The log1p function exists for a similar reason. For values of x that are close to zero, log1p(x) should be more accurate than log(1 + x).

Q: Why is the function that computes the gamma function named tgamma instead of just gamma? [p. 606]

A: At the time the C99 standard was being written, some compilers provided a function named gamma, but it computed the log of the gamma function. This function was later renamed lgamma. Choosing the name gamma for the gamma function would have conflicted with existing practice, so the C99 committee decided on the name tgamma ("true gamma") instead.

Q: Why does the description of the nextafter function say that if x and y are equal, nextafter returns y? If x and y are equal, what's the difference between returning x or y? [p. 609]

A: Consider the call nextafter(-0.0, +0.0), in which the arguments are mathematically equal. By returning y instead of x, the function has a return value of +0.0 (rather than –0.0, which would be counterintuitive). Similarly, the call nextafter(+0.0, -0.0) returns –0.0.

Q: Why does <string.h> provide so many ways to do the same thing? Do we really need four copying functions (memcpy, memmove, strcpy, and strncpy)? [p. 616]

A: Let's start with memcpy and strcpy. These functions are used for different purposes. strcpy will only copy a character array that's terminated with a null character (a string, in other words); memcpy can copy a memory block that lacks such a terminator (an array of integers, for example).

The other functions allow us to choose between safety and performance. strncpy is safer than strcpy, since it limits the number of characters that can be copied. We pay a price for safety, however, since strncpy is likely to be slower than strcpy. Using memmove involves a similar trade-off. memmove will copy bytes from one region of memory into a possibly overlapping region. memcpy isn't guaranteed to work properly in this situation; however, if we can guarantee no overlap, memcpy is likely to be faster than memmove.

Q: Why does the strspn function have such an odd name? [p. 620]

A: Instead of thinking of strspn's return value as the index of the first character that's *not* in a specified set, we could think of it as the length of the longest "span" of characters that *are* in the set.

Exercises

Section 23.3

Ⓦ 1. Extend the round_nearest function so that it rounds a floating-point number *x* to *n* digits after the decimal point. For example, the call round_nearest(3.14159, 3) would