



## 第6章 数组

### 任务1 一维数组的定义、引用及应用

主讲人 |

## 知识 目标



1. 一维数组的语法格式。
2. 一维数组的定义与引用。

## 技能 目标



1. 会定义与引用一维数组。
2. 会使用一维数组进行算法设计与实现。
3. 会调试和运行程序。

## 素质 目标



1. 学习过程中形成乐于探究的意识和敢于创新的精神。
2. 养成主动与他人交流合作的精神。

# 提纲

1.一维数组的定义

2.一维数组的初始化及元素引用

3.一维数组的应用实例



- 如何处理 一个班学生的学习成绩？
- 一行文字怎样存储？
- 一个矩阵怎样存储？



这些数据的特点：**具有相同的数据类型。**

为了很方便地使用这些数据，C语言提供了一种构造数据类型：**数组。**

**一组具有相同数据类型的数据的有序集合。**



定义:

一维数组用来存放多个相同类型数据组成的一个集合。

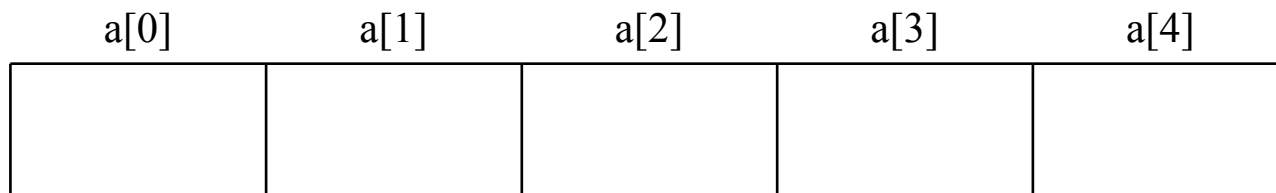
一般形式:

数据类型说明符 数组名[长度];

例如:

定义一个含有5个元素的一维数组，代码应该书写为：`int a[5];`

存储空间表示图：





### 注意说明:

- (1) 数组名的命名规则必须遵循标示符的命名规则。
- (2) 长度必须是整数，整型常量或整型常量表达式来表示，不能使用变量或含有变量的表达式，长度必须用方括号括起来。
- (3) 数据类型说明符可以是基本类型，也可以是构造类型。
- (4) 数组中的元素用数组名和下标相结合来区分，下标是从0开始。
- (5) 数组名等价于数组的首地址，也就是数组中第一个元素的地址。即a与&a[0]等价。

例如:

`int b[3+4];`是合法的。

`float c[n]; double d[n+1];` 是非法的。

`a[2]`是数组中的第3个元素。



### 注意说明:

- (1) 数组名的命名规则必须遵循标示符的命名规则。
- (2) 长度必须是整数，整型常量或整型常量表达式来表示，不能使用变量或含有变量的表达式，长度必须用方括号括起来。
- (3) 数据类型说明符可以是基本类型，也可以是构造类型。
- (4) 数组中的元素用数组名和下标相结合来区分，下标是从0开始。
- (5) 数组名等价于数组的首地址，也就是数组中第一个元素的地址。即a与&a[0]等价。

例如:

`int b[3+4];`是合法的。

`float c[n]; double d[n+1];` 是非法的。

`a[2]`是数组中的第3个元素。



## 一维数组初始化:

(1) 在定义数组时, 赋给数组各元素的初值。

例如: `int a[3]={3, 1, 5};`

(2) 数组的长度可以省略不写。

例如: `int b[]={2, 7, 9, 4};`

(3) 数组中部分元素初始化, 则长度不能省略, 其他没有赋值的元素的初始值为0。

例如: `int c[5]={5, 2, 9}; c[3]和c[4]的初值为0。`

(4) 数组中的元素初始化时, 括号中只有一个零。

例如: `float d[4]={0}; 4个元素的初值都为0.0。`





数组引用格式:

数组名[下标]

