除以 2 的幂也可以用移位运算来实现，只不过我们用的是右移，而不是左移。无符号和补码数分别使用逻辑移位和算术移位来达到目的。

整数除法总是舍入到零。为了准确进行定义，我们要引入一些符号。对于任何实数 a，定义 $\lfloor a \rfloor$ 为唯一的整数 a'，使得 $a' \leqslant a < a'+1$。例如，$\lfloor 3.14 \rfloor = 3$，$\lfloor -3.14 \rfloor = -4$ 而 $\lfloor 3 \rfloor = 3$。同样，定义 $\lceil a \rceil$ 为唯一的整数 a'，使得 $a'-1 < a \leqslant a'$。例如，$\lceil 3.14 \rceil = 4$，$\lceil -3.14 \rceil = -3$，而 $\lceil 3 \rceil = 3$。对于 $x \geqslant 0$ 和 $y > 0$，结果会是 $\lfloor x/y \rfloor$，而对于 $x < 0$ 和 $y > 0$，结果会是 $\lceil x/y \rceil$。也就是说，它将向下舍入一个正值，而向上舍入一个负值。

对无符号运算使用移位是非常简单的，部分原因是由于无符号数的右移一定是逻辑右移。

原理：除以 2 的幂的无符号除法

C 变量 x 和 k 有无符号数值 x 和 k，且 $0 \leqslant k < w$，则 C 表达式 x>>k 产生数值 $\lfloor x/2^k \rfloor$。

例如，图 2-28 给出了在 12 340 的 16 位表示上执行逻辑右移的结果，以及对它执行除以 1、2、16 和 256 的结果。从左端移入的 0 以斜体表示。我们还给出了用真正的运算做除法得到的结果。这些示例说明，移位总是舍入到零的结果，这一点与整数除法的规则一样。

k	>> k（二进制）	十进制	$12340/2^k$
0	0011000000110100	12340	12340.0
1	*0*001100000011010	6170	6170.0
4	*0000*001100000011	771	771.25
8	*00000000*00110000	48	48.203125

图 2-28　无符号数除以 2 的幂（这个例子说明了执行一个逻辑右移 k 位与除以 2^k 再舍入到零有一样的效果）

推导：除以 2 的幂的无符号除法

设 x 为位模式 $[x_{w-1}, x_{w-2}, \cdots, x_0]$ 表示的无符号整数，而 k 的取值范围为 $0 \leqslant k < w$。设 x' 为 $w-k$ 位位表示 $[x_{w-1}, x_{w-2}, \cdots, x_k]$ 的无符号数，而 x'' 为 k 位位表示 $[x_{k-1}, \cdots, x_0]$ 的无符号数。由此，我们可以看到 $x = 2^k x' + x''$，而 $0 \leqslant x'' < 2^k$。因此，可得 $\lfloor x/2^k \rfloor = x'$。

对位向量 $[x_{w-1}, x_{w-2}, \cdots, x_0]$ 逻辑右移 k 位会得到位向量

$$[0, \cdots, 0, x_{w-1}, x_{w-2}, \cdots, x_k]$$

这个位向量有数值 x'，我们看到，该值可以通过计算 x>>k 得到。■

对于除以 2 的幂的补码运算来说，情况要稍微复杂一些。首先，为了保证负数仍然为负，移位要执行的是算术右移。现在让我们来看看这种右移会产生什么结果。

原理：除以 2 的幂的补码除法，向下舍入

C 变量 x 和 k 分别有补码值 x 和无符号数值 k，且 $0 \leqslant k < w$，则当执行算术移位时，C 表达式 x>>k 产生数值 $\lfloor x/2^k \rfloor$。

对于 $x \geqslant 0$，变量 x 的最高有效位为 0，所以效果与逻辑右移是一样的。因此，对于非负数来说，算术右移 k 位与除以 2^k 是一样的。作为一个负数的例子，图 2-29 给出了对 $-12\,340$ 的 16 位表示进行算术右移不同位数的结果。对于不需要舍入的情况（$k=1$），结果是 $x/2^k$。但是当需要进行舍入时，移位导致结果向下舍入。例如，右移 4 位将会把 -771.25 向下舍入为 -772。我们需要调整策略来处理负数 x 的除法。