第6章 数 组

6.1 项目导入

前几章使用的都是属于基本类型(整型、字符型、实型)的数据,C语言还提供了构造类型的数据,它们有数组类型、结构体类型和共用体类型。构造类型数据是由基本类型数据按一定规则组成的,因此它们又被称为“导出类型”。

数组是同类型数据的有序集合。数组中的每一个元素都属于同一个数据类型。用一个统一的数组名和下标来唯一地确定数组中的元素。

本程序中对机选号码(1到35之间7个不重复的整数)的存储,就是采用数组的方式来存储的。具体操作如图6.1所示:

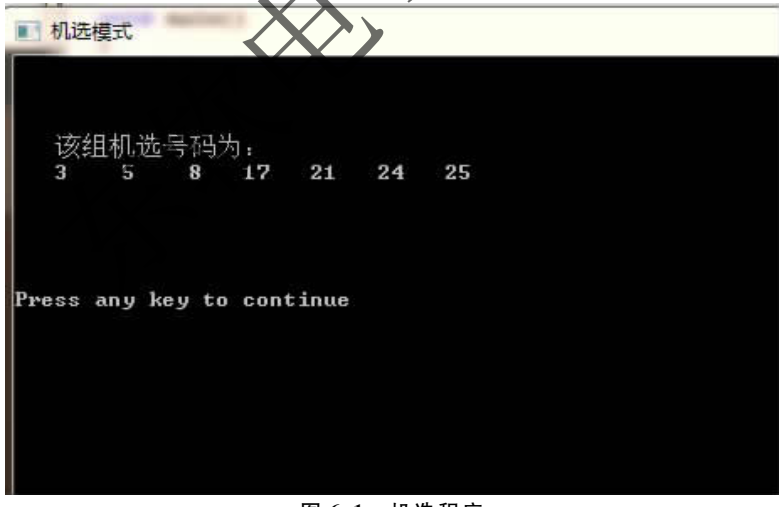

图6.1 机选程序

6.2 知识点概述

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。

6.2.1　一维数组的定义和引用

1.一维数组的定义方式

在 C 语言中使用数组必须先进行定义。

一维数组的定义方式为：

类型说明符 数组名［常量表达式］；

例如：int a[10]；

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];
```

2.一维数组元素的引用

数组必须先定义,然后使用。C语言规定只能逐个引用数组元素而不能一次引用整个数组。

数组元素的表示形式为

```
数组名[下标]
```

其中下标只能为整型常量或整型表达式。

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 6.1】 以下程序的运行结果。

```
1 #include <stdio.h>
2 void main()
3 {
4     int i, a[10];
5     for(i=0; i<=9; i++)
6         a[i]=i;
7     for(i=9; i>=0; i--)
8         printf("%d", a[i]);
9     printf("\n");
10 }
```

该程序的运行结果如图 6.2 所示:

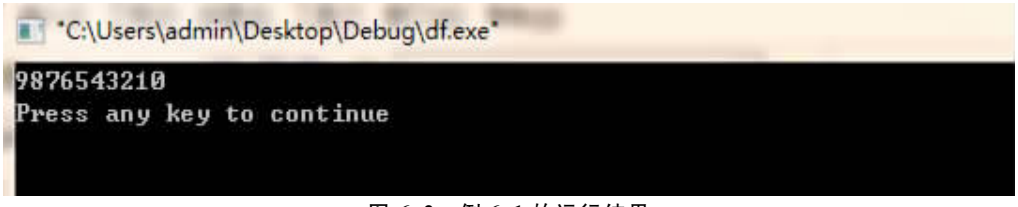

图 6.2 例 6.1 的运行结果

笔记：上两例中用一个循环语句给 a 数组各元素送入数值，然后用第二个循环语句输出各个数。在第二个 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。

3.一维数组的初始化

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。

数组初始化赋值是指在数组定义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：

```
类型说明符 数组名[常量表达式]={值,值……值};
```

其中在{ }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：

```
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 语言对数组的初始化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：

```
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：

```
int a[10]={1,1,1,1,1,1,1,1,1,1};
```

而不能写为：

```
int a[10]=1;
```

(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};
```

4.一维数组程序举例

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。

【例 6.2】 以下程序的运行结果。

```
1 #include <stdio.h>
2 void main()
3 {
4     int i,max,a[10];
5     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);
}
```

该程序的运行结果如图 6.3 所示：

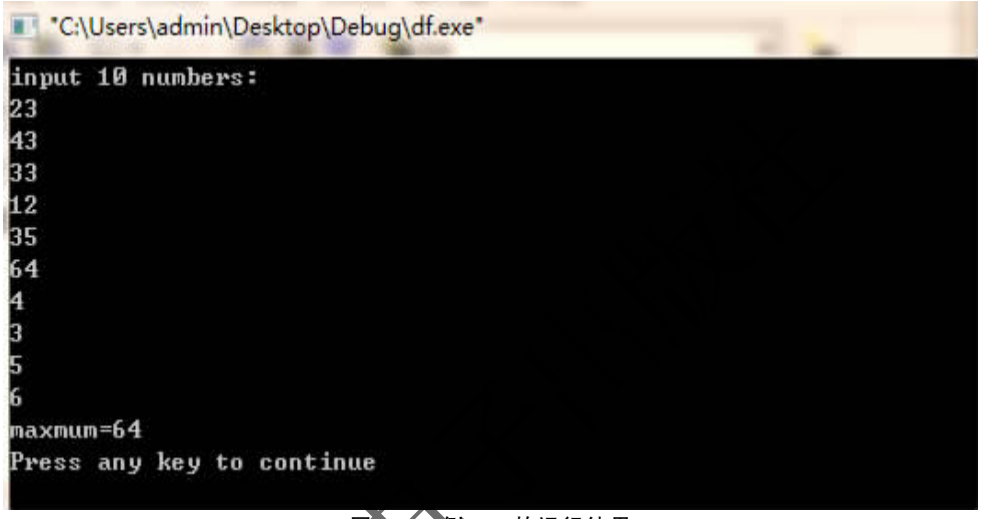

图 6.3 例 6.2 的运行结果

笔记：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。

【例 6.3】 用数组来处理求 Fibonacci 数列问题

```
#include <stdio.h>
void main()
{
    int i;
    int f[20]={1,1};
    for(i=2;i<20;i++)
        f[i]=f[i-2]+f[i-1];
    for(i=0;i<20;i++)
    {
        if(i%5==0) printf("\n");
        printf("%12d",f[i]);
    }
}
```