下标为整数，从0开始，最大值为长度-1，下标要用方括号括起来。

例如:

```
int a[5]={2, 7, 9, 4, 3};
```

a[0]	a[1]	a[2]	a[3]	a[4]
2	7	<b>9</b>	4	3

如果执行语句a[2]=8，结果会变化。

### 3. 应用实例



输入10个学生的某一门功课成绩，求出这些学生该门功课的平均成绩、最高分和最低分？

算法设计如下：

- 1、定义一个浮点型数组a，长度为10；
- 2、通过scanf语句进行数组元素赋值；
- 3、将第一个元素设为最高分和最低分；
- 4、通过累加求和得到10学生的总成绩，同时通过比较得到最高分和最低分；
- 5、总分除以10就得到平均成绩；
- 6、输出平均分、最高分和最低分；

#### 语句编写

```
main()
{
    float a[10],max,min; int i;
    float sum=0,ave;
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    max=a[0];min=a[0];
    for(i=0;i<10;i++)
    {
        sum=sum+a[i]; //累加求总成绩
        if(a[i]>max) max=a[i]; //求最高分
        if(a[i]<min) min=a[i]; //求最低分
    }
    ave=sum/10; //求平均成绩
    printf("平均分%.2f,最高分%f,最低分%f.\n",ave,max,min);
}
```

#### 运行结果

```
90 80 88 75 98 67 78 95 93 85 ✓
平均分84.90,最高分98,最低分67.
```

### 3. 应用实例



对10个数组元素依次赋值，要求按逆序输出。

算法设计如下：

1. 定义一个整型数组a，长度为10；
2. 使用循环结构通过scanf语句进行数组元素赋值；
3. 使用循环结构将数组元素逆序输出。

#### 语句编写

```
main()
{
    int a[10], i;
    printf("请输入10个数组元素：");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    printf("逆序输出为：");
    for(i=9;i>=0;i--)
    {
        printf("%-5d",a[i]);
    }
}
```

#### 运行结果

```
90 80 88 75 98 67 78 95 93 85 ✓
85 93 95 78 67 98 75 88 80 90
```

### 3. 应用实例



用冒泡算法对5个数进行从小到大排列。

算法原理：

排序过程：

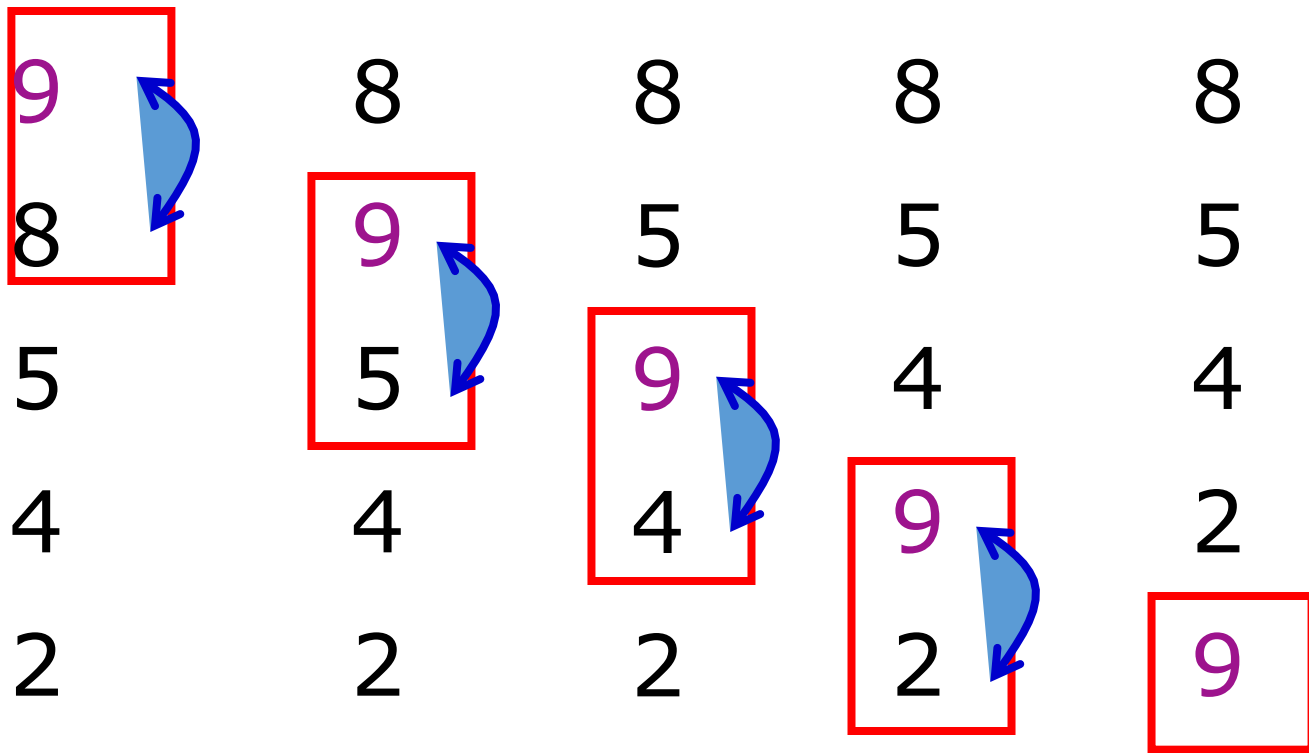
- (1) 比较第一个数与第二个数，若 $a[0] > a[1]$ ，则交换；然后比较第二个数与第三个数；依次类推，直至第4个数和第5个数比较为止——第一趟冒泡排序，结果：最大的数被安置在最后一个元素位置上。
- (2) 对前4个数进行第二趟冒泡排序，结果使次大的数被安置在第4个元素位置。
- (3) 重复上述过程，共经过4趟冒泡排序后，排序结束。

