

位、 $k=15$  个阶码位、1 个单独的整数位和  $n=63$  个小数位。整数位是 IEEE 浮点表示中隐含位的显式副本。也就是说，对于规格化的值它等于 1，对于非规格化的值它等于 0。填写下表，给出用这种格式表示的一些“有趣的”数字的近似值。

描 述	扩展精度	
	值	十进制
最小的正非规格化数		
最小的正规格化数		
最大的规格化数		

将数据类型声明为 long double，就可以把这种格式用于为与 Intel 兼容的机器编译 C 程序。但是，它会强制编译器以传统的 8087 浮点指令为基础生成代码。由此产生的程序很可能会比数据类型为 float 或 double 的情况慢上许多。

- \* 2.87 2008 版 IEEE 浮点标准，即 IEEE 754-2008，包含了一种 16 位的“半精度”浮点格式。它最初是由计算机图形公司设计的，其存储的数据所需的动态范围要高于 16 位整数可获得的范围。这种格式具有 1 个符号位、5 个阶码位 ( $k=5$ ) 和 10 个小数位 ( $n=10$ )。阶码偏置量是  $2^{5-1}-1=15$ 。

对于每个给定的数，填写下表，其中，每一列具有如下指示说明：

Hex: 描述编码形式的 4 个十六进制数字。

$M$ : 尾数的值。这应该是一个形如  $x$  或  $\frac{x}{y}$  的数，其中  $x$  是一个整数，而  $y$  是 2 的整数幂。例

如：0、 $\frac{67}{64}$  和  $\frac{1}{256}$ 。

$E$ : 阶码的整数值。

$V$ : 所表示的数字值。使用  $x$  或者  $x \times 2^z$  表示，其中  $x$  和  $z$  都是整数。

$D$ : (可能近似的)数值，用 printf 的格式规范 %f 打印。

举一个例子，为了表示数  $\frac{7}{8}$ ，我们有  $s=0$ ， $M=\frac{7}{4}$  和  $E=-1$ 。因此这个数的阶码字段为

01110<sub>2</sub> (十进制值  $15-1=14$ )，尾数字段为 1100000000<sub>2</sub>，得到一个十六进制的表示 3B00。其数值为 0.875。

标记为“—”的条目不用填写。

描 述	Hex	$M$	$E$	$V$	$D$
-0				-0	-0.0
最小的 $>2$ 的值					
512				512	512.0
最大的非规格化数					
$-\infty$		—	—	$-\infty$	$-\infty$
十六进制表示为 3B00 的数	3B00				

- \*\* 2.88 考虑下面两个基于 IEEE 浮点格式的 9 位浮点表示。

1. 格式 A

- 有一个符号位。
- 有  $k=5$  个阶码位。阶码偏置量是 15。
- 有  $n=3$  个小数位。

2. 格式 B

- 有一个符号位。
- 有  $k=4$  个阶码位。阶码偏置量是 7。