该程序的运行结果如图 6.4 所示：

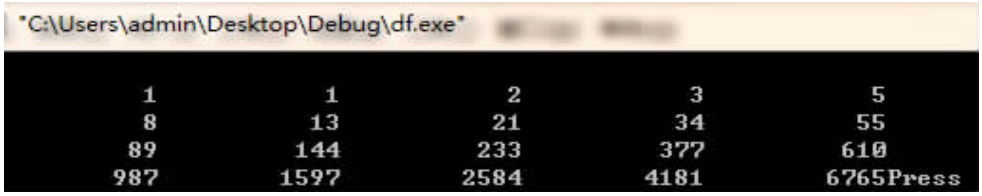

图 6.4　例 6.3 的运行结果

笔记：本例中用数组来把有规律的数列中各数据计算出来并存储于数组中，以方便下面的计算。

【例 6.4】　用起泡法对 8 个数排序（由小到大）

起泡排序也叫冒泡排序，它是一种简单的排序方法。起泡排序是依次比较相邻两个数据的排序，不符合顺序则交换记录。

设有 8 个待排序记录的排序码为(36,25,48,12,25,65,43,57)，使用从上向下的扫描的起泡法进行排序。表 6.1 给出了第一趟起泡排序过程中各数据的位置变化情况，从 R[0]开始比较相邻记录的排序码，让大的排序码“下沉”，直到比较到 R[7]为止，这样第一趟排序的结果是排序码最大的记录由 R[5]渐渐“沉入”到最低位置 R[7]，在接下来的第二趟排序中，R[7]就不再参与比较。

表 6.1　　冒泡排序第一趟排序结果

	R[0]	R[1]	R[2]	R[3]	R[4]	R[5]	R[6]	R[7]
原始数	36 ⟷	25	48	12	25	65	43	57
第一次比较	25	36 ⟷	48	12	25	65	43	57
第二次比较	25	36	48 ⟷	12	25	65	43	57
第三次比较	25	36	12	48 ⟷	25	65	43	57
第四次比较	25	36	12	25	48 ⟷	65	43	57
第五次比较	25	36	12	25	48	65 ⟷	43	57
第六次比较	25	36	12	25	48	43	65 ⟷	57
第七次比较	25	36	12	25	48	43	57	65

表 6.2 给出了整个排序过程中每一趟排序结果，其中方括号“[]”括起来的排序码是无序区，方括号后面的排序码是本趟排序中“沉底”的最大排序码。

表 6.2　　冒泡排序排序过程

	R[0]	R[1]	R[2]	R[3]	R[4]	R[5]	R[6]	R[7]
原始数	[36	25	48	12	25	65	43	57]
第一趟排序结果	[25	36	12	25	48	43	57]	65
第二趟排序结果	[25	12	25	36	43	48]	57	65
第三趟排序结果	[12	25	25	36	43]	48	57	65
第四趟排序结果	[12	25	25	36]	43	48	57	65
第五趟排序结果	[12	25	25]	36	43	48	57	65
第六趟排序结果	[12	25]	25	36	43	48	57	65
第七趟排序结果	[12]	25	25	36	43	48	57	65

```
1 void main()
2 {
3     int i,j,temp;
4     int R[]={36,25,48,12,25,65,43,57};
5     for(i=1;i<8;i++)
6         for(j=0;j<8-i;j++)
7             if(R[j]>R[j+1])
8             {
9                 temp=R[j];R[j]=R[j+1]; R[j+1]=temp;
10            }
11      for(i=0;i<8;i++)
12      printf("%4d",R[i]);
13 }
```

该程序的运行结果如图 6.5 所示：

图 6.5 例 6.4 的运行结果

6.2.2 二维数组的定义和引用

1.二维数组的定义

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。

二维数组定义的形式为

类型说明符 数组名[常量表达式][常量表达式];

例如：float a[3][4], b[5][10];

二维数组可被看作是一种特殊的一维数组：它的元素又是一个一维数组。

二维数组中元素排列的顺序是按行存放的。即在内存中先顺序存放第一行的元素，再存放第二行的元素。C 语言允许使用多维数组。

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。

即：

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）。

2.二维数组元素的引用

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]=1;

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。

【例 6.5】 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。

	张	王	李	赵	周
Math	80	61	59	85	76
C	75	65	63	87	77
Foxpro	92	71	70	90	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 average 为全组各科总平均成绩。编程如下：

```
1 void main()
2 {
3     int i,j,s=0,average,v[3],a[5][3];
4     printf("input score\n");
5     for(i=0;i<3;i++)
6     {
7         for(j=0;j<5;j++)
8         { scanf("%d",&a[j][i]);
9           s=s+a[j][i];
10        }
11        v[i]=s/5;
12        s=0;
13    }
14    average =(v[0]+v[1]+v[2])/3;
15    printf("math:%d\nc languag:%d\nbase:%d\n",v[0],v[1],v[2]);
```

```
16    printf("total: %d\n", average );
17 }
```

