check if the `-Wparentheses` option is used or if `-Wall` (all warnings) is selected. GCC allows the programmer to suppress the warning in a particular case by enclosing the `if` condition in a second set of parentheses:

```
if ((i = j)) ...
```

Q: C books seem to use several different styles of indentation and brace placement for compound statements. Which style is best?

A: According to *The New Hacker's Dictionary* (Cambridge, Mass.: MIT Press, 1996), there are four common styles of indentation and brace placement:

- The *K&R style*, used in Kernighan and Ritchie's *The C Programming Language*, is the one I've chosen for the programs in this book. In the K&R style, the left brace appears at the end of a line:

  ```
  if (line_num == MAX_LINES) {
    line_num = 0;
    page_num++;
  }
  ```

  The K&R style keeps programs compact by not putting the left brace on a line by itself. A disadvantage: the left brace can be hard to find. (I don't consider this a problem, since the indentation of the inner statements makes it clear where the left brace should be.) The K&R style is the one most often used in Java, by the way.

- The *Allman style*, named after Eric Allman (the author of `sendmail` and other UNIX utilities), puts the left brace on a separate line:

  ```
  if (line_num == MAX_LINES)
  {
    line_num = 0;
    page_num++;
  }
  ```

  This style makes it easy to check that braces come in matching pairs.

- The *Whitesmiths style*, popularized by the Whitesmiths C compiler, dictates that braces be indented:

  ```
  if (line_num == MAX_LINES)
    {
    line_num = 0;
    page_num++;
    }
  ```

- The *GNU style*, used in software developed by the GNU Project, indents the braces, then further indents the inner statements:

  ```
  if (line_num == MAX_LINES)
    {
      line_num = 0;
      page_num++;
    }
  ```