

# C 语言

## 第 10 次上机 数组与指针



张晓平

武汉大学数学与统计学院



2017 年 4 月 27 日

**例 1** 编写一个函数，对数组按从小到大进行排序。简单排序算法原理：每次从左至右扫描序列，记下最小值的位置。

```
1 #include <stdio.h>
2 void swap(double * a, double * b);
3 void sort(double arr[], int n);
4 void print_arr(double arr[], int n);
5
6 int main(void)
7 {
8     double arr[7] = {1.1, 2.2, 7.7, 4.4, 5.5, 3.3, 6.6};
9
10    print_arr(arr, 7);
11    sort(arr, 7);
12    print_arr(arr, 7);
13
14    return 0;
15 }
16
17 void swap(double * a, double * b)
```

```
18 {  
19     double temp;  
20  
21     temp = *a;  
22     *a = *b;  
23     *b = temp;  
24  
25 }  
26  
27 void sort(double arr[], int n)  
28 {  
29     int i, j, pos;  
30  
31     for(i = 0; i < n; i++)  
32     {  
33         pos = i;  
34         for(j = i+1; j < n; j++)  
35         {
```

```
36         if(arr[j] < arr[pos])
37             pos = j;
38     }
39     if (pos != i)
40         swap(&arr[i], &arr[pos]);
41 }
42 }
43
44
45 void print_arr(double arr[], int n)
46 {
47     int i;
48     for (i = 0; i < n; i++)
49         printf("%6.2f ", arr[i]);
50     putchar('\n');
51
52 }
```

**例 2** 编写一个程序，初始化一个 `double` 数组，然后把数组内容复制到另外两个数组。制作第一份拷贝的函数使用数组符号。制作第二份拷贝的函数使用指针符号，并使用指针的增量操作。把目标数组名和要复制的元素个数作为参数传递给函数。

```
double source[5] = {1.1, 2.2, 3.3, 4.4, 5.5};  
double target1[5], target2[5];  
copy_arr(source, target1, 5);  
copy_ptr(source, target2, 5);
```

```
1 #include <stdio.h>
2 #define SIZE 5
3 void copy_arr(double source[], double target[], int n);
4 void copy_ptr(double * source, double * target, int n);
5 void print_arr(double arr[], int n);
6
7 int main(void)
8 {
9     double source[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
10    double target1[5], target2[5];
11
12    copy_arr(source, target1, SIZE);
13    copy_ptr(source, target2, SIZE);
14    print_arr(source, SIZE);
15    print_arr(target1, SIZE);
16    print_arr(target2, SIZE);
17
```

```
18     return 0;
19 }
20
21 void copy_arr(double source[], double target[], int n)
22 {
23     int i;
24
25     for (i = 0; i < n; i++)
26         target[i] = source[i];
27
28 }
29
30 void copy_ptr(double * source, double * target, int n)
31 {
32     double * ptr1 = source;
33     double * ptr2 = target;
34
35     while (ptr1 < source + n)
```



```
36     *ptr2++ = *ptr1++;
37 }
38
39 void print_arr(double arr[], int n)
40 {
41     int i;
42     for (i = 0; i < n; i++)
43         printf("%6.2f ", arr[i]);
44     putchar('\n');
45
46 }
```

**例 3** 利用以上函数，把一个包含 7 个元素的数组中第 3 到第 5 个元素复制到一个包含 3 个元素的数组中。

```
1 #include <stdio.h>
2 void copy_arr(double source[], double target[], int n);
3 void print_arr(double arr[], int n);
4
5 int main(void)
6 {
7     double source[7] = {1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7};
8     double target[3];
9
10    copy_arr(source+2, target, 3);
11    print_arr(source, 7);
12    print_arr(target, 3);
13
14    return 0;
15 }
16
17 void copy_arr(double source[], double target[], int n)
```

```
18 {
19     int i;
20
21     for (i = 0; i < n; i++)
22         target[i] = source[i];
23
24 }
25
26
27 void print_arr(double arr[], int n)
28 {
29     int i;
30     for (i = 0; i < n; i++)
31         printf("%6.2f ", arr[i]);
32     putchar('\n');
33
34 }
```

例 4 编写一个函数，求一个 `double` 数组的最大值及其索引，并写一个简单驱动程序测试它。

```
1 #include <stdio.h>
2 void max_arr1d(double * arr, int n, double * max, int *
  index);
3
4 int main(void)
5 {
6     double array[4] = {2.0,-1.0,5.0,5.0};
7     double max;
8     int index;
9
10    max_arr1d(array, 4, &max, &index);
11    printf("the maximum value of array is %f "
12           "which is located at %d-th element.\n",
13           max, index+1);
14
15    return 0;
16 }
```

```
17
18 void max_arr1d(double * arr, int n, double * max, int *
    index)
19 {
20     int i;
21     *max = arr[0];
22     *index = 0;
23
24     for (i = 1; i < n; i++)
25     {
26         if (*max < arr[i])
27         {
28             *max = arr[i];
29             *index = i;
30         }
31     }
32 }
```

