

# 第6章 利用数组处理批量数据

**6.1 一维数组**

**6.2 二维数组**

**6.3 字符数组**

## 6.3 字符数组

**6.3.1 怎样定义字符数组**

**6.3.2 字符数组的初始化**

**6.3.3 怎样引用字符数组中的元素**

**6.3.4 字符串和字符串结束标志**

**6.3.5 字符数组的输入输出**

**6.3.6 善于使用字符串处理函数**

**6.3.7 字符数组应用举例**

## 6.3.1 怎样定义字符数组

- 用来存放字符数据的数组是字符数组
- 字符数组中的一个元素存放一个字符
- 定义字符数组的方法与定义数值型数组的方法类似

**char c[10];**

**c[0]='I'; c[1]=' ';**

**c[2]='a'; c[3]='m';**

**c[4]=' ';** c[5]='h';

**c[6]='a'; c[7]='p';**

**c[8]='p'; c[9]='y';**

**c[0]c[1]c[2]c[3]c[4]c[5]c[6]c[7]c[8]c[9]**

<b>I</b>		<b>a</b>	<b>m</b>		<b>h</b>	<b>a</b>	<b>p</b>	<b>p</b>	<b>y</b>
----------	--	----------	----------	--	----------	----------	----------	----------	----------

## 6.3.2 字符数组的初始化

```
char c[10]={ 'I', ' ', 'a', 'm', ' ', 'h', 'a', 'p', 'p', 'y' };  
c[0]c[1]c[2]c[3]c[4]c[5]c[6]c[7]c[8]c[9]
```

I		a	m		h	a	p	p	y
---	--	---	---	--	---	---	---	---	---

```
char c[10]={ 'c', ' ', 'p', 'r', 'o', 'g', 'r', 'a', 'm' };  
c[0]c[1]c[2]c[3]c[4]c[5]c[6]c[7]c[8]c[9]
```

c		p	r	o	g	r	a	m	\0
---	--	---	---	---	---	---	---	---	----

```
char diamond[5][5]={ { ' ', ' ', ' ', '*' },  
                      { ' ', '*', ' ', '*', ' ' },  
                      { '*', ' ', ' ', ' ', '*' },  
                      { ' ', '*', ' ', ' ', '*' },  
                      { ' ', ' ', ' ', '*', ' ' } };
```

## 6.3.3怎样引用字符数组中的元素

例**6.6** 输出一个已知的字符串。

➤ 解题思路:

- ◆ 定义一个字符数组，并用“初始化列表”对其赋以初值
- ◆ 用循环逐个输出此字符数组中的字符

```
#include <stdio.h>
int main()
{ char c[15]={'I',' ','a','m',' ','a',
             ' ','s','t','u','d','e','n','t','.'};
  int i;
  for(i=0;i<15;i++)
    printf("%c",c[i]);
  printf("\n");
  return 0;
}
```

I am a student.

## 6.3.4字符串和字符串结束标志

- 在**C**语言中，是将字符串作为**字符数组**来处理的
- 关心的是字符串的**有效长度**而不是字符数组的长度
- 为了测定字符串的实际长度，**C**语言规定了字符串结束标志**'\0'**
- **'\0'**代表**ASCII**码为**0**的字符
- 从**ASCII**码表可以查到，**ASCII**码为**0**的字符不是一个可以显示的字符，而是一个“空操作符”，即它什么也不做
- 用它作为字符串结束标志不会产生附加的操作或增加有效字符，只起一个供辨别的标志