该程序的运行结果如图 6.6 所示：

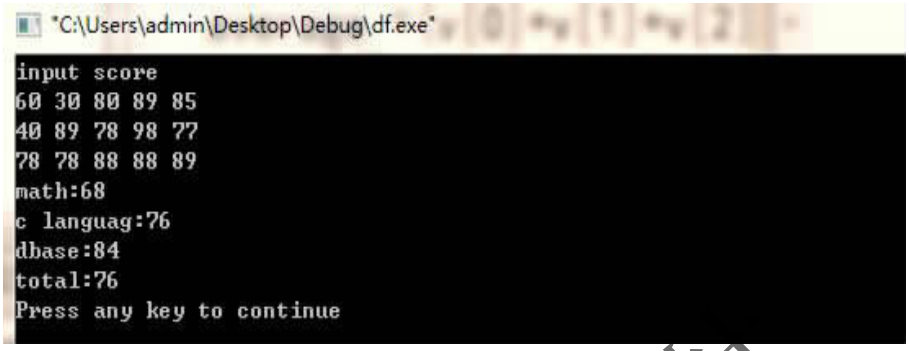

图 6.6　例 6.5 的运行结果

笔记：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。

3.二维数组的初始化

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：

```
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：

```
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。

【例 6.6】 以下程序运行结果。

```
1 void main()
2 {
3     int i,j,s=0, average,v[3];
4     int a[5][3]={{80,75,92},{61,65,71},{59,63,70},{85,87,90},{76,77,85}};
5     for(i=0;i<3;i++)
6         { for(j=0;j<5;j++)
7             s=s+a[j][i];
8           v[i]=s/5;
9           s=0;
10        }
11    average=(v[0]+v[1]+v[2])/3;
12    printf("math: %d\nc languag: %d\ndFoxpro: %d\n",v[0],v[1],v[2]);
```

```
13    printf("total: %d\n", average);
14 }
```

该程序的运行结果如图 6.7 所示：

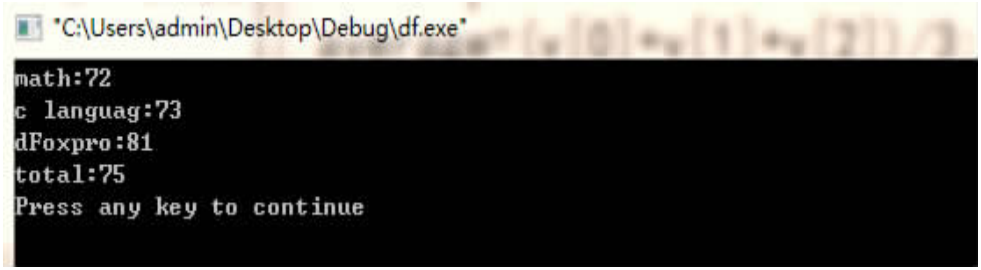

图 6.7　例 6.6 的运行结果

笔记：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：

```
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 0
2 0 0
3 0 0
```

```
int a [3][3]={{0,1},{0,0,2},{3}};
```

赋值后的元素值为：

```
0 1 0
0 0 2
3 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：

```
int a[3][3]={1,2,3,4,5,6,7,8,9};
```

可以写为：

```
int a[][3]={1,2,3,4,5,6,7,8,9};
```

(3)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。

如二维数组 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。

4.二维数组程序举例

【例 6.7】 将一个二维数组行和列的元素互换,存到另一个二维数组中。例如:

$$a=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix} \quad b=\begin{bmatrix}1 & 4\\ 2 & 5\\ 3 & 6\end{bmatrix}$$

```
#define N 2
#define M 3
void main()
{
    int a[N][M]={1,2,3,4,5,6},b[M][N];
    int i,j;
    for(i=0;i<N;i++)
      for(j=0;j<M;j++)
        b[j][i]=a[i][j];

    for(i=0;i<M;i++)
    {
        for(j=0;j<N;j++)
          printf("%d  ",b[i][j]);
        printf("\n");
    }
}
```

该程序的运行结果如图 6.8 所示:

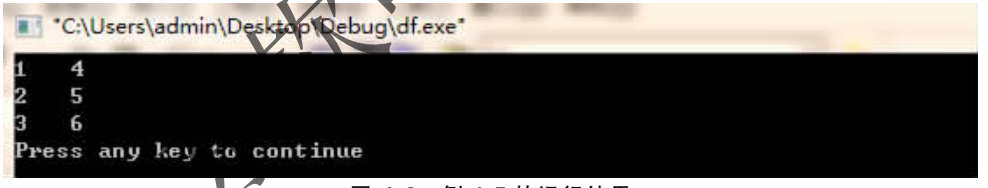

图 6.8 例 6.7 的运行结果

6.2.3 字符数组

用来存放字符量的数组称为字符数组。

1.字符数组的定义

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。

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 c[10]={'c', ' ', 'p','r','o','g','r','a','m'};
```

赋值后各元素的值为：

数组 C　　c[0]的值为'c'
　　c[1]的值为' '
　　c[2]的值为'p'
　　c[3]的值为'r'
　　c[4]的值为'o'
　　c[5]的值为'g'
　　c[6]的值为'r'
　　c[7]的值为'a'
　　c[8]的值为'm'

其中 c[9]未赋值，由自动赋予 0 值。

当对全体元素赋初值时也可以省去长度说明。

例如：

```
char c[]={'c',' ','p','r','o','g','r','a','m'};
```

这时 C 数组的长度自动定为 9。

3.字符数组的引用

【例 6.8】 输出一个字符串。

```
#include <stdio.h>
void main()
{
    char c[]={'I','a','m',",'a',",'b','o','y'};
    int i;
    for(i=0; i<10; i++)
        printf("%c", c[i]);
    printf("\n");
}
```

