**数组**

数组在程序设计中，为了处理方便， 把具有相同类型的若干变量按有序的形式组织起来。这些按序排列的同类数据元素的集合称为数组。在Ｃ语言中， 数组属于构造数据类型。一个数组可以分解为多个数组元素，这些数组元素可以是基本数据类型或是构造类型。因此按数组元素的类型不同，数组又可分为数值数组、字符数组、指针数组、结构数组等各种类别。

　　本章介绍数值数组和字符数组，其余的在以后各章陆续介绍。数组类型说明 在Ｃ语言中使用数组必须先进行类型说明。 数组说明的一般形  
式为： 类型说明符 数组名 [常量表达式]，……； 其中，类型说明符是任一种基本数据类型或构造数据类型。 数组名是用户定义的数组标识符。 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：  
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。

对于数组类型说明应注意以下几点：  
1.数组的类型实际上是指数组元素的取值类型。对于同一个数组，其所有元素的数据类型都是相同的。  
2.数组名的书写规则应符合标识符的书写规定。  
3.数组名不能与其它变量名相同，例如：  
void main()  
{  
int a;  
float a[10];  
……  
}  
是错误的。  
4.方括号中常量表达式表示数组元素的个数，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]。  
5.不能在方括号中用变量来表示元素的个数， 但是可以是符号常数或常量表达式。例如：  
#define FD 5  
void main()  
{  
int a[3+2],b[7+FD];  
……  
}  
是合法的。但是下述说明方式是错误的。  
void main()  
{  
int n=5;  
int a[n];  
……  
}  
6.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,c,d,k1[10],k2[20];

数组元素的表示方法

　　数组元素是组成数组的基本单元。数组元素也是一种变量， 其标识方法为数组名后跟一个下标。 下标表示了元素在数组中的顺序号。数组元素的一般形式为： 数组名[下标] 其中的下标只能为整型常量或整型表达式。如为小数时，C编译将自动取整。例如，a[5],a[i+j],a[i++]都是合法的数组元素。 数组元素通常也称为下标变量。必须先定义数组， 才能使用下标变量。在Ｃ语言中只能逐个地使用下标变量， 而不能一次引用整个数组。 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(i=0; i<10; i++)　　printf("%d",a[i]); 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",a);  
void main()  
{  
int i,a[10];  
for(i=0;i<10;)  
a[i++]=2\*i+1;  
for(i=9;i>=0;i--)  
printf("%d",a[i]);  
printf("/n%d %d/n",a[5.2],a[5.8]);} for(i=0;i<10;)  
a[i++]=2\*i+1; for(i=9;i>=0;i--)  
printf("%d",a[i]); printf("/n%d %d/n",a[5.2],a[5.8]);  
　　 本例中用一个循环语句给a数组各元素送入奇数值，然后用第二个循环语句从大到小输出各个奇数。在第一个 for语句中，表达式3省略了。在下标变量中使用了表达式i++，用以修改循环变量。当然第二个for语句也可以这样作， Ｃ语言允许用表达式表示下标。 程序中最后一个printf语句输出了两次a[5]的值， 可以看出当下标不为整数时将自动取整。数组的赋值给数组赋值的方法除了用赋值语句对数组元素逐个赋值外， 还可采用初始化赋值和动态赋值的方法。数组初始化赋值数组初始化赋值是指在数组说明时给数组元素赋予初值。 数组初始化是在编译阶段进行的。这样将减少运行时间，提高效率。

　　初始化赋值的一般形式为： static 类型说明符 数组名[常量表达式]={值，值……值}； 其中static表示是静态存储类型， Ｃ语言规定只有静态存储数组和外部存储数组才可作初始化赋值(有关静态存储，外部存储的概念在第五章中介绍)。在{ }中的各数据值即为各元素的初值， 各值之间用逗号间隔。例如： static int a[10]={ 0,1,2,3,4,5,6,7,8,9 }; 相当于a[0]=0;a[1]=1...a[9]=9;

