元素引用

过程：

0	1	2	3	4	5	6	7	8	9
3	55	6	4	45	10	-2	8	12	30
3	55	6	4	45	10	-2	8	12	30
3	55	6	4	45	10	-2	8	12	30
3	55	6	4	45	10	-2	8	12	30
3	55	6	4	45	10	-2	8	12	30
3	55	6	4	45	10	-2	8	12	30
3	55	6	4	45	10	-2	8	12	30
3	55	6	4	45	10	-2	8	12	30
3	55	6	4	45	10	-2	8	12	30
3	55	6	4	45	10	-2	8	12	30

成果： max= 55

min= -2

例7.5：线性查找

思绪：从数组table[10]的第一种元素开始，依次将要查找的数x和数组中元素比较，直到找到该数或找遍整个数组为止。

```
/*ch7_5.c*/
#include <stdio.h>
main()
{  int table[10]={0,2,4,6,8,10,12,14,16,18};
   int x,i,find=0;
   printf("please input data: ");
   scanf("%d",&x);
   for(i=0;i<10;i++)
     if(x==table[i])
      { find=1;
         break;
      }
   if(find==1)
     printf("%d in talbe[%d].\n",x,i);
   else
    printf("don't find %d\n",x);
}
```

例7.6 冒泡排序

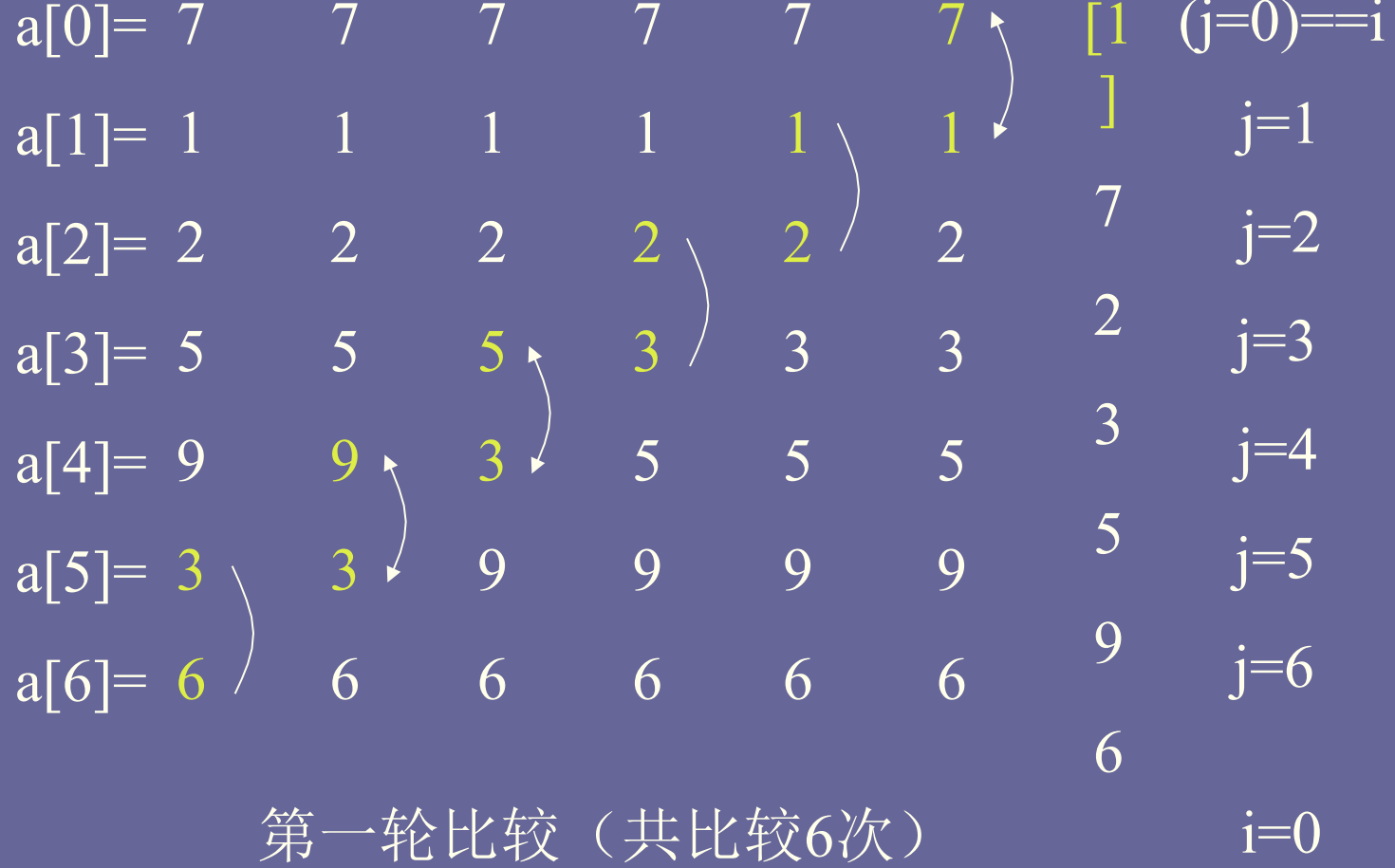
- 排序一般分为两种：
  - 升序排序：元素从小到大排列
  - 降序排序：元素从大到小排列
- 冒泡排序又称为互换排序，排序过程(升序为例)：

  1、从最终一种元素开始，两两相邻元素进行比较，将较小的元素互换到前面，直到把最小元素互换到最前面为止——第一趟冒泡排序，成果最小的数被安顿在最前面一种元素位置上

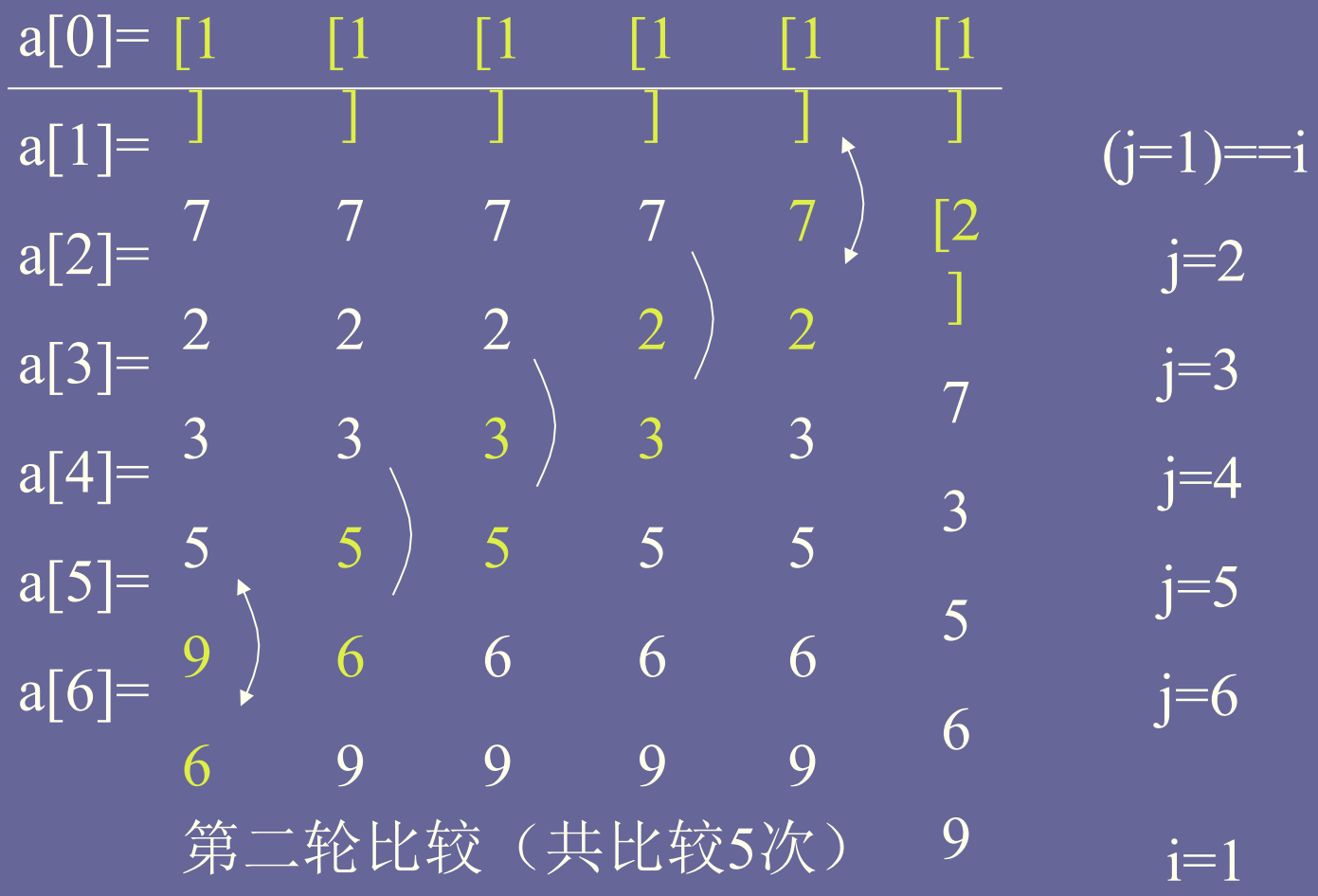
  2、对后n-1个数进行第二趟冒泡排序，成果使次小的数被安顿在第2个元素位置

  3、反复上述过程，共经过n-1趟冒泡排序后，排序结束