该程序的运行结果如下：

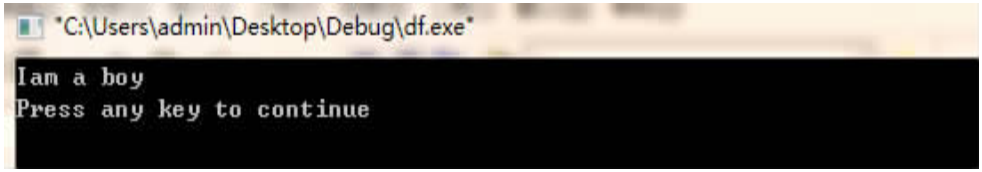

图 6.9　例 6.8 的运行结果

笔记：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，一维下标的长度可以不加以说明。

4.字符串和字符串结束标志

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。前面介绍字符串常量时,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:

```
char c[]={'C',' ','p','r','o','g','r','a','m'};
```

可写为:

```
char c[]={"C program"};
```

或去掉{}写为:

```
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		p	r	o	g	r	a	m	\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。

5.字符串数组的输入输出

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。

【例6.9】 写出以下程序的运行结果。

```
1 void main()
2 {
3     char c[]="BASIC\ndBASE";
4     printf("%s\n",c);
5 }
```

该程序的运行结果如下:

```
BASIC
dBASE
Press any key to continue
```

图6.10 例6.9的运行结果

笔记: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[]);
```

【例6.10】 写出以下程序的运行结果。

```
1 void main()
2 {
```

```
3      char st[15];
4      printf("input string:\n");
5      scanf("%s",st);
6      printf("%s\n",st);
7 }
```

该程序的运行结果如图 6.11 所示：

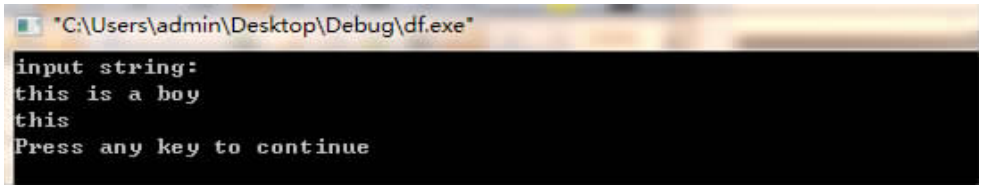

图 6.11　例 6.10 的运行结果

笔记：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。

例如当输入的字符串中含有空格时，运行情况为：

```
input string:
this is a book
```

输出为：

```
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：

【例 6.11】 写出以下程序的运行结果。

```
1 void main()
2 {    char st1[6],st2[6],st3[6],st4[6];
3      printf("input string:\n");
4      scanf("%s%s%s%s",st1,st2,st3,st4);
5       printf("%s %s %s %s\n",st1,st2,st3,st4);
6 }
```

该程序的运行结果如图 6.12 所示：

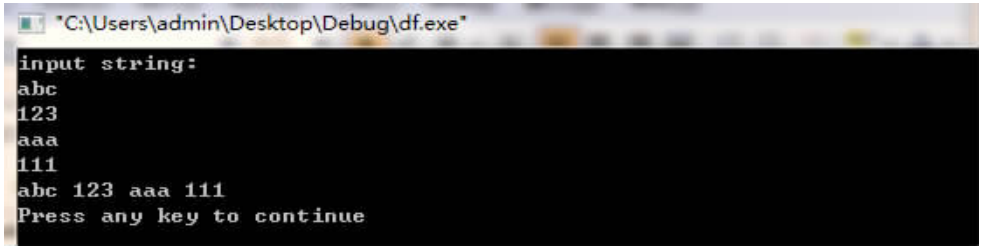

图 6.12　例 6.11 的运行结果

笔记：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。但在前例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。

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

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。

6.字符串处理函数

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

函数原型:

```
int puts(char * str);
```

功能:

把 str 指向的字符串输出到标准输出设备,将'\0'转换为换行符'\n'

返回值:

若执行成功,返回非负值,否则,返回 EOF

【例 6.12】 写出以下程序的运行结果。

```
1 #include"stdio.h"
2 void main()
3 {
4     char c[]="BASIC\ndBASE";
5     puts(c);
6 }
```

该程序的运行结果如图 6.13 所示:

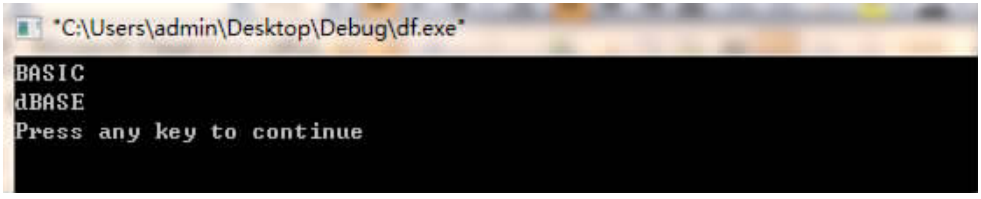

图 6.13 例 6.12 的运行结果

笔记: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

函数原型：

```
char * gets(char * str);
```

功能：

从终端输入一个字符串到字符数组 str

返回值：

若执行成功，返回字符数组的起始地址，否则，返回 NULL。

【例 6.13】 写出以下程序的运行结果。

```
1 #include"stdio.h"
2 void main()
3 {
4     char st[15];
5     printf("input string:\n");
6     gets(st);
7     puts(st);
8 }
```

该程序的运行结果如图 6.14 所示：

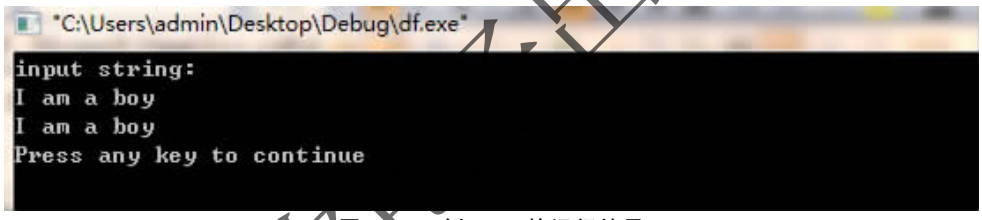

图 6.14　例 6.13 的运行结果

笔记：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

函数原型：

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

功能：

把字符串 str2 接到 str1 后面，str1 最后面的'\0'被取消

返回值：str1

【例 6.14】 写出以下程序的运行结果。

