

Character Library Functions

Except where noted, these functions are declared in `<ctype.h>` (for C) or `<cctype>` (for C++). The following test the supplied character and return 1 for true, 0 for false. In Standard C++, the returned value can be immediately converted to the corresponding bool type value.

int isalpha(char c)
returns 1 if c is in a..z or A..Z

int islower(char c)
returns 1 if c is in a..z

int isupper(char c)
returns 1 if c is in A..Z

int isdigit(char c)
returns 1 if c is in 0..9

int isxdigit(char c)
returns 1 if c is in 0..9, a..f, or A..F

int isalnum(char c)
returns 1 if c is in 0..9, a..z, or A..Z

int isgraph(char c)
returns 1 if c is a graphic character
a graphic character is any printable character other than space

int isprint(char c)
returns 1 if c is a printable character
the standard printable characters are 0x20..0xff except for 0x7f

int ispunct(char c)
returns 1 if c is a punctuation character:
~ ! @ # \$ % ^ & * () - + { } [] , < . > / ? ; : ' " \ | = _ `

int isspace(char c)
returns 1 if c is a white space character
the whitespace characters are: '\t' '\r' '\n' '\v' '\f' ' '

int iscntrl(char c)
returns 1 if c is a control character
the standard control characters are 0x0..0x1f and 0x7f

Relations between these:

isalpha(c) is true if (islower(c) || isupper(c)) is true

isalpha(c) is false if (iscntrl(c) || isdigit(c) || ispunct(c) || isspace(c)) is true

isalnum(c) is true if (isalpha(c) || isdigit(c)) is true

isprint(c) is true if iscntrl(c) is false, and vice versa

isprint(c) and isgraph(c) are the same if c is not a space character

The following transform the supplied character and return another character as an int. In Standard C++, the returned value can be immediately converted to the corresponding char type value.

int tolower(char c)

if c is an upper case alphabetic, then returns the corresponding lower case alphabetic, else returns c itself

int toupper(char c)

if c is a lower case alphabetic, then returns the corresponding upper case alphabetic, else returns c itself

Some Useful Character Input Functions

int fgetc(FILE * stream) /* declared in stdio.h */

int getc(FILE * stream)

int getchar(void)

These functions all read and return the next character from the input stream, which in the case of getchar is stdin. If an error or end-of-file occurs, EOF is returned; use `feof` to determine if it is really an EOF condition.

int ungetc(int c, FILE * stream)

This function pushes the supplied character back into the input stream at the beginning, so that it will be read by the next call to `fgetc`, `getc`, or `getchar`. If successful, c is returned, EOF if not.

int scanf("%c", &char_var) /* scanf is declared in stdio.h */

int scanf(" %c", &char_var)

These calls to `scanf` will read a single character into the char variable whose address is supplied. The first will read and store the next character even if it is whitespace. The second, with the leading whitespace in the format string, will skip leading whitespace and then read and store the first non-whitespace character. See elsewhere for the returned value for `scanf`.