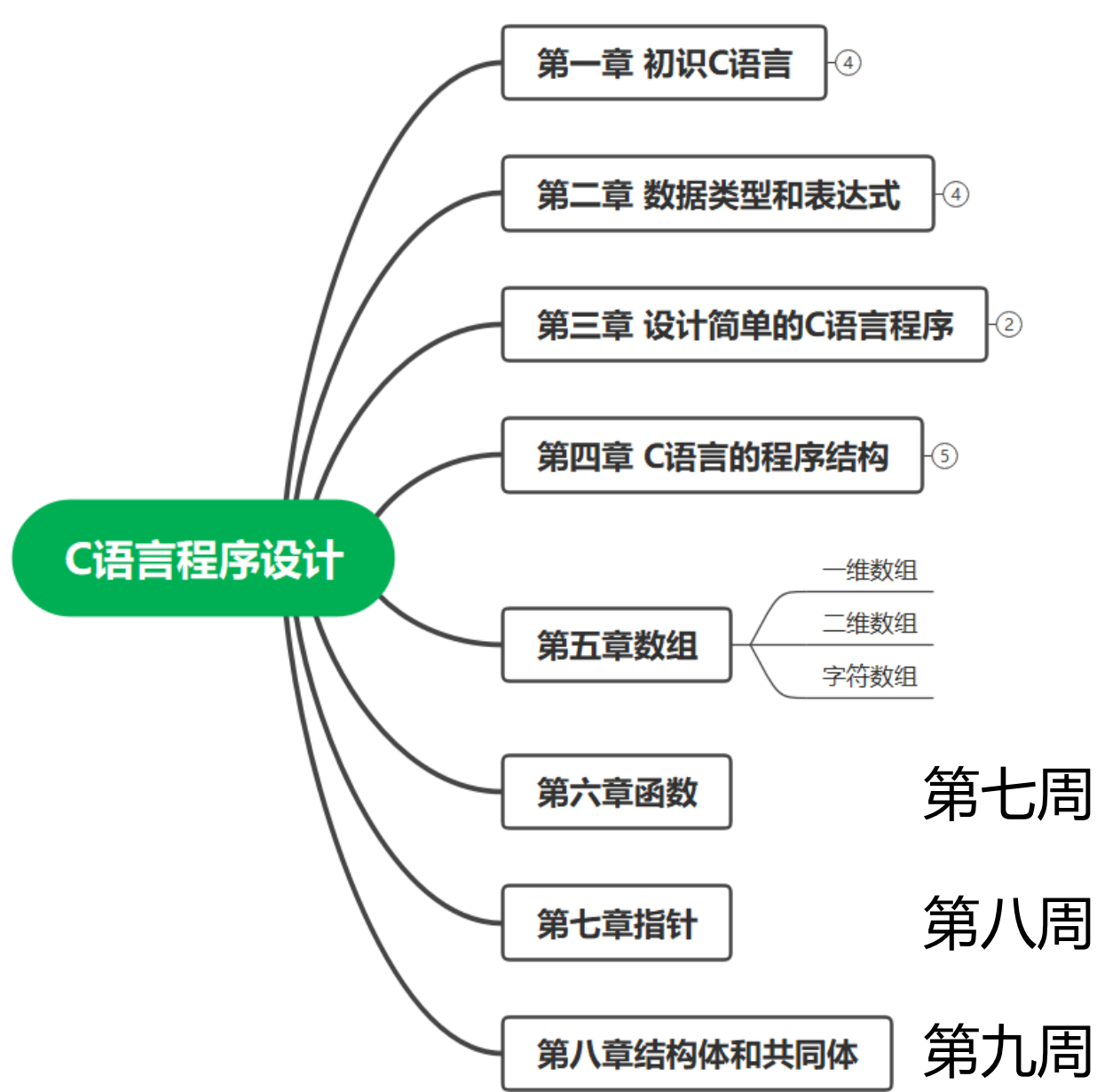


# 第5章 数组

## 《C语言程序设计新编教程》



相对简单  
训练动手能力为主

大家一定要抽出时间预习、学习、复习。  
看书、思考、动手编程

进阶学习  
课堂讲解为主，尽量抽时间给大家动手编程

# 能力要求

CAPACITY



了解一维数组基本概念



掌握C语言中数组的定义



掌握利用数组编写较复杂程序的基本方法



掌握数组元素的引用



# 内容导航

CONTENTS

## 数组



一维数组



二维数组



字符数组和字符串



项目举例

什么叫做数组？集合

**数组：**指具有相同数据类型的数据，按照顺序存储在一起组成的有序集合。

在C语言中，数据属于构造数据类型。一个数组可以分解为多个**数组元素**，数据元素具有相同的数据类型，这写数据在内存中占据一段连续的存储单元。

用一个统一的名字来标识这组数据，这个名字就叫做**数组名**，构成数组的每一个数据项就称为数组元素。

**用一个统一的数组名和下标唯一地确定数组中的元素。**

### 5.1.1 一维数组的定义

所谓**一维数组**是指数组中每个元素只带有一个下标。

在C语言中使用数组必须先进行定义。

一维数组的定义格式如下：

存储类型 类型说明符 数组名[常量表达式];

格式说明：

- ①**存储类型**说明数组元素的存储属性，可以是静态型（static）、自动型（auto）及外部型（extern）。**默认是auto型**。
- ②**类型说明符**是任一种基本数据类型或构造数据类型。
- ③**数组名**是用户定义的数组标识符。
- ④**方括号中的常量表达式**表示数据元素的个数，也称为数组的长度

例如：定义一个静态的整型的有3个元素的一维数组可以表示为：

```
static int a[3];
```

其中a为数组名，3是数组元素的个数，每个元素都是int型的数据



下面是合法的数组定义：

int a[10];       说明整型数组a，有10个元素。

float b[10],c[20];   说明实型数组b，有10个元素，实型数组c，有20个元素。

char ch[20];       说明字符数组ch，有20个元素。

#define N 6

long n[N];       说明定义了一个有6个元素的长整型数组n。

short m[8\*N];       说明定义了一个有48个元素的短整型数组m。

对于数组类型说明应注意以下几点：

- 1、数组名的书写规则和变量名的定义规则相同，应遵循标识符的命名规则。
- 2、数组名不能与其它变量名相同。





例如：

```
main()
```

```
{
```

```
    int b;
```

```
    int b[8];
```

