

23.5 The `<ctype.h>` Header: Character Handling

The `<ctype.h>` header provides two kinds of functions: character-classification functions (like `isdigit`, which tests whether a character is a digit) and character case-mapping functions (like `toupper`, which converts a lower-case letter to upper case).

Although C doesn't require that we use the functions in `<ctype.h>` to test characters and perform case conversions, it's a good idea to do so. First, these functions have been optimized for speed (in fact, many are implemented as macros). Second, we'll end up with a more portable program, since these functions work with any character set. Third, the `<ctype.h>` functions adjust their behavior when the locale is changed, which helps us write programs that run properly in different parts of the world.

locales ► 25.1

The functions in `<ctype.h>` all take `int` arguments and return `int` values. In many cases, the argument is already stored in an `int` variable (often as a result of having been read by a call of `fgetc`, `getc`, or `getchar`). If the argument has `char` type, however, we need to be careful. C can automatically convert a `char` argument to `int` type; if `char` is an unsigned type or if we're using a seven-bit character set such as ASCII, the conversion will go smoothly. But if `char` is a signed type and if some characters require eight bits, then converting such a character from `char` to `int` will give a negative result. The behavior of the `<ctype.h>` functions is undefined for negative arguments (other than EOF), potentially causing serious problems. In such a situation, the argument should be cast to `unsigned char` for safety. (For maximum portability, some programmers always cast a `char` value to `unsigned char` when passing it to a `<ctype.h>` function.)

Character-Classification Functions

```
int isalnum(int c);
int isalpha(int c);
int isblank(int c);
int iscntrl(int c);
int isdigit(int c);
int isgraph(int c);
int islower(int c);
int isprint(int c);
int ispunct(int c);
int isspace(int c);
int isupper(int c);
int isxdigit(int c);
```