C数据类型	最小值	最大值
[signed]char	-128	127
unsigned char	0	255
short	-32 768	32 767
unsigned short	0	65 535
int	-2 147 483 648	2 147 483 647
unsigned	0	4 294 967 295
long	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
unsigned long	0	18 446 744 073 709 551 615
int32_t	-2 147 483 648	2 147 483 647
uint32_t	0	4 294 967 295
int64_t	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
uint64_t	0	18 446 744 073 709 551 615

图 2-10　64 位程序上 C 语言整型数据类型的典型取值范围

图 2-9 和图 2-10 中一个很值得注意的特点是取值范围不是对称的——负数的范围比整数的范围大 1。当我们考虑如何表示负数的时候，会看到为什么会这样。

C 语言标准定义了每种数据类型必须能够表示的最小的取值范围。如图 2-11 所示，它们的取值范围与图 2-9 和图 2-10 所示的典型实现一样或者小一些。特别地，除了固定大小的数据类型是例外，我们看到它们只要求正数和负数的取值范围是对称的。此外，数据类型 int 可以用 2 个字节的数字来实现，而这几乎回退到了 16 位机器的时代。还可以看到，long 的大小可以用 4 个字节的数字来实现，对 32 位程序来说这是很典型的。固定大小的数据类型保证数值的范围与图 2-9 给出的典型数值一致，包括负数与正数的不对称性。

C数据类型	最小值	最大值
[signed]char	-127	127
unsigned char	0	255
short	-32 767	32 767
unsigned short	0	65 535
int	-32 767	32 767
unsigned	0	65 535
long	-2 147 483 647	2 147 483 647
unsigned long	0	4 294 967 295
int32_t	-2 147 483 648	2 147 483 647
uint32_t	0	4 294 967 295
int64_t	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
uint64_t	0	18 446 744 073 709 551 615

图 2-11　C 语言的整型数据类型的保证的取值范围。C 语言标准要求这些数据类型必须至少具有这样的取值范围

给 C 语言初学者　C、C++ 和 Java 中的有符号和无符号数

C 和 C++ 都支持有符号(默认)和无符号数。Java 只支持有符号数。

2.2.2　无符号数的编码

假设有一个整数数据类型有 w 位。我们可以将位向量写成 $\vec{x}$，表示整个向量，或者写成$[x_{w-1}, x_{w-2}, \cdots, x_0]$，表示向量中的每一位。把 $\vec{x}$ 看做一个二进制表示的数，就获得