```
    .....
```

```
}
```

是错误的。



- 3、方括号中常量表达式表示数组元素的个数，即数组长度。如a[5]表示数组a有 5个元素。但是其下标从0开始计算。因此5个元素分别为a[0],a[1],a[2],a[3],a[4]。
- 4、不能在方括号中用变量来表示元素的个数，但是可以是符号常量或常量表达式。例如：

```
#define MN 5
main()
{
    int a[6+4],b[9+MN];
    .....
}
```

是合法的。



但是下述说明方式是非法的。

```
main()
{
    int p=5;
    int a[p];
    .....
}
```

5、允许在同一个类型说明中，说明多个数组和多个变量。

例如：

```
int a,b, m1[5],m2[12];
```



### 5.1.2 一维数组的初始化

C语言在定义数组的同时对数组中的各个元素指定初值，这个过程就是**初始化**。

数组初始化赋值是指在数组定义时给数组元素赋予初值。数组初始化不占用运行时间，是在编译阶段进行，这样将减少运行时间，提高效率。

初始化赋值的一般形式为：

类型说明符 数组名[常量表达式]={元素值列表};

其中在{ }中的元素值列表即为各元素的初值，各值之间用逗号间隔。

例如：

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

相当于a[0]=1;a[1]=2;a[2]=3;a[3]=4;a[4]=5;



C语言对数组的初始化赋值还有以下几点规定：

1、只能给元素逐个赋值，不能给数组整体赋值。

例如：

给五个元素全部赋1值，只能写为：

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

而不能写为：

```
int a[5]=1;
```



2、可以只给部分元素赋初值。

当{ }中值的个数少于元素个数时，只给前面部分元素赋值。

例如：

```
int a[10]={4,5,6,7,8};
```

表示只给a[0] ~ a[4]中的5个元素赋值，而后5个元素系统自动赋0值。

3、如给全部元素赋值，则在数组说明中，可以不给出数组元素的个数。

例如：

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

可写为：

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

这时数组的长度就是后面赋值元素的个数。



### 5.1.3 一维数组元素的引用

数组必须先定义，后使用。C语言中只能逐个的引用数组元素，而不能引用整个的数组。数组元素是组成数组的基本单元。

数组元素引用的一般格式如下：

数组名[下标]

格式说明：

①下标只能为整型常量或整型表达式。

例如：

a[20]

a[3\*5]

都是合法的数组元素。



②数组元素通常也称为下标变量。

例如，输出有10个元素的数组必须使用循环语句逐个输出各下标变量：

```
for(k=0; k<10; k++)
```

```
    printf("%4d",c[k]);
```

而不能用一个语句输出整个数组。

下面的写法是错误的：

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







# 内容导航

CONTENTS

## 数组



一维数组



二维数组



字符数组和字符串



项目举例

### 5.2.1 二维数组的定义

二维数组可以看作是多个相同类型的一维数组。本节只介绍二维数组，多维数组可由二维数组类推而得到。

二维数组定义的一般格式如下：

存储类型 类型说明符 数组名[常量表达式1][常量表达式2]

**格式说明：**

①常量表达式1表示第一维下标的长度，常量表达式2 表示第二维下标的长度。

例如：

```
int s[3][3];
```

定义了一个三行三列的整型数组，数组名为s，其数组元素的类型为整型。该数组的元素共有 $3 \times 3$ 个，即：



`s[0][0],s[0][1],s[0][2]`

`s[1][0],s[1][1],s[1][2]`

`s[2][0],s[2][1],s[2][2]`

② **二维数组相当于数学中的“矩阵”。**“常量表达式1”代表矩阵的行数，“常量表达式2”代表矩阵的列数。而实际的硬件存储器却是连续编址的，也就是说存储器单元是按一维线性排列的。如何在一维存储器中存放二维数组，可有两种方式：一种是按行排列，即放完一行之后顺次放入第二行。另一种是按列排列，即放完一列之后再顺次放入第二列。

在C语言中，二维数组是按行排列的，即：

二维数组`s[3][3]`可以看作是有3个元素的一维数组，元素为`s[0]`，`s[1]`，`s[2]`。而每个元素又可以看作是有3个元素的一维数组，如`s[0]`中有3个元素，分别为`s[0][0]`，`s[0][1]`，`s[0][2]`，以此类推。

先存放`s[0]`行，再存放`s[1]`行，最后存放`s[2]`行。每行中有三个元素也是依次存放。由于数组`s`说明为`int`类型，该类型在`vc++6.0`环境中占四个字节的内存空间，所以每个元素均占有四个字节。



### 5.2.2 二维数组的初始化

二维数组初始化和一维数组类似也是在类型说明时给各下标变量赋以初值。二维数组可按行连续赋值，也可按行分段赋值。

例如：

对数组s[3][5]赋值：

1. 按行连续赋值可写为：

```
int s[3][5]={ 87,90,76,77,80,75,92,61,65,85,71,59,63,70,85};
```

2. 按行分段赋值可写为：

```
int s[3][5]={ {87,90,76,77,80},{75,92,61,65,85},{71,59,63,70,85} };
```

这两种赋初值的结果是完全相同的。



### 5.2.3 二维数组元素的引用

二维数组元素的引用格式如下：

数组名[下标1][下标2]