例：采用冒泡法对七个数排序



例：采用冒泡法对七个数排序

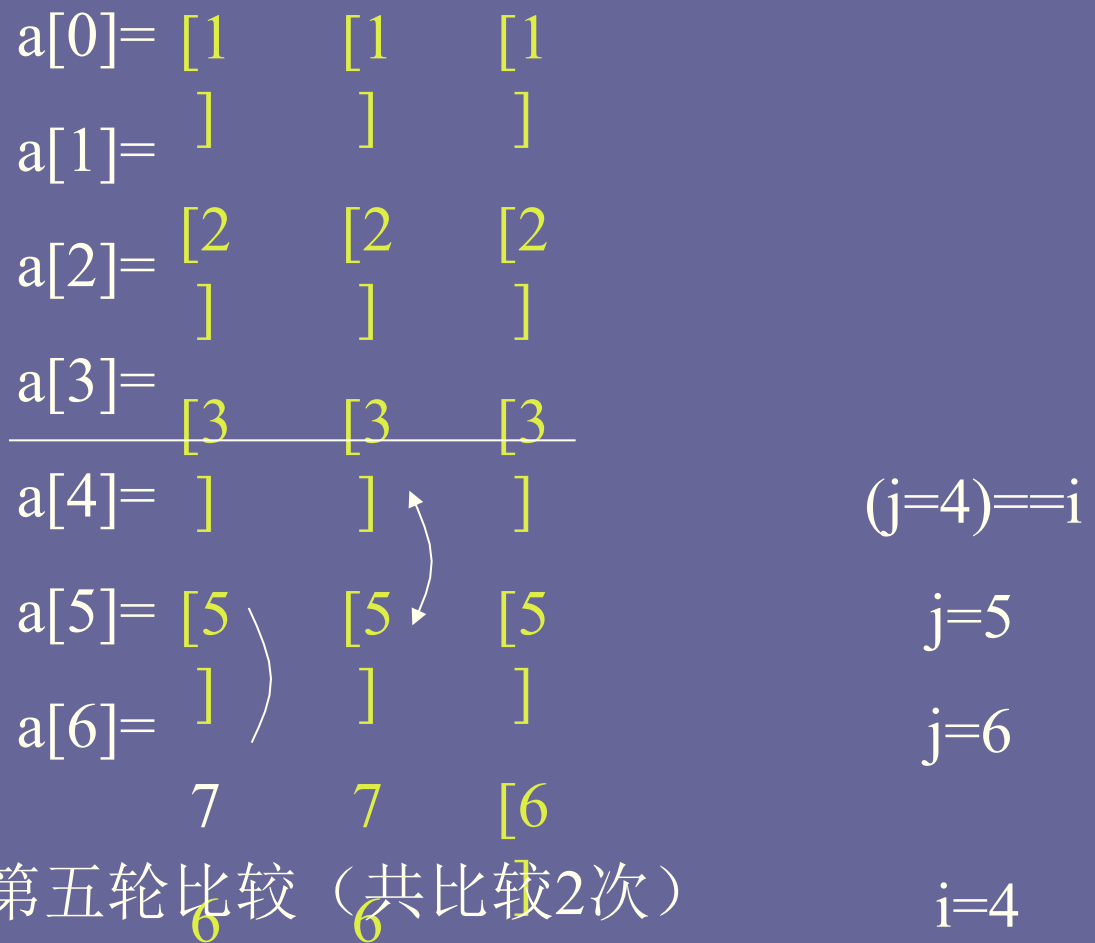


例：采用冒泡法对七个数排序

例：采用冒泡法对七个数排序

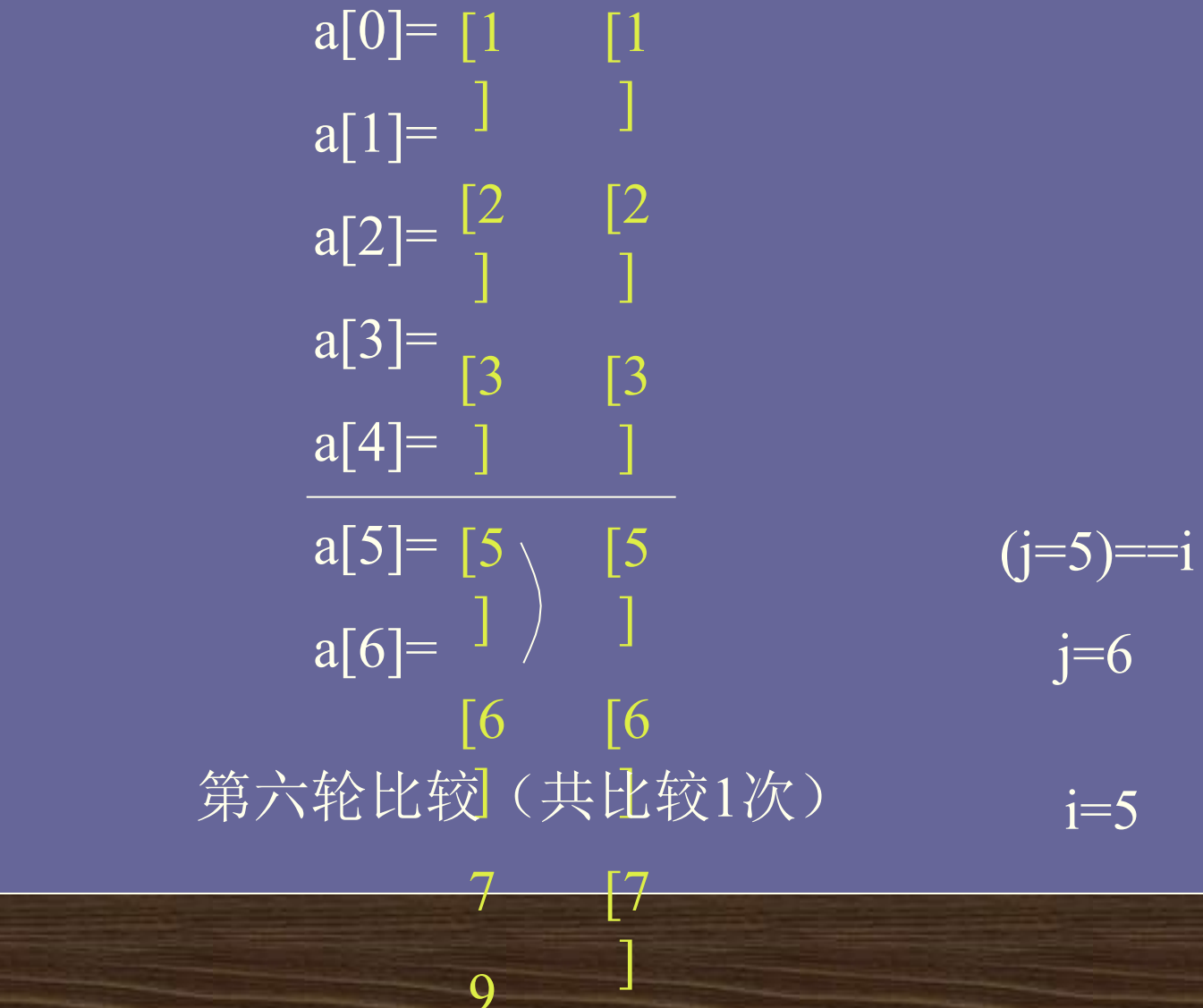
第四轮比较（共比较3次）

例：采用冒泡法对七个数排序



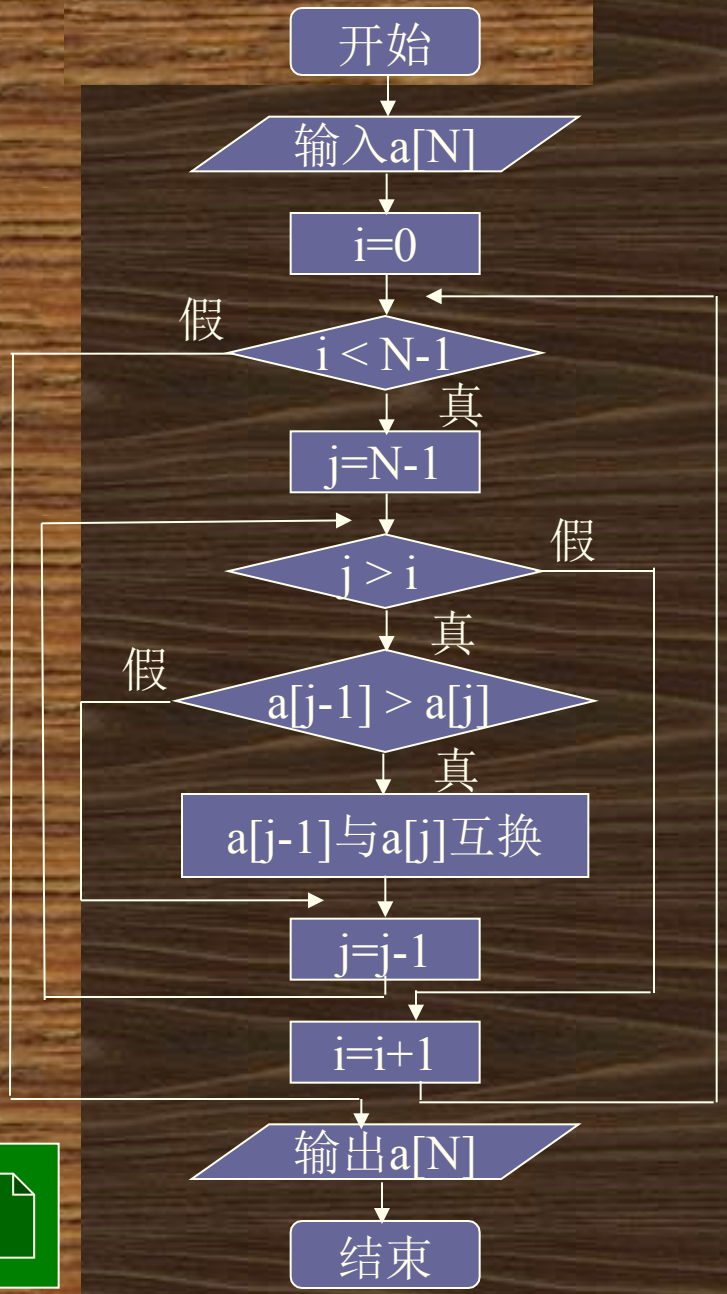
第五轮比较（共比较2次）

例：采用冒泡法对七个数排序



第六轮比较（共比较1次）

例 冒泡排序法对N个元素进行升序排序




```
/*ch7_10.c*/
#include  <stdio.h>
#define N 10
main()
{  int i,j,k;
   int a[N]={7,3,2,5,9,1,6,10,4,8};
   for(i=0;i<N-1;i++)
      for(j=N-1;j>i;j--)
         if(a[j-1]>a[j])
          {  k=a[j-1];
             a[j-1]=a[j];
             a[j]=k;
          }
   printf("\n");
   for(i=0;i<N;i++)
     printf("%d",a[i]);
}
```

