第 4 章　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型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，其余的在以后各章陆续介绍。

本章要点：

- 一维数组在程序中的应用
- 二维数组在程序中的应用

4.1　一维数组

4.1.1　一维数组的定义

在 C 语言中使用数组必须先进行定义。一维数组的定义形式如下：

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[5];              //说明整型数组 a，有 5 个元素
float b[5],c[20];      //说明实型数组 b，有 5 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）数组名不能与其他变量名相同。

例如：

```
main()
{
    int b;
    float b[5];
    ……
}
```

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]，如图 4-1 所示。

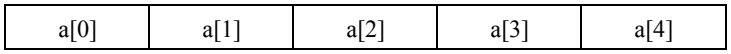

图 4-1　数组示例

5）不能在方括号中用变量来表示元素的个数，但是可以是符号常量或常量表达式。

例如：

```
#define FM 5
main()
{
    int a[3+2],b[7+FM];
    ……
}
```

是合法的。

但是下述说明方式是错误的。

```
main()
{
    int n=10;
    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];
```

4.1.2　一维数组元素的引用

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]
```

其中下标只能为整型常量或整型表达式。如为小数时，C 编译系统将自动取整。

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);
```

例 4.1

```
main()
{
  int i,a[10];
  for(i=0;i<10;i++)
      a[i]=2*i+1;
  for(i=0;i<=9;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输出各个奇数。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。

4.1.3 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 10 个元素全部赋值 1，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};      //系统自动根据初值的个数确定长度为 5
```

4.1.4　一维数组程序设计举例

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 4.2

```
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);
}
```

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 4.3

```
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]);
   }
}
```

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 赋予 p，而把该下标变量值 a[i]赋予 q。然后进入小循环，从 a[i+1]起到最后一个元素逐个与 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。

4.2 二维数组

4.2.1 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 表示第一维长度，即行数；常量表达式 2 表示第二维的长度，即列数。

例如：

```
int a[3][4];
```

说明了一个 3 行 4 列的整型数组，数组名为 a。该数组的元素共有 3×4 即 12 个，即：

```
a[0][0],a[0][1],a[0][2],a[0][3]   第 0 行
a[1][0],a[1][1],a[1][2],a[1][3]   第 1 行
a[2][0],a[2][1],a[2][2],a[2][3]   第 2 行
```

在 C 语言中，二维数组是按行存放的。

即：先存放第 0 行，再存放第 1 行，最后存放第 2 行。每行中有 4 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。

说明：我们可以把二维数组 a 看作是一个特殊的一维数组，包含 3 个数组元素 a[0]、a[1]、a[2]，每个数组元素又是一个一维数组，分别包含 4 个 int 类型的元素。

4.2.2 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，下标是从 0 开始的。

例如：

```
int a[3][4];
```

数组元素中 a[2][3]表示 a 数组中第 2 行第 3 列的元素。

说明：引用二维数组元素时，行下标值的下限为 0，上限为定义时的行数减 1；列下标值的下限为 0，上限为定义时的列数减 1；在引用时要确认下标不越界，如在定义了 int a[3][4]; 之后，绝对没有 a[3][4]这个元素可以引用，因为此时下标越界。

4.2.3 二维数组的初始化

二维数组初始化也是在类型说明时给数组各元素赋以初值。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 4.4

```
main()
{
   int i,j,s=0, average,v[3];
   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;
        }
   average=(v[0]+v[1]+v[2])/3;
   printf("math:%d\nc languag:%d\ndFoxpro:%d\n",v[0],v[1],v[2]);
   printf("total:%d\n", average);
}
```

该实例求数组的第 0 列、第 1 列、第 2 列的平均值以及整个二维数组的平均值。

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
```

例如：

```
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};
```

系统会自动根据赋初值的情况确定行数，一般情况下，行数的确定基于最小化原则：若初值个数能被“常量表达式 2”（即第二维的大小）整除，那么商就是第一维的大小，否则即为商加 1。

4.2.4 二维数组程序设计举例

例 4.5 求一个 3×3 矩阵对角线元素之和。

算法分析：利用双重 for 循环控制输入二维数组，再将 a[i][i]累加后输出。

程序如下：