格式说明：

- ① 下标应为整型常量或整型表达式，必须有确定的值。
- ② 下标从0开始变化，其值分别小于数组定义中的“常量表达式1”和“常量表达式2”。

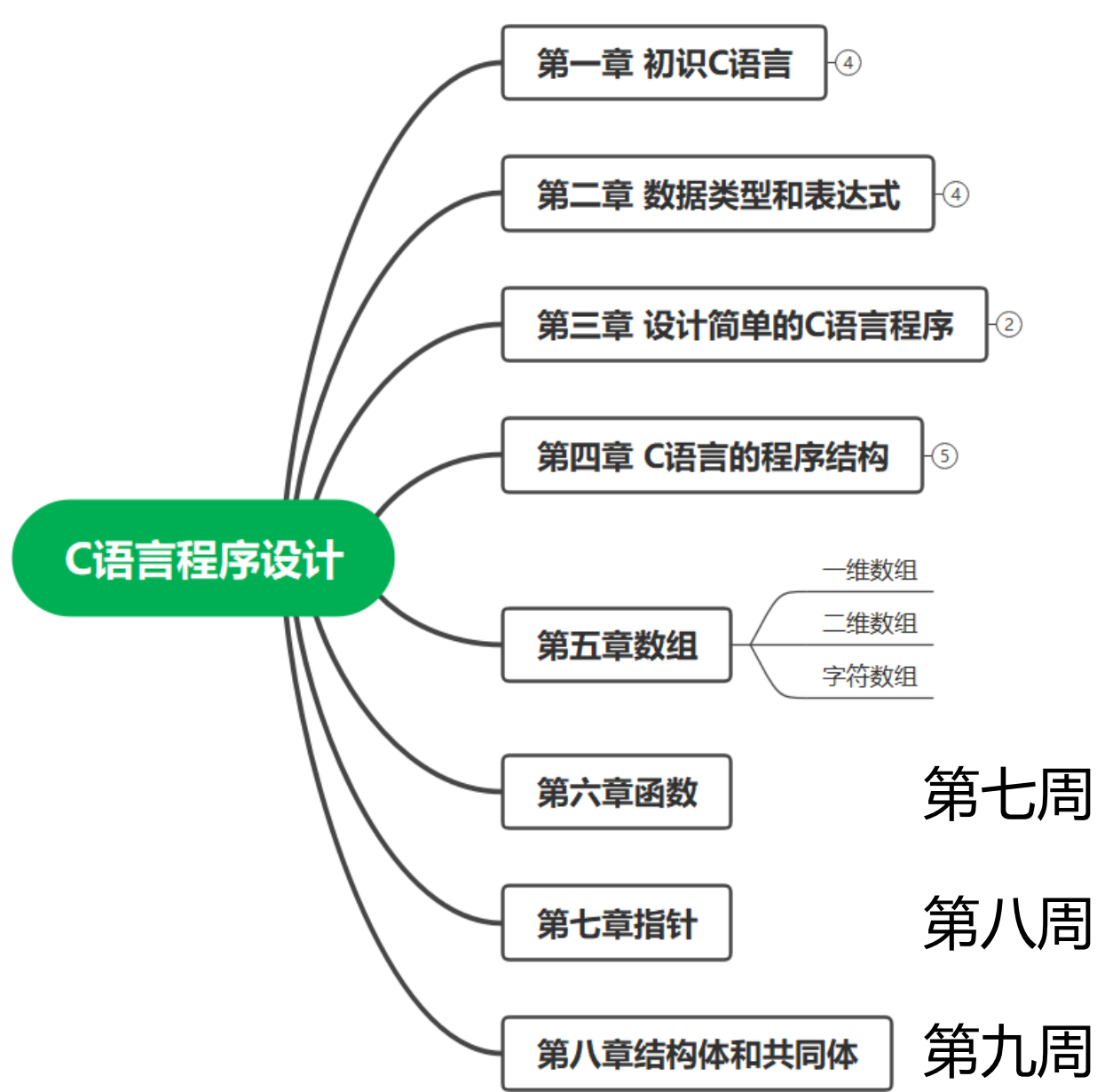
例如：

a[3][5]