## 6.3.4字符串和字符串结束标志

`char c[]={ "I am happy" }; 可写成`

`char c[]="I am happy"; 相当于`

`char c[11]={ "I am happy" }; 可写成`

`char c[10]={ "China" }; 可写成`

`char c[10]="China";`

从`c[5]`开始，元素值均为`\0`

C	h	i	n	a	\0	\0	\0	\0	\0
---	---	---	---	---	----	----	----	----	----

只显示

`printf("%s",c);`



## 6.3.5 字符数组的输入输出

- 字符数组的输入输出可以有两种方法：
  - ◆ 逐个字符输入输出 (**%c**)
  - ◆ 整个字符串一次输入输出 (**%s**)
- 输出的字符中不包括结束符'**\0**'
- 用**%s**输出字符串时，**printf**函数中的输出项是字符数组名，不是数组元素名
- 如果一个字符数组中包含多个'**\0**'，则遇第一个'**\0**'时输出就结束
- 可以用**scanf**函数输入一个字符串
- **scanf**函数中的输入项**c**是已定义的字符数组名，输入的字符串应短于已定义的字符数组的长度

## 6.3.5 字符数组的输入输出

```
char c[6];
```

```
scanf("%s",c); China✓
```

系统自动在**China**后面加一个'\0'

```
char str1[5],str2[5],str3[5];
```

```
scanf("%s%s%s",str1,str2,str3);
```

How are you? ✓

str1	H	o	w	\0	\0
str2	a	r	e	\0	\0
str3	y	o	u	?	\0

## 6.3.6 善于使用字符串处理函数

- 在C函数库中提供了一些用来专门处理字符串的函数，使用方便

### 1. puts函数----输出字符串的函数

- 其一般形式为：**puts (字符数组)**
- 作用是将一个字符串输出到终端

**char str[20]="China"; puts(str);**      输出**China**

### 2. gets函数----输入字符串的函数

- 其一般形式为：**gets(字符数组)**
- 作用是输入一个字符串到字符数组

**char str[20]; gets(str);      Computer✓**

## 6.3.6 善于使用字符串处理函数

### 3. **strcat**函数-----字符串连接函数

➤ 其一般形式为：

**strcat(字符串数组1, 字符串数组2)**

➤ 其作用是把两个字符串连接起来，把字符串**2**接到字符串**1**的后面，结果放在字符串数组**1**中

```
char str1[30]="People";      //要足够大
```

```
char str2[]="China";
```

```
printf("%s", strcat(str1,str2));
```

输出：**PeopleChina**

使用字符串函数时,在程序开头用**#include <string.h>**

## 6.3.6 善于使用字符串处理函数

### 4. strcpy和strncpy函数-字符串复制

➤ **strcpy**一般形式为:

**strcpy**(字符数组**1**,字符串**2**)

➤ 作用是将字符串**2**复制到字符数组**1**中去

```
char str1[10],str2[]="China";  
strcpy(str1,str2);
```

数组名

数组名或字符串常量  
**strcpy(str1, "China");**

要足够大

str1

C	h	i	n	a	\0	\0	\0	\0	\0
---	---	---	---	---	----	----	----	----	----

## 6.3.6 善于使用字符串处理函数

### 4. **strcpy**和**strncpy**函数-字符串复制

- 可以用**strncpy**函数将字符串**2**中前面**n**个字符复制到字符数组**1**中去
- **strncpy(str1, str2, 2);**
  - ◆ 作用是将**str2**中最前面**2**个字符复制到**str1**中，取代**str1**中原有的最前面**2**个字符
  - ◆ 复制的字符个数**n**不应多于**str1**中原有的字符

## 6.3.6 善于使用字符串处理函数

### 5. strcmp函数-----字符串比较函数

- 其一般形式为: **strcmp(字符串1, 字符串2)**
  - ◆ 作用是比较字符串**1**和字符串**2**
  - ◆ **strcmp(str1,str2);**
  - ◆ **strcmp("China","Korea");**
  - ◆ **strcmp(str1,"Beijing");**
- 字符串比较的规则是
  - ◆ 将两个字符串自左至右逐个字符相比, 直到出现不同的字符或遇到'**\0**'为止
  - ◆ 如全部字符相同, 认为两个字符串相等
  - ◆ 若出现不相同的字符, 则以第一对不相同的字符的比较结果为准

