There are two ways to write a universal character name (\u*dddd* and \U*dddddddd*), where each *d* is a hexadecimal digit. In the form \U*dddddddd*, the *d*'s form an eight-digit hexadecimal number that identifies the UCS code point of the desired character. The form \u*dddd* can be used for characters whose code points have hexadecimal values of FFFF or less, which includes all characters in the Basic Multilingual Plane.

For example, the UCS code point for the Greek letter β is 000003B2, so the universal character name for this character is \U000003B2 (or \U000003b2, since the case of hexadecimal digits doesn't matter). Because the first four hexadecimal digits of the UCS code point are 0, we can also use the \u notation, writing the character as \u03B2 or \u03b2. The code point values for UCS (which match those for Unicode) can be found at *www.unicode.org/charts/*.

Not all universal character names may be used in identifiers; the C99 standard contains a list of which ones are allowed. Also, an identifier may not begin with a universal character name that represents a digit.

25.5 The <wchar.h> Header (C99) Extended Multibyte and Wide-Character Utilities

The <wchar.h> header provides functions for wide-character input/output and wide-string manipulation. The vast majority of functions in <wchar.h> are wide-character versions of functions from other headers (primarily <stdio.h> and <string.h>).

The <wchar.h> header declares several types and macros, including the following:

- mbstate_t — A value of this type can be used to store the conversion state when a sequence of multibyte characters is converted to a sequence of wide characters or vice versa.
- wint_t — An integer type whose values represent extended characters.
- WEOF — A macro representing a wint_t value that's different from any extended character. WEOF is used in much the same way as EOF, typically to indicate an error or end-of-file condition.

EOF macro ➤*22.2*

Note that <wchar.h> provides functions for wide characters but not multibyte characters. That's because C's ordinary library functions are capable of dealing with multibyte characters, so no special functions are needed. For example, the fprintf function allows its format string to contain multibyte characters.

Most wide-character functions behave the same as a function that belongs to another part of the standard library. Usually, the only changes involve arguments and return values of type wchar_t instead of char (or wchar_t * instead of char *). In addition, arguments and return values that represent character counts are measured in wide characters rather than bytes. In the remainder of this section, I'll indicate which other library function (if any) corresponds to each