matches any string containing only the letters a, b, and c, while %[^abc] matches any string that doesn't contain a, b, or c.

Many of the ...scanf conversion specifiers are closely related to the numeric conversion functions in <stdlib.h>. These functions convert strings (like "-297") to their equivalent numeric values (–297). The d specifier, for example, looks for an optional + or - sign, followed by a series of decimal digits; this is exactly the same form that the strtol function requires when asked to convert a string to a decimal number. Table 22.13 shows the correspondence between conversion specifiers and numeric conversion functions.

numeric conversion functions ➤ 26.2

Table 22.13 Correspondence between ...scanf Conversion Specifiers and Numeric Conversion Functions

Conversion Specifier	*Numeric Conversion Function*
d	strtol with 10 as the base
i	strtol with 0 as the base
o	strtoul with 8 as the base
u	strtoul with 10 as the base
x, X	strtoul with 16 as the base
a, A, e, E, f, F, g, G	strtod

It pays to be careful when writing calls of scanf. An invalid conversion specification in a scanf format string is just as bad as one in a printf format string; either one causes undefined behavior.

C99 C99 Changes to ...scanf Conversion Specifications

The conversion specifications for scanf and fscanf have undergone some changes in C99, but the list isn't as extensive as it was for the ...printf functions:

- *Additional length modifiers.* C99 adds the hh, ll, j, z, and t length modifiers. These correspond to the length modifiers in ...printf conversion specifications.
- *Additional conversion specifiers.* C99 adds the F, a, and A conversion specifiers. They're provided for symmetry with ...printf; the ...scanf functions treat them the same as e, E, f, g, and G.
- *Ability to read infinity and NaN.* Just as the ...printf functions can write infinity and NaN, the ...scanf functions can read these values. To be read properly, they should have the same appearance as values written by the ...printf functions, with case being ignored. (For example, either INF or inf will be read as infinity.)
- *Support for wide characters.* The ...scanf functions are able to read multibyte characters, which are then converted to wide characters for storage. The %lc conversion specification is used to read a single multibyte character or a