千锋嵌入式学院C语言培训

数组专题

提纲

1.一维数组定义
2.一维数组用法和注意事项
3.一维数组总结
4.二维数组定义
5.二维数组用法和注意事项

数组的引进

- 数组是相同类型变量的有序集合。
- 一个数组里集合了多个数据对象，这些数据对象被称为数组中的元素。
- 数组中的每一个元素都属于同一种数据类型，它们用一个统一的数组名称和下标（元素在数组中的位置）来唯一确定。
- 数组也是变量。需要先定义后引用。

数组举例:

```
int  a[6];
```

- 上面定义了一个整型数组。
  - 数组名称为a
  - 数组中有6个整型变量
  - 数组中的6个基本元素为:
    a[0]，a[1]，a[2]，a[3]，a[4]，a[5]。
  - 上面方括号中的0～5是数组下标，表明各个元素在数组中的位置。

定义一维数组时，说明几点：

1. “数据类型名”可以是任意的数据类型。
2. “数组名”必须是合法的标识符。
3. “常量表达式”表示数组长度（即数组中的元素个数），一定要用方括号括起来，且里面不允许包含变量。
4. 一维数组中每个元素只有一个下标，并且第一个元素的下标为“0”，最后一个元素的下标为“数组长度减1”。

用法和常见错误

- 错误：数组相互赋值

```
int a[5] = { 4, 3, 2, 1 }; // 定义数组a
int b[5] = a; // 错误，把数组a赋值给数组b
```

一维数组的引用

- 引用格式：**数组名[下标表达式]**
- 说明：
  - “下标表达式”的值必须是一个整型的量。
  - 对于所引用的数组元素，使用方法和前面学过的简单数据类型变量使用方法相同，包括取地址等运算。
  - C语言没有语法结构能同时引用数组的所有元素，只能一个一个的引用每一个数组元素。

从键盘输入5个整数，将它们反序输出。

```
int main(void)
{
    int i;
    int nums[5];
    for(i=0;i<5;i++)
        scanf("%d",&nums[i]);
    for(i=4;i>=0;i—)
        printf("%5d",nums[i]);
    return 0;
}
```

一维数组的初始化

初始化，即赋初值。C语言允许定义数组时直接对数组进行初始化。

一般形式：

类型说明符 数组名[常量表达式] = {数值表};

举例：

```
float var[8]={0.1, 1.2, 2.6, 3.5,
                    4.3, 5.5, 3.5, 5.5};
int   a[5]={2, 3, 4, 5, 6};
```

初始化时，各元素的值要顺序放在一对花括号里面，各元素值之间用逗号间隔。

说明

当花括号中的数值表中的数据个数少于数组定义中的元素个数时，C语言将这些数据方别赋给数组的前几个元素，其余数组元素自动被初始化为0。

当对数组元素赋初值时，可省略数组长度。

例如:

```
float var[]={0.1, 1.2, 2.6, 3.5, s
                    4.3, 5.5, 3.5, 5.5};
int   a[]={2, 3, 4, 5, 6};
```

系统将认为数组的元素个数就是后面初始化时所提供的数据的个数。

为数组中若干元素赋相同初值时，要注意不能随意简化。

例如:

```
float var[8]={ 0.1, 0.1, 0.1, 0.1, 0.1,
                                    5.5, 3.5, 5.5};
```

不能简化为:

```
float var[8]={ 0.1*5, 5.5, 3.5, 5.5};
```

否则，系统将认为数组中的各元素的值分别为:

