By default, **sort** arranges lines of text alphabetically. Many options control the sorting, and Table 5-2 lists some of them.

Table 5-2. Some sort options

Option	Description
-n	Sort numerically (example: 10 sorts after 2), ignore blanks and tabs.
-r	Reverse the sorting order.
-f	Sort upper- and lowercase together.
+*x*	Ignore first *x* fields when sorting.

More than two commands may be linked up into a pipe. Taking a previous pipe example using **grep**, we can further sort the files modified in August by order of size. The following pipe uses the commands **ls**, **grep**, and **sort**:

```
$ ls -l | grep "Aug" | sort +4n
-rw-rw-r--  1 carol doc      1605 Aug 23 07:35 macros
-rw-rw-r--  1 john  doc      2488 Aug 15 10:51 intro
-rw-rw-rw-  1 john  doc      8515 Aug  6 15:30 ch07
-rw-rw-rw-  1 john  doc     11008 Aug  6 14:10 ch02
$
```

This pipe sorts all files in your directory modified in August by order of size, and prints them to the terminal screen. The **sort** option **+4n** skips four fields (fields are separated by blanks), then sorts the lines in numeric order. So, the output of **ls**, filtered by **grep**, is sorted by the file size (this is the fifth column, starting with 1605). Both **grep** and **sort** are used here as filters to modify the output of the **ls -l** command. If you wanted to email this listing to someone, you could add a final pipe to the **mail** program. Or you could print the listing by piping the **sort** output to your printer command (either **lp** or **lpr**).

Piping to a Pager

The **less** program, which you saw in the section "Looking Inside Files with less" in Chapter 3, can also be used as a filter. A long output normally zips by you on the screen, but if you run text through **less**, the display stops after each screenful of text.

Let's assume that you have a long directory listing. (If you want to try this example and need a directory with lots of files, use **cd** first to change to a