

本章知识点小结

内容	备注
何谓指针	指针是一种特殊的数据类型。
何谓指针变量	指针变量是具有指针类型的变量，是专门用于存储变量的地址值的变量。 指针变量与其他类型变量相比，其共性是都在内存中占据一定大小的存储单元；都需要先定义、后使用。指针变量的特殊性表现在：它的内容只能是地址，而不能是数据；必须初始化后才能使用，否则指向不确定的存储单元；只能指向同一基类型的变量；只能参与加（减）整数，自增、自减、关系、赋值等运算。
指针变量的定义和初始化	使用指针变量之前必须先对其进行初始化，在定义指针变量的同时为其初始化是一种良好的编程习惯。
取地址运算符&	用于获取变量的内存地址。
间接寻址运算符*	用于访问变量的内容。
指针变量做函数参数	指针做函数参数是按地址调用，需要将函数外的某个变量的地址作为实参传给函数相应的指针形参 这为函数提供了在函数内修改实参变量的手段。而普通变量作函数参数是按值调用，函数内的代码不能通过普通类型的形参改变函数外的实参变量的值。
使用指针变量时应恪守的原则	永远清楚每个指针指向什么位置；永远清楚每个指针指向的位置中的内容是什么；永远不要使用未初始化的指针变量。
字符串常量，例如"Hello"	字符串常量是由一对双引号括起来的一个字符序列。而用一对单引号括起来的一个字符，称为字符常量。
字符数组，例如： char str[]={ 'H','e','l','l','o','\0'}; char str[] = "Hello";	可用下面两种方法对字符数组进行初始化： 1) 用字符型数据对数组进行初始化 2) 用字符串常量直接对数组初始化
字符指针，例如 char *ptr = "Hello";	字符指针是指向字符型数据的指针变量
字符串的输入输出，例如： gets(str); puts(str);	使用 puts()函数时，不需要加换行符'\n'。 gets()不能限制输入字符串的长度。而 fgets()能够限制输入字符串的长度
字符串处理函数，例如 strlen(str); strcpy(str1,str2); strcmp(str1,str2); strcat(str1,str2);	计算字符串长度 字符串复制 字符串比较 字符串连接

本章常见错误小结

常见错误实例	常见错误描述	错误类型
	混淆指针与地址的概念，误以为指针就是地址，地址就是指针。	理解错误
int *pa, pb;	误以为用来声明指针变量的星号（*）会对同一个声明语句中的所有指针变量都起作用，而省略了其他指针变量名前的星号前缀。	理解错误
int *p; scanf("%d", p); ... *p = 1;	在没有对指针变量进行初始化,或没有将指针变量指向内存中某一个确定的存储单元的情况下,就利用这个指针变量去访问它所指向的存储单元,从而造成非法内存访问,即代码访问了不该访问的内存地址,指针未初始化和数组下标越界都会引起非法内存访问错误。	运行时错误
... Swap(a, b); ... void Swap(int *x, int *y) { ... }	没有意识到某些函数形参是传地址调用,把变量的值而非变量的地址当作实参传给了这些形参。	运行时错误
int *pa = &a; float *pb = &b; pa = pb;	在不同基类型的指针变量之间赋值。	编译错误

<code>int *p; p = 100;</code>	用非地址值为指针变量赋值。	提示 warning
<code>int i; float *p; p=&i;</code>	将指针变量指向与其类型不同的变量。	提示 warning
<code>void *p = NULL; *p = 100;</code>	试图用一个 void 类型的指针变量去访问内存。	编译错误
<code>"a"</code>	误以为用单个字符构成的字符串只占 1 个字节的内存。	理解错误
	误以为在函数中定义的静态 (static) 局部数组中的元素, 在每次函数被调用时都会被初始化为零。	理解错误
<code>char str[6]="secret";</code>	没有定义一个足够大的字符数组来保存字符串结束标志'\0'。	运行时错误
<code>char str[4]; strcpy(str, "secretabc");</code>	在执行字符串处理操作时, 没有提供空间足够大的空间用于存储处理后的字符串。	运行时错误
	在执行字符串处理操作时, 忘记在字符串的末尾添加字符串结束标志'\0'。	运行时错误
	在输入字符串时, 没有提供空间足够大的字符数组来存储, 即用户从键盘键入的字符个数超过了字符数组所含元素的个数。	运行时错误
	打印一个不包含字符串结束标志'\0'的字符数组。	运行时错误
<code>scanf("%s", &a);</code>	用 scanf()读取字符串时, 代表地址值的数组名前面添加了取地址符&。	运行时错误
	用 scanf()而非 gets()输入带空格的字符串。	运行时错误
<code>char *p; char s[] = "abc"; strcpy(p, s);</code>	字符串指针未被初始化就使用。	运行时错误
<code>char str[10]; str++;</code>	将数组名当做指针变量进行增 1 和减 1 操作, 使其使其指向字符串中的某个字符。	编译错误
<code>if (str1 == str2)</code>	直接使用关系运算符而未使用函数 strcmp()来比较字符串大小。	运行时错误
<code>str = "secret";</code>	直接使用赋值运算符而未使用函数 strcpy()对字符数组进行赋值。	编译错误
	误以为函数 strcpy(str1,str2)是将字符串 str1 复制到字符串 str2 中。	理解错误