　　Ｃ语言对数组的初始赋值还有以下几点规定：  
1.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： static int a[10]={0,1,2,3,4};表示只给a[0]～a[4]5个元素赋值，而后5个元素自动赋0值。  
2.只能给元素逐个赋值，不能给数组整体赋值。 例如给十个元素全部赋1值，只能写为： static int a[10]={1,1,1,1,1,1,1,1,1,1};而不能写为： static int a[10]=1;  
3.如不给可初始化的数组赋初值，则全部元素均为0值。  
4.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： static int a[5]={1,2,3,4,5};可写为： static int a[]={1,2,3,4,5};动态赋值可以在程序执行过程中，对数组作动态赋值。 这时可用循环语句配合scanf函数逐个对数组元素赋值。  
void main()  
{  
int i,max,a[10];  
printf("input 10 numbers:/n");  
for(i=0;i<10;i++)  
scanf("%d",&a[i]);  
max=a[0];  
for(i=1;i<10;i++)  
if(a[i]>max) max=a[i];  
printf("maxmum=%d/n",max);  
}  
for(i=0;i<10;i++)  
scanf("%d",&a[i]);  
max=a[0];  
for(i=1;i<10;i++)  
if(a[i]>max) max=a[i];  
printf("maxmum=%d/n",max);  
　　 本例程序中第一个for语句逐个输入10个数到数组a中。 然后把a[0]送入max中。在第二个for语句中，从a[1]到a[9]逐个与max中的内容比较，若比max的值大，则把该下标变量送入max中，因此max总是在已比较过的下标变量中为最大者。比较结束，输出max的值。  
void main()  
{  
int i,j,p,q,s,a[10];  
printf("/n input 10 numbers:/n");  
for(i=0;i<10;i++)  
scanf("%d",&a[i]);  
for(i=0;i<10;i++){  
p=i;q=a[i];  
for(j=i+1;j<10;j++)  
if(q<a[j]) { p=j;q=a[j]; }  
if(i!=p)  
{s=a[i];  
a[i]=a[p];  
a[p]=s; }  
printf("%d",a[i]);  
}  
}  
for(i=0;i<10;i++)  
scanf("%d",&a[i]);  
for(i=0;i<10;i++){  
p=i;q=a[i];  
for(j=i+1;j<10;j++)  
if(q<a[j]) { p=j;q=a[j]; }  
if(i!=p)  
{ s=a[i];  
a[i]=a[p];  
a[p]=s; }  
printf("%d",a[i]);  
}  
　　 本例程序中用了两个并列的for循环语句，在第二个for 语句中又嵌套了一个循环语句。第一个for语句用于输入10个元素的初值。第二个for语句用于排序。本程序的排序采用逐个比较的方法进行。在i次循环时，把第一个元素的下标i赋于p，而把该下标变量值a[i]赋于q。然后进入小循环，从a[i+1]起到最后一个元素止逐个与a[i]作比较，有比a[i]大者则将其下标送p，元素值送q。 一次循环结束后，p即为最大元素的下标，q则为该元素值。若此时i≠p，说明p,q值均已不是进入小循环之前所赋之值，则交换a[i]和a[p]之值。 此时a[i]为已排序完毕的元素。输出该值之后转入下一次循环。对i+1以后各个元素排序。

二维数组

　　前面介绍的数组只有一个下标，称为一维数组， 其数组元素也称为单下标变量。在实际问题中有很多量是二维的或多维的， 因此Ｃ语言允许构造多维数组。多维数组元素有多个下标， 以标识它在数组中的位置，所以也称为多下标变量。 本小节只介绍二维数组，多维数组可由二维数组类推而得到。二维数组类型说明二维数组类型说明的一般形式是：  
类型说明符 数组名[常量表达式1][常量表达式2]…；  
其中常量表达式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 a[3][4]; 说明了一个三行四列的数组，数组名为a，其下标变量的类型为整型。该数组的下标变量共有3×4个，即： a[0][0],a[0][1],a[0][2],a[0][3]  
a[1][0],a[1][1],a[1][2],a[1][3]  
a[2][0],a[2][1],a[2][2],a[2][3]  
　　 二维数组在概念上是二维的，即是说其下标在两个方向上变化， 下标变量在数组中的位置也处于一个平面之中， 而不是象一维数组只是一个向量。但是，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。 在图4.1中，按行顺次存放，先存放a[0]行，再存放a[1]行，最后存放a[2]行。每行中有四个元素也是依次存放。由于数组a说明为  
int类型，该类型占两个字节的内存空间，所以每个元素均占有两个 字节(图中每一格为一字节)。

