



程序设计基础（C）——第6章数组

郑州大学软件学院/网络空间安全学院



Lecturer: 宋轩

Office : 行政楼-306

Email : songxuan@zzu.edu.cn

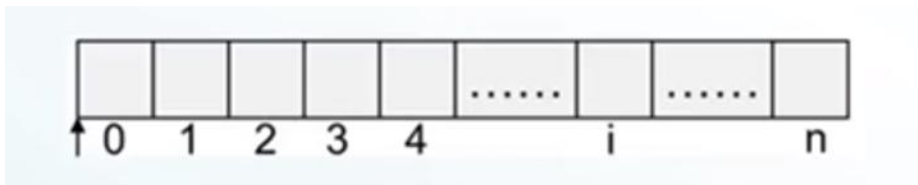
第6章 数组——数组定义

数组概述

数组是由基本数据类型按照一定的规则组成的。

数组由一系列元素构成，这些元素均属于同一数据类型。

数组是有序数据的集合，各元素在内存中是连续存放的。



数组概述

- ◆ 根据数组的组成规则，可分为
 - ◆ 一维数组、二维数组和 multidimensional 数组。
- ◆ 根据元素数据类型的不同，可以分为
 - ◆ 整型数组、实型数组、字符型数组等。
- ◆ C语言中把字符串定义为字符数组，即数组元素为字符型的数组。

一维数组定义

例如：

```
int a[20]
```

```
char c[10]
```

```
float x[15]
```

```
int d, array[]
```

定义形式：

类型说明符 数组名[数组长度]

说明

数组名是标识符，符合标识符命名规则，
以及作用域和存储类别规则

数组长度标识数组中含有元素的个数

一维数组定义

针对数组长度，下列定义是合法的

```
int a[10];
```

```
int b[10+10];
```

```
float x[15-5];
```

数组长度可以是常量，或者是由常量构成的式子，即常量表达式

一维数组的引用

- ◆ 数组必须**先定义引用**

```
int a[10];
```

- ◆ 数组元素引用形式:

数组名[下标]

如a[0], a[1], a[2], ..., a[i], a[i+1], ...

- ◆ 数组定义和数组元素引用的形式相似, 但意义不同



引用时下标允许是常
量、变量或表达式

```
int i , a[3];  
a[0]=10;  
a[1]=20;  
a[2]=30;  
for (i=0;i<=2;i++)  
    printf( "%d" , a[i])
```

```
int i , a[3];  
a[1]=10;  
a[2]=20;  
a[3]=30;  
for (i=1;i<=3;i++)  
    printf( "%d" , a[i])
```

原则上说，如果数组长度为n，则下标范围应界于0和n-1。

但C编译系统没有对下标进行越界检查。

一维数组的引用

- ◆ C语言规定不能一次引用整个数组
- ◆ 允许使用a[i] 形式访问每个元素
- ◆ 可以像使用普通变量一样使用
a[0], a[1], a[2], ..., a[9]

```
#include<stdio.h>

int main()
{
    int i,a[10];
    for (i=0;i<10;i++)
        a[i]=i;
    for (i=9;i>=0;i--)
        printf(" %d",a[i]);
    printf("\n");
    return 0;
}
```

一维数组的初始化

- ◆ 定义数组时可对数组元素赋初值。
 - ⊙ 如：`int a[3] = {0, 1, 2};`
- ◆ 如果数组没有初始化，系统会用默认值对它初始化。即外部数组或静态数组赋0值，自动数组赋随机值。
- ◆ 初始化数组时，初始值的个数可以比数组元素的个数少，未提供初始值的元素被置为0。如：

```
int a[10] = {1, 2, 3, 4, 5};
```

```
int b[10] = {0};
```



一维数组的初始化

- ◆ 如果对全部数组元素赋初始值时，可以不指定数组长度。

例如：

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

等价于

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

但不能省略[]

```
int a = {1, 2, 3, 4, 5}; //×
```

- ◆ 对数组进行初始化时，不允许初始化的元素个数比定义的数组长度大，如：

```
int a[5] = {0, 1, 2, 3, 4, 5}; //×
```



一维数组的初始化

◆ 假设下面的变量和数组均是动态存储方式，各程序片段是否等价？

片段1	片段2	等价否
<code>int a=3;</code>	<code>int a; a=3;</code>	✓
<code>int a[3]={0,0,0};</code>	<code>int a[3]={0};</code>	✓
<code>int a[3]={0};</code>	<code>int a[3];</code>	✗
<code>int a[3]={0,1,2};</code>	<code>int a[3]; a[3]={0,1,2};</code>	✗
<code>int a[3]={0,1,2};</code>	<code>int a[3]; a[0]=0;a[1]=1;a[2]=2;</code>	✓

