but not statements. The solution is to wrap the statements in a do loop whose condition is false (and which therefore will be executed just once):

```
do { ... } while (0)
```

Notice that the do statement isn't complete—it needs a semicolon at the end. To see this trick (ahem, technique) in action, let's incorporate it into our ECHO macro:

```
#define ECHO(s)      \
        do {         \
          gets(s);   \
          puts(s);   \
        } while (0)
```

When ECHO is used, it must be followed by a semicolon, which completes the do statement:

```
ECHO(str);
  /* becomes do { gets(str); puts(str); } while (0); */
```

Predefined Macros

C has several predefined macros. Each macro represents an integer constant or string literal. As Table 14.1 shows, these macros provide information about the current compilation or about the compiler itself.

Table 14.1
Predefined Macros

Name	*Description*
`__LINE__`	Line number of file being compiled
`__FILE__`	Name of file being compiled
`__DATE__`	Date of compilation (in the form `"Mmm dd yyyy"`)
`__TIME__`	Time of compilation (in the form `"hh:mm:ss"`)
`__STDC__`	1 if the compiler conforms to the C standard (C89 or C99)

The `__DATE__` and `__TIME__` macros identify when a program was compiled. For example, suppose that a program begins with the following statements:

```
printf("Wacky Windows (c) 2010 Wacky Software, Inc.\n");
printf("Compiled on %s at %s\n", __DATE__, __TIME__);
```

Each time it begins to execute, the program will print two lines of the form

```
Wacky Windows (c) 2010 Wacky Software, Inc.
Compiled on Dec 23 2010 at 22:18:48
```

This information can be helpful for distinguishing among different versions of the same program.

We can use the `__LINE__` and `__FILE__` macros to help locate errors. Consider the problem of detecting the location of a division by zero. When a C program terminates prematurely because it divided by zero, there's usually no indication of which division caused the problem. The following macro can help us pinpoint the source of the error: