

# 第六章 利用数组处理批量数据

作者：石璞东

参考资料：《C程序设计（第四版）》谭浩强

## 前言

数组：同一类性质的数据

- 数组是一组有序数据的集合，数组中各数据的排列是有一定规律的，下标代表数据在数组中的序号。
- 用一个数组名和下标来唯一地确定数组中的元素。
- 数组中的每一个元素都属于同一个数据类型，不能把不同类型的数据（如学生的成绩和学生的性别）放在同一个数组中。

## 6.1 怎样定义和引用一维数组

### 6.1.1 怎样定义一维数组

要使用数组，必须在程序中先定义数组，即通知计算机：由哪些数据组成数组，数组中有多少元素，属于哪个数据类型，定义数组的一般形式为：类型符 数组名[常量表达式]；。

c 语言不允许对数组的大小做动态定义，即数组的大小不依赖于程序运行过程中变量的值。

如果在被调用的函数（不包括主函数）中定义数组，其长度可以是变量或非常量表达式，如以下代码所示：

```
1 void func(int n){
2     int a[2*n];
3     ...
4 }
```

在调用函数 `func` 时，形参 `n` 从实参得到值，这种情况称为**可变长数组**，允许在每次调用 `func` 函数时，`n` 有不同的值，但是在执行函数时，`n` 的值是不变的，数组长度是固定的。

如果指定数组为静态存储方式，即 `static int a[2*n]`，则不能用可变长数组。

## 6.1.2 怎样引用一维数组元素

只能引用数组元素而不能一次整体调用整个数组全部元素的值，其引用形式为：数组名[下标]。

例6.1 逆序输出。

```
1  #include<stdio.h>
2  int main(){
3      int i,num[10];
4      float avg = 0.0,sum = 0.0;
5      printf("请输入数组元素:");
6      for(i=0;i<10;i++){
7          scanf("%d",&num[i]);
8      }
9      for(int j = 9;j>=0;j--){
10         sum += num[j];
11         printf("%3d",num[j]);
12     }
13     printf("\n");
14     avg = sum/10;
15     printf("平均值为:%f\n",avg);
16     return 0;
17 }
```

## 6.1.3 一维数组的初始化

- 对全部数组元素赋予初值，如 `int a[5]={1,2,3,4,5}`。
- 可以只给数组中的一部分元素赋值，如 `int a[10]={0,1,2,3,4}` 定义 a 数组有10个元素，但花括号内只提供5个初值，这表示只给前面5个元素赋初值，系统自动给后面5个元素赋初值为0。
- 如果想使一个数组中全部元素值为0，可以写成 `int a[10]={0,0,0,0,0,0,0,0,0,0}` 或 `int a[10]={0}`。
- 在对全部数组元素赋初值时，由于数据的个数已经确定，因此可以不指定数组长度，如 `int a[5]={1,2,3,4,5}` 可以写成 `int []={1,2,3,4,5}`。

【注】：如果在定义数值型数组时，指定了数组的长度并对其进行初始化，凡未被“初始化列表”指定的数组元素，系统会自动将它们初始化为0，如果是字符型数组，则初始化为 `'\0'`，如果是指针型数组，则初始化为 `NULL`，即空指针。

## 6.1.4 一维数组程序举例

例6.2 斐波那契数列

```
1  #include<stdio.h>
2  int main(){
3      int i;
4      int f[20]={1,1};
```

```

5     printf("斐波那契数列前20项为: \n");
6     for (i = 2; i < 20; i++) {
7         f[i] = f[i-2]+f[i-1];
8     }
9     for (i = 0; i < 20; i++) {
10        if (i%5 == 0) {
11            printf("\n");
12        }
13        printf("%12d",f[i]);
14    }
15    printf("\n");
16    return 0;
17 }

