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。

```c
#include <stdio.h>
void swap(double * a, double * b);
void sort(double arr[], int n);
void print_arr(double arr[], int n);

int main(void)
{
  double arr[7] = {1.1, 2.2, 7.7, 4.4, 5.5, 3.3, 6.6};

  print_arr(arr, 7);
  sort(arr, 7);
  print_arr(arr, 7);

  return 0;
}

void swap(double * a, double * b)
```

```c
{
  double temp;

  temp = *a;
  *a = *b;
  *b = temp;

}

void sort(double arr[], int n)
{
  int i, j, pos;

  for(i = 0; i < n;i++)
  {
    pos = i;
    for(j = i+1; j < n; j++)
    {
      if(arr[j] < arr[pos])
        pos = j;
    }
    if (pos != i)
      swap(&arr[i], &arr[pos]);
  }
}


void print_arr(double arr[], int n)
{
  int i;
  for (i = 0; i < n; i++)
    printf("%6.2f ", arr[i]);
  putchar('\n');

}
```

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);
```

```c
#include <stdio.h>
#define SIZE 5
void copy_arr(double source[], double target[], int n);
void copy_ptr(double * source, double * target, int n);
void print_arr(double arr[], int n);

int main(void)
{
  double source[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
  double target1[5], target2[5];

  copy_arr(source, target1, SIZE);
  copy_ptr(source, target2, SIZE);
  print_arr(source, SIZE);
  print_arr(target1, SIZE);
  print_arr(target2, SIZE);

  return 0;
}

void copy_arr(double source[], double target[], int n)
{
  int i;

  for (i = 0; i < n; i++)
    target[i] = source[i];

}

void copy_ptr(double * source, double * target, int n)
{
  double * ptr1 = source;
  double * ptr2 = target;

  while (ptr1 < source + n)
```

```c
    *ptr2++ = *ptr1++;
}

void print_arr(double arr[], int n)
{
  int i;
  for (i = 0; i < n; i++)
    printf("%6.2f ", arr[i]);
  putchar('\n');

}
```

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。

```c
#include <stdio.h>
void copy_arr(double source[], double target[], int n);
void print_arr(double arr[], int n);

int main(void)
{
  double source[7] = {1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7};
  double target[3];

  copy_arr(source+2, target, 3);
  print_arr(source, 7);
  print_arr(target, 3);

  return 0;
}

void copy_arr(double source[], double target[], int n)
```

```c
{
  int i;

  for (i = 0; i < n; i++)
    target[i] = source[i];

}


void print_arr(double arr[], int n)
{
  int i;
  for (i = 0; i < n; i++)
    printf("%6.2f ", arr[i]);
  putchar('\n');

}
```

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。

```c
#include <stdio.h>
void max_arr1d(double * arr, int n, double * max, int * index);

int main(void)
{
  double array[4] = {2.0,-1.0,5.0,5.0};
  double max;
  int index;

  max_arr1d(array, 4, &max, &index);
  printf("the maximum value of array is %f "
         "which is located at %d-th element.\n",
         max, index+1);

  return 0;
}
```

```c

void max_arr1d(double * arr, int n, double * max, int * index)
{
  int i;
  *max = arr[0];
  *index = 0;

  for (i = 1; i < n; i++)
  {
    if (*max < arr[i])
    {
      *max = arr[i];
      *index = i;
    }
  }
}
```

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。

```c
#include <stdio.h>
void sum_arr1d(int * arr1, int * arr2, int * arr3, int n);
void print_arr1d(int arr1d[], int n);

int main(void)
{
  int arr1[4] = {2,4,5,8};
  int arr2[4] = {1,0,4,6};
  int arr3[4];

  sum_arr1d(arr1, arr2, arr3, 4);
  print_arr1d(arr1, 4);
  print_arr1d(arr2, 4);
  print_arr1d(arr3, 4);

  return 0;
}
```

```c

void sum_arr1d(int * arr1, int * arr2, int * arr3, int n)
{
  int i;
  for (i = 0; i < n; i++)
    arr3[i] = arr1[i] + arr2[i];

}

void print_arr1d(int arr1d[], int n)
{
  int i;
  for (i = 0; i < n; i++)
    printf("%3d ", arr1d[i]);
  putchar('\n');
}
```

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 3×5 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

```c
#include <stdio.h>

#define COLS 5
#define ROWS 3
void input_array2d(double arr2d[][COLS], int row);
void print_array1d(double arr1d[], int n);
void print_array2d(double arr2d[][COLS], int row);
double aver_array1d(double * arr1d, int n);
void aver_array2d_row(double arr2d[][COLS], int row, double average[]);
double aver_array2d(double arr2d[][COLS], int row);
double max_array2d(double arr2d[][COLS], int row);
```

```c
#include "ex10_12.h"

int main(void)
{
  double arr2d[ROWS][COLS];
  double average_row[ROWS];
  double average;
  double max;

  printf("Please input 3 sets of five double numbers each.\n");
  input_array2d(arr2d, ROWS);

  printf("The array is:\n");
  print_array2d(arr2d, ROWS);

  aver_array2d_row(arr2d, ROWS, average_row);
  printf("average of each row:\n");
  print_array1d(average_row, ROWS);

  average = aver_array2d(arr2d, ROWS);
  printf("average of arr2d is %5.2f\n", average);

  max = max_array2d(arr2d, ROWS);
  printf("max of arr2d is %5.2f\n", max);

  return 0;
}

void input_array2d(double arr2d[][COLS], int row)
{
  int r, c;
  for (r = 0; r < row; r++)
    for (c = 0; c < COLS; c++)
      scanf("%lf", &arr2d[r][c]);
```

```c
}

void print_array1d(double arr1d[], int n)
{
  int i;
  for (i = 0; i < n; i++)
    printf("%8.2f", arr1d[i]);
  putchar('\n');
}

void print_array2d(double arr2d[][COLS], int row)
{
  int r;
  for (r = 0; r < row; r++)
    print_array1d(arr2d[r], COLS);
}

double aver_array1d(double * arr1d, int n)
```

```c
{
  int i;
  double total = 0.0;

  for (i = 0; i < n; i++)
    total += arr1d[i];

  return total/(double) n;
}

void aver_array2d_row(double arr2d[][COLS], int row,
                      double average[])
{
  int i;

  for (i = 0; i < row; i++)
    average[i] = aver_array1d(arr2d[i], COLS);
}
```

```c

double aver_array2d(double arr2d[][COLS], int row)
{
  int i, j;
  double total = 0.0;

  for (i = 0; i < row; i++)
    for (j = 0; j < COLS; j++)
      total += arr2d[i][j];

  return total/(double) (row*COLS);
}

double max_array2d(double arr2d[][COLS], int row)
{
  int i, j;
  double max = arr2d[0][0];

  for (i = 0; i < row; i++)
    for (j = 0; j < COLS; j++)
      if (max < arr2d[i][j])
        max = arr2d[i][j];

  return max;
}
```

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。