**ANSI C library header files**

|  |  |
| --- | --- |
| [**<assert.h>**](https://en.wikipedia.org/wiki/Assert.h) | Contains the assert macro, used to assist with detecting logical errors and other types of bug in debugging versions of a program. |
| [**<complex.h>**](https://en.wikipedia.org/wiki/Complex.h) | A set of functions for manipulating complex numbers. (New with **C99**) |
| [**<ctype.h>**](https://en.wikipedia.org/wiki/Ctype.h) | This header file contains functions used to classify characters by their types or to convert between upper and lower case in a way that is independent of the used character set (typically ASCII or one of its extensions, although implementations utilizing EBCDIC are also known). |
| [**<errno.h>**](https://en.wikipedia.org/wiki/Errno.h) | For testing error codes reported by library functions. |
| [**<fenv.h>**](https://en.wikipedia.org/wiki/Fenv.h) | For controlling floating-point environment. (New with **C99**) |
| [**<float.h>**](https://en.wikipedia.org/wiki/Float.h) | Contains defined constants specifying the implementation-specific properties of the floating-point library, such as the minimum difference between two different floating-point numbers (\_EPSILON), the maximum number of digits of accuracy (\_DIG) and the range of numbers which can be represented (\_MIN, \_MAX). |
| [**<inttypes.h>**](https://en.wikipedia.org/wiki/Inttypes.h) | For precise conversion between integer types. (New with **C99**) |
| [**<iso646.h>**](https://en.wikipedia.org/wiki/Iso646.h) | For programming in ISO 646 variant character sets. (New with **NA1**) |
| [**<limits.h>**](https://en.wikipedia.org/wiki/Limits.h) | Contains defined constants specifying the implementation-specific properties of the integer types, such as the range of numbers which can be represented (\_MIN, \_MAX). |
| [**<locale.h>**](https://en.wikipedia.org/wiki/Locale.h) | For setlocale() and related constants. This is used to choose an appropriate locale. |
| [**<math.h>**](https://en.wikipedia.org/wiki/Math.h) | For computing common mathematical functions  -- see [Further math](https://en.wikibooks.org/wiki/C_Programming/Further_math) or [C++ Programming/Code/Standard C Library/Math](https://en.wikibooks.org/wiki/C%2B%2B_Programming/Code/Standard_C_Library/Math) for details. |
| [**<setjmp.h>**](https://en.wikipedia.org/wiki/Setjmp.h) | setjmp and longjmp, which are used for non-local exits |
| [**<signal.h>**](https://en.wikipedia.org/wiki/Signal.h) | For controlling various exceptional conditions |
| [**<stdarg.h>**](https://en.wikipedia.org/wiki/Stdarg.h) | For accessing a varying number of arguments passed to functions. |
| [**<stdbool.h>**](https://en.wikipedia.org/wiki/Stdbool.h) | For a boolean data type. (New with **C99**) |
| [**<stdint.h>**](https://en.wikipedia.org/wiki/Stdint.h) | For defining various integer types. (New with **C99**) |
| [**<stddef.h>**](https://en.wikipedia.org/wiki/Stddef.h) | For defining several useful types and macros. |
| [**<stdio.h>**](https://en.wikipedia.org/wiki/Stdio.h) | Provides the core input and output capabilities of the C language. This file includes the venerable printf function. |
| [**<stdlib.h>**](https://en.wikipedia.org/wiki/Stdlib.h) | For performing a variety of operations, including conversion, pseudo-random numbers, memory allocation, process control, environment, signalling, searching, and sorting. |
| [**<string.h>**](https://en.wikipedia.org/wiki/String.h) | For manipulating several kinds of strings. |
| [**<tgmath.h>**](https://en.wikipedia.org/wiki/Tgmath.h) | For type-generic mathematical functions. (New with **C99**) |
| [**<time.h>**](https://en.wikipedia.org/wiki/Time.h) | For converting between various time and date formats. |
| [**<wchar.h>**](https://en.wikipedia.org/wiki/Wchar.h) | For manipulating wide streams and several kinds of strings using wide characters - key to supporting a range of languages. (New with **NA1**) |
| [**<wctype.h>**](https://en.wikipedia.org/wiki/Wctype.h) | For classifying wide characters. (New with **NA1**) |

| **Function** | **System Include File** | **Function Prototype** | **Description** |
| --- | --- | --- | --- |
| abort | stdlib.h | void abort(void); | Stops a program abnormally. |
| abs | stdlib.h | int abs(int *n*); | Calculates the absolute value of an integer argument *n*. |
| acos | math.h | double acos(double *x*); | Calculates the arc cosine of *x*. |
| asctime | time.h | char \*asctime(const struct tm \**time*); | Converts the *time* that is stored as a structure to a character string. |
| asctime\_r | time.h | char \*asctime\_r (const struct tm \*tm, char \*buf); | Converts *tm* that is stored as a structure to a character string. (Restartable version of asctime.) |
| asin | math.h | double asin(double *x*); | Calculates the arc sine of *x*. |
| assert | assert.h | void assert(int *expression*); | Prints a diagnostic message and ends the program if the expression is false. |
| atan | math.h | double atan(double *x*); | Calculates the arc tangent of *x*. |
| atan2 | math.h | double atan2(double *y*, double *x*); | Calculates the arc tangent of *y/x*. |
| atexit | stdlib.h | int atexit(void (\**func*)(void)); | Registers a function to be called at normal termination. |
| atof | stdlib.h | double atof(const char \**string*); | Converts *string* to a double-precision floating-point value. |
| atoi | stdlib.h | int atoi(const char \**string*); | Converts *string* to an integer. |
| atol | stdlib.h | long int atol(const char \**string*); | Converts *string* to a long integer. |
| bsearch | stdlib.h | void \*bsearch(const void \**key*, const void \**base*, size\_t *num*, size\_t *size*, int (\**compare*) (const void \**element1*, const void \**element2*)); | Performs a binary search on an array of *num* elements, each of *size* bytes. The array must be sorted in ascending order by the function pointed to by *compare*. |
