

result. In addition, `ERANGE` is stored in `errno`. The C99 standard describes a more complicated set of actions when overflow occurs:

- If default rounding is in effect or if the return value is an “exact infinity” (such as `log(0.0)`), then the function returns either `HUGE_VAL`, `HUGE_VALF`, or `HUGE_VALL`, depending on the function’s return type. (`HUGE_VALF` and `HUGE_VALL`—the float and long double versions of `HUGE_VAL`—are new in C99. Like `HUGE_VAL`, they may represent positive infinity.) The value returned has the sign of the correct result.
- If the value of `math_errhandling & MATH_ERRNO` is nonzero, `ERANGE` is stored in `errno`.
- If the value of `math_errhandling & MATH_ERREXCEPT` is nonzero, the *divide-by-zero* floating-point exception is raised if the mathematical result is an exact infinity. Otherwise, the *overflow* exception is raised.

Underflow. If the magnitude is too small to represent, the C89 standard requires the function to return zero; some implementations may also store `ERANGE` in `errno`. The C99 standard prescribes a somewhat different set of actions:

- The function returns a value whose magnitude is less than or equal to the smallest normalized positive number belonging to the function’s return type. (This value might be zero or a subnormal number.)
- If the value of `math_errhandling & MATH_ERRNO` is nonzero, an implementation may store `ERANGE` in `errno`.
- If the value of `math_errhandling & MATH_ERREXCEPT` is nonzero, an implementation may raise the *underflow* floating-point exception.

Notice the word “may” in the latter two cases. For reasons of efficiency, an implementation is not required to modify `errno` or raise the *underflow* exception.

Functions

We’re now ready to tackle the functions that C99 adds to `<math.h>`. I’ll present the functions in groups, using the same categories as the C99 standard. These categories differ somewhat from the ones in Section 23.3, which came from the C89 standard.

One of the biggest changes in the C99 version of `<math.h>` is the addition of two more versions of most functions. In C89, there’s only a single version of each math function; typically, it takes at least one argument of type `double` and/or returns a `double` value. In C99, however, there are two additional versions: one for `float` and one for `long double`. The names of these functions are identical to the name of the original function except for the addition of an `f` or `l` suffix. For example, the original `sqrt` function, which takes the square root of a `double` value, is now joined by `sqrtf` (the `float` version) and `sqrtl` (the `long double` version). I’ll list the prototypes for the new versions (in *italics*, as is my custom for functions that are new in C99). I won’t describe the functions further, though, since they’re virtually identical to their C89 counterparts.