

programs, examining the output of the preprocessor may prove useful for locating this kind of error.

14.2 Preprocessing Directives

Most preprocessing directives fall into one of three categories:

- *Macro definition.* The `#define` directive defines a macro; the `#undef` directive removes a macro definition.
- *File inclusion.* The `#include` directive causes the contents of a specified file to be included in a program.
- *Conditional compilation.* The `#if`, `#ifdef`, `#ifndef`, `#elif`, `#else`, and `#endif` directives allow blocks of text to be either included in or excluded from a program, depending on conditions that can be tested by the preprocessor.

The remaining directives—`#error`, `#line`, and `#pragma`—are more specialized and therefore used less often. We'll devote the rest of this chapter to an in-depth examination of preprocessing directives. The only directive we won't discuss in detail is `#include`, since it's covered in Section 15.2.

Before we go further, let's look at a few rules that apply to all directives:

- *Directives always begin with the # symbol.* The `#` symbol need not be at the beginning of a line, as long as only white space precedes it. After the `#` comes the name of the directive, followed by any other information the directive requires.
- *Any number of spaces and horizontal tab characters may separate the tokens in a directive.* For example, the following directive is legal:

```
#      define      N      100
```

- *Directives always end at the first new-line character, unless explicitly continued.* To continue a directive to the next line, we must end the current line with a `\` character. For example, the following directive defines a macro that represents the capacity of a hard disk, measured in bytes:

```
#define DISK_CAPACITY (SIDES *          \
                        TRACKS_PER_SIDE * \
                        SECTORS_PER_TRACK * \
                        BYTES_PER_SECTOR)
```

- *Directives can appear anywhere in a program.* Although we usually put `#define` and `#include` directives at the beginning of a file, other directives are more likely to show up later, even in the middle of function definitions.
- *Comments may appear on the same line as a directive.* In fact, it's good practice to put a comment at the end of a macro definition to explain the meaning of the macro:

```
#define FREEZING_PT 32.0f /* freezing point of water */
```