

# 实验 5

## 数组与字符串

### 5.1 实验目的

- (1) 熟练掌握一维数组、二维数组的定义、初始化和输入输出方法。
- (2) 熟练掌握字符数组和字符串函数的使用。
- (3) 掌握与数组有关的常用算法(如查找、排序等)。

### 5.2 实验指导

本实验指导中将编写两个程序,程序的要求和目标如下:

- (1) 在一个存放 10 个元素的一维整型数组中,找出数组元素的最大值和最小值并输出。熟悉一维数组的定义、初始化及其使用方法。
- (2) 输入一个  $4 \times 4$  的二维数组,输出此二维数组后,再分别输出其主对角线与副对角线的和。熟悉二维数组的定义、初始化及其使用方法。

#### 1. 程序一

编写程序,在一个存放 10 个元素的一维整型数组中,找出数组元素的最大值和最小值并输出。

程序代码如下:

```
#include <stdio.h>
main()
{
    int a[10],i,max,min;
    printf("请输入 10 个整数:\n");
    for (i=0;i<10;i++)
    {
        scanf("%d",&a[i]);
    }
    max=a[0];
```

```

min=a[0];
for (i=1;i<10;i++)
{
    if (a[i]>max)
        max=a[i];
    if (a[i]<min)
        min=a[i];
}
for (i=0;i<10;i++)
{
    printf("a[%d]=%d\n",i,a[i]);
}
printf("最大值是 %d, 最小值是 %d\n",max,min);
}

```

#### 分析：

- (1) 有关查找数据的最大值和最小值的程序在前面已经编写过，在设计思想上没有什么变化。
- (2) 对于一维数组编程，通常涉及数据输入、数据处理、数据输出 3 个独立部分，每一部分由一个循环来完成。
- (3) 使用 scanf 函数实现数组元素的输入，在输入前给出必要的提示是用户界面友好的一种表现形式。
- (4) 输出时，首先输出数组的 10 个元素，然后输出最大值和最小值。
- (5) 编译运行程序，输入 10 个整数，验证程序的正确性。例如输入

21 37 6 17 9 12 89 76 35 59

运行结果如下：

```

a[0]=21
a[1]=37
a[2]=6
a[3]=17
a[4]=9
a[5]=12
a[6]=89
a[7]=76
a[8]=35
a[9]=59
最大值是 89, 最小值是 6

```

## 2. 程序二

编写程序，输入一个  $4 \times 4$  的二维数组，输出此二维数组后，再分别输出其主对角线与副对角线的和。

程序代码如下：

```

#include <stdio.h>
main()
{
    int a[4][4], i, j, sum1=0, sum2=0;
    printf("请输入 4*4 的二维整数数组:\n");
    for (i=0; i<4; i++)
        for (j=0; j<4; j++)
            scanf("%d", &a[i][j]);
    for (i=0; i<4; i++)
    {
        sum1=sum1+a[i][i];
        sum2=sum2+a[i][3-i];
    }
    for (i=0; i<4; i++)
    {
        for (j=0; j<4; j++)
            printf("%d\t", a[i][j]);
        printf("\n");
    }
    printf("主对角线之和是 %d\n 副对角线之和是 %d\n", sum1, sum2);
}

```

### 分析:

(1) 二维维数组的数据输入、数据处理、数据输出 3 个部分通常都各由一个双重循环来完成,本示例的数据处理部分不涉及该行的每一个数值,只与主对角线或副对角线上的值有关,因此仅使用针对行的单重循环来完成。

(2) 在二维数组的数据输出时,应在每行结束处加一个换行标记。

(3) 编译运行程序。输入数字 1~16,运行结果如下:

```

1     2     3     4
5     6     7     8
9    10    11    12
13   14   15   16

```

主对角线之和是 34

副对角线之和是 34

输入其他值,验证程序的正确性。

## 5.3 实验内容

(1) 对于一个存放任意 10 个元素的一维数组,将数组元素进行对调(即第一个元素和最后一个元素对调,第二个元素和第九个元素对调……)。要求初始化一个一维数组,

输出原始数组内容和对调后的结果。

(2) 在一个有序的整型数组中,插入一个整型数据并保持原来的排序顺序不变(提示:原有序数组为 1,2,3,7,8,9,插入数据 5 后的排序为 1,2,3,5,7,8,9)。要求:初始化一个有序数组,从键盘读入一个整型数据,输出该数组以及插入数据后的数组。

(3) 首先输入一个大于 2 且小于 10 的整数 n,然后定义一个二维整型数组( $n \times n$ ),初始化该数组,将数组中最大元素所在的行和最小元素所在的行对调。

**要求:**

①  $n \times n$  数组元素的值由 scanf 函数从键盘输入(假定最大值与最小值不在同一行上),然后输出该数组。

② 查找最大值与最小值所在行。

③ 将数组中最大元素所在的行和最小元素所在的行对调,并输出对调后的数组。

④ 为直观起见,数组按 n 行 n 列的方式输出。

(4) 将 3 个学生 4 门课程的成绩分别存放在  $4 \times 5$  数组的前  $3 \times 4$  的位置,计算每个学生的总成绩,存放在该数组的最后一列的对应行上;计算单科成绩的平均分,存放在最后一行的对应列上。

**要求:**

① 数组类型定义为实数类型,成绩由 scanf 函数从键盘输入。

② 输出原始成绩数据( $3 \times 4$ )。

③ 计算每个学生的总成绩以及单科成绩的平均分,并按要求填入数组中,输出结果数组( $4 \times 5$ )。

④ 数据保留一位小数。

(5) 输入一个字符串,编写程序,求字符串中的数字之和。

**要求:**

① 输出字符串中的各个数字之和。

② 如果字符串中没有数字字符,输出提示信息。

(6) 在给定的字符串中查找指定的子字符串。

**要求:**

① 输入两个字符串,给出友好的提示信息。

② 在字符串中查找指定的子字符串,如果存在,输出该子字符串在字符串中首次出现的位置。

③ 如果在给定的字符串中不存在指定的子字符串,则给出相应的说明信息。