```

### 例6.3 输入10个数字，并按照从小到大顺序输出。

```

1  #include<stdio.h>
2  int main(){
3      int arr[10],t;
4      printf("请输入数组元素:");
5      for (int i = 0; i < 10; i++) {
6          scanf("%d",&arr[i]);
7      }
8      printf("顺序输出结果为: ");
9      for (int i = 0; i < 10; i++) {
10         printf("%2d",arr[i]);
11     }
12     printf("\n");
13     printf("排序后的结果（由小到大）为: ");
14     for (int i = 0; i < 9; i++) {
15         for (int j = 0; j < 9-i; j++) {
16             if(arr[j]>=arr[j+1]){
17                 t = arr[j];
18                 arr[j] = arr[j+1];
19                 arr[j+1] = t;
20             }
21         }
22     }
23     for (int i =0; i < 10; i++) {
24         printf("%2d",arr[i]);
25     }
26     printf("\n");
27     return 0;
28 }

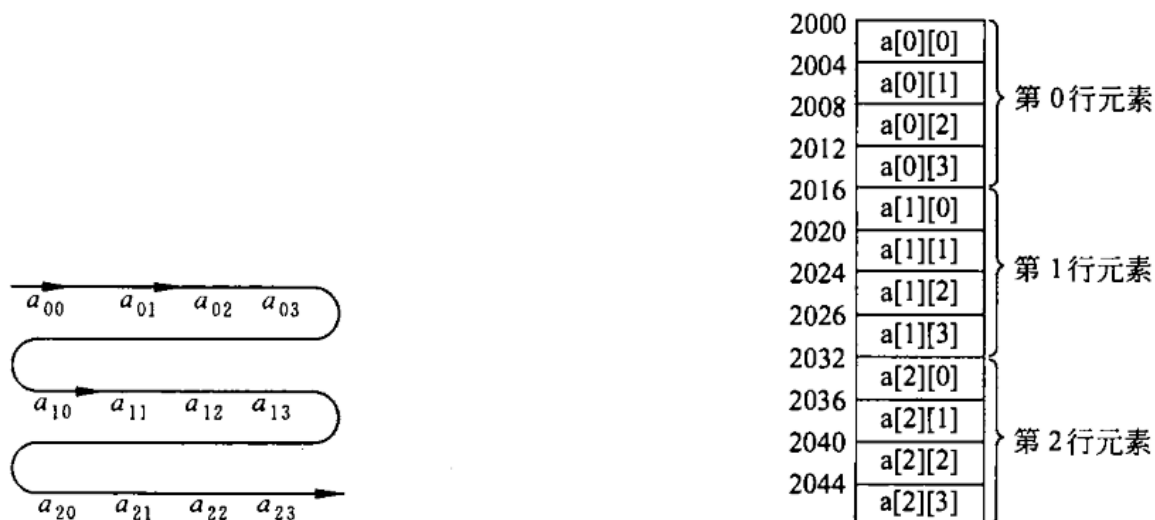
```

## 6.2 怎样定义和引用二维数组

二维数组常称为**矩阵**，可以把二维数组写成行和列的排列形式。

### 6.2.1 怎样定义二维数组

二维数组定义的一般形式为：`类型说明符 数组名[常量表达式][常量表达式];`，二维数组中元素的排列顺序是按行存放的，即在内存中先顺序存放第1行元素，接着再存放第2行的元素，如图所示。



在内存中，各元素是连续存放的，不是二维的，是线性的。

### 6.2.2 怎样引用二维数组的元素

二维数组元素的表示形式为：`数组名[下标][下标]`，C语言中对数组越界不进行检查。

```
20
21 #include<stdio.h>
22 int main(){
23     int arr[10]={0};
24     printf("%d\n",arr[11]);
25     return 0;
26 }
27
```

Array index 11 is past the end of the array (which contains 10 elements)

□ □

435812772

Program ended with exit code: 0

### 6.2.3 二维数组的初始化

- 分行给二维数组赋初值，如 `int a[3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}}`（与第二种方法相比，更推荐第一种方法）
- 将所有数据写在一个花括号内，按数组元素在内存中的排列顺序对各元素赋初值，如 `int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12}`
- 可以对部分元素赋初值。

```

1  #include<stdio.h>
2  int main(){
3      int arr[4][4]={{1},{1,2},{1,2,3},{1,2,3,4}};
4      for (int i = 0; i < 4; i++) {
5          for(int j = 0;j < 4;j++){
6              printf("%2d",arr[i][j]);
7          }
8          printf("\n");
9      }
10     printf("\n");
11     return 0;
12 }

```

- 如果对全部元素都赋初值，则定义数组时对第1维的长度可以不指定，但第2维的长度不能省。

```

1  #include<stdio.h>
2  int main(){
3      int arr[][4]={1,2,3,4,5,6,7,8,9,10,11,12};
4      for (int i = 0; i < 3; i++) {
5          for(int j = 0;j < 4;j++){
6              printf("%4d",arr[i][j]);
7          }
8          printf("\n");
9      }
10     printf("\n");
11     return 0;
12 }

```

## 6.2.4 二维数组程序举例

### 例6.4 矩阵转置

```

1  #include<stdio.h>
2  int main(){
3      int arr[2][3]={{1,2,3},{4,5,6}},arr_T[3][2];
4      printf("原二维数组为: \n");
5      for (int i = 0; i < 2; i++) {
6          for(int j = 0;j < 3;j++){
7              printf("%2d",arr[i][j]);
8          }
9          printf("\n");
10     }
11     for(int i = 0;i < 3;i++){
12         for(int j = 0;j < 2;j++){
13             arr_T[i][j] = arr[j][i];
14         }
15     }
16 }