第6章 数组——一维数组使用

例1.求Fibonacci数列前20项

◆ Fibonacci数列的定义如下：

$$f_1=1$$

$$f_2=1$$

$$f_n=f_{n-1}+f_{n-2} \quad (n \geq 3)$$



例1.求Fibonacci数列前20项

◆ 分析：

1. 定义数组 `int f[21]` ;
(不使用元素 `f[0]`)
2. 给各元素赋值，即求出数列的各项;
3. 输出

```
#include <stdio.h>
int main ()
{
    int i, f[21];
    f[1]=1;
    f[2]=1;
    for (i = 3; i<=20; i++)
        f[i] = f[i-1] + f[i-2];
    for (i=1; i<=20; i++)
    {   printf ( "%d\t", f[i]);
        if (i%5 == 0)
            printf ( "\n" );
    }
    return 0;
}
```

例2.统计分数

由键盘输入50个学生的考试成绩，统计出各分数段的人数。

以10分为一个分数段，即0~9，10~19，.....，90~99，100，共11个分数段

◆ 分析：

- ④ 定义int score,num[11]; 用于存放各分数段的人数

- ④ 重复执行

 - 读入一个成绩score;

 - 对应分数段的人数加1;

- ④ 输出各分数段人数



例2.统计分数

◆ 对应分数段的人数加1;

```
num[score/10] ++;
```

```
.....  
if(score>=0&&score<=9)  
    num[0] ++;  
if(score>=10&&score<=19)  
    num[1] ++;  
if(score>=20score<=29)  
    num[2] ++;  
.....  
  
if(score>=90score<=99)  
    num[9] ++;  
if(score==100)  
    num[10] ++;
```

```
#include <stdio.h>
int main()
{
    int i,score;
    int num[11]={0};           //数组num为各分数段人数，初始化为0
    for(i=1;i<=50;i++)
    {
        scanf("%d",&score);
        num[score/10]++;       //统计各分数段人数
    }
    printf("分数段:\t人数\n");
    for(i=0;i<10;i++)
        printf("%d~%d:\t%d\n", i*10, i*10+9, num[i]);
    printf("100:\t%d\n", num[10]);
    return 0;
}
```



例3.输入10个整数，找出其中的最大数和最小数。

中国大学MOOC

- 定义数组 `int a[10];`
- 定义两个变量 `int max, min;`
- 算法：
 - 循环输入10个整数，存放在数组中
 - `max, min`设初值
 - 循环遍历数组每个元素，找出最大和最小者

```
#include<stdio.h>
int main()
{
    int a[10], i, max, min;
    for(i=0; i<10; i++)
        scanf("%d",&a[i]);
    max = min = a[0];
    for(i=1; i<10;i++)
    {
        if (a[i] > max)
            max = a[i];
        if (a[i] < min)
            min = a[i];
    }
    printf( "max=%d\n ", max);
    printf( "min=%d\n ", min);
    return 0;
}
```

例4.用冒泡法对10个整数排序

假如规定整数从小到大是正序

冒泡排序的思想：

相邻两个数比较，一旦逆序就交换，两两比较一轮后，
确定了最大值；

然后在剩下的数中进行第二轮的两两比较，确定次大
值.....



例4.用冒泡法对10个整数排序

6	5	5	5	5
5	6	6	6	6
8	8	8	4	4
4	4	4	8	1
1	1	1	1	8



例4.用冒泡法对10个整数排序

6	5	5	5	5
5	6	6	6	6
8	8	8	4	4
4	4	4	8	1
1	1	1	1	8

5	5	5	5
6	6	4	4
4	4	6	1
1	1	1	6
8	8	8	8



例4.用冒泡法对10个整数排序

6	5	5	5	5
5	6	6	6	6
8	8	8	4	4
4	4	4	8	1
1	1	1	1	8

5	5	5	5
6	6	4	4
4	4	6	1
1	1	1	6
8	8	8	8

5	4	4
4	5	1
1	1	5
6	6	6
8	8	8



例4.用冒泡法对10个整数排序

6	5	5	5	5
5	6	6	6	6
8	8	8	4	4
4	4	4	8	1
1	1	1	1	8

5	5	5	5
6	6	4	4
4	4	6	1
1	1	1	6
8	8	8	8

5	4	4
4	5	1
1	1	5
6	6	6
8	8	8

4	1
1	4
5	5
6	6
8	8

对于5个数，共进行了4轮排序

第一轮在5个数中比较了4次，确定最大值

第二轮在4个数中比较了3次，确定次大值

第三轮在3个数中比较了2次，确定第三大者

第四轮在2个数中比较了1次，确定次小值

例4.用冒泡法对10个整数排序

对于 n 个数，共进行 $n-1$ 轮排序

第一轮在 n 个数中比较 $n-1$ 次，确定最大值

第二轮在 $n-1$ 个数中比较 $n-2$ 次，确定次大值

.....

