Going a step further, we might use the macros in <limits.h> to help a program *choose* how to represent a type. Let's say that variables of type Quantity must be able to hold integers as large as 100,000. If INT_MAX is at least 100,000, we can define Quantity to be int; otherwise, we'll need to make it long int:

```
#if INT_MAX >= 100000
typedef int Quantity;
#else
typedef long int Quantity;
#endif
```

23.3 The <math.h> Header (C89): Mathematics

The functions in the C89 version of <math.h> fall into five groups:

Trigonometric functions
Hyperbolic functions
Exponential and logarithmic functions
Power functions
Nearest integer, absolute value, and remainder functions

C99 adds a number of functions to these groups as well as introducing other categories of math functions. The C99 changes to <math.h> are so extensive that I've chosen to cover them in a separate section that follows this one. That way, readers who are primarily interested in the C89 version of the header—or who are using a compiler that doesn't support C99—won't be overwhelmed by all the C99 additions.

Before we delve into the functions provided by <math.h>, let's take a brief look at how these functions deal with errors.

Errors

The <math.h> functions handle errors in a way that's different from other library functions. When an error occurs, most <math.h> functions store an error code in a special variable named errno (declared in the <errno.h> header). In addition, when the return value of a function would be larger than the largest double value, the functions in <math.h> return a special value, represented by the macro HUGE_VAL (defined in <math.h>). HUGE_VAL is of type double, but it isn't necessarily an ordinary number. (The IEEE standard for floating-point arithmetic defines a value named "infinity"—a logical choice for HUGE_VAL.)

<errno.h> header ➤24.2

infinity ➤23.4

The functions in <math.h> detect two kinds of errors:

- *Domain error:* An argument is outside a function's domain. If a domain error occurs, the function's return value is implementation-defined and EDOM