### 3. 应用实例



执行过程：第一趟

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



语句编写

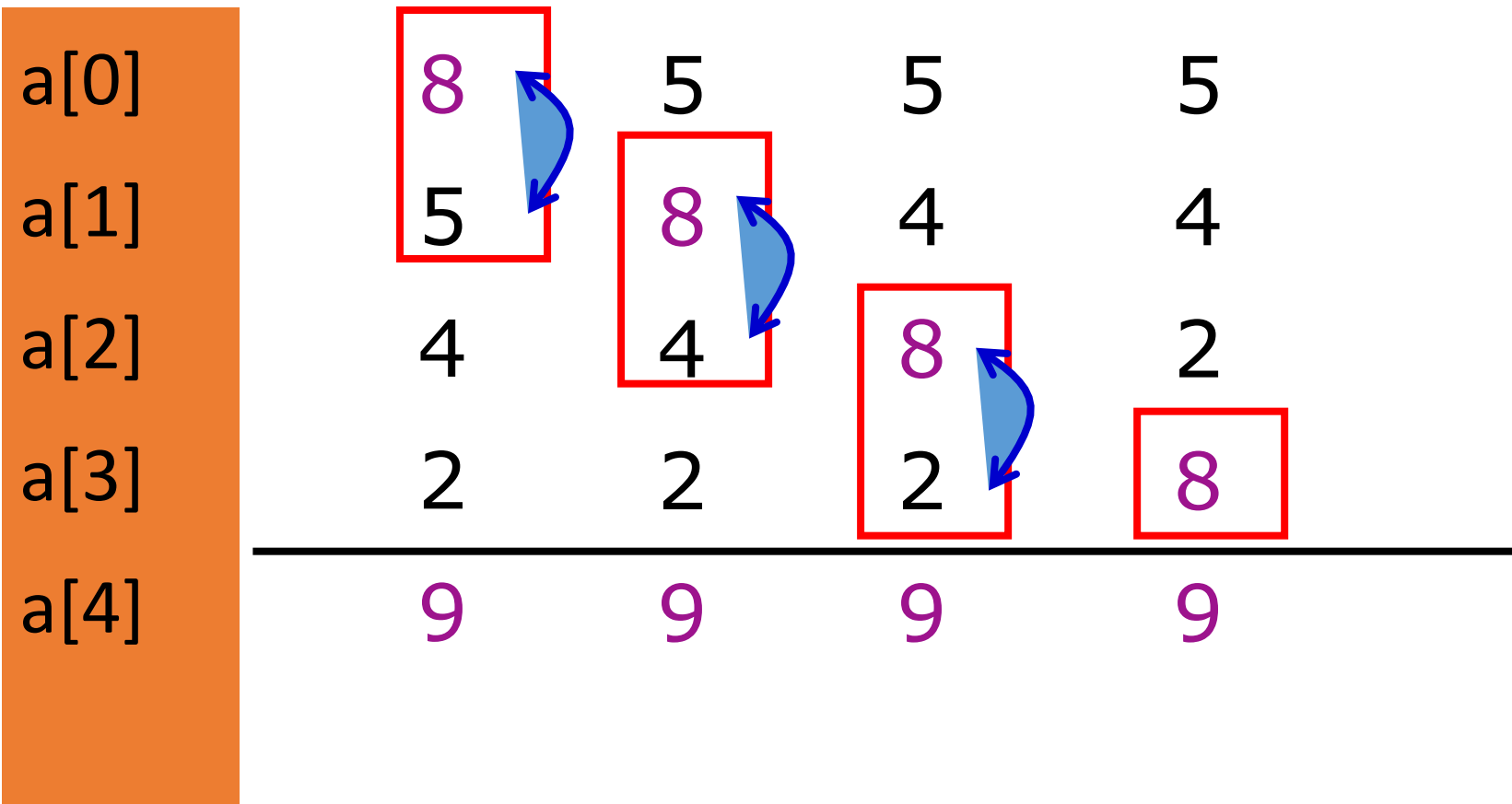
```
for(i=0;i<4;i++)  
    if (a[i]>a[i+1])  
    {  
        t=a[i];  
        a[i]=a[i+1];  
        a[i+1]=t;  
    }
```