二维数组元素的表示方法

　　二维数组的元素也称为双下标变量，其表示的形式为： 数组名[下标][下标] 其中下标应为整型常量或整型表达式。例如： a[3][4] 表示a数组三行四列的元素。下标变量和数组说明在形式中有些相似，但这两者具有完全不同的含义。 数组说明的方括号中给出的是某一维的长度，即可取下标的最大值； 而数组元素中的下标是该元素在数组中的位置标识。前者只能是常量， 后者可以是常量，变量或表达式。  
一个学习小组有5个人，每个人有三门课的考试成绩。求全组分科的平均成绩和各科总平均成绩。  
课程 成绩姓名 Math　C　DBASE  
张　　　　　　80　　75　92  
王　　　　　　61　　65　71  
李　　　　　　59　　63　70  
赵　　　　　　85　　87　90  
周　　　　　　76　　77　85  
　　 可设一个二维数组a[5][3]存放五个人三门课的成绩。再设一个一维数组v[3]存放所求得各分科平均成绩，设变量l为全组各科总平均成绩。编程如下：  
void main()  
{  
int i,j,s=0,l,v[3],a[5][3];  
printf("input score/n");  
for(i=0;i<3;i++){  
for(j=0;j<5;j++)  
{ scanf("%d",&a[j][i]);  
s=s+a[j][i];}  
v[i]=s/5;  
s=0;  
}  
l=(v[0]+v[1]+v[2])/3;  
printf("math:%d/nc languag:%d/ndbase:%d/n",v[0],v[1],v[2]);  
printf("total:%d/n",l);  
} for(i=0;j<3;i++)  
for(j=0;j<5;j++)  
{ scanf("%d",&a[j][i]);  
s=s+a[j][i];}  
v[i]=s/5;  
s=0;  
}  
l=(v[0]+v[1]+v[2])/3;  
　　 程序中首先用了一个双重循环。 在内循环中依次读入某一门课程的各个学生的成绩，并把这些成绩累加起来， 退出内循环后再把该累加成绩除以5送入v[i]之中，这就是该门课程的平均成绩。外循环共循环三次，分别求出三门课各自的平均成绩并存放在v数组之中。退出外循环之后，把v[0],v[1],v[2]相加除以3即得到各科总平均成绩。最后按题意输出各个成绩。

