

计算机等级考试二级C语言运算符 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/135/2021\\_2022\\_\\_E8\\_AE\\_A1\\_E7\\_AE\\_97\\_E6\\_9C\\_BA\\_E7\\_c97\\_135275.htm](https://www.100test.com/kao_ti2020/135/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E7_c97_135275.htm)

C语言的内部运算符很丰富，运算符是告诉编译程序执行特定算术或逻辑操作的符号。C语言有三大运算符：算术、关系与逻辑、位操作。另外，C还有一些特殊的运算符，用于完成一些特殊的任务。

2.6.1 算术运算符表2-5列出了C语言中允许的算术运算符。在C语言中，运算符“+”、“-”、“\*”和“/”的用法与大多数计算机语言的相同，几乎可用于所有C语言内定义的数据类型。当“/”被用于整数或字符时，结果取整。例如，在整数除法中， $10/3=3$ 。一元减法的实际效果等于用-1乘单个操作数，即任何数值前放置减号将改变其符号。模运算符“%”在C语言中也同它在其它语言中的用法相同。切记，模运算取整数除法的余数，所以“%”不能用于float和double类型。最后一行打印一个0和一个1，因为1/2整除时为0，余数为1，故 $1\%2$ 取余数1。

2.6.2 自增和自减C语言中有两个很有用的运算符，通常在其它计算机语言中是找不到它们的自增和自减运算符，和--。运算符“++”是操作数加1，而“--”是操作数减1，换句话说： $x=x+1$ 同 $x.x=x-1$ 同 $--x$ 。自增和自减运算符可用在操作数之前，也可放在其后，例如： $x=x+1$ ；可写成 $x++$ ；或 $++x$ ；但在表达式中这两种用法是有区别的。自增或自减运算符在操作数之前，C语言在引用操作数之前就先执行加1或减1操作；运算符在操作数之后，C语言就先引用操作数的值，而后再进行加1或减1操作。请看下例： $x=10.y=x$ 。此时 $y=11$ 。如果程序改为： $x=10.y=x++$ 。则 $y=10$ 。在这两种情况下

，x都被置为11，但区别在于设置的时刻，这种对自增和自减发生时刻的控制是非常有用的。在大多数C编译程序中，为自增和自减操作生成的程序代码比等价的赋值语句生成的代码要快得多，所以尽可能采用加1或减1运算符是一种好的选择。下面是算术运算符的优先级：：最高、----（一元减）\*、/、%最低、-编译程序对同级运算符按从左到右的顺序进行计算。当然，括号可改变计算顺序。C语言处理括号的方法与几乎所有的计算机语言相同：强迫某个运算或某组运算的优先级升高。2.6.3关系和逻辑运算符关系运算符中的“关系”二字指的是一个值与另一个值之间的关系，逻辑运算符中的“逻辑”二字指的是连接关系的方式。因为关系和逻辑运算符常在一起使用，所以将它们放在一起讨论。关系和逻辑运算符概念中的关键是True（真）和Flase（假）。C语言中，非0为True，0为Flase。使用关系或逻辑运算符的表达式对Flase和Ture分别返回值0或1(见表2-6)。表2-6给出于关系和逻辑运算符，下面用1和0给出逻辑真值表。关系和逻辑运算符的优先级比算术运算符低，即像表达式10>1 12的计算可以假定是对表达式10>(1 12)的计算，当然，该表达式的结果为Flase。在一个表达式中允许运算的组合。例如

：10>5&&! (10这一表达式的结果为True。。下表给出了关系和逻辑运算符的相对优先级：：最高！！>= === !==&最低||同算术表达式一样，在关系或逻辑表达式中也使用括号来修改原计算顺序。切记，所有关系和逻辑表达式产生的结果不是0就是1，所以下面的程序段不仅正确而且将在屏幕上打印数值1。int x..x=100..printf("%d",x>10).2.6.4位操作符与其它语言不同，C语言支持全部的位操作符（BitwiseOperators）。

因为C语言的设计目的是取代汇编语言，所以它必须支持汇编语言所具有的运算能力。位操作是对字节或字中的位（bit）进行测试、置位或移位处理，这里字节或字是针对C标准中的char和int数据类型而言的。位操作不能用于float、double、longdouble、void或其它复杂类型。表2-7给出了位操作的操作符。位操作中的AND、OR和NOT（1的补码）的真值表与逻辑运算等价，唯一不同的是位操作是逐位进行运算的。如表2-8所示，当且仅当一个操作数为True时，异或的输出为True，否则为False。位操作通常用于设备驱动程序，例如调制解调器程序、磁盘文件管理程序和打印机驱动程序。这是因为位操作可屏蔽掉某些位，如奇偶校验位（奇偶校验位用于确保字节中的其它位不会发生错误通常奇偶校验位是字节的最高位）。通常我们可把位操作AND作为关闭位的手段，这就是说两个操作数中任一为0的位，其结果中对应位置为0。例如，下面的函数通过调用函数read\_modem()，从调制解调器端口读入一个字符，并将奇偶校验位置成0。字节的位8是奇偶位，将该字节与一个位1到位7为1、位8为0的字节进行与操作，可将该字节的奇偶校验位置成0。表达式`ch&0x7F`却为False(0)。记住，关系和逻辑操作符结果不是0就是1。而相似的位操作通过相应处理，结果可为任意值。换言之，位操作可以有0或1以外的其它值，而逻辑运算符的计算结果总是0或1。移位操作符`>>`和`<<`右移位数

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](http://www.100test.com)