| btowc | stdio.h wchar.h | wint\_t btowc(int *c*); | Determines whether *c* constitues a valid multibyte character in the initial shift state. |
| calloc | stdlib.h | void \*calloc(size\_t *num*, size\_t *size*); | Reserves storage space for an array of *num* elements, each of size *size*, and initializes the values of all elements to 0. |
| catclose6 | nl\_types.h | int catclose (nl\_catd catd); | Closes a previously opened message catalog. |
| catgets6 | nl\_types.h | char \*catgets(nl\_catd catd, int set\_id, int msg\_id, const char \*s); | Retrieves a message from an open message catalog. |
| catopen6 | nl\_types.h | nl\_catd catopen (const char \*name, int oflag); | Opens a message catalog, which must be done before a message can be retrieved. |
| ceil | math.h | double ceil(double *x*); | Calculates the double value representing the smallest integer that is greater than or equal to *x*. |
| clearerr | stdio.h | void clearerr(FILE \**stream*); | Resets the error indicators and the end-of-file indicator for *stream*. |
| clock | time.h | clock\_t clock(void); | Returns the processor time that has elapsed since the job was started. |
| cos | math.h | double cos(double *x*); | Calculates the cosine of *x*. |
| cosh | math.h | double cosh(double *x*); | Calculates the hyperbolic cosine of *x*. |
| ctime | time.h | char \*ctime(const time\_t \**time*); | Converts *time* to a character string. |
| ctime64 | time.h | char \*ctime64(const time64\_t \**time*); | Converts *time* to a character string. |
| ctime\_r | time.h | char \*ctime\_r(const time\_t \*time, char \*buf); | Converts *time* to a character string. (Restartable version of ctime.) |
| ctime64\_r | time.h | char \*ctime64\_r(const time64\_t \*time, char \*buf); | Converts *time* to a character string. (Restartable version of ctime64.) |
| difftime | time.h | double difftime(time\_t *time2*, time\_t *time1*); | Computes the difference between *time2* and *time1*. |
| difftime64 | time.h | double difftime64(time64\_t *time2*, time64\_t *time1*); | Computes the difference between *time2* and *time1*. |
| div | stdlib.h | div\_t div(int *numerator*, int *denominator*); | Calculates the quotient and remainder of the division of *numerator* by *denominator*. |
| erf | math.h | double erf(double *x*); | Calculates the error function of *x*. |
| erfc | math.h | double erfc(double *x*); | Calculates the error function for large values of *x*. |
| exit | stdlib.h | void exit(int *status*); | Ends a program normally. |
| exp | math.h | double exp(double *x*); | Calculates the exponential function of a floating-point argument *x*. |
| fabs | math.h | double fabs(double *x*); | Calculates the absolute value of a floating-point argument *x*. |
| fclose | stdio.h | int fclose(FILE \**stream*); | Closes the specified *stream*. |
| fdopen5 | stdio.h | FILE \*fdopen(int *handle*, const char *\*type*); | Associates an input or output stream with the file identified by handle. |
| feof | stdio.h | int feof(FILE \**stream*); | Tests whether the end-of-file flag is set for a given *stream*. |
| ferror | stdio.h | int ferror(FILE \**stream*); | Tests for an error indicator in reading from or writing to *stream*. |
| fflush1 | stdio.h | int fflush(FILE \**stream*); | Writes the contents of the buffer associated with the output *stream*. |
| fgetc1 | stdio.h | int fgetc(FILE \**stream*); | Reads a single unsigned character from the input *stream*. |
| fgetpos1 | stdio.h | int fgetpos(FILE \**stream*, fpos\_t \**pos*); | Stores the current position of the file pointer associated with *stream* into the object pointed to by *pos*. |
| fgets1 | stdio.h | char \*fgets(char \**string*, int *n*, FILE \**stream*); | Reads a string from the input *stream*. |
| fgetwc6 | stdio.h wchar.h | wint\_t fgetwc(FILE \**stream*); | Reads the next multibyte character from the input stream pointed to by *stream*. |
| fgetws6 | stdio.h wchar.h | wchar\_t \*fgetws(wchar\_t \**wcs*, int *n*, FILE \**stream*); | Reads wide characters from the stream into the array pointed to by *wcs*. |
| fileno5 | stdio.h | int fileno(FILE *\*stream*); | Determines the file handle currently associated with *stream*. |
| floor | math.h | double floor(double *x*); | Calculates the floating-point value representing the largest integer less than or equal to *x*. |
| fmod | math.h | double fmod(double *x*, double *y*); | Calculates the floating-point remainder of *x/y*. |
| fopen | stdio.h | FILE \*fopen(const char \**filename*, const char \**mode*); | Opens the specified file. |
| fprintf | stdio.h | int fprintf(FILE \**stream*, const char \**format-string*, *arg-list*); | Formats and prints characters and values to the output *stream*. |
| fputc1 | stdio.h | int fputc(int *c*, FILE \**stream*); | Prints a character to the output *stream*. |
| fputs1 | stdio.h | int fputs(const char \**string*, FILE \**stream*); | Copies a string to the output *stream*. |
| fputwc6 | stdio.h wchar.h | wint\_t fputwc(wchar\_t *wc*, FILE *\*stream*); | Converts the wide character *wc* to a multibyte character and writes it to the output stream pointed to by *stream* at the current position. |
| fputws6 | stdio.h wchar.h | int fputws(const wchar\_t \**wcs*, FILE *\*stream*); | Converts the wide-character string *wcs* to a multibyte-character string and writes it to *stream* as a multibyte character string. |
| fread | stdio.h | size\_t fread(void \**buffer*, size\_t *size*, size\_t *count*, FILE \**stream*); | Reads up to *count* items of *size* length from the input *stream*, and stores them in *buffer*. |
| free | stdlib.h | void free(void \**ptr*); | Frees a block of storage. |
| freopen | stdio.h | FILE \*freopen(const char \**filename*, const char \**mode*, FILE \**stream*); | Closes *stream*, and reassigns it to the file specified. |
