```
#define CHECK_ZERO(divisor) \
  if (divisor == 0) \
    printf("*** Attempt to divide by zero on line %d " \
           "of file %s ***\n", __LINE__, __FILE__)
```

The CHECK_ZERO macro would be invoked prior to a division:

```
CHECK_ZERO(j);
k = i / j;
```

If `j` happens to be zero, a message of the following form will be printed:

```
*** Attempt to divide by zero on line 9 of file foo.c ***
```

Error-detecting macros like this one are quite useful. In fact, the C library has a general-purpose error-detecting macro named `assert`.

assert macro ➤24.1

The `__STDC__` macro exists and has the value 1 if the compiler conforms to the C standard (either C89 or C99). By having the preprocessor test this macro, a program can adapt to a compiler that predates the C89 standard (see Section 14.4 for an example).

C99 Additional Predefined Macros in C99

C99 provides a few additional predefined macros (Table 14.2).

Table 14.2 Additional Predefined Macros in C99

Name	*Description*
`__STDC__HOSTED__`	1 if this is a hosted implementation; 0 if it is freestanding
`__STDC__VERSION__`	Version of C standard supported
`__STDC_IEC_559__` †	1 if IEC 60559 floating-point arithmetic is supported
`__STDC_IEC_559_COMPLEX__` †	1 if IEC 60559 complex arithmetic is supported
`__STDC_ISO_10646__` †	*yyyymm*L if `wchar_t` values match the ISO 10646 standard of the specified year and month

†Conditionally defined

To understand the meaning of `__STDC__HOSTED__`, we need some new vocabulary. An ***implementation*** of C consists of the compiler plus other software necessary to execute C programs. C99 divides implementations into two categories: hosted and freestanding. A ***hosted implementation*** must accept any program that conforms to the C99 standard, whereas a ***freestanding implementation*** doesn't have to compile programs that use complex types or standard headers beyond a few of the most basic. (In particular, a freestanding implementation doesn't have to support the `<stdio.h>` header.) The `__STDC__HOSTED__` macro represents the constant 1 if the compiler is a hosted implementation; otherwise, the macro has the value 0.

complex types ➤27.3

Q&A

The `__STDC__VERSION__` macro provides a way to check which version of the C standard is recognized by the compiler. This macro first appeared in Amendment 1 to the C89 standard, where its value was specified to be the long