二维数组的初始化  
　　 二维数组初始化也是在类型说明时给各下标变量赋以初值。 二维数组可按行分段赋值，也可按行连续赋值。 例如对数组a[5][3]：  
1.按行分段赋值可写为static int a[5][3]={ {80,75,92},{61,65,71},{59,63,70},{85,87,90},{76,77,85} };  
2.按行连续赋值可写为static int a[5][3]={ 80,75,92,61,65,71,59,63,70,85,87,90,76,77,85 };  
　　 这两种赋初值的结果是完全相同的。  
void main()  
{  
int i,j,s=0,l,v[3];  
static int a[5][3]={ {80,75,92},{61,65,71},{59,63,70},  
{85,87,90},{76,77,85} };  
for(i=0;i<3;i++)  
{ for(j=0;j<5;j++)  
s=s+a[j][i];  
v[i]=s/5;  
s=0;  
}  
l=(v[0]+v[1]+v[2])/3;  
printf("math:%d/nc languag:%d/ndbase:%d/n",v[0],v[1],v[2]);  
printf("total:%d/n",l);  
}  
　　 对于二维数组初始化赋值还有以下说明：  
1.可以只对部分元素赋初值，未赋初值的元素自动取0值。  
例如： static int a[3][3]={{1},{2},{3}}; 是对每一行的第一列元素赋值，未赋值的元素取0值。 赋值后各元素的值为： 1 0 02 0 03 0 0  
static int a [3][3]={{0,1},{0,0,2},{3}}; 赋值后的元素值为 0 1 00 0 23 0 0  
2.如对全部元素赋初值，则第一维的长度可以不给出。  
例如： static int a[3][3]={1,2,3,4,5,6,7,8,9}; 可以写为：static int a[][3]={1,2,3,4,5,6,7,8,9};  
　　 数组是一种构造类型的数据。 二维数组可以看作是由一维数组的嵌套而构成的。设一维数组的每个元素都又是一个数组， 就组成了二维数组。当然，前提是各元素类型必须相同。根据这样的分析，一个二维数组也可以分解为多个一维数组。 Ｃ语言允许这种分解有二维数组a[3][4]，可分解为三个一维数组，其数组名分别为a[0],a[1],a[2]。对这三个一维数组不需另作说明即可使用。这三个一维数组都有4个元素，例如：一维数组a[0]的元素为a[0][0],a[0][1],a[0][2],a[0][3]。必须强调的是，a[0],a[1],a[2]不能当作下标变量使用，它们是数组名，不是一个单纯的下标变量。

字符数组

　　用来存放字符量的数组称为字符数组。 字符数组类型说明的形式与前面介绍的数值数组相同。例如： char c[10]; 由于字符型和整型通用，也可以定义为int c[10]但这时每个数组元素占2个字节的内存单元。字符数组也可以是二维或多维数组，例如： char c[5][10];即为二维字符数组。 字符数组也允许在类型说明时作初始化赋值。例如： static char c[10]={`c`,` `,`p`,`r`,o`,g`,r`,`a`,`m`};赋值后各元素的值为： 数组C c[0]c[1]c[2]c[3]c[4]c [5]c[6]c[7]c[8]c[9]其中c[9]未赋值，由系统自动赋予0值。 当对全体元素赋初值时也可以省去长度说明。例如： static char c[]={`c`,` `,`p`,`r`,`o`,`g`,`r`,`a`,`m`};这时C数组的长度自动定为9。  
main()  
{  
int i,j;  
char a[][5]={{'B','A','S','I','C',},{'d','B','A','S','E'}};  
for(i=0;i<=1;i++)  
{  
for(j=0;j<=4;j++)  
printf("%c",a[i][j]);  
printf("/n");  
}  
}  
　　 本例的二维字符数组由于在初始化时全部元素都赋以初值， 因此一维下标的长度可以不加以说明。字符串在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。在2.1.4节介绍字符串常量时，已说明字符串总是以'/0'作为串的结束符。因此当把一个字符串存入一个数组时， 也把结束符'/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：  
static char c[]={'c', ' ','p','r','o','g','r','a','m'}; 可写为：  
static char c[]={"C program"}; 或去掉{}写为：  
sratic char c[]="C program";  
　　 用字符串方式赋值比用字符逐个赋值要多占一个字节， 用于存放字符串结束标志'/0'。上面的数组c在内存中的实际存放情况为： C program/0`/0'是由C编译系统自动加上的。由于采用了`/0'标志，所以在用字符串赋初值时一般无须指定数组的长度， 而由系统自行处理。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。 除了上述用字符串赋初值的办法外，还可用printf函数和scanf函数一次性输出输入一个字符数组中的字符串， 而不必使用循环语句逐个地输入输出每个字符。  
void main()  
{  
static char c[]="BASIC/ndBASE";  
printf("%s/n",c);  
} printf("%s/n",c);  
注意在本例的printf函数中，使用的格式字符串为“%s”， 表示输出的是一个字符串。而在输出表列中给出数组名则可。 不能写为： printf("%s",c[]);  
void main()  
{  
char st[15];  
printf("input string:/n");  
scanf("%s",st);  
printf("%s/n",st);  
} char st[15];  
　　 本例中由于定义数组长度为15， 因此输入的字符串长度必须小于15，以留出一个字节用于存放字符串结束标志`/0`。 应该说明的是，对一个字符数组，如果不作初始化赋值，则必须说明数组长度。还应该特别注意的是，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。例如运行例4.8，当输入的字符串中含有空格时，运行情况为： input string:this is a book this 从输出结果可以看出空格以后的字符都未能输出。 为了避免这种情况， 可多设几个字符数组分段存放含空格的串。程序可改写如下：  
Lesson  
void main()  
{  
char st1[6],st2[6],st3[6],st4[6];  
printf("input string:/n");  
scanf("%s%s%s%s",st1,st2,st3,st4);  
printf("%s %s %s %s/n",st1,st2,st3,st4);  
}  
　　 本程序分别设了四个数组， 输入的一行字符的空格分段分别装入四个数组。然后分别输出这四个数组中的字符串。在前面介绍过，scanf的各输入项必须以地址方式出现，如 &a,&b等。但在例4.8中却是以数组名方式出现的，这是为什么呢?这是由于在Ｃ语言中规定，数组名就代表了该数组的首地址。 整个数组是以首地址开头的一块连续的内存单元。如有字符数组char c[10]，在内存可表示如图4.2。设数组c的首地址为2000，也就是说c[0]单元地址为2000。则数组名c就代表这个首地址。因此在c前面不能再加地址运算符&。如写作scanf("%s",&c);则是错误的。 在执行函数printf("%s",c) 时，按数组名c找到首地址，然后逐个输出数组中各个字符直到遇到字符串终止标志'/0'为止。

