As an example, assume that the following variables have been declared:

```
int i;
float f;
double d;
long double ld;
float complex fc;
double complex dc;
long double complex ldc;
```

For each macro invocation in the left column below, the corresponding function call appears in the right column:

Macro Invocation	*Equivalent Function Call*
`sqrt(i)`	`sqrt(i)`
`sqrt(f)`	`sqrtf(f)`
`sqrt(d)`	`sqrt(d)`
`sqrt(ld)`	`sqrtl(ld)`
`sqrt(fc)`	`csqrtf(fc)`
`sqrt(dc)`	`csqrt(dc)`
`sqrt(ldc)`	`csqrtl(ldc)`

Note that writing `sqrt(i)` causes the `double` version of `sqrt` to be called, not the `float` version.

These rules also cover macros with more than one parameter. For example, the macro invocation `pow(ld, f)` will be replaced by the call `powl(ld, f)`. Both of `pow`'s parameters are generic; because one of the arguments has type `long double`, rule 1 states that the `long double` version of `pow` will be called.

27.6 The `<fenv.h>` Header (C99): Floating-Point Environment

IEEE Standard 754 is the most widely used representation for floating-point numbers. (This standard is also known as IEC 60559, which is how the C99 standard refers to it.) The purpose of the `<fenv.h>` header is to give programs access to the floating-point status flags and control modes specified in the IEEE standard. Although `<fenv.h>` was designed in a general fashion that allows it to work with other floating-point representations, supporting the IEEE standard was the reason for the header's creation.

A discussion of why programs might need access to status flags and control modes is beyond the scope of this book. For good examples, see "What every computer scientist should know about floating-point arithmetic" by David Goldberg (*ACM Computing Surveys*, vol. 23, no. 1 (March 1991): 5–48), which can be found on the Web.