| frexp | math.h | double frexp(double *x*, int \**expptr*); | Separates a floating-point number into its mantissa and exponent. |
| fscanf | stdio.h | int fscanf(FILE \**stream*, const char \**format-string*, *arg-list*); | Reads data from *stream* into locations given by *arg-list*. |
| fseek1 | stdio.h | int fseek(FILE \**stream*, long int *offset*, int *origin*); | Changes the current file position associated with *stream* to a new location. |
| fsetpos1 | stdio.h | int fsetpos(FILE \**stream*, const fpos\_t \**pos*); | Moves the current file position to a new location determined by *pos*. |
| ftell1 | stdio.h | long int ftell(FILE \**stream*); | Gets the current position of the file pointer. |
| fwide6 | stdio.h wchar.h | int fwide(FILE *\*stream*, int *mode*); | Determines the orientation of the stream pointed to by *stream*. |
| fwprintf6 | stdio.h wchar.h | int fwprintf(FILE *\*stream*, const wchar\_t *\*format*, *arg-list*); | Writes output to the stream pointed to by *stream*. |
| fwrite | stdio.h | size\_t fwrite(const void \**buffer*, size\_t *size*,size\_t *count*, FILE \**stream*); | Writes up to *count* items of *size* length from *buffer* to *stream*. |
| fwscanf6 | stdio.h wchar.h | int fwscanf(FILE *\*stream*, const wchar\_t *\*format*, *arg-list*) | Reads input from the stream pointed to by *stream*. |
| gamma | math.h | double gamma(double *x*); | Computes the Gamma Function |
| getc1 | stdio.h | int getc(FILE \**stream*); | Reads a single character from the input *stream*. |
| getchar1 | stdio.h | int getchar(void); | Reads a single character from *stdin.* |
| getenv | stdlib.h | char \*getenv(const char \**varname*); | Searches environment variables for *varname*. |
| gets | stdio.h | char \*gets(char \**buffer*); | Reads a string from *stdin*, and stores it in *buffer*. |
| getwc6 | stdio.h wchar.h | wint\_t getwc(FILE *\*stream*); | Reads the next multibyte character from *stream*, converts it to a wide character and advances the associated file position indicator for *stream*. |
| getwchar6 | wchar.h | wint\_t getwchar(void); | Reads the next multibyte character from stdin, converts it to a wide character, and advances the associated file position indicator for stdin. |
| gmtime | time.h | struct tm \*gmtime(const time\_t \**time*); | Converts a *time* value to a structure of type tm. |
| gmtime64 | time.h | struct tm \*gmtime64(const time64\_t \**time*); | Converts a *time* value to a structure of type tm. |
| gmtime\_r | time.h | struct tm \*gmtime\_r (const time\_t \*time, struct tm \*result); | Converts a *time* value to a structure of type tm. (Restartable version of gmtime.) |
| gmtime64\_r | time.h | struct tm \*gmtime64\_r (const time64\_t \*time, struct tm \*result); | Converts a *time* value to a structure of type tm. (Restartable version of gmtime64.) |
| hypot | math.h | double hypot(double *side1*, double *side2*); | Calculates the hypotenuse of a right-angled triangle with sides of length *side1* and *side2*. |
| isalnum | ctype.h | int isalnum(int *c*); | Tests if *c* is alphanumeric. |
| isalpha | ctype.h | int isalpha(int *c*); | Tests if *c* is alphabetic. |
| isascii4 | ctype.h | int isascii(int *c*); | Tests if *c* is within the 7-bit US-ASCII range. |
| isblank | ctype.h | int isblank(int *c*); | Tests if *c* is a blank or tab character. |
| iscntrl | ctype.h | int iscntrl(int *c*); | Tests if *c* is a control character. |
| isdigit | ctype.h | int isdigit(int *c*); | Tests if *c* is a decimal digit. |
| isgraph | ctype.h | int isgraph(int *c*); | Tests if *c* is a printable character excluding the space. |
| islower | ctype.h | int islower(int *c*); | Tests if *c* is a lowercase letter. |
| isprint | ctype.h | int isprint(int *c*); | Tests if *c* is a printable character including the space. |
| ispunct | ctype.h | int ispunct(int *c*); | Tests if *c* is a punctuation character. |
| isspace | ctype.h | int isspace(int *c*); | Tests if *c* is a whitespace character. |
| isupper | ctype.h | int isupper(int *c*); | Tests if *c* is an uppercase letter. |
| iswalnum4 | wctype.h | int iswalnum (wint\_t wc); | Checks for any alphanumeric wide character. |
| iswalpha4 | wctype.h | int iswalpha (wint\_t wc); | Checks for any alphabetic wide character. |
| iswblank4 | wctype.h | int iswblank (wint\_t wc); | Checks for any blank or tab wide character. |
| iswcntrl4 | wctype.h | int iswcntrl (wint\_t wc); | Tests for any control wide character. |
| iswctype4 | wctype.h | int iswctype(wint\_t wc, wctype\_t wc\_prop); | Determines whether or not the wide character wc has the property wc\_prop. |
| iswdigit4 | wctype.h | int iswdigit (wint\_t wc); | Checks for any decimal-digit wide character. |
| iswgraph4 | wctype.h | int iswgraph (wint\_t wc); | Checks for any printing wide character except for the wide-character space. |
| iswlower4 | wctype.h | int iswlower (wint\_t wc); | Checks for any lowercase wide character. |
| iswprint4 | wctype.h | int iswprint (wint\_t wc); | Checks for any printing wide character. |
| iswpunct4 | wctype.h | int iswpunct (wint\_t wc); | Test for a wide non-alphanumeric, non-space character. |
| iswspace4 | wctype.h | int iswspace (wint\_t wc); | Checks for any wide character that corresponds to an implementation-defined set of wide characters for which iswalnum is false. |
| iswupper4 | wctype.h | int iswupper (wint\_t wc); | Checks for any uppercase wide character. |
| iswxdigit4 | wctype.h | int iswxdigit (wint\_t wc); | Checks for any hexadecimal digit character. |
| isxdigit4 | wctype.h | int isxdigit(int *c*); | Tests if *c* is a hexadecimal digit. |
| j0 | math.h | double j0(double *x*); | Calculates the Bessel function value of the first kind of order 0. |
| j1 | math.h | double j1(double *x*); | Calculates the Bessel function value of the first kind of order 1. |
| jn | math.h | double jn(int *n*, double *x*); | Calculates the Bessel function value of the first kind of order *n*. |