大数沉底，小数起泡

### 3. 应用实例



执行过程：第二趟



语句编写

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

### 3. 应用实例



执行过程：第三趟

a[0]	5	4	4
a[1]	4	5	2
a[2]	2	2	5
a[3]	8	8	8
a[4]	9	9	9

The diagram illustrates the third pass of a sorting algorithm. The array elements are arranged in a grid. The first three columns represent the current state. Red boxes highlight the elements being compared: (a[0], a[1]), (a[1], a[2]), and (a[2], a[3]). Blue arrows indicate the swap of elements in each comparison. A horizontal line is drawn below the first three rows. The values 5, 4, 2, 8, 9 are shown in purple, while the values 4, 5, 2, 8, 9 are shown in black.

语句编写

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

### 3. 应用实例



执行过程：第四趟

a[0]	4	2
a[1]	2	4
a[2]	5	5
a[3]	8	8
a[4]	9	9

The diagram shows an array with two columns of values. The first column contains 4, 2, 5, 8, 9 and the second column contains 2, 4, 5, 8, 9. A red box highlights the 4 and 2 in the first row, with a blue arrow pointing from the 4 to the 2. Another red box highlights the 4 in the second row. A horizontal line is drawn below the second row.

语句编写

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



### 3. 应用实例

第一  
趟

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

第二  
趟

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

第三  
趟

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

第四  
趟

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

算法设计如下（双层循环）：

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

### 3. 应用实例



用冒泡算法对5个数进行从小到大排列。

算法设计如下：

- 1、数据的输入；
- 2、数据的输出；
- 3、每一趟比较程序设计；
- 4、趟数控制的程序设计；

#### 语句编写

```
main()
{
    int a[5]={9,8,5,4,2},i,j,t;
    for(j=0;j<4;j++)
    {
        for(i=0;i<4-j;i++)
            if(a[i]>a[i+1])
                {t=a[i];a[i]=a[i+1];a[i+1]=t;}
    }
    for(i=0;i<=4;i++)
        printf("%d ",a[i]);
}
```

