strcmp(str1, str2) < 0

is true if str1 precedes str2; and the expression

strcmp(str1, str2) > 0

is true if str1 follows str2. Thus, the strcmp() function can play the role of the ==, !=, <, and > operators, depending upon how you set up a test condition.

Next, compstr.cpp uses the increment operator to march the variable ch through the alphabet:

ch++

You can use the increment and decrement operators with character variables, because type char really is an integer type, so the operation actually changes the integer code stored in the variable. Also, note that using an array index makes it simple to change individual characters in a string:

word[0] = ch;

Finally, unlike most of the for loops to date, this loop isn't a counting loop. That is, it doesn't execute a block of statements a specified number of times. Instead, the loop watches for a particular circumstance (word being "mate") to signal that it's time to stop. More typically, C++ programs use while loops for this second kind of test, so let's examine that form now.

**The while Loop**

The while loop is a for loop stripped of the initialization and update parts; it has just a test condition and a body:

while (test-condition)  
  body

First, a program evaluates the test-condition expression. If the expression evaluates to