Table 23.8 Relationship between FLT_EVAL_METHOD and the float_t and double_t Types

Value of FLT_EVAL_METHOD	*Meaning of* float_t	*Meaning of* double_t
0	float	double
1	double	double
2	long double	long double
Other	Implementation-defined	Implementation-defined

Macros

C99 adds a number of macros to <math.h>. I'll mention just two of them at this point. INFINITY represents the float version of positive or unsigned infinity. (If the implementation doesn't support infinity, then INFINITY represents a float value that overflows at compile time.) The NAN macro represents the float version of "not a number." More specifically, it represents a "quiet" NaN (one that doesn't raise an exception if used in an arithmetic expression). If quiet NaNs aren't supported, the NAN macro won't be defined.

I'll cover the function-like macros in <math.h> later in the section, along with ordinary functions. Macros that are relevant only to a specific function will be described with the function itself.

Errors

For the most part, the C99 version of <math.h> deals with errors in the same way as the C89 version. However, there are a few twists that we'll need to discuss.

First, C99 provides several macros that give implementations a choice of how errors are signaled: via a value stored in errno, via a floating-point exception, or both. The macros MATH_ERRNO and MATH_ERREXCEPT represent the integer constants 1 and 2, respectively. A third macro, math_errhandling, represents an int expression whose value is either MATH_ERRNO, MATH_ERREXCEPT, or the bitwise OR of the two values. (It's also possible that math_errhandling isn't really a macro; it might be an identifier with external linkage.) The value of math_errhandling can't be changed within a program.

Now, let's see what happens when a domain error occurs during a call of one of the functions in <math.h>. The C89 standard says that EDOM is stored in errno. The C99 standard, on the other hand, states that if the expression math_errhandling & MATH_ERRNO is nonzero (i.e., the MATH_ERRNO bit is set), then EDOM is stored in errno. If the expression math_errhandling & MATH_ERREXCEPT is nonzero, the *invalid* floating-point exception is raised. Thus, either or both actions are possible, depending on the value of math_errhandling.

Finally, let's turn to the actions that take place when a range error is detected during a function call. There are two cases, based on the magnitude of the function's return value.

Overflow. If the magnitude is too large, the C89 standard requires the function to return positive or negative HUGE_VAL, depending on the sign of the correct