#### 运行结果

2 4 5 8 9



我学会了什么？

- 1.学会定义与引用一维数组。
- 2.学会使用一维数组进行算法设计与实现。
- 3.学会调试和运行程序。



## 第6章 数组

### 任务2 字符数组的定义、引用及应用

主讲人 |

知识  
目标



1. 字符数组的语法格式。
2. 字符数组的定义与引用。

技能  
目标



1. 会定义与引用字符数组。
2. 会使用字符数组进行算法设计与实现。
3. 会调试和运行程序。

素质  
目标



1. 学习过程中形成乐于探究的意识和敢于创新的精神。
2. 养成主动与他人交流合作的精神。

# 提纲

1. 字符数组的定义、初始化及元素引用

2. 字符串定义及相关函数

3. 字符数组应用实例



一行文字怎样存储？

其数组元素的数据类型为字符型变量，关键字为char。

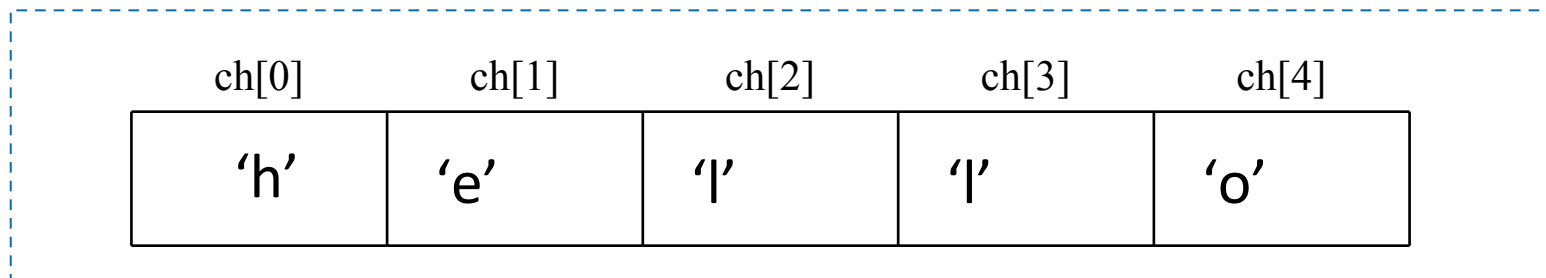
一般形式：

char 数组名[长度]；

例如：

char ch[5];该代码表示定义ch为一个字符数组，在内存中占据连续5个字节的存储单元，每个存储单元分别存放一个字符。

存储空间表示图：





定义:

其数组元素的数据类型为字符型变量，关键字为**char**。

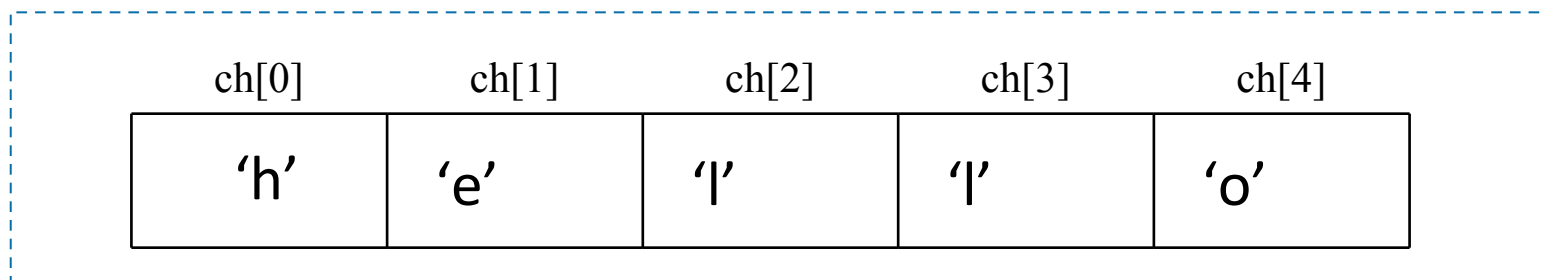
一般形式:

`char 数组名[长度];`

例如:

`char ch[5];`该代码表示定义ch为一个字符数组，在内存中占据连续5个字节的存储单元，每个存储单元分别存放一个字符。

存储空间表示图：





以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/955232104342011222>