

C Library - <stdlib.h>

The **stdlib.h** header defines four variable types, several macros, and various functions for performing general functions.

Library Variables

Following are the variable types defined in the header `stdlib.h` –

Sr.No.	Variable & Description
1	size_t This is the unsigned integral type and is the result of the sizeof keyword.
2	wchar_t This is an integer type of the size of a wide character constant.
3	div_t This is the structure returned by the div function.
4	ldiv_t This is the structure returned by the ldiv function.

Library Macros

Following are the macros defined in the header `stdlib.h` –

Sr.No.	Macro & Description
1	NULL This macro is the value of a null pointer constant.
2	EXIT_FAILURE This is the value for the exit function to return in case of failure.
3	EXIT_SUCCESS This is the value for the exit function to return in case of success.
4	RAND_MAX This macro is the maximum value returned by the rand function.
5	MB_CUR_MAX This macro is the maximum number of bytes in a multi-byte character set which cannot be larger than MB_LEN_MAX.

Library Functions

Following are the functions defined in the header `stlib.h` –

Sr.No.	Function & Description
1	double atof(const char *str) Converts the string pointed to, by the argument <i>str</i> to a floating-point number (type double).
2	int atoi(const char *str) Converts the string pointed to, by the argument <i>str</i> to an integer (type int).
3	long int atol(const char *str) Converts the string pointed to, by the argument <i>str</i> to a long integer (type long int).
4	double strtod(const char *str, char **endptr) Converts the string pointed to, by the argument <i>str</i> to a floating-point number (type double).
5	long int strtol(const char *str, char **endptr, int base) Converts the string pointed to, by the argument <i>str</i> to a long integer (type long int).
6	unsigned long int strtoul(const char *str, char **endptr, int base) Converts the string pointed to, by the argument <i>str</i> to an unsigned long integer (type unsigned long int).
7	void *calloc(size_t nitems, size_t size) Allocates the requested memory and returns a pointer to it.
8	void free(void *ptr) Deallocates the memory previously allocated by a call to <i>calloc</i> , <i>malloc</i> , or <i>realloc</i> .
9	void *malloc(size_t size) Allocates the requested memory and returns a pointer to it.
10	void *realloc(void *ptr, size_t size) Attempts to resize the memory block pointed to by <i>ptr</i> that was previously allocated with a call to <i>malloc</i> or <i>calloc</i> .
11	void abort(void) Causes an abnormal program termination.
12	int atexit(void (*func)(void))

	Causes the specified function func to be called when the program terminates normally.
13	void exit(int status) Causes the program to terminate normally.
14	char *getenv(const char *name) Searches for the environment string pointed to by name and returns the associated value to the string.
15	int system(const char *string) The command specified by string is passed to the host environment to be executed by the command processor.
16	void *bsearch(const void *key, const void *base, size_t nitems, size_t size, int (*compar)(const void *, const void *)) Performs a binary search.
17	void qsort(void *base, size_t nitems, size_t size, int (*compar)(const void *, const void *)) Sorts an array.
18	int abs(int x) Returns the absolute value of x.
19	div_t div(int numer, int denom) Divides numer (numerator) by denom (denominator).
20	long int labs(long int x) Returns the absolute value of x.
21	ldiv_t ldiv(long int numer, long int denom) Divides numer (numerator) by denom (denominator).
22	int rand(void) Returns a pseudo-random number in the range of 0 to <i>RAND_MAX</i> .
23	void srand(unsigned int seed) This function seeds the random number generator used by the function rand .

24	<pre>int mblen(const char *str, size_t n)</pre> <p>Returns the length of a multibyte character pointed to by the argument <i>str</i>.</p>
25	<pre>size_t mbstowcs(schar_t *pwcs, const char *str, size_t n)</pre> <p>Converts the string of multibyte characters pointed to by the argument <i>str</i> to the array pointed to by <i>pwcs</i>.</p>
26	<pre>int mbtowc(wchar_t *pwc, const char *str, size_t n)</pre> <p>Examines the multibyte character pointed to by the argument <i>str</i>.</p>
27	<pre>size_t wcstombs(char *str, const wchar_t *pwcs, size_t n)</pre> <p>Converts the codes stored in the array <i>pwcs</i> to multibyte characters and stores them in the string <i>str</i>.</p>
28	<pre>int wctomb(char *str, wchar_t wchar)</pre> <p>Examines the code which corresponds to a multibyte character given by the argument <i>wchar</i>.</p>