```
1 #include"string.h"
2 void main()
3 {
4    static char st1[30]="My name is ";
5     int st2[10];
6     printf("input your name:\n");
7     gets(st2);
```

```
8      strcat(st1,st2);
9      puts(st1);
10 }
```

该程序的运行结果如图 6.15 所示：

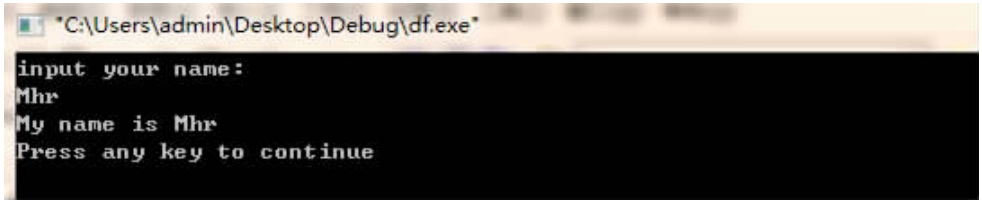

图 6.15　例 6.14 的运行结果

笔记：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

函数原型：

```
char * strcpy(char * str1, char * str2);
```

功能：

把 str2 指向的字符串复制到 str1 中去

返回值：

str1

【例 6.15】 写出以下程序的运行结果。

```
1 #include"string.h"
2 void main()
3 {
4      char st1[15],st2[]="C Language";
5      strcpy(st1,st2);
6      puts(st1);printf("\n");
7 }
```

该程序的运行结果如图 6.16 所示：

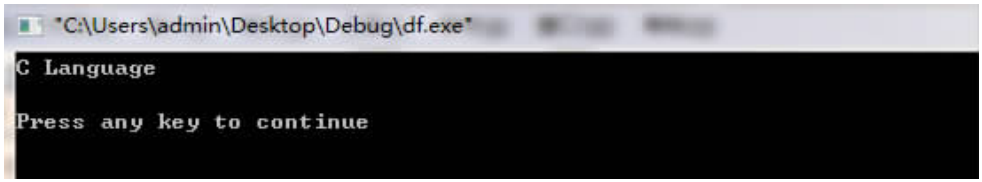

图 6.16　例 6.15 的运行结果

笔记：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

函数原型：

```
int strcmp(char * str1, char * str2);
```

功能：

比较两个字符串 str1、str2

返回值：str1＜str2，返回负数；

str1＝str2，返回 0；

str1＞str2，返回正数

【例 6.16】 写出以下程序的运行结果。

```
1 #include"string.h"
2 void main()
3 {
4     int k;
5     static char st1[15],st2[]="C Language";
6     printf("input a string:\n");
7     gets(st1);
8     k=strcmp(st1,st2);
9     if(k==0) printf("st1=st2\n");
10    if(k>0) printf("st1>st2\n");
11    if(k<0) printf("st1<st2\n");
12 }
```

该程序的运行结果如图 6.17 所示：

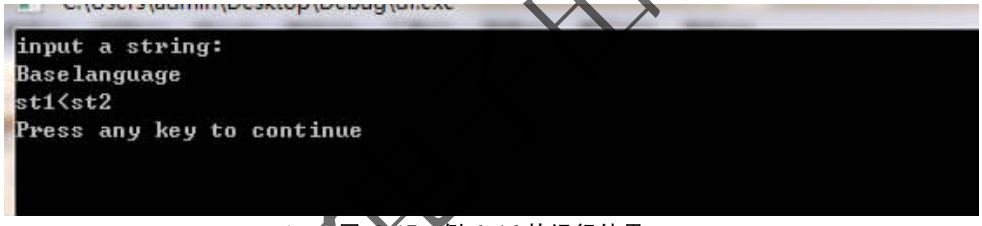

图 6.17　例 6.16 的运行结果

笔记：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

函数原型：

```
unsigned int strlen(char * str);
```

功能：

统计字符串 str 中字符的个数(不包括终止符‘\0’)

返回值：

返回字符个数

【例 6.17】 写出以下程序的运行结果。

```
1 #include"string.h"
2 void main()
3 {
4     int k;
5     static char st[]="C language";
6     k=strlen(st);
```

```
7      printf("The lenth of the string is %d\n",k);
8 }
```

该程序的运行结果如图 6.18 所示：

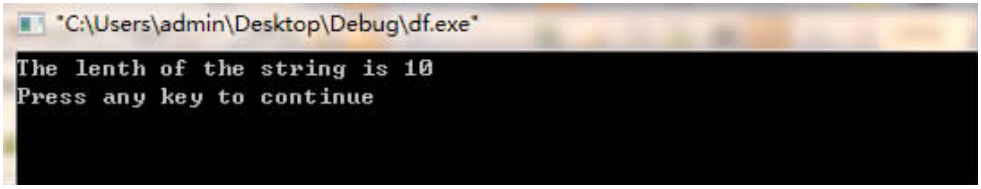

图 6.18　例 6.17 的运行结果

(7)strlwr 函数

函数原型：

```
char * strlwr(char * str);
```

功能：

将字符串 str 中大写字母换成小写字母

返回值：str

(8)strupr 函数

函数原型：

```
char * strupr(char * str);
```

功能：

将字符串 str 中小写字母换成大写字母

返回值：str

6.3　项目结合

应用数组结构来存储一组机选号码(1 到 35 之间 7 个不重复的整数)。数组就是用来存储同一类型的有序数据的集合的。本项目中需要存储 1 到 35 之间 7 个不重复的整数，所以数组的长度应该至少为 7，数组的类型为整数类型 int。

彩票系统机选功能，首先我们进行功能分析：

1. 首先系统开辟数组空间，接收随机产生的 7 个不重复的数。

