C语言辅导:A I／ISO标准 PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/285/2021_2022_C_E8_AF_AD_E8_A8_80_E8_BE_85_c97_285290.htm 如果你不理解C语言标准的价值，你就不会知道你是怎样地幸运。 一个C程序员会期望一个C程序无论是在哪里开发的，在另一个编译程序中都能通过编译。实际上不能完全做到这一点，因为许多头文件和函数库都是针对某些特定的编译程序或平台的。有些(很少!)语言扩充性能，例如基于Intel的编译程序所使用的near和far关键字以及寄存器伪变量，也只不过是某种平台的开发商们所认可的一种标准。 如果你认为靠一种标准走遍天下是理所当然的，就象左脚踩加速器，右脚踩刹车一样，那么你的视野未免有些狭窄。有两种不同的BASIC标准，但都没有得到广泛的支持；世界上最流行的Pascal编译程序并不符合正式的标准；现在正在发展的C 标准，由于变化太快，也没有得到广泛的支持；有些实现遵循一种严格的Ada标准，但Ada标准也没能大规模地占领世界市场。 从技术上讲有两种C语言标准，一种来自ANSI(American National Standard Institute，美国国家标准协会)X3J11委员会，另一种来自ISO(International Standard Organization，国际标准协会)98991990。由于ISO标准中的某些改进优于ANSI标准，而ANSI标准也接受了这个国际版本，因此"ANSI／ISO标准 ”是一种正确的说法。 那么，这种标准对你有什么帮助呢?你可以买到一份该标准的副本，即Herbert Schildt所著的((The Annotated ANSI C Standard》(Osborne McGraw-Hill出版，ISBN O-07-881952-O)一书，该书对语言和库都作了介绍，并带有

注释。这本书比大多数正式标准要便宜多了，后者由ANSI和ISO出售，以解决建立标准所需的部分费用。并不是每一个C程序员都需要这样一本书，但它是最权威的。 最重要的一点是，ANSI／ISO标准是对“ 什么是c? ”这一问题的权威解答。如果编译程序开发商所做的某些实现不符合这一标准，你可以把它作为错误指出来，这不会引起争论。 ANSI／ISO标准也不是包罗万象的。具体地说，它没有涉及c程序可能会做的许多有趣的事情，例如图形或多任务。许多兼容性不强的标准包含了这些内容，其中的一些将来可能会成为权威的标准，因此你不必完全拘泥于ANSI／ISO标准。 顺便提一句，除编程语言之外，还有许多东西也有ANSI标准，其中的一种就是ANSI为全屏幕文本操作的退出序列集合而写的标准，在第17章中所介绍的MSDOS的"ANSI驱动程序 ”指的就是这种标准(有趣的是，MS-DOS的ANSI．SYS只实现了ANSI标准序列中的一小部分)。 16．1 运算符的优先级总能起作用吗? 有关运算符优先级的规则稍微有点复杂。在大多数情况下，这些规则确实是你所需要的，然而，有人也指出其中的一些规则本来是可以设计得更好的。 让我们快速地回顾一些有关内容：“ 运算符优先级 ”是这样一些规则的集合这些规则规定了“ 运算符 ”(例如，-，等等)的优先性，即哪一种运算符先参加运算。在数学中，表达式“ 2×3 4×5 ”和“ (2×3) (4×5) ”是等价的，因为乘法运算在加法运算之前进行，也就是说乘法的优先级比加法高。 在c中，有16级以上的运算符优先级。尽管这么多的规则有时使c程序不易阅读，但也使C程序写起来容易多了。虽然这不是唯一的一种折衷方法，但这就是C所采用的方法。表16．1总结了运算符的优先级。 表16

．1运算符优先级总结(从高到低)

---------- 优先级 运算符

---------- 1 x[y](下标) x(y)(函数调用) x．y(访问成员) x->y(访问成员指针) x (后缀自增) x--(后缀自减)-- 2 x(自增) --x(自减) amp.y(按位与) 10 x^y(按位异或) ． 11 x | y(按位或) 12 xamp.y(逻辑与) 13 x||y(逻辑或) 14 x?y：z(条件) x=y，x*=y，x／=y，x =y，x-=y，>=，&amp.=，^=，|=(赋值，右结合性) 16 x，y(逗号)

---------- 优先级最高的是后缀表达式，即运算符跟在一个表达式后面；其次是前缀或单目表达式，即运算符位于一个表达式的前面；再次是强制类型转换表达式。 注意：关于运算符优先级，最重要的是知道*p 和*(p)是等价的。也就是说，在*p 中， 运算符作用在指针上，而不是作用在指针所指向的对象上。象 “ *p =*q ；这样的代码在C中是随处可见的，其中的优先级和 “ (*(p))=(*(q)) ” 中的是相同的。这个表达式的含义是 “ q 1，但仍用q原来的值找到q所指向的对象；p加1，但仍用p原来的值；把q所指向的对象赋给p所指向的对象 ”，整个表达式的值就是原来q所指向的对象。在C中你会经常看到这样的代码，并且你会有许多机会去写这样的代码。对于其它运算符，如果你记不住其优先级，可以查阅有关资料，但是，一个好的c程序员应该连想都不用想就能明白*p的含义。 最初的C编译程序是为这样一种计算机编写的它的某些指令对象*p 和*p =*q 这样的代码的处理效率高得令人难

以置信，因此，很多C代码就写成这种形式了。进一步地，因为象这样的C代码实在太多了，所以新机型的设计者会保证提供能非常高效地处理这些C代码的指令。 再下一级的优先级是乘法、除法和求余(也叫取模)，再往后是加法和减法。与数学中的表达式相同，“ 2*3 4*5 ”和“ (2*3) (4*5) ”是等价的。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com