Table 25.6 Typical Values of Monetary `lconv` Members for U.S.A. and Finland

Member	*U.S.A.*	*Finland*
`mon_decimal_point`	`"."`	`","`
`mon_thousands_sep`	`","`	`" "`
`mon_grouping`	`"\3"`	`"\3"`
`positive_sign`	`""`	`""`
`negative_sign`	`"-"`	`"-"`
`currency_symbol`	`"$"`	`"EUR"`
`frac_digits`	`2`	`2`
`p_cs_precedes`	`1`	`0`
`n_cs_precedes`	`1`	`0`
`p_sep_by_space`	`0`	`2`
`n_sep_by_space`	`0`	`2`
`p_sign_posn`	`1`	`1`
`n_sign_posn`	`1`	`1`
`int_curr_symbol`	`"USD "`	`"EUR "`
`int_frac_digits`	`2`	`2`
`int_p_cs_precedes`	`1`	`0`
`int_n_cs_precedes`	`1`	`0`
`int_p_sep_by_space`	`1`	`2`
`int_n_sep_by_space`	`1`	`2`
`int_p_sign_posn`	`1`	`1`
`int_n_sign_posn`	`1`	`1`

Here's how the monetary quantity 7593.86 would be formatted in the two locales, depending on the sign of the quantity and whether the formatting is local or international:

	U.S.A.	*Finland*
Local format (positive)	`$7,593.86`	`7 593,86 EUR`
Local format (negative)	`-$7,593.86`	`- 7 593,86 EUR`
International format (positive)	`USD 7,593.86`	`7 593,86 EUR`
International format (negative)	`-USD 7,593.86`	`- 7 593,86 EUR`

Keep in mind that none of C's library functions are able to format monetary quantities automatically. It's up to the programmer to use the information in the `lconv` structure to accomplish the formatting.

25.2 Multibyte Characters and Wide Characters

Latin-1 ➤7.3

One of the biggest problems in adapting programs to different locales is the character-set issue. ASCII and its extensions, which include Latin-1, are the most popular character sets in North America. Elsewhere, the situation is more complicated. In many countries, computers employ character sets that are similar to ASCII, but lack certain characters; we'll discuss this issue further in Section 25.3. Other countries, especially those in Asia, face a different problem: written languages that require a very large character set, usually numbering in the thousands.