## 6.3.6 善于使用字符串处理函数

### 5. strcmp函数-----字符串比较函数

`"A"<"B"`    `"a">"A"`    `"computer">"compare"`  
`"these">"that"`    `"1A">"$20"`  
`"CHINA">"CANADA"`    `"DOG"<"cat"`  
`"Tsinghua">"TSINGHUA"`

➤ 比较的结果由函数值带回

- ◆如果字符串**1**=字符串**2**，则函数值为**0**
- ◆如果字符串**1**>字符串**2**，则函数值为一个正整数
- ◆如果字符串**1**<字符串**2**，则函数值为一个负整数

`if(str1>str2) printf("yes");` // 错误

`if(strcmp(str1,str2)>0) printf("yes");` // 正确



## 6.3.6 善于使用字符串处理函数

### 6. **strlen**函数-----测字符串长度的函数

➤ 其一般形式为：**strlen (字符数组)**

- ◆它是测试字符串长度的函数
- ◆函数的值为字符串中的实际长度
- ◆**char str[10]="China";**
- ◆**printf("%d",strlen(str));**
- ◆输出结果是**5**
- ◆也可以直接测试字符串常量的长度
- ◆**strlen("China");**

## 6.3.6 善于使用字符串处理函数

### 7. **strlwr**函数-----转换为小写的函数

➤ 其一般形式为: **strlwr** (字符串)

◆函数的作用是将字符串中大写字母换成小写字母

◆**strlwr("China");** //得到: **china**

### 8. **strupr**函数-----转换为大写的函数

➤ 其一般形式为: **strupr** (字符串)

