我们可以看到N是8；当y是负数时，加上偏置量7，并且右移3位。

2.44　这些“C的谜题”清楚地告诉程序员必须理解计算机运算的属性。

A. (x > 0) || ((x-1) < 0)

假。设x等于−2 147 483 648($TMin_{32}$)。那么，我们有x-1等于2147483647($TMax_{32}$)。

B. (x & 7) != 7 || (x << 29 < 0)

真。如果(x & 7) != 7这个表达式的值为0，那么我们必须有位x_2等于1。当左移29位时，这个位将变成符号位。

C. (x * x) >= 0

假。当x为65 535(0xFFFF)时，x * x为−131 071(0xFFFE0001)。

D. x < 0 || -x <= 0

真。如果x是非负数，则-x是非正的。

E. x > 0 || -x >= 0

假。设x为−2 147 483 648($TMin_{32}$)。那么x和-x都为负数。

F. x+y == uy+ux

真。补码和无符号乘法有相同的位级行为，而且它们是可交换的。

G. x*~y + uy*ux == -x

真。~y等于-y-1。uy*ux等于x*y。因此，等式左边等价于x*-y-x+x*y。

2.45　理解二进制小数表示是理解浮点编码的一个重要步骤。这个练习让你试验一些简单的例子。

小数值	二进制表示	十进制表示
$\frac{1}{8}$	0.001	0.125
$\frac{3}{4}$	0.11	0.75
$\frac{25}{16}$	1.1001	1.5625
$\frac{43}{16}$	10.1011	2.6875
$\frac{9}{8}$	1.001	1.125
$\frac{47}{8}$	101.111	5.875
$\frac{51}{16}$	11.0011	3.1875

考虑二进制小数表示的一个简单方法是将一个数表示为形如$\frac{x}{2^k}$的小数。我们将这个形式表示为二进制的过程是：使用x的二进制表示，并把二进制小数点插入从右边算起的第k个位置。举一个例子，对于$\frac{25}{16}$，我们有$25_{10}=11001_2$。然后我们把二进制小数点放在从右算起的第4位，得到1.1001_2。

2.46　在大多数情况中，浮点数的有限精度不是主要的问题，因为计算的相对误差仍然是相当低的。然而在这个例子中，系统对于绝对误差是很敏感的。

A. 我们可以看到$0.1-x$的二进制表示为：

$$0.000000000000000000000001100[1100]\cdots_2$$

B. 把这个表示与$\frac{1}{10}$的二进制表示进行比较，我们可以看到这就是$2^{-20}\times\frac{1}{10}$，也就是大约$9.54\times10^{-8}$。

C. $9.54\times10^{-8}\times100\times60\times60\times10\approx0.343$秒。

D. $0.343\times2000\approx687$米。

2.47　研究字长非常小的浮点表示能够帮助澄清IEEE浮点是怎样工作的。要特别注意非规格化数和规格化数之间的过渡。