```
int a[7];
```

2. 设置标题为“机选模式”。

```
system("title 机选模式");
```

3. 先随机产生一个数 1 到 35 之间的整数，放到数组的第一个元素中。

```
srand((unsigned)time(NULL));//时间种子，避免每次都一样。
a[0]=rand() % 35+1;
```

4. 依次循环产生其他几个数

5. 程序结束

接下来程序分步实现：

1. 初始化工作。

```
int a[7];
int i=1,j=0,t,x;

system("title 机选模式");

srand((unsigned)time(NULL));
a[0]=rand()%35+1;
```

2. 下面要循环生成其他几个不重复的数。判断是否到达 7 个，如果到达，循环结束。

```
if(i==7)
    break;
```

3. 每产生一个随机数，判断是否重复，如果重复就 break。挨个元素比较后，如果不重复就把该数放到数组的下一个位置。

```
x=rand()%35+1;
for(j=0;j<=i;j++)
{
    if(x==a[j])
    {  break;}
}
if(j>i)
{
    a[i]=x;
    i++;
}
```

4. 对随机生成的 7 个不重复的数据，进行排序。

```
for(j=0;j<6;j++)
    for(i=0;i<6-j;i++)
        if(a[i]>a[i+1])
        {
            t=a[i];
            a[i]=a[i+1];
            a[i+1]=t;
        }
```

5. 打印排好序的 7 个数。

```
printf("\n\n\n 该组机选号码为:\n");
for(i=0;i<7;i++)
    printf("%4d ",a[i]);
printf("\n\n\n\n\n");
```

程序源码如下：

```
#include<stdio.h>
#include<windows.h>
#include<time.h>
void main()
{
    int a[7];
    int i=1,j=0,t,x;

    system("title 机选模式");

    srand((unsigned)time(NULL));
    a[0]=rand()%35+1;
    while(1)
    {
        if(i==7)
            break;
        x=rand()%35+1;
        for(j=0;j<=i;j++)
            {
                if(x==a[j])
                {
                    break;
                }
            }
        if(j>i)
        {
            a[i]=x;
            i++;
        }

    }

    for(j=0;j<6;j++)
        for(i=0;i<6-j;i++)
            if(a[i]>a[i+1])
            {
                t=a[i];
                a[i]=a[i+1];
                a[i+1]=t;
            }

    printf("\n\n\n该组机选号码为:\n");
```

```
    for(i=0;i<7;i++)
        printf("%4d",a[i]);
    printf("\n\n\n\n\n");

}
```

6.4　实作强化

题目 1:有一个已排好序的数组,要求输入一个数后,按原来排序的规律将它插入数组中。

```
void main()
{
    int a[10]={2,4,6,8,10,12,14,16};
    inti,x;
    printf("请输入要插入的数:");
    scanf("%d",&x);
    for(i=7;i>=0;i--)
        if(a[i]>x)
            a[i+1]=a[i];
        else
            break;
    a[i+1]=x;
    for(i=0;i<9;i++)
        printf("%d  ",a[i]);
}
```

题目 2:将一个数组中的值按逆序重新存放。例如,原来顺序为 8,6,5,4,1。要求改为 1,4,5,6,8。

```
#define N 5
void main()
{
    int a[N]={8,6,5,4,1};
    int i,t;
    for(i=0;i<N/2;i++)
    {
        t=a[i];
        a[i]=a[N-i-1];
        a[N-i-1]=t;
    }
    for(i=0;i<N;i++)
        printf("%d  ",a[i]);
}
```

题目3:输出以下的杨辉三角形(要求输出10行)

```
1
1     1
1     2     1
1     3     3     1
1     4     6     4     1
1     5     10    10    5     1
……
```

```
#define N 10
void main()
{
    int a[N][N];
    int i,j;
    a[0][0]=1;
    a[1][0]=1;
a[1][1]=1;
for(i=2;i<N;i++)
    {
    a[i][0]=1;
    for(j=1;j<i;j++)
        a[i][j]=a[i-1][j-1]+a[i-1][j];
    a[i][i]=1;
    }
    for(i=0;i<N;i++)
    {
    for(j=0;j<=i;j++)
    printf("%4d",a[i][j]);
        printf("\n");
    }
}
```

题目4:有3个字符串,要求找出其中最大者

```
#include <stdio.h>
#include <string.h>
void main()
{
    char string[20];
    char str[3][20];
    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 string is:\n%s\n", string);
}
```

题目 5: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:五个国家名可由一个二维字符数组来存放。其中二维数组的每一行存放一个国家名(作为一个字符串)。使用字符串比较函数比较各行字符串的大小,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");
}
```

题目 6:求一个 3×3 的整型矩阵对角线元素之和。

```
#define N 3
void main()
{
```

```
    int a[N][N]={1,2,3,4,5,6,7,8,9};
    int i,j;
    intsum=0;
    for(i=0;i<N;i++)
    {
        sum=sum+a[i][i];
    }
    printf("sum= %d\n",sum);
}
```

6.5 精选练习

1. C语言中,引用数组元素时,其数组下标的数据类型允许是_____。

A)整型常量　　B)整型表达式

C)整型常量或整型表达式　　D)任何类型的表达式

2. 在C语言中,一维数组的定义方式是:类型说明符　数组名[_____];。

A)常量表达式　　B)整型表达式

C)整型常量或整型表达式　　D)整型常量

3. 以下程序段给数组所有的元素输入数据,请选择正确答案填入。

```
#include <stdio.h>
void main()
{
    int a[10],i=0;
    while(i<10)
      scanf("%d",_____);
    …
    }
```

A)a+(i++)　　B)&a[i+1]　　C)a+i　　D)&a[++i]