第 i 轮在 $n-i+1$ 个数中比较 $n-i$ 次，确定第 i 大值

.....

第 $n-1$ 轮在2个数中比较1次，确定次小值



例4.用冒泡法对10个整数排序

```
#include <stdio.h>
int main()
{   int a[10], i, j, t;
    printf("input 10 numbers:\n");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    for(i=1;i<=9;i++)
        for(j=1;j<=10-i;j++)
            if(a[j-1]>a[j])
                { t=a[j-1]; a[j-1]=a[j]; a[j]=t; }
    printf("the sorted numbers:\n");
    for(i=0;i<10;i++)
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}
```

对于 n 个数，共进行 $n-1$ 轮排序

第一轮在 n 个数中比较 $n-1$ 次，确定最大值

第二轮在 $n-1$ 个数中比较 $n-2$ 次，确定次大值

.....

第 i 轮在 $n-i+1$ 个数中比较 $n-i$ 次，确定第 i 大值

.....

第 $n-1$ 轮在2个数中比较1次，确定次小值



例4.用冒泡法对10个整数排序

```
#include <stdio.h>
int main()
{   int a[10], i, j, t;
    printf("input 10 numbers:\n");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);

    for(i=1;i<=9;i++)
        for(j=1;j<=10-i;j++)
            if(a[j-1]>a[j])
                { t=a[j-1]; a[j-1]=a[j]; a[j]=t; }

    printf("the sorted numbers:\n");
    for(i=0;i<10;i++)
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}
```

对于n个数，共进行n-1轮排序

第一轮在n个数中比较n-1次，确定最大值

第二轮在n-1个数中比较n-2次，确定次大值

.....

第i轮在n-i+1个数中比较n-i次，确定第i大值

.....

第n-1轮在2个数中比较1次，确定次小值



例4.用冒泡法对10个整数排序

```
#include <stdio.h>
int main()
{   int a[10], i, j, t;
    printf("input 10 numbers:\n");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);

    for(i=1;i<=9;i++)
        for(j=0;j<10-i;j++)
            if(a[j]>a[j+1])
                {t=a[j];a[j]=a[j+1];a[j+1]=t;}

    printf("the sorted numbers:\n");
    for(i=0;i<10;i++)
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}
```

对于n个数，共进行n-1轮排序

第一轮在n个数中比较n-1次，确定最大值

第二轮在n-1个数中比较n-2次，确定次大值

.....

第i轮在n-i+1个数中比较n-i次，确定第i大值

.....

第n-1轮在2个数中比较1次，确定次小值



例4.用冒泡法对10个整数排序

```
#include <stdio.h>
int main()
{   int a[10], i, j, t;
    printf("input 10 numbers:\n");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);

    for(i=1;i<=9;i++)
        for(j=0;j<10-i;j++)
            if(a[j]>a[j+1])
                {t=a[j];a[j]=a[j+1];a[j+1]=t;}

    printf("the sorted numbers:\n");
    for(i=0;i<10;i++)
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}
```

对于n个数，共进行n-1轮排序

第一轮在n个数中比较n-1次，确定最大值

第二轮在n-1个数中比较n-2次，确定次大值

.....

第i轮在n-i+1个数中比较n-i次，确定第i大值

.....

第n-1轮在2个数中比较1次，确定次小值

1 2 3 4 5 6 7 8 9 10

这个程序有待改进！



```
#include<stdio.h>
int main()
{   int a[10], i, j, t, flag;
    .....
    for(i=1,flag=0;i<=9;i++,flag=0)
    {
        for(j=1;j<=10-i;j++)
            if(a[j-1]>a[j])
                { t=a[j-1]; a[j-1]=a[j]; a[j]=t;
                  flag=1; }

        if(flag==0)
            break;

    }

    .....
}
```

如果一轮两两比较下来没有发生元素交换，则停止下一轮的比较。

```
#include<stdio.h>
int main()
{   int a[10], i, j, t;
    printf("input 10 numbers:\n");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);

    for(i=1;i<=9;i++)
        for(j=1;j<=10-i;j++)
            if(a[j-1]>a[j])
                { t=a[j-1]; a[j-1]=a[j]; a[j]=t; }

    printf("the sorted numbers:\n");
    for(i=0;i<10;i++)
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}
```

Questions & Answers

A large, bold, red capital letter 'C' is centered within a white square. The square is positioned on the right side of the slide, overlapping the dark blue background and the light gray horizontal band.