The whole painful spiel on why we have signed integers

The signed-unsigned business all has to do with how numbers are stored inside a computer. The secret is that everything, no matter how it looks on the screen or in your program, is stored in the binary tongue inside the computer. That's counting in base 2 (ones and zeroes).

Binary numbers are composed of *bits,* or *b*inary dig*its*. Suppose that your C language compiler uses two bytes to store an integer. Those two bytes contain 16 binary digits, or bits. (Eight bits are in a byte.) For example:

```
0111 0010 1100 0100
```

This value is written as 29,380 in decimal (the human counting system). In binary, the ones and zeroes represent various multiples of two, which can get quite complex before your eyes, but is like eating ice cream to the computer.

Look at this number:

```
0111 1111 1111 1111
```

It's the value 32,767 — almost a solid bank of ones. If you add 1 to this value, you get the following amazing figure:

```
1000 0000 0000 0000
```

How the computer interprets this binary value depends on how you define your variable. For a signed value, a `1` in the far left position of the number isn't a `1` at all. *It's a minus sign.* The preceding number becomes –32,768 in binary math. If the variable is an unsigned value, it's interpreted as positive 32,768.

The deal with signed and unsigned numbers all depends on that pesky first bit in the computer's binary counting tongue. If you're working with a signed number, the first bit is the minus sign. Otherwise, the first bit is just another droll bit in the computer, happy to be a value and not a minus sign.

Table 9-2 What Signed and Unsigned Variables Can Hold

Signed	*Range*	*Unsigned*	*Range*
char	–128 to 127	unsigned char	0 to 255
int	–32768 to 32,767	unsigned int	0 to 65,535
long	–2,147,483,648	unsigned long	0 to 4,294,967,295 to 2,147,483,647

- Floating-point numbers (numbers with a decimal part or fractions) can be positive or negative without regard to any signed or unsigned nonsense.
- Floating-point numbers are covered in the following section.
- Normally, the differences between signed and unsigned values shouldn't bother you.