Each character-classification function returns a nonzero value if its argument has a particular property. Table 23.10 lists the property that each function tests.

Table 23.10
Character-Classification Functions

Function	*Test*
`isalnum(c)`	Is c alphanumeric?
`isalpha(c)`	Is c alphabetic?
`isblank(c)`	Is c a blank?[†]
`iscntrl(c)`	Is c a control character?[††]
`isdigit(c)`	Is c a decimal digit?
`isgraph(c)`	Is c a printing character (other than a space)?
`islower(c)`	Is c a lower-case letter?
`isprint(c)`	Is c a printing character (including a space)?
`ispunct(c)`	Is c punctuation?[†††]
`isspace(c)`	Is c a white-space character?[††††]
`isupper(c)`	Is c an upper-case letter?
`isxdigit(c)`	Is c a hexadecimal digit?

[†] The standard blank characters are space and horizontal tab (`\t`). This function is new in C99.

[††] In ASCII, the control characters are `\x00` through `\x1f` plus `\x7f`.

[†††] All printing characters except those for which `isspace` or `isalnum` are true are considered punctuation.

[††††] The white-space characters are space, form feed (`\f`), new-line (`\n`), carriage return (`\r`), horizontal tab (`\t`), and vertical tab (`\v`).

C99

The C99 definition of `ispunct` is slightly different than the one in C89. In C89, `ispunct(c)` tests whether `c` is a printing character but not a space or a character for which `isalnum(c)` is true. In C99, `ispunct(c)` tests whether `c` is a printing character for which neither `isspace(c)` nor `isalnum(c)` is true.

PROGRAM Testing the Character-Classification Functions

The following program demonstrates the character-classification functions (with the exception of `isblank`, which is new in C99) by applying them to the characters in the string `"azAZ0 !\t"`.

tclassify.c

```
/* Tests the character-classification functions */

#include <ctype.h>
#include <stdio.h>

#define TEST(f) printf("  %c  ", f(*p) ? 'x' : ' ')

int main(void)
{
  char *p;

  printf("     alnum     cntrl     graph     print"
         "     space    xdigit\n"
         "          alpha     digit     lower     punct"
         "     upper\n");
```