字符串常用函数

　　Ｃ语言提供了丰富的字符串处理函数， 大致可分为字符串的输入、输出、合并、修改、比较、转换、复制、搜索几类。 使用这些函数可大大减轻编程的负担。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。 即在屏幕上显示该字符串  
#include"stdio.h"  
main()  
{  
static char c[]="BASIC/ndBASE";  
puts(c);  
}  
static char c[]="BASIC/ndBASE";  
puts(c);  
　　 从程序中可以看出puts函数中可以使用转义字符， 因此输出结果成为两行。puts函数完全可以由printf函数取代。 当需要按一定格式输出时，通常使用printf函数。  
2.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个字符串。 本函数得到一个函数值，即为该字符数组的首地址。  
#include"stdio.h"  
main()  
{  
char st[15];  
printf("input string:/n");  
gets(st);  
puts(st);  
}  
　　 可以看出当输入的字符串中含有空格时，输出仍为全部字符串。说明gets函数并不以空格作为字符串输入结束的标志， 而只以回车作为输入结束。这是与scanf函数不同的。  
3.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后的串标志“/0”。本函数返回值是字符数组1的首地址。  
#include"string.h"  
main()  
{  
static char st1[30]="My name is ";  
int st2[10];  
printf("input your name:/n");  
gets(st2);  
strcat(st1,st2);  
puts(st1);  
}  
static char st1[30]="My name is ";  
int st2[10];  
printf("input your name:/n");  
gets(st2);  
strcat(st1,st2);  
本程序把初始化赋值的字符数组与动态赋值的字符串连接起来。 要注意的是，字符数组1应定义足够的长度，否则不能全部装入被连接的字符串  
4.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。字符数名2， 也可以是一个字符串常量。这时相当于把一个字符串赋予一个字符数组。  
#include"string.h"  
main()  
{  
static char st1[15],st2[]="C Language";  
strcpy(st1,st2);  
puts(st1);printf("/n");  
}  
static char st1[15],st2[]="C Language";  
strcpy(st1,st2);  
本函数要求字符数组1应有足够的长度，否则不能全部装入所拷贝的字符串。  
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＝0；  
字符串2〉字符串2，返回值〉0；  
字符串1〈字符串2，返回值〈0。  
本函数也可用于比较两个字符串常量，或比较数组和字符串常量。  
#include"string.h"  
main()  
{ int k;  
static char st1[15],st2[]="C Language";  
printf("input a string:/n");  
gets(st1);  
k=strcmp(st1,st2);  
if(k==0) printf("st1=st2/n");  
if(k>0) printf("st1>st2/n");  
if(k<0) printf("st1<st2/n");  
}  
{ int k;  
static char st1[15],st2[]="C Language";  
printf("input a string:/n");  
gets(st1);  
k=strcmp(st1,st2);  
if(k==0) printf("st1=st2/n");  
if(k>0) printf("st1>st2/n");  
if(k<0) printf("st1<st2/n");  
}  
　　 本程序中把输入的字符串和数组st2中的串比较，比较结果返回到k中，根据k值再输出结果提示串。当输入为dbase时，由ASCII 码可知“dBASE”大于“C Language”故k〉0,输出结果“st1>st2”。  
6.测字符串长度函数strlen 格式： strlen(字符数组名) 功能：测字符串的实际长度(不含字符串结束标志‘/0’) 并作为函数返回值。  
#include"string.h"  
main()  
{ int k;  
static char st[]="C language";  
k=strlen(st);  
printf("The lenth of the string is %d/n",k);  
}

