Sequence	*Shortcut for or Equivalent to*
\t	Tab
\v	Vertical tab (moves the cursor down a line)
\\	The backslash character
\'	The apostrophe
\"	The double-quote character
\?	The question mark
\0	The "null" byte (that's 0, not the letter *O*)
\O*nn*	A character value in octal (base 8)
\x*nnn*	A character value in hexadecimal (base 16)

The final two items in Table 24-1 are the most flexible. They allow you to insert any character into any text after you know that character's octal or hexadecimal code value.

Suppose that you need to display the Escape character in your text. A quick look into Appendix B shows that the hexadecimal code for Escape is `1b`. As an escape sequence — an Escape escape sequence — that would be written as

```
\x1b
```

Here's how it would look in `printf()`:

```
printf("On some consoles, this clears the screen \x1b[2J");
```

You may want to flag this page with a sticky note or dog-ear the corner. The page has stuff no one remembers, so you wind up referring to Table 24-1 often.

The `printf()` escape-sequence testing program deluxe

To see how some of these characters work, create the PRINTFUN.C program, listed next. You modify the `printf()` statement at the core of the program to demonstrate how the various escape sequences affect text: