

第7章 用函数实现模块化程序设计

7.1为什么要用函数

7.2怎样定义函数

7.3调用函数

7.4对被调用函数的声明和函数原型

7.5函数的多级嵌套调用

7.6递归函数设计

7.7数组作为函数参数

7.8局部变量和全局变量

7.9变量的存储类别和生存期

7.10函数说明符

7.11 内部函数和外部函数

7.7数组作为函数参数

7.7.1数组元素作函数实参

7.7.2数组名作函数参数

7.7.3多维数组名作函数参数

7.7.1 数组元素作函数实参

例7.9 输入**10**个数，要求输出所有值中的**最大值**。

➤ 解题思路：

- ◆ 存储：定义数组**a**，用来存放**10**个数
- ◆ 前面已设计函数**max**，用来求两个数中的大者
- ◆ 在主函数中定义变量**maxValue**，初值为**a[0]**，每次调用**max**函数后的返回值存放在**maxValue**中。
- ◆ 用“打擂台”算法，依次将数组元素从**a[1]**到**a[9]**与**maxValue**比较(调用**max**函数)，最后得到的**maxValue**值就是**10**个数中的最大者

```
#include <stdio.h>
int max(int x,int y);
int main()
```

```
{
    int a[10], maxValue, i;
    printf("请依次输入10个数(数间以空格分隔):\n");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    maxValue = a[0];
    for(i=1; i<10; i++)
    {
        maxValue = max(a[i], maxValue);
    }
    printf("最大值为:%d\n", maxValue);
    return 0;
}
int max(int x, int y){
    return x>y?x:y;
}
```

请依次输入10个数(数间以空格分隔):
7 -12 21 -76 56 32 29 12 -8 39
最大值为:56

7.7.2数组名作函数参数

- 上例中，用数组元素作为函数实参，若将上例改为求**10个数中的第几个数最大**，应该如何修改呢？
- 换言之：能不能写一个函数，求一个长度为**10**的数组中最大元素所在的位置(下标)呢？

◆输入：长度为**10**的数组

◆输出：最大值所在下标(**int**)

```
int getIndex(int a[10])  
{  
    int max=a[0], pos=0;  
    for(int i=1; i<10; i++)  
        if(a[i] > a[pos]) pos = i;  
    return pos;  
}
```

7.7.2数组名作函数参数

- 上述函数如何调用呢？实参是什么？
 - ◆ **int index = getIndex(实参?);**
- 长度为**10**的数组！如何表示？
- 解决方法：
 - ◆ 数组名做函数调用的实参
 - ◆ 数组名表示数组元素在内存中的首地址
 - ◆ 通过数组名(地址运算)可以访问到数组中的所有元素
- 主函数

```
int main()
{
    int a[10]={7,-12,21,76,56,32,29,12,-8,39}, index;
    index = getIndex(a);
    printf("第%d个数最大, 值为:%d\n", index, a[index]);
    return 0;
}
```

7.7.2数组名作函数参数

- 函数: **int getIndex(int a[10]);**
- 形参决定了数组的长度为**10**, 不适用于其他长度的数组, 那如果数组长度为**20**, ...等其他值时, **如何解决?**

- 解决方法:

◆ 变长数组

```
int getIndex(int a[], int n)
{
    int max=a[0], pos=0;
    for(int i=1; i<n; i++)
        if(a[i] > a[pos]) pos = i;
    return pos;
}
```

- 调用:

```
index = getIndex(a, 20);
index = getIndex(a, 5);
```

7.7.2数组名作函数参数

- 例**7.10** 编写一个对任意长度的数组进行排序的函数
- 冒泡排序

```
void sort(int a[], int n)
{
    int i, j;
    for(i=0; i<n-1; i++)
        for(j=0; j<n-1-i; j++)
            if(a[j]>a[j+1]){
                int t=a[j];a[j]=a[j+1];a[j+1]=t;
            }
}
```

- 调用

```
int main( )
{
    int a[10] = {0,9,8,7,6,5,4,3,2,1};
    sort(a, 10);
    .....
}
```


7.7.2数组名作函数参数

- 例**7.10** 编写一个对任意长度的数组进行排序的函数
- 冒泡排序

```
void sort(int a[], int n)
{
    int i, j;
    for(i=0; i<n-1; i++)
        for(j=0; j<n-1-i; j++)
            if(a[j]>a[j+1]){
                int t=a[j];a[j]=a[j+1];a[j+1]=t;
            }
}
```

- 调用

```
int main( )
{
    int a[10] = {0,9,8,7,6,5,4,3,2,1};
    sort(a, 10);
    .....
}
```