| labs | stdlib.h | long int labs(long int *n*); | Calculates the absolute value of *n*. |
| ldexp | math.h | double ldexp(double *x*, int *exp*); | Returns the value of *x* multiplied by (2 to the power of *exp*). |
| ldiv | stdlib.h | ldiv\_t ldiv(long int *numerator*, long int *denominator*); | Calculates the quotient and remainder of *numerator*/*denominator*. |
| localeconv | locale.h | struct lconv \*localeconv(void); | Formats numeric quantities in struct lconv according to the current locale. |
| localtime | time.h | struct tm \*localtime(const time\_t \**timeval*); | Converts *timeval* to a structure of type tm. |
| localtime64 | time.h | struct tm \*localtime64(const time64\_t \**timeval*); | Converts *timeval* to a structure of type tm. |
| localtime\_r | time.h | struct tm \*localtime\_r (const time\_t \*timeval, struct tm \*result); | Converts a *time* value to a structure of type *tm*. (Restartable version of localtime.) |
| localtime64\_r | time.h | struct tm \*localtime64\_r (const time64\_t \*timeval, struct tm \*result); | Converts a *time* value to a structure of type *tm*. (Restartable version of localtime64.) |
| log | math.h | double log(double *x*); | Calculates the natural logarithm of *x*. |
| log10 | math.h | double log10(double *x*); | Calculates the base 10 logarithm of *x*. |
| longjmp | setjmp.h | void longjmp(jmp\_buf *env*, int *value*); | Restores a stack environment previously set in *env* by the setjmp function. |
| malloc | stdlib.h | void \*malloc(size\_t *size*); | Reserves a block of storage. |
| mblen | stdlib.h | int mblen(const char \**string*, size\_t *n*); | Determines the length of a multibyte character *string*. |
| mbrlen4 | wchar.h | int mbrlen (const char \*s, size\_t n, mbstate\_t \*ps); | Determines the length of a multibyte character. (Restartable version of mblen.) |
| mbrtowc4 | wchar.h | int mbrtowc (wchar\_t \*pwc, const char \*s, size\_t n, mbstate\_t \*ps); | Convert a multibyte character to a wide character (Restartable version of mbtowc.) |
| mbsinit4 | wchar.h | int mbsinit (const mbstate\_t \*ps); | Test state object *\*ps* for initial state. |
| mbsrtowcs4 | wchar.h | size\_t mbsrtowc (wchar\_t \*dst, const char \*\*src, size\_t len, mbstate\_t \*ps); | Convert multibyte string to a wide character string. (Restartable version of mbstowcs.) |
| mbstowcs | stdlib.h | size\_t mbstowcs(wchar\_t \**pwc*, const char \**string*, size\_t *n*); | Converts the multibyte characters in *string* to their corresponding wchar\_t codes, and stores not more than *n* codes in *pwc*. |
| mbtowc | stdlib.h | int mbtowc(wchar\_t \**pwc*, const char \**string*, size\_t *n*); | Stores the wchar\_t code corresponding to the first *n* bytes of multibyte character *string* into the wchar\_t character *pwc*. |
| memchr | string.h | void \*memchr(const void \**buf*, int *c*, size\_t *count*); | Searches the first *count* bytes of *buf* for the first occurrence of *c* converted to an unsigned character. |
| memcmp | string.h | int memcmp(const void \**buf1*, const void \**buf2*, size\_t *count*); | Compares up to *count* bytes of *buf1* and *buf2*. |
| memcpy | string.h | void \*memcpy(void \**dest*, const void \**src*, size\_t *count*); | Copies *count* bytes of *src* to *dest*. |
| memmove | string.h | void \*memmove(void \**dest*, const void \**src*, size\_t *count*); | Copies *count* bytes of *src* to *dest*. Allows copying between objects that overlap. |
| memset | string.h | void \*memset(void \**dest*, int *c*, size\_t *count*); | Sets *count* bytes of *dest* to a value *c*. |
| mktime | time.h | time\_t mktime(struct tm \**time*); | Converts local *time* into calendar time. |
| mktime64 | time.h | time64\_t mktime64(struct tm \**time*); | Converts local *time* into calendar time. |
| modf | math.h | double modf(double *x*, double \**intptr*); | Breaks down the floating-point value *x* into fractional and integral parts. |
| nextafter | math.h | double nextafter(double *x*, double *y*); | Calculates the next representable value after *x* in the direction of *y*. |
| nextafterl | math.h | long double nextafterl(long double *x*, long double *y*); | Calculates the next representable value after *x* in the direction of *y*. |
| nexttoward | math.h | double nexttoward(double *x*, long double *y*); | Calculates the next representable value after *x* in the direction of *y*. |
| nexttowardl | math.h | long double nexttowardl(long double *x*, long double *y*); | Calculates the next representable value after *x* in the direction of *y*. |
| nl\_langinfo4 | langinfo.h | char \*nl\_langinfo(nl\_item *item*); | Retrieve from the current locale the string that describes the requested information specified by *item*. |
| perror | stdio.h | void perror(const char \**string*); | Prints an error message to stderr. |
| pow | math.h | double pow(double *x*, double *y*); | Calculates the value *x* to the power *y*. |
| printf | stdio.h | int printf(const char \**format-string*, *arg-list*); | Formats and prints characters and values to stdout. |
| putc1 | stdio.h | int putc(int *c*, FILE \**stream*); | Prints *c* to the output *stream*. |
| putchar1 | stdio.h | int putchar(int *c*); | Prints *c* to stdout. |
| putenv | stdlib.h | int \*putenv(const char \**varname*); | Sets the value of an environment variable by altering an existing variable or creating a new one. |
| puts | stdio.h | int puts(const char \**string*); | Prints a string to stdout. |
| putwc6 | stdio.h wchar.h | wint\_t putwchar(wchar\_t *wc*, FILE *\*stream*); | Converts the wide character *wc* to a multibyte character, and writes it to the stream at the current position. |
| putwchar6 | wchar.h | wint\_t putwchar(wchar\_t *wc*); | Converts the wide character *wc* to a multibyte character and writes it to stdout. |