◆函数的作用是将字符串中小写字母换成大写字母

◆**strupr("China");** //得到: **China**

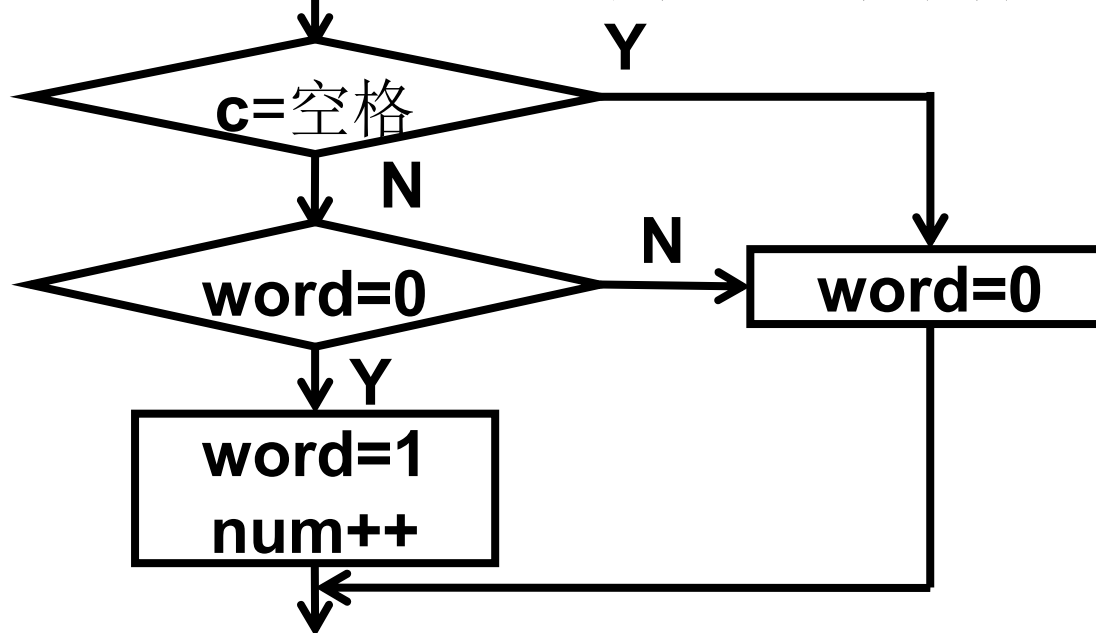
## 6.3.7 字符数组应用举例

例**6.8** 输入一行字符，统计其中有多少个单词，单词之间用空格分隔开。

- 解题思路：问题的关键是怎样确定“出现一个新单词了”
  - ◆ 从第**1**个字符开始逐个字符进行检查，判断此字符是否是新单词的开头，如果是，就使变量**num**的值加**1**，最后得到的**num**的值就是单词总数
  - ◆ 判断是否出现新单词，可以由是否有空格出现来决定(连续的若干个空格作为出现一次空格；一行开头的空格不统计在内)
  - ◆ 如果测出某一个字符为非空格，而它的前面的字符是空格，则表示“新的单词开始了”，此时使**num**累加**1**
  - ◆ 如果当前字符为非空格而其前面的字符也是非空格，则**num**不应再累加**1**

## 6.3.7 字符数组应用举例

- ◆ 用变量 **word** 作为判别当前是否开始了一个新单词的标志，若 **word=0** 表示未出现新单词，如出现了新单词，就把 **word** 置成 **1**
- ◆ 前面一个字符是否空格可以从 **word** 的值看出来，若 **word** 等于 **0**，则表示前一个字符是空格；如果 **word** 等于 **1**，意味着前一个字符为非空格



```
if(c==' ')  
    word=0;  
else if(word==0)  
{  
    word=1;  
    num++;  
}
```

当前字符	I		a	m		a		b	o	y	.
是否空格	否	是	否	否	是	否	是	否	否	否	否
word原值	0	1	0	1	1	0	1	0	1	1	1
新单词开始否	是	否	是	否	否	是	否	是	否	否	否
word新值	1	0	1	1	0	1	0	1	1	1	1
num值	1	1	2	2	2	3	3	4	4	4	4

```
#include <stdio.h>
```

```
int main(){
```

```
    char string[81],c;  int i,num=0,word=0;
```

```
    gets(string);
```

```
    for (i=0;(c=string[i])!='\0';i++)
```

```
        if(c==' ') word=0;
```

```
        else if(word==0)
```

```
            { word=1;
```

```
              num++;
```

```
            }
```

```
    printf("%d words\n",num);
```

```
    return 0
```

```
}
```

一定要设初始

相当于  
`c=string[i];`  
`c!='\0'`

```
I am a boy.  
4 words
```

例**6.9** 有**3**个字符串,要求找出其中最大者。

➤ 解题思路: 设一个二维的字符数组**str**,大小为**3×10**。每一行存放一个字符串 **char str[3][10];**

➤ 可以把**str[0],str[1],str[2]**看作**3**个一维字符数组,可以把它们如同一维数组那样进行处理

```
for (i=0;i<3;i++)  
    gets (str[i]);
```



str[0]	C	h	i	n	a	\0	\0	\0	\0	\0
str[1]	J	a	p	a	n	\0	\0	\0	\0	\0
str[2]	I	n	d	i	a	\0	\0	\0	\0	\0

➤ 经过两次比较,就可得到值最大者,把它放在一维字符数组**string**中

```
if (strcmp(str[0],str[1])>0) strcpy(string,str[0]);  
else
```

```
    strcpy(string,str[1]);
```

```
if (strcmp(str[2],string)>0) strcpy(string,str[2]);
```

```
#include<stdio.h>
#include<string.h>
int main ( )
{
    char str[3][10]; char string[10]; int i;
    for (i=0;i<3;i++)    gets (str[i]);
    if (strcmp(str[0],str[1])>0)
        strcpy(string,str[0]);
    else
        strcpy(string,str[1]);
    if (strcmp(str[2],string)>0)
        strcpy(string,str[2]);
    printf("\nthe largest:\n%s\n",string);
    return 0;
}
```



```
China
Japan
India

the largest:
Japan
```