

**ceil**    The `ceil` (“ceiling”) function returns—as a `double` value—the smallest integer that’s greater than or equal to its argument. **floor** returns the largest integer that’s less than or equal to its argument:

```
ceil(7.1)    ⇒ 8.0
ceil(7.9)    ⇒ 8.0
ceil(-7.1)   ⇒ -7.0
ceil(-7.9)   ⇒ -7.0
```

```
floor(7.1)   ⇒ 7.0
floor(7.9)   ⇒ 7.0
floor(-7.1)  ⇒ -8.0
floor(-7.9)  ⇒ -8.0
```

In other words, `ceil` “rounds up” to the nearest integer, while `floor` “rounds down.” C89 lacks a standard function that rounds to the nearest integer, but we can easily use `ceil` and `floor` to write our own:

```
double round_nearest(double x)
{
    return x < 0.0 ? ceil(x - 0.5) : floor(x + 0.5);
}
```

**C99** C99 provides several functions that round to the nearest integer, as we’ll see in the next section.

**fabs**    `fabs` computes the absolute value of a number:

```
fabs(7.1)    ⇒ 7.1
fabs(-7.1)   ⇒ 7.1
```

**fmod**    `fmod` returns the remainder when its first argument is divided by its second argument:

```
fmod(5.5, 2.2) ⇒ 1.1
```

C doesn’t allow the `%` operator to have floating-point operands, but `fmod` is a more-than-adequate substitute.

## 23.4 The <math.h> Header (C99): Mathematics

The C99 version of the <math.h> header includes the entire C89 version, plus a host of additional types, macros, and functions. The changes to this header are so numerous that I’ve chosen to cover them separately. There are several reasons why the standards committee added so many capabilities to <math.h>:

- *Provide better support for the IEEE floating-point standard.* C99 doesn’t mandate the use of the IEEE standard; other ways of representing floating-point