| qsort | stdlib.h | void qsort(void \**base*, size\_t *num*, size\_t *width*, int(\**compare*)(const void \**element1*, const void \**element2*)); | Performs a quick sort of an array of *num* elements, each of *width* bytes in size. |
| quantexpd32 | math.h | \_Decimal32 quantized32(\_Decimal32 *x*, \_Decimal32 *y*); | Compute the quantum exponent of a single-precision decimal floating-point value. |
| quantexpd64 | math.h | \_Decimal64 quantized64(\_Decimal64 *x*, \_Decimal64 *y*); | Compute the quantum exponent of a double-precision decimal floating-point value. |
| quantexpd128 | math.h | \_Decimal128 quantized128(\_Decimal128 *x*, \_Decimal128 *y*); | Compute the quantum exponent of a quad-precision decimal floating-point value. |
| quantized32 | math.h | int quantexpd32(\_Decimal32 *x*); | Set the quantum exponent of a single-precision decimal floating-point value to the quantum exponent of another single-precision decimal floating-point value. |
| quantized64 | math.h | int quantexpd64(\_Decimal64 *x*); | Set the quantum exponent of a double-precision decimal floating-point value to the quantum exponent of another double-precision decimal floating-point value. |
| quantized128 | math.h | int quantexpd128(\_Decimal128 *x*); | Set the quantum exponent of a quad-precision decimal floating-point value to the quantum exponent of another quad-precision decimal floating-point value. |
| samequantumd32 | math.h | \_\_bool\_\_ samequantumd32(\_Decimal32 *x*, \_Decimal32 *y*); | Determine if the quantum exponents of two single-precision decimal floating-point values are the same. |
| samequantumd64 | math.h | \_\_bool\_\_ samequantumd64(\_Decimal64 *x*, \_Decimal64 *y*); | Determine if the quantum exponents of two double-precision decimal floating-point values are the same. |
| samequantumd128 | math.h | \_\_bool\_\_ samequantumd128(\_Decimal128 *x*, \_Decimal128 *y*); | Determine if the quantum exponents of two quad-precision decimal floating-point values are the same. |
| raise | signal.h | int raise(int *sig*); | Sends the signal *sig* to the running program. |
| rand | stdlib.h | int rand(void); | Returns a pseudo-random integer. |
| rand\_r | stdlib.h | int rand\_r(void); | Returns a pseudo-random integer. (Restartable version) |
| realloc | stdlib.h | void \*realloc(void \**ptr*, size\_t *size*); | Changes the *size* of a previously reserved storage block. |
| regcomp | regex.h | int regcomp(regex\_t *\*preg*, const char *\*pattern*, int *cflags*); | Compiles the source regular expression pointed to by *pattern* into an executable version and stores it in the location pointed to by *preg*. |
| regerror | regex.h | size\_t regerror(int *errcode*, const regex\_t *\*preg*, char *\*errbuf*, size\_t *errbuf\_size*); | Finds the description for the error code *errcode* for the regular expression *preg*. |
| regexec | regex.h | int regexec(const regex\_t *\*preg*, const char *\*string*, size\_t *nmatch*, regmatch\_t *\*pmatch*, int *eflags*); | Compares the null-ended string *string* against the compiled regular expression *preg* to find a match between the two. |
| regfree | regex.h | void regfree(regex\_t \*preg); | Frees any memory that was allocated by regcomp to implement the regular expression *preg*. |
| remove | stdio.h | int remove(const char \**filename*); | Deletes the file specified by *filename*. |
| rename | stdio.h | int rename(const char \**oldname*, const char \**newname*); | Renames the specified file. |
| rewind1 | stdio.h | void rewind(FILE \**stream*); | Repositions the file pointer associated with *stream* to the beginning of the file. |
| scanf | stdio.h | int scanf(const char \**format-string*, *arg-list*); | Reads data from stdin into locations given by *arg-list*. |
| setbuf | stdio.h | void setbuf(FILE \**stream*, char \**buffer*); | Controls buffering for *stream*. |
| setjmp | setjmp.h | int setjmp(jmp\_buf *env*); | Saves a stack environment that can be subsequently restored by longjmp. |
| setlocale | locale.h | char \*setlocale(int *category*, const char \**locale*); | Changes or queries variables defined in the *locale*. |
| setvbuf | stdio.h | int setvbuf(FILE \**stream*, char \**buf*, int *type*, size\_t *size*); | Controls buffering and buffer *size* for *stream*. |
| signal | signal.h | void(\*signal (int *sig*, void(\**func*)(int))) (int); | Registers func as a signal handler for the signal sig. |
| sin | math.h | double sin(double *x*); | Calculates the sine of *x*. |
| sinh | math.h | double sinh(double *x*); | Calculates the hyperbolic sine of *x*. |
| snprintf | stdio.h | int snprintf(char \*outbuf, size\_t n, const char\*, ...) | Same as sprintf except that the function will stop after n characters have been written to outbuf. |
| sprintf | stdio.h | int sprintf(char \**buffer*, const char \**format-string*, *arg-list*); | Formats and stores characters and values in *buffer*. |
| sqrt | math.h | double sqrt(double *x*); | Calculates the square root of *x*. |
| srand | stdlib.h | void srand(unsigned int *seed*); | Sets the *seed* for the pseudo-random number generator. |
| sscanf | stdio.h | int sscanf(const char \**buffer*, const char \**format*, *arg-list*); | Reads data from *buffer* into the locations given by *arg-list*. |
| strcasecmp | strings.h | int srtcasecmp(const char \*string1, const char \*string2); | Compares strings without case sensitivity. |
| strcat | string.h | char \*strcat(char \**string1*, const char \**string2*); | Concatenates *string2* to *string1*. |
| strchr | string.h | char \*strchr(const char \**string*, int *c*); | Locates the first occurrence of *c* in *string*. |
| strcmp | string.h | int strcmp(const char \**string1*, const char \**string2*); | Compares the value of *string1* to *string2*. |
| strcoll | string.h | int strcoll(const char \**string1*, const char \**string2*); | Compares two strings using the collating sequence in the current locale. |