表示a数组第三行第五列的元素。

- ③ 需要注意的是下标值应在已定义的数组大小的范围内，否则就越界了。





相对简单  
训练动手能力为主

大家一定要抽出时间预习、学习、复习。  
看书、思考、动手编程

进阶学习  
课堂讲解为主，尽量抽时间给大家动手编程

回顾：何为数值？何为字符？

何为数组？

具有相同数据类型的数据，按照顺序存储在一起组成的有序集合

变量 `x=1`

一维数组 `a[5]={1,2,3,4,5}`

二维数组 `b[3][3]={ 1,2,3  
4,5,6  
7,8,9}`

字符数组：

`char str[15]={'H', 'e', 'l', 'l', 'o', '!', 'W', 'o', 'r', 'l', 'd', '!'};`

数组使用最重要的两点：

1、定义及初始化

2、引用

`for(k=0; k<10; k++)`

`printf("%4d",c[k]);`

而不能用一个语句输出整个数组。

下面的写法是错误的：

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



# 内容导航

CONTENTS

## 数组



一维数组



二维数组



字符数组和字符串



项目举例



回顾：何为数值？何为字符？

变量  $x=1$

一维数组  $a[5]=\{1,2,3,4,5\}$

二维数组  $b[3][3]=\{$   
1,2,3  
4,5,6  
7,8,9 $\}$

提问：如何把 “hello world” 这句话存起来？

### 5.3.1 字符数组的定义

所谓字符数组就是**用来存放字符数据**的数组，其中一个数组元素存放一个字符。

定义与前面介绍的数组定义类似，只是类型说明符用char。

例如：

**char str[7];** (a) 

ch[0]	ch[1]	ch[2]	ch[4]	ch[5]	ch[6]
H	e	l	l	o	

定义str为字符数组，包含7个元素，即**str[0],str[1],str[2],str[3],str[4],str[5],str[6]**。

**字符数组也可以是二维或多维数组。**

例如：

char str[4][5];

即为二维字符数组。



arr[0]→	arr[0][0]	arr[0][1]	arr[0][2]
arr[1]→	arr[1][0]	arr[1][1]	arr[1][2]
arr[2]→	arr[2][0]	arr[2][1]	arr[2][2]
arr[3]→	arr[3][0]	arr[3][1]	arr[3][2]



### 5.3.2 字符数组的初始化

**字符数组也允许在定义时作初始化赋值。**

例如：

```
char str[15]={‘H’, ‘e’, ‘l’, ‘l’, ‘o’, ‘!’, ‘W’, ‘o’, ‘r’, ‘l’, ‘d’, ‘!’};
```

赋值后各元素的值为：

str[0]的值为 ‘H’， str[1]的值为 ‘e’， str[2]的值为 ‘l’， str[3]的值为 ‘l’， str[4]的值为 ‘o’， str[5]的值为 ‘!’， str[6]的值为 ‘W’， str[7]的值为 ‘o’， str[8]的值为 ‘r’， str[9]的值为 ‘l’， str[10]的值为 ‘d’， str[11]的值为 ‘!’。

其中str[12] str[13] str[14]未赋值， **系统自动赋予空字符（‘ \0’）**。

	ch[0]	ch[1]	ch[2]	ch[4]	ch[5]	ch[6]
(a)	H	e	l	l	o	
(b)	H	e	l	\0	\0	
(c)	H	e	l	l	o	\0
(d)	H	e	l	\0	\0	\0

图 5-5 字符数组初始化示意图



**需要注意的是：**

如果花括弧中的字符个数大于数组的长度，则作语法错误处理。

如果初值的个数小于数组的长度，则将这些字符赋给数组中前面的元素，其余的元素系统自动定为空字符（'\0'）。



### 5.3.3 字符数组元素的引用

根据字符数组的下标引用字符数组中的元素，得到一个字符。

【例5.4】输出一行字符串。

```
char str[15]={'H', 'e', 'l', 'l', 'o', '!', 'W', 'o', 'r', 'l', 'd', '!'};
```

如何输出？

字符数组可以按照单个元素依次引用输出



### 程序代码：

```
/*ex5_4.C:输出一行字符串*/
#include<stdio.h>
int main()
{
    int i;
    char str[15]={'H', 'e', 'l', 'l', 'o', '!', 'W', 'o', 'r', 'l', 'd', '!'};
    for(i=0;i<15;i++)
        printf("%c",str[i]);
    printf("\n");
}
```



### 5.3.4 字符数组的输入和输出

在C语言中没有专门的字符串变量，通常用一个字符数组来存放一个字符串。

当把一个字符串存入一个数组时，也把字符串的结束符‘\0’存入数组，并以此作为该字符串是否结束的标志。有了‘\0’标志后，就不必再用字符数组的长度来判断字符串的长度了。

C语言允许用字符串的方式对数组作初始化赋值。

例如：

```
char str[15]={'H', 'e', 'l', 'l', 'o', '!', 'W', 'o', 'r', 'l', 'd', '!'};
```

可写为：

```
char str[]={"Hello!World!"};
```

或去掉{}写为：

```
char str[]="Hello!World!";
```



## 字符数组的输入输出有两种方法：

- (1) 用格式符“%c”逐个字符输入输出。如例5.4所示。
- (2) 在采用字符串方式后，字符数组的输入输出将变得简单方便。  
用格式符“%s”将整个字符串一次性地输入或输出。