4. 执行下面的程序段后,变量k中的值为_____。

```
int  k=3, s[2];
s[0]=k; k=s[1]*10;
```

A)不定值　　B)33　　C)30　　D)10

5. 若有以下说明:

```
int a[12]={1,2,3,4,5,6,7,8,9,10,11,12};
char c='a',d,g;
```

则数值为4的表达式是_____。

A)a[g-c]　　B)a[4]　　C)a['d'-'c']　　D)a['d'-c]

6. 请读程序:

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

```
void main( )
{
    int  N[2],I,J,K;
    for ( I=0;I<2;I++)
        N[J]=N[I]+1;
    printf(" %d\n", N[K]) ;
}
```

上面程序的输出结果是_____。

A)不确定的值　　B)3　　C)2　　D)1

7. 有如下程序

```
void main()
{
    int  n[5]={0 ,0 ,0},i ,k=2;
    for(i=0;i<k ;i++) n[i]=n[i]+1 ;
    printf(" %d\n",n[k]) ;
}
```

该程序的输出结果是_____。

A)不确定的值　　B)2　　C)1　　D)0

8. 以下程序的输出结果是 _____。

```
void  main()
{
    int   i ,k ,a[10] ,p[3];
    k=5;
    for(i=0;i<10;i++)
        a[i]=i;
    for(i=0;i<3;i++)
        p[i]=a[i*(i+1)];
    for(i=0;i<3;i++)
        k+=p[i]*2;
    printf(" %d\n",k) ;
}
```

A)20　　B)21　　C)22　　D)23

9. 阅读下列程序：

```
void  main( )
{
    int n[3],i,j,k;
    for(i=0;i<3;i++)
        n[i]=0;
    k=2;
    for (i=0;i<k;i++)
        for (j=0;j<k;j++)
```

```
            n[j]=n[i]+1;
    printf("%d\n",n[1]) ;
}
```

下述程序运行后输出结果是_____。

A)2　　B)1　　C)0　　D)3

10.对以下说明语句的正确理解是_____。

```
int  a[10]={6,7,8,9,10};
```

A)将5个初值依次赋给a[1]至a[5]

B)将5个初值依次赋给a[0]至a[4]

C)将5个初值依次赋给a[6]至a[10]

D)因为数组长度与初值的个数不相同,所以此语句不正确

11.有如下程序

```
void main( )
{
    int a[3][3]={{1,2},{3,4},{5,6}},i,j,s=0;
    for(i=1 ;i<3 ;i++)
        for(j=0 ;j<=i ;j++)
            s+=a[i][j];
        printf(" %d\n",s) ;
    }
```

该程序的输出结果是_____。

A)18　　B)19　　C)20　　D)21

12.有定义如下:

```
int  i , x[3][3]={1,2,3,4,5,6,7,8,9};
```

则下面语句的输出结果是_____。

```
for(i=0;i<3;i++) printf("%d",x[i][2-i]) ;
```

A)159　　B)147　　C)357　　D)369

13.以下能对二维数组a进行正确初始化的语句是_____。

A) int a[2][ ]={{1,0,1},{5,2,3}};

B) int a[ ][3]={{1,2,3},{4,5,6}};

C) int a[2][4]={{1,2,3},{4,5},{6}};

D) int a[ ][3]={{1,0,1}{},{1,1}};

14.合法的数组定义是_____。

A)int a[]="string" ;　　B)int a[5]={0 ,1 ,2 ,3 ,4 ,5} ;

C)char s="string" ;　　D)char a[]={0 ,1 ,2 ,3 ,4 ,5} ;

15.若有说明:int a[3][4]={0};则下面正确的叙述是_____。

A)只有元素a[0][0]可得到初值0

B) 此说明语句不正确

C) 数组a中各元素都可得到初值,但其值不一定为0

D) 数组 a 中每个元素均可得到初值 0

16. 若有说明:int　a[3][4];则数组 a 中各元素_____。

A) 可在程序的运行阶段得到初值 0

B) 可在程序的编译阶段得到初值 0

C) 不能得到确定的初值

D) 可在程序的编译或运行阶段得到初值 0

17. 不能把字符串:Hello! 赋给数组 b 的语句是_____。

A)char b[10]={'H','e','l','l','o','! '};

B)char b[10];b="Hello!";

C)char b[10];strcpy(b,"Hello!") ;

D)char b[10]="Hello!";

18. 以下不能对二维数组 a 进行正确初始化的语句是_____。

A) int　a[2][3]={0};

B) int　a[ ][3]={{1,2},{0}};

C) int　a[2][3]={{1,2},{3,4},{5,6}};

D) int　a[ ][3]={1,2,3,4,5,6};

19. 若有定义和语句:

```
char  s[10] ;
s="abcd" ;printf(" % s\n" ,s) ;
```

则结果是(以下 u 代表空格) _____。

A)输出 abcd　　B)输出 a　　C)输出 abcduuuuu　D)编译不通过

20. 设有数组定义:char　array [ ]="China"; 则数组 array 所占的空间为_____。

A)4 个字节　　B)5 个字节　　C)6 个字节　　D)7 个字节

21. 下述对 C 语言字符数组的描述中错误的是_____。

A)字符数组可以存放字符串

B)字符数组中的字符串可以整体输入、输出

C)可以在赋值语句中通过赋值运算符"="对字符数组整体赋值

D)不可以用关系运算符对字符数组中的字符串进行比较

22. 下列描述中不正确的是_____。

A)字符型数组中可以存放字符串

B)可以对字符型数组进行整体输入、输出

C)可以对整型数组进行整体输入、输出

D)不能在赋值语句中通过赋值运算符"="对字符型数组进行整体赋值

23. 请读程序片段(字符串内没有空格字符):

```
printf(" % d\n", strlen("ATS\N012\1\\") ) ;
```

上面程序片段的输出结果是_____。

A)11　　B)10　　C)9　　D)8

24.若有以下程序片段：

```
char str[ ]="ab\n\012\\\"" ;
printf("%d\n" ,strlen(str) ) ;
```