| strcpy | string.h | char \*strcpy(char \**string1*, const char \**string2*); | Copies *string2* into *string1*. |
| strcspn | string.h | size\_t strcspn(const char \**string1*, const char \**string2*); | Returns the length of the initial substring of *string1* consisting of characters not contained in *string2*. |
| strerror | string.h | char \*strerror(int *errnum*); | Maps the error number in *errnum* to an error message string. |
| strfmon4 | wchar.h | int strfmon (char \*s, size\_t maxsize, const char \*format, ...); | Converts monetary value to string. |
| strftime | time.h | size\_t strftime (char \**dest*, size\_t *maxsize*, const char \**format*, const struct tm \**timeptr*); | Stores characters in an array pointed to by *dest*, according to the string determined by *format*. |
| strlen | string.h | size\_t strlen(const char \**string*); | Calculates the length of *string*. |
| strncasecmp | strings.h | int strncasecmp(const char \*string1, const char \*string2, size\_t count); | Compares strings without case sensitivity. |
| strncat | string.h | char \*strncat(char \**string1*, const char \**string2*, size\_t *count*); | Concatenates up to *count* characters of *string2* to *string1*. |
| strncmp | string.h | int strncmp(const char \**string1*, const char \**string2*, size\_t *count*); | Compares up to *count* characters of *string1* and *string2*. |
| strncpy | string.h | char \*strncpy(char \**string1*, const char \**string2*, size\_t *count*); | Copies up to *count* characters of *string2* to *string1*. |
| strpbrk | string.h | char \*strpbrk(const char \**string1*, const char \**string2*); | Locates the first occurrence in *string1* of any character in *string2*. |
| strptime4 | time.h | char \*strptime (const char \*buf, const char \*format, struct tm \*tm); | Date and time conversion |
| strrchr | string.h | char \*strrchr(const char \**string*, int *c*); | Locates the last occurrence of *c* in *string*. |
| strspn | string.h | size\_t strspn(const char \**string1*, const char \**string2*); | Returns the length of the initial substring of *string1* consisting of characters contained in *string2*. |
| strstr | string.h | char \*strstr(const char \**string1*, const char \**string2*); | Returns a pointer to the first occurrence of *string2* in *string1*. |
| strtod | stdlib.h | double strtod(const char \**nptr*, char \*\**endptr*); | Converts *nptr* to a double precision value. |
| strtod32 | stdlib.h | \_Decimal32 strtod32(const char \**nptr*, char \*\**endptr*); | Converts *nptr* to a single-precision decimal floating-point value. |
| strtod64 | stdlib.h | \_Decimal64 strtod64(const char \**nptr*, char \*\**endptr*); | Converts *nptr* to a double-precision decimal floating-point value. |
| strtod128 | stdlib.h | \_Decimal128 strtod128(const char \**nptr*, char \*\**endptr*); | Converts *nptr* to a quad-precision decimal floating-point value. |
| strtof | stdlib.h | float strtof(const char \**nptr*, char \*\**endptr*); | Converts *nptr* to a float value. |
| strtok | string.h | char \*strtok(char \**string1*, const char \**string2*); | Locates the next token in *string1* delimited by the next character in *string2*. |
| strtok\_r | string.h | char \*strtok\_r(char \**string*, const char \**seps*, char \*\**lasts*); | Locates the next token in *string* delimited by the next character in *seps*. (Restartable version of strtok.) |
| strtol | stdlib.h | long int strtol(const char \**nptr*, char \*\**endptr*, int *base*); | Converts *nptr* to a signed long integer. |
| strtold | stdlib.h | long double strtold(const char \**nptr*, char \*\**endptr*); | Converts *nptr* to a long double value. |
| strtoul | stdlib.h | unsigned long int strtoul(const char \**string1*, char \*\**string2*, int *base*); | Converts *string1* to an unsigned long integer. |
| strxfrm | string.h | size\_t strxfrm(char \**string1*, const char \**string2*, size\_t *count*); | Converts *string2* and places the result in *string1*. The conversion is determined by the program's current locale. |
| swprintf | wchar.h | int swprintf(wchar\_t *\*wcsbuffer*, size\_t *n*, const wchar\_t *\*format*, *arg-list*); | Formats and stores a series of wide characters and values into the wide-character buffer *wcsbuffer*. |
| swscanf | wchar.h | int swscanf (const wchar\_t *\*buffer*, const wchar\_t *\*format*, *arg-list*) | Reads data from *buffer* into the locations given by*arg-list*. |
| system | stdlib.h | int system(const char \**string*); | Passes *string* to the system command analyzer. |
| tan | math.h | double tan(double *x*); | Calculates the tangent of *x*. |
| tanh | math.h | double tanh(double *x*); | Calculates the hyperbolic tangent of *x*. |
| time | time.h | time\_t time(time\_t \**timeptr*); | Returns the current calendar time. |
| time64 | time.h | time64\_t time64(time64\_t \**timeptr*); | Returns the current calendar time. |
| tmpfile | stdio.h | FILE \*tmpfile(void); | Creates a temporary binary file and opens it. |
| tmpnam | stdio.h | char \*tmpnam(char \**string*); | Generates a temporary file name. |
| toascii | ctype.h | int toascii(int *c*); | Converts *c* to a character in the 7-bit US-ASCII character set. |
| tolower | ctype.h | int tolower(int *c*); | Converts *c* to lowercase. |
| toupper | ctype.h | int toupper(int *c*); | Converts *c* to uppercase. |
| towctrans | wctype.h | wint\_t towctrans(wint\_t *wc*, wctrans\_t *desc*); | Translates the wide character *wc* based on the mapping described by *desc*. |
| towlower4 | wctype.h | wint\_t towlower (wint\_t wc); | Converts uppercase letter to lowercase letter. |
| towupper4 | wctype.h | wint\_t towupper (wint\_t wc); | Converts lowercase letter to uppercase letter. |
| ungetc1 | stdio.h | int ungetc(int *c*, FILE \**stream*); | Pushes *c* back onto the input *stream*. |
| ungetwc6 | stdio.h wchar.h | wint\_t ungetwc(wint\_t *wc*, FILE *\*stream*); | Pushes the wide character *wc* back onto the input stream. |