- 【例5.5】例5.4的改进，一次性的输出一行字符串。
- **程序代码：**
- `/*ex5_5.C:一次性输出一行字符串*/`
- `#include<stdio.h>`
- `main()`
- `{`
- `char str[15];`
- `printf("请输入字符串:\n");`
- `scanf("%s",str);`
- `printf("请输出字符串:\n");`
- `printf("%s\n",str);`
- `}`



## 程序说明:

- ①当用scanf函数输入字符串时，字符串中不能含有空格，否则将以空格作为串的结束符。



【例5.6】程序【例5.5】的改进。

程序代码：

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

```
int main()
```

```
{
```

```
    char st1[6],st2[6],st3[6],st4[6];
```

```
    printf("请输入字符串:\n");
```

```
    scanf("%s%s%s%s",st1,st2,st3,st4);
```

```
    printf("请输出字符串:\n");
```

```
    printf("%s %s %s %s\n",st1,st2,st3,st4);
```

```
}
```



### 5.3.5 常用字符串处理函数

C语言的库函数提供了丰富的字符串处理函数，使用这些函数可大大减轻编程的负担。大致可分为字符串的输入、输出、合并、修改、比较、复制、转换等等。用于输入输出的字符串函数，在使用前应包含头文件<stdio.h>，使用其它字符串函数则应包含头文件<string.h>。

下面介绍几个最常用的字符串函数。

#### 1. 字符串输入函数gets

格式： gets (字符数组名)

功能：从标准输入设备键盘上输入一行字符串。

本函数得到一个函数值，即为该字符数组的首地址。

## 2. 字符串输出函数 puts

格式: puts (字符数组名)

功能: 把字符数组中的字符串输出到显示器。



**【例5.7】**用字符串输入函数gets输入一行字符串。

**程序代码：**

```
/*ex5_7.C: 用字符串输入函数gets输入一行字符串并输出*/
```

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

```
#include<string.h>
```

```
int main()
```

```
{
```

```
    char str[15];
```

```
    printf("请输入字符串:\n");
```

```
    gets(str);
```

```
    printf("请输出字符串:\n");
```

```
    puts(str);
```

```
}
```

【例5.8】用字符串输出函数puts输出一行字符串。

**程序代码：**

```
/*ex5_8.C: 用字符串输出函数puts输出一行字符串*/
```

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

```
#include<string.h>
```

```
void main()
```

```
{
```

```
    char str[]="Hello!\nWorld!";
```

```
    printf("请输出字符串:\n");
```

```
    puts(str);
```

```
}
```



### 3. 字符串拷贝函数strcpy

格式: `strcpy (字符数组名1, 字符数组名2)`

功能: 把字符数组2中的字符串拷贝到字符数组1中。

串结束标志' \0'也一同拷贝。字符数组2, 也可以是一个字符串常量。这相当于把一个字符串赋予一个字符数组。





```
#include <stdio.h>
#include <string.h>
#define LEN 20
int main()
{
    char str1[LEN],str2[LEN];
    printf("Input one strings:\n");
    gets(str2);
    strcpy(str1,str2);
    puts(str1);
    return 0;
}
```



#### 4. 字符串连接函数strcat

格式： strcat (字符数组名1, 字符数组名2)

功能：把字符数组2中的字符串连接到字符数组1中字符串的后面，并删去字符串1后的串结束标志' \0'。本函数返回值是字符数组1的首地址。

【例5.10】用字符串连接函数strcat连接并输出一行字符串。



程序代码：

```
/*ex5_10.C: 用字符串连接函数strcat连接并输出一行字符串*/
```

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

```
#include<string.h>
```

```
int main()
```

```
{
```

```
    static char str1[30]="My English name is ";
```

```
    char str2[10];
```

```
    printf("请输入你的英文名字:\n");
```

```
    gets(str2);
```

```
    strcat(str1,str2);
```

```
    printf("输出字符串:\n");
```

```
    puts(str1);
```

```
}
```