```

```

15     printf("\n");
16 }
17 printf("转置之后的数组为: \n");
18 for (int i = 0; i < 3; i++) {
19     for(int j = 0;j < 2;j++){
20         printf("%2d",arr_T[i][j]);
21     }
22     printf("\n");
23 }
24 printf("\n");
25 return 0;
26 }

```

### 例6.5 求二维数组最大值

```

1  #include<stdio.h>
2  int main(){
3      int row=0,column=0,max;
4      int arr[3][4] = {{13,24,53,1},{2,6,7,5},{67,54,63,45}};
5      printf("原二维数组为: \n");
6      for (int i = 0; i < 3; i++) {
7          for(int j = 0;j < 4;j++){
8              printf("%5d",arr[i][j]);
9          }
10         printf("\n");
11     }
12     max = arr[0][0];
13     for (int i = 0; i < 3; i++) {
14         for(int j = 0;j < 3;j++){
15             if(arr[i][j] > max){
16                 max = arr[i][j];
17                 row = i;
18                 column = j;
19             }
20         }
21     }
22     printf("原二维数组中的最大值为%d, 在第%d行第%d列\n",max,(row+1),(column+1));
23     return 0;
24 }

```

## 6.3 字符数组

**c**语言中没有字符串类型，字符串是存放在字符型数组中的。

### 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，由于字符型数据是以整数形式（ASCII代码）存放的，因此也可以用整型数组来存放字符数据。

### 6.3.2 字符数组的初始化

如果在定义字符数组时不进行初始化，则数组中各元素的值是不可预料的，如果花括号中提供的初值个数大于数组长度，则出现语法错误，如果初值个数小于数组长度，则只将这些字符赋给数组中前面那些元素，其余的元素自动定位空字符，即 `'\0'`。

### 6.3.3 怎样引用字符数组中的元素

#### 例6.6 定义字符数组

```
1  #include<stdio.h>
2  int main(){
3      char str[10] = {'H','e','l','l','o','W','o','r','l','d'};
4      for(int i = 0;i < 10;i++){
5          printf("%c",str[i]);
6      }
7      printf("\n");
8      return 0;
9  }
```

#### 例6.7 定义字符数组\_plus

```
1  #include<stdio.h>
2  int main(){
3      char str[] = "helloworld";
4      printf("%s",str);
5      printf("\n");
6      return 0;
7  }
```

### 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，并规定了一个“字符串结束标志”，以字符 `'\0'` 作为结束标志。

C系统在用字符数组存储字符串常量时会自动加一个 `'\0'` 作为结束符，如'C program'共有9个字符，字符串是存放在一维数组中的，在数组中它占10个字节，最后一个字节 `'\0'` 是由系统自动加上去的。

三种等价形式：

```
1 char c[] = {"I am happy"};
2 char c[] = "I am happy";
3 char c[] = {'I', ' ', 'a', 'm', ' ', 'h', 'a', 'p', 'p', 'y'};
```

### 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式：

- 逐个字符输入输出，用格式符 `%c` 输入或输出一个字符；
- 将整个字符串一次输入或输出，用 `%s` 格式符

【注】：

- 如果一个字符数组中包含一个以上 `\0`，则遇到第一个 `\0` 时输出就结束。
- `scanf` 函数中的输入项如果是字符数组名，不要再加地址符 `&`。

### 6.3.6 使用字符串处理函数

在使用字符串处理函数时，应该引入 `#include<string.h>` 头文件。

- `puts` 函数 - 输出字符串的函数

一般形式为：`puts(字符数组)`，其作用是将一个字符串输出到终端。

- `gets` 函数 - 输入字符串的函数

一般形式为：`gets(字符数组)`，其作用是从终端输入一个字符串到字符数组，并且得到一个函数值，该函数值是字符数组的起始地址。



```

1  #include <stdio.h>
2  #include<string.h>
3  int main(){
4      char str[50];
5      printf("请输入一个字符串: ");
6      gets(str);
7      puts(str);
8      return 0;
9  }

```

- **strcat** 函数 - 字符串连接函数

一般形式为：`strcat(字符数组1, 字符数组2)`，其作用是把两个字符数组中的字符串连接起来，把字符串2接到字符串1后面，结果放在字符数组1中，函数调用后得到一个函数值，即字符数组1的地址。

```

1  #include <stdio.h>
2  #include<string.h>
3  int main(){
4      char str1[100] = "Hello,";
5      char str2[] = "i am hahaCoder!";
6      printf("str1和str2合并后的结果为:%s\n",strcat(str1, str2));
7      return 0;
8  }

