Trigraphs can be freely substituted for their ASCII equivalents. For example, the program

```
#include <stdio.h>

int main(void)
{
  printf("hello, world\n");
  return 0;
}
```

could be written

```
??=include <stdio.h>

int main(void)
??<
  printf("hello, world??/n");
  return 0;
??>
```

Compilers that conform to the C89 or C99 standards are required to accept trigraphs, even though they're rarely used. Occasionally, this feature can cause problems.

⚠ Be careful about putting ?? in a string literal—it's possible that the compiler will treat it as the beginning of a trigraph. If this should happen, turn the second ? character into an escape sequence by preceding it with a \ character. The resulting ?\? combination can't be mistaken for the beginning of a trigraph.

C99 Digraphs

Acknowledging that trigraphs are difficult to read, Amendment 1 to the C89 standard added an alternative notation known as ***digraphs.*** As the name implies, a digraph requires just two characters instead of three. Digraphs are available as substitutes for the six tokens shown in Table 25.9. tokens ➤2.8

Table 25.9
Digraphs

Digraph	*Token*
<:	[
:>	]
<%	{
%>	}
%:	#
%:%:	##

Digraphs—unlike trigraphs—are *token* substitutes, not *character* substitutes. Thus, digraphs won't be recognized inside a string literal or character constant. For example, the string "<::>" has length four; it contains the characters: <, :, :,