```
long double nextafterl(long double x, long double y);

double nexttoward(double x, long double y);
float nexttowardf(float x, long double y);
long double nexttowardl(long double x,
                        long double y);
```

The mysteriously named "manipulation functions" are all new in C99. They provide access to the low-level details of floating-point numbers.

copysign The `copysign` function copies the sign of one number to another number. The call `copysign(x, y)` returns a value with the magnitude of `x` and the sign of `y`.

nan The `nan` function converts a string to a NaN value. The call `nan("`*n-char-sequence*`")` is equivalent to `strtod("NAN(`*n-char-sequence*`)", (char**) NULL)`. (See the discussion of `strtod` for a description of the format of *n-char-sequence*.) The call `nan("")` is equivalent to `strtod("NAN()", (char**) NULL)`. If the argument in a call of `nan` doesn't have the value "*n-char-sequence*" or "", the call is equivalent to `strtod("NAN", (char**) NULL)`. If quiet NaNs aren't supported, `nan` returns zero. Calls of `nanf` and `nanl` are equivalent to calls of `strtof` and `strtold`, respectively. This function is used to construct a NaN value containing a specific binary pattern. (Recall from earlier in this section that the fraction part of a NaN value is arbitrary.)

strtod function ➤28.2

nextafter The `nextafter` function determines the next representable value of a number `x` (if all values of `x`'s type were listed in order, the number that would come just before or just after `x`). The value of `y` determines the direction: if `y` < `x`, then the function returns the value just before `x`; if `x` < `y`, it returns the value just after `x`. If `x` and `y` are equal, `nextafter` returns `y`.

Q&A

nexttoward The `nexttoward` function is the same as `nextafter`, except that the `y` parameter has type `long double` instead of `double`. If `x` and `y` are equal, `nexttoward` returns `y` converted to the function's return type. The advantage of `nexttoward` is that a value of any (real) floating type can be passed as the second argument without the danger of it being incorrectly converted to a narrower type.

Maximum, Minimum, and Positive Difference Functions

```
double fdim(double x, double y);
float fdimf(float x, float y);
long double fdiml(long double x, long double y);

double fmax(double x, double y);
float fmaxf(float x, float y);
long double fmaxl(long double x, long double y);

double fmin(double x, double y);
float fminf(float x, float y);
long double fminl(long double x, long double y);
```