上面程序片段的输出结果是_____。

A)3　　B)4　　C)6　　D)12

25.函数调用:strcat(strcpy(str1 ,str2) ,str3)的功能是_____。

A)将串 str1 复制到串 str2 中后再连接到串 str3 之后

B)将串 str1 连接到串 str2 之后再复制到串 str3 之后

C)将串 str2 复制到串 str1 中后再将串 str3 连接到串 str1 之后

D)将串 str2 连接到串 str1 之后再将串 str1 复制到串 str3 中

26.请选出以下语句的输出结果_____。

```
printf("%d\n" ,strlen("\t\"\065\xff\n") ) ;
```

A)5　　B)14

C)8　　D) 输出项不合法 ,无正常输出

27.给出以下定义：

```
char x[ ]="abcdefg";
char y[ ]={'a' ,'b' ,'c' ,'d' ,'e' ,'f','g'};
```

则正确的叙述为

A)数组 X 和数组 Y 等价

B) 数组 x 和数组 Y 的长度(空间)相同

C)数组 X 的长度(空间)大于数组 Y 的长度(空间)

D)数组 X 的长度(空间)小于数组 Y 的长度(空间)

28.请选出以下程序段的输出结果。

```
#include<stdio.h>
void main()
{
    char   s1[10] ,s2[10] ,s3[10] ,s4[10] ;
    scanf("%s%s" ,s1 ,s2) ;gets(s3) ; gets(s4) ;
    puts(s1) ; puts(s2) ; puts(s3) ; puts(s4) ;
}
```

输入数据如下：(此处<CR>代表回车符)

```
aaaa bbbb<CR>
cccc dddd<CR>
```

A) aaaa　　B) aaaa　　C) aaaa　　D) aaaa

bbbb　　bbbb　　bbbb　　bbbb

cccc　　cccc　　dddd

cccc　　dddd　　dddd　　cccc dddd

29.下面程序的运行结果是_____。

```
void  main()
```

```
{
    char  ch[7]={"65ab21"};
    int  i ,s =0 ;
    for(i=0 ;ch[i]>='0'&&ch[i]<'9' ;i+=2)
        s=10 * s+ch[i]-'0' ;
    printf(" % d\n",s) ;
}
```

A)12ba56　　B)6521　　C)6　　D)62

30. 下列程序的输出结果是_____。

```
void  main( )
{
    char ch[2][5]={"6934,"8254"};
    int i,j,s=0;
    for(i=0;i<2;i + +)
        for(j=0;j<4;j++)
        s=10 * s+ch[i][j]-'0';
    printf(" % d\n",s) ;
}
```

A) 6385　　B)69825　　C) 63825　　D) 69348254

二、填空题

1. 以下程序的功能是:从键盘上输入若干个学生的成绩,统计计算出平均成绩,并输出低于平均分的学生成绩,用输入负数结束输入,请填空。

```
void  main( )
{
    float x[1000], sum=0.0, ave, a;
    int n=0, i;
    printf("Enter mark:\n") ;
    scanf(" % f",&a) ;
    while(a>=0.0&& n<1000)
    {
            sum+ _____;
            x[n]= _____ ;
            n++;
            scanf(" % f",&a) ;
    }
    ave= _____ ;
    printf("Output:\n") ;
    printf("ave= % f\n",ave) ;
    for( i=0;i<n;i++)
        if (_____ )
                printf(" % d   ",x[i]) ;
}
```

2.阅读下列程序：

```
#include<stdio.h>
void  main()
{
    int i , j , row , col ,m ;
    static  int  a[3][3]={{100,200,300} ,{28,72,-30} ,{-850,  2, 6}};
    m=a[0][0] ;
    for (i=0 ; i<3 ; i++)
        for (j=0 ; j<3 ; i++)
            if (a[i][j]<m)
                {
                    m=a[i][j] ;
                      row=i ;
                      col=j ;
                }
        printf(" %d , %d , %d\n" ,m ,row ,col) ;
}
```

上述程序的输出结果是_____。

3.下面程序的功能是：将字符数组 a 中下标值为偶数的元素从小到大排列，其它元素不变，请填空。

```
#include <stdio.h>
#include <string.h>
void  main()
{
    char a[ ]="clanguage" ,t ;
    int   i ,j ,k;
    k=strlen(a) ;
    for(i=0 ;i<=k-2 ;i+=2)
        for(j=i+2 ;j<=k ; _____ )
            if(_____ )
            {
              t=a[i] ;a[i]=a[j] ;a[j]=t ;
            }
    puts(a) ;
    printf("\n") ;
}
```

4.设有下列程序：

```
#include <stdio.h>
#include <string.h>
void  main()
{
```

```
    int  i ;
    char  str[10] ,temp[10] ;
    gets(temp) ;
    for (i=0 ; i<4 ; i++)
    {
        gets(str) ;
        if (strcmp(temp ,str)< 0) strcpy(temp ,str) ;
    }
    printf(" % s\n" ,temp) ;
}
```

上述程序运行后，如果从键盘上输入(在此<CR>代表回车符)：

```
C++<CR>
BASIC<CR>
QuickC<CR>
Ada<CR>
Pascal<CR>
```

则程序的输出结果是_____。

三、编程题

1. 一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如 6=1+2+3. 编程找出 1000 以内的所有完数。

2. 将一个数组逆序输出。

3. 用选择法对 10 个数进行从大到小排序。

4. 有一行电文译文下面规律译成密码：

A->Z　a->z　B->Y　b->y　C->X　c->x

即第一个字母变成第 26 个字母，第 i 个字母变成第(26-i+1)个字母。非字母字符不变，要求编程序将密码回原文，并打印出密码和原文。

5. 从键盘输入若干个整数，其值在 0 至 4 范围内，用-1 作为输入结束的标志。统计每个整数的个数。