## 5. 字符串比较函数strcmp

格式： `strcmp(字符数组名1, 字符数组名2)`

功能：按照ASCII码顺序比较两个数组中的字符串，并由函数返回值返回比较结果。

字符串1 = 字符串2，返回值为0；

字符串1 > 字符串2，返回值为一正整数；

字符串1 < 字符串2，返回值为一负整数。

本函数也可用于比较两个字符串常量，或比较字符数组和字符串常量



```
#include <stdio.h>
#include <string.h>
#define LEN 20
int main()
{
    char str1[LEN],str2[LEN];
    int result;
    printf("Input two strings:\n");
    gets(str1);
    gets(str2);
    result=strcmp(str1,str2);
    if(result==0)
        printf("str1==str2\n");
    else if(result>0)
        printf("str1>str2\n");
    else
        printf("str1<str2\n");
}
```



## 56. 求字符串长度函数strlen

格式: `strlen(字符数组名)`

功能: 函数返回字符串的实际长度(不含字符串结束标志' \0')。

**【例5.12】**用字符串长度函数strlen求字符串的实际长度。



程序代码：

```
/*ex5_12.C: 用字符串长度函数strlen求字符串的实际长度*/
```

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

```
#include<string.h>
```

```
int main()
```

```
{
```

```
int k;
```

```
    static char str[]="Hello! World!";
```

```
    k=strlen(str);
```

```
    printf("字符串的实际长度是： %d\n",k);
```

```
}
```





# 内容导航

CONTENTS

## 数组



一维数组



二维数组



字符数组和字符串



项目举例



### 1. 项目描述

设有一位学生的5门课成绩如下表，要求求这位学生5科成绩的平均值。

课程 姓名	数学	英语	语文	物理	C语言
张三	99	80	92	71	88



## 2. 项目分析

### (1) 功能分析

根据项目描述，就是给定学生5门课的成绩，编写程序求5门课程的平均成绩。

### (2) 数据分析

根据功能要求，需要定义一个一维数组，这个一维数组中有5个元素分别用来存放5门课程的成绩。为简单起见，设这个数组的类型为整型。

## 3. 设计思想

(1) 定义数组和三个变量。数组名为score，数组中有5个元素，变量sum是存放5门课的总成绩，变量average 是存放平均值的，还有个循环变量i；

(2) 从键盘中给数组元素赋值；

(3) 求5门课的总成绩，赋给变量sum；

(4) 求5门课的平均成绩，赋给变量average。



## 5.4.2 求M位学生N门课程的平均成绩问题

## 1. 项目描述

已知有三位学生，每位学生有5门课程，编写程序求每位学生的平均成绩。

课程 姓名	数学	英语	语文	物理	C语言
张三	99	80	92	71	88
李四	77	66	87	90	85
王五	85	91	88	76	100



## 2. 项目分析

### (1) 功能分析

根据功能描述，程序实现的功能就是对三位学生，分别求他们的平均成绩。

### (2) 数据分析

本程序需要一个存储值的二维数组，用于存储三位学生的5科成绩，还有两个循环变量和一个存放三位学生平均值的一维数组。

## 3. 设计思想

- (1) 定义变量和数组。两个整型变量为*i*和*j*，一个二维数组score[3][5]存放三位学生的5科成绩，一个一维数组average[M]存放学生平均值；
- (2) 定义符号常量。M为3，N为5；
- (3) 求每位学生的总成绩存放到average[3]；
- (4) 输出每位学生的平均成绩；



### 5.4.3 按字母顺序排列输出的问题

#### 1. 项目描述

输入三个国家的名称按字母顺序排列输出。

#### 2. 项目分析

##### (1) 功能分析

根据功能描述，程序实现的功能就是将三个字符串按字母的顺序给它排列出来。

##### (2) 数据分析

本程序需要一个二维字符数组来存储三个国家名。然而C语言规定可以把一个二维数组看成多个一维数组处理。因此本题又可以按三个一维数组处理，而每一个一维数组存放一个国家名字符串。用字符串比较函数比较各个一维数组的大小，并排序，输出结果即可。



### 3. 设计思想

- (1) 定义数组和变量。两个循环变量为*i*,*j*, 一个中间变量*p*, 一个一维数组*a*[20]和一个二维数组*b*[3][20];
- (2) 输入三个字符串。通过循环输入三个字符串*gets*(*b*[*i*]);
- (3) 把字符串存放到数组*a*中进行比较大小。
- (4) 输出按ASCII码从小到大的字符串。



---

# THANKS

---