**例 5** 编写一个函数，将两个长度相同的数组相加，结果存储到第三个数组中，并用一个简单的驱动程序测试它。



```
1 #include <stdio.h>
2 void sum_arr1d(int * arr1, int * arr2, int * arr3, int n);
3 void print_arr1d(int arr1d[], int n);
4
5 int main(void)
6 {
7     int arr1[4] = {2,4,5,8};
8     int arr2[4] = {1,0,4,6};
9     int arr3[4];
10
11     sum_arr1d(arr1, arr2, arr3, 4);
12     print_arr1d(arr1, 4);
13     print_arr1d(arr2, 4);
14     print_arr1d(arr3, 4);
15
16     return 0;
17 }
```

```
18
19 void sum_arr1d(int * arr1, int * arr2, int * arr3, int n)
20 {
21     int i;
22     for (i = 0; i < n; i++)
23         arr3[i] = arr1[i] + arr2[i];
24
25 }
26
27 void print_arr1d(int arr1d[], int n)
28 {
29     int i;
30     for (i = 0; i < n; i++)
31         printf("%3d ", arr1d[i]);
32     putchar('\n');
33 }
```

**例 6** 编写一个函数，求两个三维向量的内积和外积，并用一个简单的驱动程序测试它。

设

$$\vec{u} = (a_1, a_2, a_3)^T, \quad \vec{v} = (b_1, b_2, b_3)^T$$

则内积为

$$\vec{u} \cdot \vec{v} = a_1 b_1 + a_2 b_2 + a_3 b_3$$

外积为

$$\vec{u} \times \vec{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)^T.$$

**例 7** 编写一个函数，提示用户输入三个数集，每个数集包括 5 个 `double` 值。程序应当实现以下功能：

1. 把输入信息存储到一个  $3 \times 5$  的数组中
2. 计算出每个数集的平均值
3. 计算所有数的平均值
4. 找出这 15 个数中的最大值
5. 打印出结果

```
1 #include <stdio.h>
2
3 #define COLS 5
4 #define ROWS 3
5 void input_array2d(double arr2d[][COLS], int row);
6 void print_array1d(double arr1d[], int n);
7 void print_array2d(double arr2d[][COLS], int row);
8 double aver_array1d(double * arr1d, int n);
9 void aver_array2d_row(double arr2d[][COLS], int row, double
    average[]);
10 double aver_array2d(double arr2d[][COLS], int row);
11 double max_array2d(double arr2d[][COLS], int row);
```

```
1  #include "ex10_12.h"
2
3  int main(void)
4  {
5      double arr2d[ROWS][COLS];
6      double average_row[ROWS];
7      double average;
8      double max;
9
10     printf("Please input 3 sets of five double numbers each.\n");
11     input_array2d(arr2d, ROWS);
12
13     printf("The array is:\n");
14     print_array2d(arr2d, ROWS);
15
16     aver_array2d_row(arr2d, ROWS, average_row);
```

```
17     printf("average of each row:\n");
18     print_array1d(average_row, ROWS);
19
20     average = aver_array2d(arr2d, ROWS);
21     printf("average of arr2d is %5.2f\n", average);
22
23     max = max_array2d(arr2d, ROWS);
24     printf("max of arr2d is %5.2f\n", max);
25
26     return 0;
27 }
28
29 void input_array2d(double arr2d[][COLS], int row)
30 {
31     int r, c;
32     for (r = 0; r < row; r++)
33         for (c = 0; c < COLS; c++)
34             scanf("%lf", &arr2d[r][c]);
```

```
35 }
36
37 void print_array1d(double arr1d[], int n)
38 {
39     int i;
40     for (i = 0; i < n; i++)
41         printf("%8.2f", arr1d[i]);
42     putchar('\n');
43 }
44
45 void print_array2d(double arr2d[][COLS], int row)
46 {
47     int r;
48     for (r = 0; r < row; r++)
49         print_array1d(arr2d[r], COLS);
50 }
51
52 double aver_array1d(double * arr1d, int n)
```



```
53 {  
54     int i;  
55     double total = 0.0;  
56  
57     for (i = 0; i < n; i++)  
58         total += arr1d[i];  
59  
60     return total/((double) n);  
61 }  
62  
63 void aver_array2d_row(double arr2d[][COLS], int row,  
64                     double average[])  
65 {  
66     int i;  
67  
68     for (i = 0; i < row; i++)  
69         average[i] = aver_array1d(arr2d[i], COLS);  
70 }
```

```
71
72 double aver_array2d(double arr2d[][COLS], int row)
73 {
74     int i, j;
75     double total = 0.0;
76
77     for (i = 0; i < row; i++)
78         for (j = 0; j < COLS; j++)
79             total += arr2d[i][j];
80
81     return total/(double) (row*COLS);
82 }
83
84 double max_array2d(double arr2d[][COLS], int row)
85 {
86     int i, j;
87     double max = arr2d[0][0];
88
```

```
89     for (i = 0; i < row; i++)
90         for (j = 0; j < COLS; j++)
91             if (max < arr2d[i][j])
92                 max = arr2d[i][j];
93
94     return max;
95 }
```

例 8 用变长数组重写以上程序。

例 9 用一维数组重写以上程序。