

本章知识点小结

内容	实例	备注
数组名的特殊含义		不带下标的数组名是一个常量指针，它代表数组元素的首地址。让声明为相同基类型的指针变量指向数组元素的首地址以后，既可用下标法也可用指针法引用数组元素。
多维数组	<pre>int a[3][4];</pre>	C 语言中的多维数组是按行存放的，因此，用指针法引用数组元素时，必须已知数组的列的维数。
二维数组的行指针	<pre>int (*p)[4]; p = a;</pre>	定义二维数组的行指针时，需指定它所指向的二维数组的列数。二维数组的行指针需用相同基类型的二维数组的第 0 行的首地址（即二维数组名）来初始化。
二维数组的列指针	<pre>int *p; p = *a;</pre>	定义二维数组的列指针时，无需指定它所指向的二维数组的列数。二维数组的列指针需用相同基类型的二维数组的第 0 行第 0 列的首地址（即二维数组名前面加*）来初始化。
向函数传递一维数组	<pre>void Input(int b[], int n);</pre>	一维数组作函数形参时，不必在方括号内显式指定数组的长度，通常用一个整型形参来指定数组的长度。
向函数传递二维数组	<pre>void Input(int a[][4], int m);</pre>	二维数组作函数形参时，需在第二个方括号内显式指定数组的列数，但不必在第一个方括号内显式指定数组的行数，通常用一个整型形参来指定数组的行数。
指针数组	<pre>char *ptr[]={ "China", "Japan"}; void main(int argc, char *argv[]) { }</pre>	指针数组的两个重要应用是： 1) 对多个字符串进行处理操作（如排序等）； 2) 用作不带参数的 main() 函数的参数，指向命令行中的字符串。

本章常见错误小结

常见错误描述	错误类型
对并没有指向数组中某个元素的指针变量进行算术运算	无意义操作
对并非指向同一数组元素的两个指针进行相减或比较运算	无意义操作
每个数组都有上/下两个边界，在对指向数组的指针进行算术运算时，使指针超出了数组的边界，而发生越界访问内存的错误	运行时错误
试图以指针运算的方式来改写一个数组名所代表的地址	编译错误