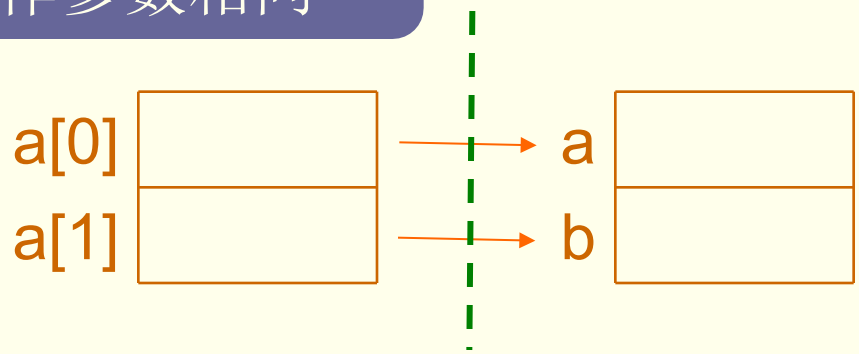
数组作为函数参数

- 数组元素作函数实参——值传递

例:在main函数中有下列调用

```
ave=fun1(    a[0],   a[1]);      /*主函数中的调用函数语句*/
  ....
float fun1(float a,float b)      /*定义一种fun1函数*/
{ float sum.aver;
  sum=a+b;
  aver=sum/2.0;
  return(aver);
}
```

使用方法与变量作参数相同



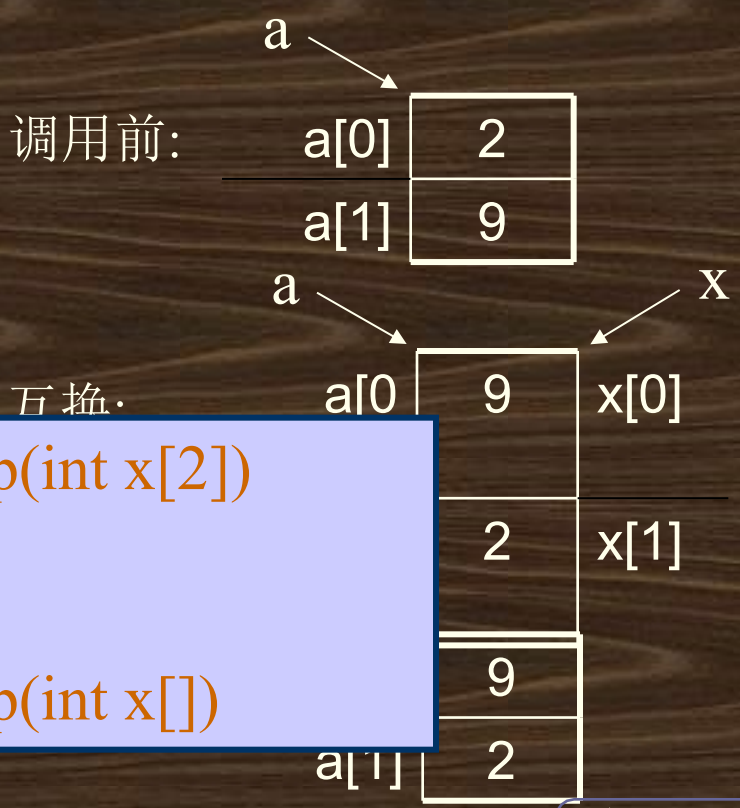
数组作为函数参数

即把实参数组的起始地址传给形参数组，形参数组和实参数组共占用一段内存单元

- 数组名作函数实参
  - 在主调函数与被调函数分别定义数组，且类型应一致
  - 形参数组大小(多维数组第一维)可不指定
  - 形参数组名是地址变量

```
例:
main()
{ void swap(int x[2]);
  int a[2]={2,9};
  swap(a);

printf("%d,%d\n",a[0
;
}
void swap(int x[2])
```



void swap(int x[2])

可写成:

void swap(int x[])

二维数组

★二维数组的阐明

❖一般形式：

<类型标识符> <数组名>[<常量体现式>][<常量体现式>];

行数　列数

❖数组元素的存储顺序

●原因

●二维

例：

例 int a[3][4];

float b[2][5];

int a[3,4];

元素个数=行数*列数

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

a[0][0]	a[0][1]
a[1][0]	a[1][1]
a[2][0]	a[2][1]

★二维数组了解

❖ 二维数组可看成是一种特殊的一维数组，该一维数组的每个数据元素也是一种一维数组

二维数组a可看成由3个元素构成的一维数组

例 int a[3][4];

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

每个元素a[i]又是一种包括4个元素的一维数组

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

★二维数组元素的引用

形式： <数组名>[<常量体现式>][<常量体现式>]

★二维数组元素的初始化

- 分行初始化
- 按元素排列顺序初始化

第一维长度省略初始化

第一维长度省略初始化

例 static int a[][3]={1,2,3,4,5};

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