

本章知识点小结

内容	实例	备注
下标运算符	[]	和圆括号一样，优先级最高。
数组类型		数组是一组具有相同类型的变量的集合，它是一种构造数据类型。其他两种构造数据类型（结构体和共用体）在第 12 章介绍。
一维数组的定义和初始化	<pre>int days[12] = {31,28,31,30,31,30,31,31,30,31,30,31};</pre>	
二维数组的定义和初始化	<pre>int days[2][12] = {{31,28,31,30,31,30,31,31,30,31,30,31}, {31,29,31,30,31,30,31,31,30,31,30,31}};</pre>	
向函数传递一维数组和二维数组	<pre>void DataSort(int score[], int n); int ReadScore(int score[][COURSE_N], long num[]);</pre>	数组作函数参数传给函数的是数组的首地址，因此属于按引用传参。普通变量作函数参数传给函数的是变量的内容，因此属于按值传参。按值传递实参时，该实参的值是不能在函数内被改变的。为了在函数内改变实参的值，必须按引用传参。
常用算法	排序、查找、求最大最小值	排序算法介绍了交换法和选择法，冒泡法在习题中介绍 查找算法介绍了顺序查找和折半查找

本章常见错误小结

常见错误实例	常见错误描述	错误类型
<code>arr(5)</code>	使用圆括号引用数组元素	编译错误
<code>matrix(2,3)</code>	使用圆括号且将行下标和列下标写在一个圆括号内引用数组元素	编译错误
<code>matrix[2,3]</code>	将行下标和列下标写在一个方括号内引用数组元素。即用形如 <code>a[x, y]</code> 而非 <code>a[x][y]</code> 的形式，来访问一个二维数组中的元素，C 编译器会将把 <code>a[x, y]</code> 解释为 <code>a[y]</code> ，并不认为这是一个编译错误。	运行时错误
<code>arr = {1,2,3,4,5};</code>	试图用数组名接收对数组元素的整体赋值。	编译错误
	忘记对需要进行元素初始化的数组进行初始化，将导致运行结果错误。	运行时错误
<code>int arr[4] = {1,2,3,4,5};</code>	在对数组元素进行初始化时提供的初值个数多于数组元素的个数。	编译错误
<code>int arr[n];</code>	使用变量而非整型常量来定义数组的长度。	编译错误
	没有意识到数组的下标都是从 0 开始的，在访问数组元素时发生下标“多 1”或者“少 1”的操作，从而引发越界访问内存错误。	运行时错误
<code>max=FindMax(score[],n);</code>	函数调用时实参数组名后跟着一对空的方括号。	编译错误
<code>max=FindMax(int score[],int n);</code>	按照函数定义首部的形参列表书写函数调用语句中实参列表。	编译错误
	函数原型、函数定义和函数调用，在形参和实参的数量以及返回值的类型上未严格保持一致。	编译错误
	函数原型、函数定义和函数调用，在形参和实参的类型和顺序上未严格保持一致。	运行时错误
<code>int ReadScore(int score[][], long num[]);</code>	多维数组作函数形参时，省略了除第一维以外的后面所有维的长度声明。	编译错误
	误以为在每次函数被调用时，在函数中定义的静态局部数组的元素都会被初始化为零。	理解错误