```

【注】：

1. 字符数组1必须足够大，以便容纳连接后的新字符串。
2. 连接前两个字符串的后面都有 `'\0'`，连接时将字符串1后面的 `'\0'` 取消，只在新串最后保留 `'\0'`。

- **strcpy** 和 **strncpy** 字符串复制函数

一般形式为：`strcpy(字符数组1, 字符串2)`，其作用是将字符串2复制到字符数组1中去。

```

1  #include <stdio.h>
2  #include <string.h>
3  int main (){
4      char str1[]="Sample string";
5      char str2[40];
6      char str3[40];
7      strcpy (str2,str1);
8      strcpy (str3,"copy successful");
9      printf ("str1: %s\nstr2: %s\nstr3: %s\n",str1,str2,str3);
10     return 0;
11 }

```

【注】：

1. 字符数组1必须定义得足够大，以便容纳被复制的字符串2。
2. 字符数组1必须被写成数组名形式，字符串2可以是字符数组名，也可以是一个字符串常量。
3. 不能用赋值语句将一个字符串常量或字符数组直接给一个字符数组，只能用 `strcpy` 函数将一个字符串复制到另一个字符数组中去，用赋值语句只能将一个字符赋给一个字符型变量或字符数组元素。

```
1  #include <stdio.h>
2  #include <string.h>
3  int main(){
4      char src[40];
5      char dest[12];
6      strcpy(src, "This is runoob.com");
7      strncpy(dest, src, 10);
8      printf("最终的目标字符串: %s\n", dest);
9      return(0);
10 }
```

`strncpy(str1, str2, 2)`：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。

- `strcmp` 函数 - 字符串比较函数

一般形式为：`strcmp(字符串1, 字符串2)`，其作用是比较字符串1和字符串2，比较规则是将两个字符串自左至右逐个字符按 `ASCII` 码值大小进行比较，直到出现不同的字符或遇到 `'\0'` 为止。

`str1 = str2`，函数值为0；`str1 > str2`，函数值为正数；`str1 < str2`，函数值为负数。

```
1  #include <stdio.h>
2  #include <string.h>
3  int main (){
4      char str1[15];
5      char str2[15];
6      printf("请输入字符: ");
7      scanf("%s %s", str1, str2);
8      int ret;
9      ret = strcmp(str1, str2);
10     if(ret < 0){
11         printf("str1<str2\n");
12     }
13     else if(ret > 0){
14         printf("str1>str2\n");
15     }
16     else{
17         printf("str1=str2\n");
18     }
19     return 0;
20 }
```

- `strlen` 函数 - 测试字符串长度的函数

一般形式为：`strlen(字符数组)`，其作用是测试字符串的长度，函数值为字符串中的实际长度，不包括 `'\0'` 在内。

```
1 #include <stdio.h>
2 #include <string.h>
3 int main (){
4     char str[50];
5     strcpy(str, "www.shipudong.com");
6     printf("字符串%s的长度是%d\n",str,strlen(str));
7     return 0;
8 }
```

- `strlwr` 函数 - 转换为小写的函数

一般形式为：`strlwr(字符串)`，其作用是将字符串中大写字母换成小写字母。

- `strupr` 函数 - 转换为大写的函数

一般形式为：`strupr(字符串)`，其作用是将字符串中小写字母转换成大写字母。

## 6.3.7 字符数组应用举例

### 例6.8 统计单词个数

```
1 #include<stdio.h>
2 #include<string.h>
3 int main(){
4     char string[100];
5     int num = 0,word = 0;
6     char c;
7     printf("请输入文本内容: ");
8     fgets(string,sizeof(string)/sizeof(string[0]),stdin);
9     for (int i = 0; (c = string[i])!='\0'; i++) {
10         if(c == ' '){
11             word = 0;
12         }else if (word == 0){
13             word = 1;
14             num++;
15         }
16     }
17     printf("共有%d个单词\n",num);
18     return 0;
19 }
```

### 例6.9 求3个字符串中的最大者

```
1  #include<stdio.h>
2  #include<string.h>
3  int main(){
4      char str[3][20];
5      char string[20];
6      printf("请输入三个字符串: ");
7      for (int i = 0; i < 3; i++) {
8          scanf("%s",str[i]);
9      }
10     if (strcmp(str[0],str[1])>0) {
11         strcpy(string, str[0]);
12     }else{
13         strcpy(string, str[1]);
14     }
15     if(strcmp(string, str[2])<0){
16         strcpy(string, str[2]);
17     }
18     printf("最大字符串为:%s\n",string);
19     return 0;
20 }
```

课后题: 3、4、5、6、7、9、10、12、13、14