var[0]=0.1*5 , var[1]=5.5 , var[2]=3.5 , var[3]=5.5 , var[4]=0 , …… , var[7]=0。

关于排序

程序设计中的常见算法

- 选择法
- 冒泡法
- ……

用选择法对10个整数按照从小到大的顺序排列

从所有的数中找出最小的一个，将其放在最前面；接着在余下的数中找出最小的一个，将其放在第二位，依次类推，数列由前往后逐渐成型。

下面以6个数(8、6、5、4、3、1)为例，用图示说明。

选择法第一轮：先找出序列中最小的一个。

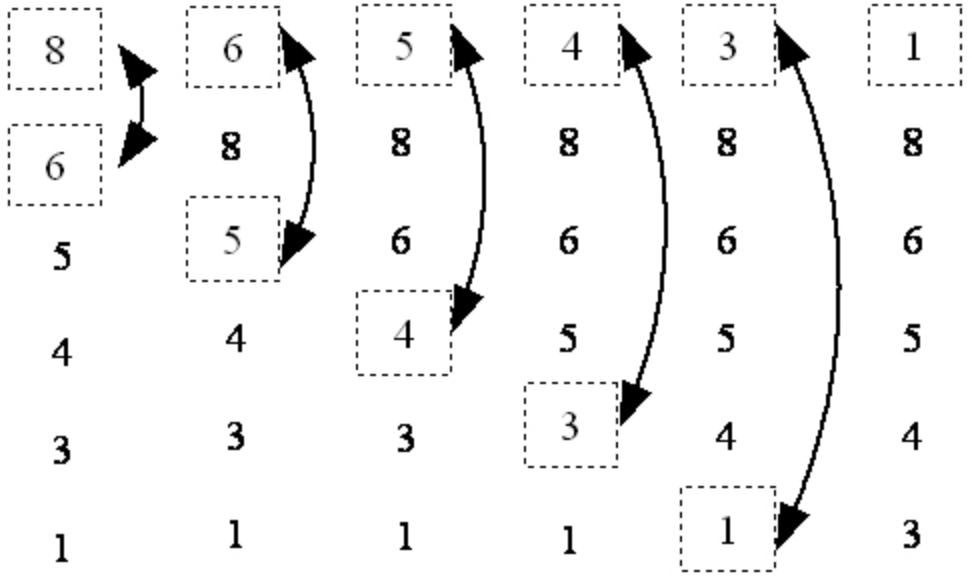


图 7.1 选择法第一轮

选择法第二轮：找出余下序列中最小的一个。

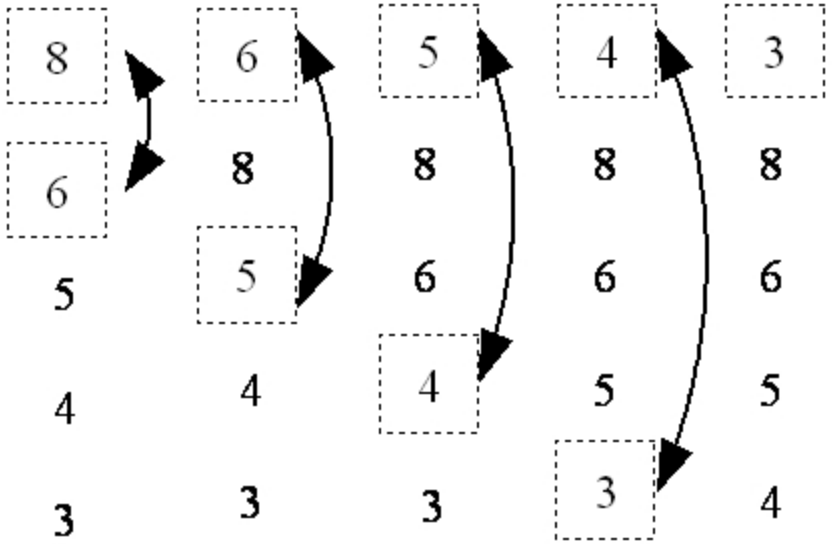


图 7.2　选 择 法第 二轮

```
int main(void)
{
    int num[10];int i,j,temp;
    printf("Please input 10 numbers:\n");
    for(i=0;i<10;i++) scanf("%d",&num[i]);
    for(i=0;i<10;i++)
        for(j=i;j<10;j++)
            if(num[j]<num[i])
            {
                temp=num[i];
                num[i]=num[j];
                num[j]=temp;
            }
    printf("The sorted numbers:\n");
    for(i=0;i<10;i++)
        printf("%4d",num[i]);
    return 0;
}
```

用冒泡法对10个整数排序（从小到大）

思路

对相邻两个数进行比较，将较小的调到前面，两两比较一轮之后，最大的一个数被放置在最后面；接着从头开始重复执行以上操作，次大的数被放置在倒数第二位，依次类推，数列由后往前逐渐成型。