| va\_arg | stdarg.h | *var\_type* va\_arg(va\_list *arg\_ptr*, var\_type); | Returns the value of one argument and modifies *arg\_ptr* to point to the next argument. |
| va\_copy | stdarg.h | void *va\_copy*(va\_list *dest*, va\_list *src*); | Initializes *dest* as a copy of *src*. |
| va\_end | stdarg.h | void *va\_end*(va\_list *arg\_ptr*); | Facilitates normal return from variable argument list processing. |
| va\_start | stdarg.h | void *va\_start*(va\_list *arg\_ptr, variable\_name*); | Initializes *arg\_ptr* for subsequent use by *va\_arg* and *va\_end*. |
| vfprintf | stdio.h stdarg.h | int vfprintf(FILE \**stream*, const char \**format*, *va\_list arg\_ptr*); | Formats and prints characters to the output *stream* using a variable number of arguments. |
| vfscanf | stdio.h stdarg.h | int vfscanf(FILE *\*stream*, const char *\*format*, va\_list *arg\_ptr*); | Reads data from a specified stream into locations given by a variable number of arguments. |
| vfwprintf6 | stdarg.h stdio.h wchar.h | int vfwprintf(FILE *\*stream*, const wchar\_t *\*format*, *va\_list arg*); | Equivalent to fwprintf, except that the variable argument list is replaced by *arg*. |
| vfwscanf | stdio.h stdarg.h | int vfwscanf(FILE *\*stream*, const wchar\_t *\*format*, *va\_list arg\_ptr*); | Reads wide data from a specified stream into locations given by a variable number of arguments. |
| vprintf | stdio.h stdarg.h | int vprintf(const char \**format*, va\_list *arg\_ptr*); | Formats and prints characters to stdout using a variable number of arguments. |
| vscanf | stdio.h stdarg.h | int vscanf(const char *\*format*, *va\_list arg\_ptr*); | Reads data from stdin into locations given by a variable number of arguments. |
| vsprintf | stdio.h stdarg.h | int vsprintf(char \**target-string*, const char \**format*, va\_list *arg\_ptr*); | Formats and stores characters in a buffer using a variable number of arguments. |
| vsnprintf | stdio.h | int vsnprintf(char \*outbuf, size\_t n, const char\*, va\_list); | Same as vsprintf except that the function will stop after n characters have been written to outbuf. |
| vsscanf | stdio.h stdarg.h | int vsscanf(const char*\*buffer*, const char *\*format*, *va\_list arg\_ptr*); | Reads data from a buffer into locations given by a variable number of arguments. |
| vswprintf | stdarg.h wchar.h | int vswprintf(wchar\_t *\*wcsbuffer*, size\_t *n*, const wchar\_t *\*format*, *va\_list arg*); | Formats and stores a series of wide characters and values in the buffer *wcsbuffer*. |
| vswscanf | stdio.h wchar.h | int vswscanf(const wchar\_t *\*buffer*, const wchar\_t *\*format*, *va\_list arg\_ptr*); | Reads wide data from a buffer into locations given by a variable number of arguments. |
| vwprintf6 | stdarg.h wchar.h | int vwprintf(const wchar\_t *\*format*, *va\_list arg*); | Equivalent to wprintf, except that the variable argument list is replaced by *arg*. |
| vwscanf | stdio.h wchar.h | int vwscanf(const wchar\_t *\*format*, *va\_list arg\_ptr*); | Reads wide data from stdin into locations given by a variable number of arguments. |
| wcrtomb4 | wchar.h | int wcrtomb (char \*s, wchar\_t wchar, mbstate\_t \*pss); | Converts a wide character to a multibyte character. (Restartable version of wctomb.) |
| wcscat | wchar.h | wchar\_t \*wcscat(wchar\_t *\*string1*, const wchar\_t *\*string2*); | Appends a copy of the string pointed to by *string2* to the end of the string pointed to by *string1*. |
| wcschr | wchar.h | wchar\_t \*wcschr(const wchar\_t *\*string*, wchar\_t *character*); | Searches the wide-character string pointed to by *string* for the occurrence of *character*. |
| wcscmp | wchar.h | int wcscmp(const wchar\_t *\*string1*, const wchar\_t *\*string2*); | Compares two wide-character strings, *\*string1* and *\*string2*. |
| wcscoll4 | wchar.h | int wcscoll (const wchar\_t \*wcs1, const wchar\_t \*wcs2); | Compares two wide-character strings using the collating sequence in the current locale. |
| wcscpy | wchar.h | wchar\_t \*wcscpy(wchar\_t *\*string1*, const wchar\_t *\*string2*); | Copies the contents of *\*string2* (including the ending wchar\_t null character) into *\*string1*. |
| wcscspn | wchar.h | size\_t wcscspn(const wchar\_t *\*string1*, const wchar\_t *\*string2*); | Determines the number of wchar\_t characters in the initial segment of the string pointed to by *\*string1* that do not appear in the string pointed to by *\*string2*. |
| wcsftime | wchar.h | size\_t wcsftime(wchar\_t *\*wdest*, size\_t *maxsize*, const wchar\_t *\*format*, const struct tm *\*timeptr*); | Converts the time and date specification in the *timeptr* structure into a wide-character string. |
| wcslen | wchar.h | size\_t wcslen(const wchar\_t *\*string*); | Computes the number of wide-characters in the string pointed to by *string*. |
| wcslocaleconv | locale.h | struct wcslconv \*wcslocaleconv(void); | Formats numeric quantities in struct wcslconv according to the current locale. |
| wcsncat | wchar.h | wchar\_t \*wcsncat(wchar\_t *\*string1*, const wchar\_t *\*string2*, size\_t *count*); | Appends up to *count* wide characters from *string2* to the end of *string1*, and appends a wchar\_t null character to the result. |
| wcsncmp | wchar.h | int wcsncmp(const wchar\_t *\*string1*, const wchar\_t *\*string2*, size\_t *count*); | Compares up to *count* wide characters in *string1* to *string2*. |
| wcsncpy | wchar.h | wchar\_t \*wcsncpy(wchar\_t *\*string1*, const wchar\_t *\*string2*, size\_t *count*); | Copies up to *count* wide characters from *string2* to *string1*. |
| wcspbrk | wchar.h | wchar\_t \*wcspbrk(const wchar\_t *\*string1*, const wchar\_t *\*string2*); | Locates the first occurrence in the string pointed to by *string1* of any wide characters from the string pointed to by *string2*. |