程序举例

　　把一个整数按大小顺序插入已排好序的数组中。 为了把一个数按大小插入已排好序的数组中， 应首先确定排序是从大到小还是从小到大进行的。设排序是从大到小进序的， 则可把欲插入的数与数组中各数逐个比较， 当找到第一个比插入数小的元素i时，该元素之前即为插入位置。然后从数组最后一个元素开始到该元素为止，逐个后移一个单元。最后把插入数赋予元素i即可。如果被插入数比所有的元素值都小则插入最后位置。  
main()  
{  
int i,j,p,q,s,n,a[11]={127,3,6,28,54,68,87,105,162,18};  
for(i=0;i<10;i++)  
{ p=i;q=a[i];  
for(j=i+1;j<10;j++)  
if(q<a[j]) {p=j;q=a[j];}  
if(p!=i)  
{  
s=a[i];  
a[i]=a[p];  
a[p]=s;  
}  
printf("%d ",a[i]);  
}  
printf("/ninput number:/n");  
scanf("%d",&n);  
for(i=0;i<10;i++)  
if(n>a[i])  
{for(s=9;s>=i;s--) a[s+1]=a[s];  
break;}  
a[i]=n;  
for(i=0;i<=10;i++)  
printf("%d ",a[i]);  
printf("/n");  
}  
scanf("%d",&n);  
for(i=0;i<10;i++)  
if(n>a[i])  
{ for(s=9;s>=i;s--) a[s+1]=a[s];  
break; }  
a[i]=n; 本程序首先对数组a中的10个数从大到小排序并输出排序结果。然后输入要插入的整数n。再用一个for语句把n和数组元素逐个比较，如果发现有n>a[i]时，则由一个内循环把i以下各元素值顺次后移一个单元。后移应从后向前进行(从a[9]开始到a[i]为止)。 后移结束跳出外循环。插入点为i，把n赋予a[i]即可。 如所有的元素均大于被插入数，则并未进行过后移工作。此时i=10，结果是把n赋于a[10]。最后一个循环输出插入数后的数组各元素值。程序运行时，输入数47。从结果中可以看出47已插入到54和 28之间。