- 冒泡法的核心：小数上浮，大数下沉。
- 冒泡法第一轮：使最大的数放在最后一个位置上

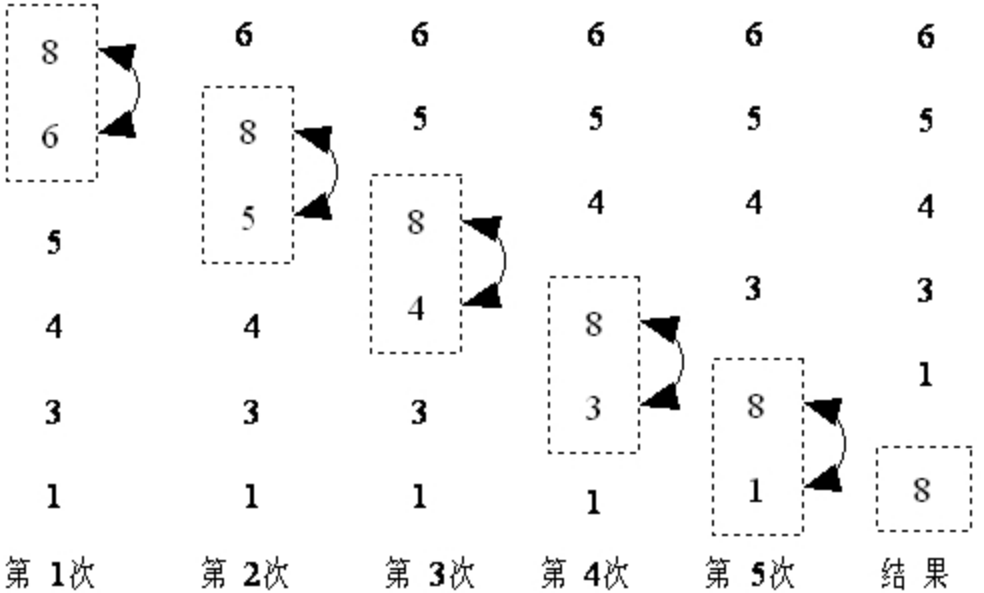


图7.3 冒泡法(第一轮比较)

```
  t main()
{
    int num[10];int i,j,temp;
    printf("Please input 10 numbers:\n");
    for(i=0;i<10;i++)   scanf("%d",&num[i]);
    for(i=0;i<10;i++)
        for(j=0;j<10-i;j++)
            if(num[j]>num[j+1])
            {
                temp=num[j];
                num[j]=num[j+1];
                num[j+1]=temp;
            }
    printf("The sorted numbers:\n");
        for(i=0;i<10;i++)
            printf("%4d",num[i]);
    return 0;
}
```

二维数组的定义

- 二维数组定义的一般形式:

  数据类型名 数组名[常量表达式1][常量表达式2];

- 例如:

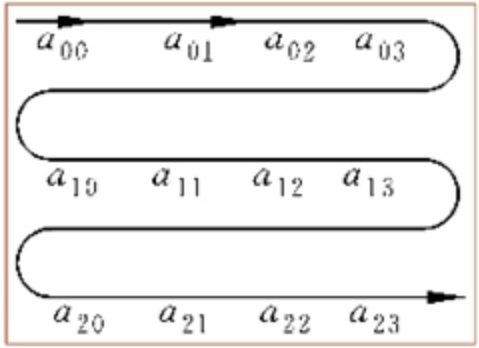
```
double    a[3][3], b[2][2];
int       f[2][3];
```

- 二维数组可以理解为特殊的一维数组，如 int a[3][4],可以理解为数组a有3个元素，但每个元素又是一个一维数组。

二维数组

- 二维数组在内存中的存放，存放原则是“按行存放”，即先存放第一行再依次存放第二行…

下图表示对a［3］［4］数组存放的顺序



二维数组的初始化

方法1: 直接赋初值。适用于数据较少时。

- 将数组元素的所有初始值都写在一个花括号内，由编译系统按照数组元素在内存中存放的顺序对各元素赋初值。
- 例如:

```
int num[3][4]={11, 2, 23, 4, 51, 6, 7, 8, 92, 10, 0, 12};
```

方法2：分行为二维数组赋值。适用于每一行的初始值个数少于每一行中的数组元素个数时。

例如：

```
int num[3][4]={{11, 2},
               {51, 6, 7},
               {92, 10, 0, 12}};
```

每一行中后面的剩余元素被自动赋值为0 ！

二维数组

- **二维数组的初始化**
  可以写在一个大括号中集中初始化：
  例如： int a[3][2] = { 0, 1, 2, 3, 4, 5};
  也可以部分初始化：
  如： int a[3][4] = {{1}, {5}, {9}};

1	0	0	0
5	0	0	0
9	**0**	**0**	**0**

int a[3][4] = {{1}, {0, 6}, {0, 0, 11}};

1	0	0	0
0	6	0	0
0	0	0	11

二维数组的长度不能随意省略，必须遵守如下规则

- 使用第一种方式，只有当为二维数组的所有元素赋初值时，所定义数组的第一维的长度才可以省略。
- 例如：

```
int num[ ][4]={11, 2, 0, 0, 51,
               6, 7, 0, 92, 10, 0, 12};
```

- 使用第二种方式，无论是为二维数组的所有元素赋初值，还是为二维数组的部分元素赋初值，定义数组时都可以省略第一维的长度。

- 无论采用上述哪一种方式为二维数组元素赋初值，在定义数组时第二维的长度都绝对不可以省略。

二维数组

- **二维数组的初始化**
  如果已经对所有元素进行了赋值，可以省略数组的行数，但不能省略列数。
  例如： int a[3][2] = {
  {0,1}, {2,3},{4,5}
  };

也可以省略行数写成int a[][2] = {...};

二维数组

- **二维数组的引用**

  格式： 数组名[下标][下标]，但注意千万不要越界访问！如：

  int a[3][4]

  可以引用a[0][3],但是不能引用a[0][4].

  可以引用a[2][3],但是不能引用a[3][3].

  行跟列都不能越界！

作业

- 用二维数组来写一个简单的描雷程序
- 用5*5的二维数组表示地图
- 随机生成有雷的位置
- 玩家从键盘输入一个坐标
- 如果这个位置上无雷，就显示出上下左右总雷数
- 获胜：玩家把非雷位置全部探测出来
- 失败：玩家触雷