| wcsptime | wchar.h | wchar\_t \*wcsptime ( const wchar\_t \*buf, const wchar\_t \*format, struct tm \*tm ); | Date and time conversion. Equivalent to strptime(), except that it uses wide characters. |
| wcsrchr | wchar.h | wchar\_t \*wcsrchr(const wchar\_t *\*string*, wchar\_t *character*); | Locates the last occurrence of *character* in the string pointed to by *string*. |
| wcsrtombs4 | wchar.h | size\_t wcsrtombs (char \*dst, const wchar\_t \*\*src, size\_t len, mbstate\_t \*ps); | Converts wide character string to multibyte string. (Restartable version of wcstombs.) |
| wcsspn | wchar.h | size\_t wcsspn(const wchar\_t *\*string1*, const wchar\_t *\*string2*); | Computes the number of wide characters in the initial segment of the string pointed to by *string1*, which consists entirely of wide characters from the string pointed to by *string2*. |
| wcsstr | wchar.h | wchar\_t \*wcsstr(const wchar\_t *\*wcs1*, const wchar\_t *\*wcs2*); | Locates the first occurrence of *wcs2* in *wcs1*. |
| wcstod | wchar.h | double wcstod(const wchar\_t *\*nptr*, wchar\_t *\*\*endptr*); | Converts the initial portion of the wide-character string pointed to by *nptr* to a double value. |
| wcstod32 | wchar.h | \_Decimal32 wcstod32(const wchar\_t *\*nptr*, wchar\_t *\*\*endptr*); | Converts the initial portion of the wide-character string pointed to by *nptr* to a single-precision decimal floating-point value. |
| wcstod64 | wchar.h | \_Decimal64 wcstod64(const wchar\_t *\*nptr*, wchar\_t *\*\*endptr*); | Converts the initial portion of the wide-character string pointed to by *nptr* to a double-precision decimal floating-point value. |
| wcstod128 | wchar.h | \_Decimal128 wcstod128(const wchar\_t *\*nptr*, wchar\_t *\*\*endptr*); | Converts the initial portion of the wide-character string pointed to by *nptr* to a quad-precision decimal floating-point value. |
| wcstof | wchar.h | float wcstof(const wchar\_t *\*nptr*, wchar\_t *\*\*endptr*); | Converts the initial portion of the wide-character string pointed to by *nptr* to a float value. |
| wcstok | wchar.h | wchar\_t \*wcstok(wchar\_t *\*wcs1*, const wchar\_t *\*wcs2*, wchar\_t *\*\*ptr*) | Breaks *wcs1* into a sequence of tokens, each of which is delimited by a wide character from the wide string pointed to by *wcs2*. |
| wcstol | wchar.h | long int wcstol(const wchar\_t *\*nptr*, wchar\_t *\*\*endptr*, int *base*); | Converts the initial portion of the wide-character string pointed to by *nptr* to a long integer value. |
| wcstold | wchar.h | long double wcstold(const wchar\_t *\*nptr*, wchar\_t *\*\*endptr*); | Converts the initial portion of the wide-character string pointed to by *nptr* to a long double value. |
| wcstombs | stdlib.h | size\_t wcstombs(char \*dest, const wchar\_t \**string*, size\_t *count*); | Converts the wchar\_t *string* into a multibyte string *dest*. |
| wcstoul | wchar.h | unsigned long int wcstoul(const wchar\_t *\*nptr*, wchar\_t *\*\*endptr*, int *base*); | Converts the initial portion of the wide-character string pointed to by *nptr* to an unsigned long integer value. |
| wcsxfrm4 | wchar.h | size\_t wcsxfrm (wchar\_t *\*wcs1*, const wchar\_t *\*wcs2*, size\_t *n*); | Transforms a wide-character string to values which represent character collating weights and places the resulting wide-character string into an array. |
| wctob | stdarg.h wchar.h | int wctob(wint\_t *wc*); | Determines whether *wc* corresponds to a member of the extended character set whose multibyte character representation is a single byte when in the initial shift state. |
| wctomb | stdlib.h | int wctomb(char \**string*, wchar\_t *character*); | Converts the wchar\_t value of *character* into a multibyte *string*. |
| wctrans | wctype.h | wctrans\_t wctrans(const char *\*property*); | Constructs a value with type wctrans\_t that describes a mapping between wide characters identified by the string argument property. |
| wctype4 | wchar.h | wctype\_t wctype (const char *\*property*); | Obtains handle for character property classification. |
| wcwidth | wchar.h | int wcswidth(const wchar\_t *\*pwcs*, size\_t *n*); | Determine the display width of a wide character string. |
| wmemchr | wchar.h | wchar\_t \*wmemchr(const wchar\_t *\*s*, wchar\_t *c*, size\_t *n*); | Locates the first occurrence of *c* in the initial *n* wide characters of the object pointed to by *s.* |
| wmemcmp | wchar.h | int wmemcmp(const wchar\_t *\*s1*, const wchar\_t *\*s2*, size\_t *n*); | Compares the first *n* wide characters of the object pointed to by *s1* to the first *n* characters of the object pointed to by *s2*. |
| wmemcpy | wchar.h | wchar\_t \*wmemcpy(wchar\_t *\*s1*, const wchar\_t *\*s2*, size\_t *n*); | Copies *n* wide characters from the object pointed to by *s2* to the object pointed to by *s1*. |
| wmemmove | wchar.h | wchar\_t \*wmemmove(wchar\_t *\*s1*, const wchar\_t *\*s2*, size\_t *n*); | Copies *n* wide characters from the object pointed to by *s2* to the object pointed to by *s1*. |
| wmemset | wchar.h | wchar\_t \*wmemset(wchar\_t *\*s*, wchar\_t *c*, size\_t *n*); | Copies the value of *c* into each of the first *n* wide characters of the object pointed to by *s*. |
| wprintf6 | wchar.h | int wprintf(const wchar\_t *\*format*, *arg-list*); | Equivalent to fwprintf with the argument stdout interposed before the arguments to wprintf. |
| wscanf6 | wchar.h | int wscanf(const wchar\_t *\*format*, *arg-list*); | Equivalent to fwscanf with the argument stdin interposed before the arguments of wscanf. |
| y0 | math.h | double y0(double *x*); | Calculates the Bessel function value of the second kind of order 0. |
| y1 | math.h | double y1(double *x*); | Calculates the Bessel function value of the second kind of order 1. |
| yn | math.h | double yn(int *n*, double *x*); | Calculates the Bessel function value of the second kind of order *n.* |
| Table 1. Standard C Library Functions | | | |