　　在二维数组a中选出各行最大的元素组成一个一维数组b。 a=3 16 87 65 4 32 11 108 10 25 12 37b=(87 108 37) 本题的编程思路是，在数组A的每一行中寻找最大的元素，找到之后把该值赋予数组B相应的元素即可。程序如下：  
main()  
{  
static int a[][4]={3,16,87,65,4,32,11,108,10,25,12,27};  
int b[3],i,j,l;  
for(i=0;i<=2;i++)  
{ l=a[i][0];  
for(j=1;j<=3;j++)  
if(a[i][j]>l) l=a[i][j];  
b[i]=l;}  
printf("/narray a:/n");  
for(i=0;i<=2;i++)  
{ for(j=0;j<=3;j++)  
printf("%5d",a[i][j]);  
printf("/n");}  
printf("/narray b:/n");  
for(i=0;i<=2;i++)  
printf("%5d",b[i]);  
printf("/n");  
}  
for(i=0;i<=2;i++){  
l=a[i][0];  
for(j=1;j<=3;j++)  
if(a[i][j]>l) l=a[i][j];  
b[i]=l;  
}  
　　 程序中第一个for语句中又嵌套了一个for语句组成了双重循环。外循环控制逐行处理，并把每行的第0列元素赋予l。进入内循环后，把l与后面各列元素比较，并把比l大者赋予l。内循环结束时l 即为该行最大的元素，然后把l值赋予b[i]。等外循环全部完成时，数组b中已装入了a各行中的最大值。后面的两个 for语句分别输出数组a和数组b。

　　输入五个国家的名称按字母顺序排列输出。  
　　 本题编程思路如下：五个国家名应由一个二维字符数组来处理。然而Ｃ语言规定可以把一个二维数组当成多个一维数组处理。 因此本题又可以按五个一维数组处理， 而每一个一维数组就是一个国家名字符串。用字符串比较函数比较各一维数组的大小，并排序， 输出结果即可。  
编程如下：  
void main()  
{  
char st[20],cs[5][20];  
int i,j,p;  
printf("input country's name:/n");  
for(i=0;i<5;i++)  
gets(cs[i]);  
printf("/n");  
for(i=0;i<5;i++)  
{ p=i;strcpy(st,cs[i]);  
for(j=i+1;j<5;j++)  
if(strcmp(cs[j],st)<0) {p=j;strcpy(st,cs[j]);}  
if(p!=i)  
{  
strcpy(st,cs[i]);  
strcpy(cs[i],cs[p]);  
strcpy(cs[p],st);  
}  
puts(cs[i]);}printf("/n");  
}  
for(i=0;i<5;i++)  
{ p=i;strcpy(st,cs[i]);  
for(j=i+1;j<5;j++)  
if(strcmp(cs[j],st)<0) { p=j;strcpy(st,cs[j]);}  
if(p!=i)  
{  
strcpy(st,cs[i]);  
strcpy(cs[i],cs[p]);  
strcpy(cs[p],st);  
}  
　　 本程序的第一个for语句中，用gets函数输入五个国家名字符串。上面说过Ｃ语言允许把一个二维数组按多个一维数组处理， 本程序说明cs[5][20]为二维字符数组，可分为五个一维数组cs[0]，cs[1]，cs[2]，cs[3]，cs[4]。因此在gets函数中使用cs[i]是合法的。 在第二个for语句中又嵌套了一个for语句组成双重循环。 这个双重循环完成按字母顺序排序的工作。在外层循环中把字符数组cs[i]中的国名字符串拷贝到数组st中，并把下标i赋予P。 进入内层循环后，把st与cs[i]以后的各字符串作比较，若有比st小者则把该字符串拷贝到st中，并把其下标赋予p。内循环完成后如p不等于 i 说明有比cs[i]更小的字符串出现，因此交换cs[i]和st的内容。 至此已确定了数组cs的第i号元素的排序值。然后输出该字符串。在外循环全部完成之后即完成全部排序和输出。

本章小结

1.数组是程序设计中最常用的数据结构。数组可分为数值数组(整数组，实数组)，字符数组以及后面将要介绍的指针数组，结构数组等。

2.数组可以是一维的，二维的或多维的。

3.数组类型说明由类型说明符、数组名、数组长度 (数组元素个数)三部分组成。数组元素又称为下标变量。 数组的类型是指下标变量取值的类型。

4.对数组的赋值可以用数组初始化赋值， 输入函数动态赋值和赋值语句赋值三种方法实现。 对数值数组不能用赋值语句整体赋值、输入或输出，而必须用循环语句逐个对数组元素进行操作。