

数组名作为函数参数-1

杨振平

数组名作为函数参数

- 数组元素作实参，对应的形参为变量，一次传递一个元素，采用值传递。
- 数组名作实参，对应的形参为一个数组，一次传递整个数组。
- 数组作参数，其参数传递可理解为形参数组与实参数组共用同一数组空间（即共用实参数组空间）。因此，在函数中，使用形数组就是在使用实数组，改变形数组元素的值就是在改变实数组元素的值，这一点与引用传递非常相似。

数组名作为函数参数（续）

1. 一维数组的传递

- ▶ 一维数组作形参的声明格式：

<类型> <数组名>[]

其中，[]中可以省略数组的长度值。（可认为形参数组与实参数组长度相同）

- ▶ 对应的实参应为同类型的一维数组名。（仅用数组名）

说明：为了使函数知道需要处理的数组元素的个数，通常给函数再传递一个表示元素个数的整型数。

数组名作为函数参数（续）

例如：一维数组名作为函数的参数。编写函数，计算一个整型数组中从第m个元素（m从0开始）开始的n个元素之和。

函数设计：

函数原型：**int fun(int b[],int m,int n);**

功能：计算数组b中从第m个元素开始的n个元素之和。

主函数设计：

定义并初始化一个整型数组a。

测试1：**fun(a,0,10);** // 求从第0个元素开始的10个元素之和

测试2：**fun(a,3,5);** // 求从第3个元素开始的5个元素之和

数组名作为函数参数（续）

函数代码：

```
int fun(int b[],int m,int n)
{
    int i,s=0;
    for(i=m;i<m+n;i++)
        s=s+b[i];
    return s;
}
```

主函数代码：

```
int main()
{
    int x,a[]={0,1,2,3,4,5,6,7,8,9};
    x=fun(a,0,10);
    cout<<x<<endl;
    x=fun(a,3,5);
    cout<<x<<endl;
    return 0;
}
```

45
25

例如：编写一个函数sort，对n个元素的一维整型数组b进行从小到大排序。

- 函数定义如下：

```
void sort(int b[],int n) ;//b为一维数组， n为元素的个数
{
    //选用一种排序方法对n个元素的b数组排序（代码略）
    //对b数组排序，实际就是对实参数组排序
}
```

- 主函数中的函数调用语句：

```
sort(a,10);
//调用sort函数，对10个元素的整型数组a进行排序
//函数调用后，并输出排序结果，即输出a数组中各个元素值。
```