Table 25.19
Wide-Character Classification Functions

Function	*Test*
`iswalnum(wc)`	Is wc alphanumeric?
`iswalpha(wc)`	Is wc alphabetic?
`iswblank(wc)`	Is wc a blank?†
`iswcntrl(wc)`	Is wc a control character?
`iswdigit(wc)`	Is wc a decimal digit?
`iswgraph(wc)`	Is wc a printing character (other than a space)?
`iswlower(wc)`	Is wc a lower-case letter?
`iswprint(wc)`	Is wc a printing character (including a space)?
`iswpunct(wc)`	Is wc punctuation?
`iswspace(wc)`	Is wc a white-space character?
`iswupper(wc)`	Is wc an upper-case letter?
`iswxdigit(wc)`	Is wc a hexadecimal digit?

†The standard blank wide characters are space (`L' '`) and horizontal tab (`L'\t'`).

leaving open the possibility that more than one wide character is considered to be a "space." See Appendix D for more detailed descriptions of these functions.

In most cases, the wide-character classification functions are consistent with the corresponding functions in `<ctype.h>`: if a `<ctype.h>` function returns a nonzero value (indicating "true") for a particular character, then the corresponding `<wctype.h>` function will return true for the wide version of the same character. The only exception involves white-space wide characters (other than space) that are also printing characters, which may be classified differently by `iswgraph` and `iswpunct` than by `isgraph` and `ispunct`. For example, a character for which `isgraph` returns true may cause `iswgraph` to return false.

Extensible Wide-Character Classification Functions

```
int iswctype(wint_t wc, wctype_t desc);
wctype_t wctype(const char *property);
```

Each of the wide-character classification functions just discussed is able to test a single fixed condition. The `wctype` and `iswctype` functions—which are designed to be used together—make it possible to test for other conditions as well.

wctype

The `wctype` function is passed a string describing a class of wide characters: it returns a `wctype_t` value that represents this class. For example, the call

```
wctype("upper")
```

returns a `wctype_t` value representing the class of upper-case letters. The C99 standard requires that the following strings be allowed as arguments to `wctype`:

```
"alnum"  "alpha"  "blank"  "cntrl"  "digit"  "graph"
"lower"  "print"  "punct"  "space"  "upper"  "xdigit"
```

Additional strings may be provided by an implementation. Which strings are legal arguments to `wctype` at a given time depends on the `LC_CTYPE` category of the