```
main()
{
   float a[3][3],sum=0;
   int i,j;
   printf("please input rectangle element:\n");
   for(i=0;i<3;i++)
      for(j=0;j<3;j++)
         scanf("%f",&a[i][j]);
   for(i=0;i<3;i++)
      sum=sum+a[i][i];
   printf("duijiaoxian he is %6.2f",sum);
}
```

例 4.6 从键盘上输入一个 2×3 的矩阵，将其转置后形成 3×2 的矩阵输出。

算法分析如下：A 矩阵用 2 行 3 列的二维数组存放，B 矩阵用 3 行 2 列的二维数组存放。外层用次数为 2 的循环控制行（控制变量用 i），内层用次数为 3 的循环控制列（控制变量用 j），循环体中将 A_{ji} 存入 B_{ij}。

程序如下：

```
main()
{
   int a[2][3],b[3][2],i,j;
   for(i=0;i<2;i++)
       for(j=0;j<3;j++)
           scanf("%d",&a[i][j]);
   for(i=0;i<3;i++)
      for(j=0;j<2;j++)
          b[i][j]=a[j][i];
   for(i=0;i<3;i++)
   {
       for(j=0;j<2;j++)
       printf("%8d",b[i][j]);
          printf("\n");
   }
}
```

4.3 本章实例

例 4.7 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行的，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");
}
```

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 赋予 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。

例 4.8　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,
    10,25,12,37)
b=(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()
{
    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");
}
```

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。

4.4 习题

一、选择题

1．下列数组定义正确的是（　）。

A．#define N 8　int n;　　B．int a[5];

C．int a(10);　　D．int n=10,a[n];

2．对以下说明语句的正确理解是（　）。

```
int a[10]={3,4,5,6,7};
```

A．因为数组长度与初值的个数不相同，所以此语句不正确

B．将 5 个初值依次赋给 a[1]至 a[5]

C．将 5 个初值依次赋给 a[6]至 a[10]

D．将 5 个初值依次赋给 a[0]至 a[4]

3．若有 int a[5];定义，则对数组中第三个元素赋值 15 的正确表达式是（　）。

A．a[10%5]=15　　B．a[3]=15

C．a(2)=15　　D．a[7-5]=15

4．若有说明 int a[3][4];则对其数组元素的正确引用是（　）。

A．a(2)(3)　　B．a[3][4]　　C．a[2,3]　　D．a[1][2]

5．下列二维数组的初始化语句中，正确的是（　）。

A．float a[3][]={1,2,3,4,5};

B．int a[][3]={1,2,3,4,5};

C．int a[2][3]={{0,1},{2,3},{5,4}};

D．int a[2][3]={(1,2),(3,4)};

二、填空题

1．以下程序的运行结果是_____。

```
main()
{
   int a[4][4]={{4,2,-3,-4},{0,-11,-13,14},{-22,24,0,-24},{-31,32,-33,0}};
   int i,j,s=0;
   for(i=0;i<=3;i++ )
     for(j=0;j<=3;j++ )
       { if(a[i][j]%2!=0)break;
         if(a[i][j]==0)continue;
         s=s+a[i][j];
       }
   printf("s=%d\n",s);
}
```

2．以下程序的运行结果是_____。

```
main()
{
   int i,t[][3]={9,8,7,6,5,4,3,2,1};
   for(i=0;i<3;i++)
       printf("%d",t[2-i][i]);
}
```

三、程序设计题

1．编写程序，输入 10 个整数存入一维数组，按逆序重新存放后再输出。

2．编写程序，输入整型一维数组 a[8]，计算并输出 a 数组中所有元素的平均值。

3．输入一个 3 行 4 列的整数矩阵，输出其中最大值、最小值和它们的行列坐标。

4．编写程序，输入一个 5 行 5 列的整数矩阵，判断该矩阵是否为对称矩阵，是则输出 yes；否则输出 no（说明：对称矩阵的定义是所有第 i 行第 j 列的元素值均等于第 j 行第 i 列元素的值）。