7.7.2数组名作函数参数

➤ 例**7.11** 选择法排序的函数

➤ 选择排序原理

◆选择法就是先将所有数中最小的与**a[0]**对换；再将从**a[1]**开始的所有数中最小的与**a[1]**对换.....每进行一轮，找出一个未经排序的数中最小的一个

◆共进行**n-1**轮

```
void sort(int a[], int n)
```

```
{
```

```
    int i, j;
```

```
    for(i=0; i<n-1; i++){
```

```
        int k = i;
```

```
        for(j=i+1; j<n; j++) //找第i小的数
```

```
            if(a[j]<a[k]) k = j;
```

```
        if(k!=i){          //将第i小的数换到位置i处
```

```
            int t=a[i]; a[i]=a[k]; a[k]=t;
```

```
        }
```

```
    }
```

```
}
```

7.7.2数组名作函数参数

参数传递的两种形式

➤ 值传递（传值调用）

- ◆实参的值复制到形参（值的赋值过程，将实参的值赋值给形参）

- ◆实参和形参是分离的，二者不再存在联系

➤ 地址传递（传地址调用）

- ◆传递的地址，主调函数和被调函数操作的是同一对象

- ◆实参和形参共享内存空间

➤ 数组名做函数参数和数组元素做函数参数的区别

- ◆数组名做函数参数——传地址调用

- 通过上述排序的例子可以看出，调用**sort**函数后，主调函数(**main**函数)中的数组就有序了。

- ◆数组元素做函数参数——传值调用

- 通过求最大值的函数**max**可以验证

7.7.3 多维数组名作函数参数

例**7.12** 写一个函数求一个 3×4 的矩阵中所有元素的最大值。

➤ 解题思路:

◆ 存储: 3×4 的矩阵用**3**行**4**列的二维数组表示

◆ 算法: 和求一维数组最大值类似。

```
int getMax(int a[][4], int m)
{
    int i, j, max=a[0][0];
    for(i=0; i<m; i++)
        for(j=0; j<4; j++)
            if(a[i][j] > max) max = a[i][j];
    return max;
}

int main(){
    int a[3][4]={.....}, max;
    max = getMax(a, 3);
    .....
}
```

7.8局部变量和全局变量

7.8.1 局部变量

7.8.2 全局变量

7.8.1 局部变量

- 定义变量可能有三种情况：
 - ◆ 在函数的内部定义
 - ◆ 在函数内的复合语句内定义
 - ◆ 在函数的外部定义
- 在函数内部或复合语句内部定义的变量称为“局部变量”
- 局部变量的作用域
 - ◆ 在一个函数内部定义的变量只在本函数范围内有效
 - ◆ 在复合语句内定义的变量只在本复合语句范围内有效

```
float f1( int a)
{ int b,c;
  .....
}
```

a、b、c仅在此
函数内有效

```
char f2(int x,int y)
{ int i,j;
  .....
}
```

x、y、i、j仅在此
函数内有效

```
int main( )
{ int m,n;
  .....
  return 0;
}
```

m、n仅在此函
数内有效

```
float f1( int a)
```

```
{ int b,c;
```

```
.....
```

```
{ int i,j,c;
```

```
.....
```

```
}
```

```
}
```

```
int main( )
```

```
{ int a,b;
```

```
.....
```

```
return 0;
```

```
}
```

类似于不同班
同名学生

i、**j**、**c**仅在此复
合语句内有效

a、**b**也仅在此函
数内有效


```
int main ( )  
{ int a,b;
```

.....

```
{ int c;  
  c=a+b;
```

.....

```
}
```

.....

```
}
```

a、**b**仅在此复合语句内有效

c仅在此复合语句内有效

7.8.2全局变量

- 在函数内定义的变量是局部变量，而在函数之外定义的变量称为**外部变量**
- 外部变量是全局变量(也称全程变量)
- 全局变量可以为本文件中其他函数所共用
- 有效范围为从定义变量的位置开始到本源文件结束

```
int p=1,q=5
```

```
float f1(int a)
```

```
{ int b,c; ..... }
```

```
char c1,c2;
```

```
char f2 (int x, int y)
```

```
{.....}
```

p、q的有效范围

p、q、c1、c2为
全局变量

c1、c2的有效范围

例7.13 有一个一维数组，内放**10**个学生成绩，写一个函数，当主函数调用此函数后，能求出平均分、最高分和最低分。

- 解题思路：调用一个函数最多可得一个返回值，现在希望通过函数调用能得到**3**个结果。除一个可以使用函数返回外，另外两个可以利用全局变量来达到此目的。

```
float max, min;
float avg(int score[], int n)
{
    float sum;
    int i;
    sum=max=min=score[0];
    for(i=1; i<n; i++){
        if(score[i]>max) max=score[i];
        if(score[i]<min) min = score[i];
        sum += score[i];
    }
    return sum/n;
}
```