Null-terminated byte strings

A null-terminated byte string (NTBS) is a sequence of nonzero bytes followed by a byte with value zero (the terminating null character). Each byte in a byte string encodes one character of some character set. For example, the character array `{'\x63','\x61','\x74','\0'}` is an NTBS holding the string `"cat"` in ASCII encoding.

Functions

Character classification

Defined in header <ctype.h>

isalnum	checks if a character is alphanumeric (function)
isalpha	checks if a character is alphabetic (function)
islower	checks if a character is lowercase (function)
isupper	checks if a character is an uppercase character (function)
isdigit	checks if a character is a digit (function)
isxdigit	checks if a character is a hexadecimal character (function)
iscntrl	checks if a character is a control character (function)
isgraph	checks if a character is a graphical character (function)
isspace	checks if a character is a space character (function)
isblank (C99)	checks if a character is a blank character (function)
isprint	checks if a character is a printing character (function)
ispunct	checks if a character is a punctuation character (function)

Character manipulation

tolower	converts a character to lowercase (function)
toupper	converts a character to uppercase (function)

Note: additional functions whose names begin with either **to** or **is**, followed by a lowercase letter, may be added to the header ctype.h in future and should not be defined by programs that include that header.

ASCII values			characters	iscntrl iswcntrl	isprint iswprint	isspace iswspace	isblank iswblank	isgraph iswgraph	ispunct iswpunct	isalnum iswalnum	isalpha iswalpha	isupper iswupper	islower iswlower	isdigit iswdigit	isxdigit iswxdigit
decimal	hexadecimal	octal													
0–8	\x0-\x8	\0-\10	control codes (NUL, etc.)	≠0	0	0	0	0	0	0	0	0	0	0	0
9	\x9	\11	tab (\t)	≠0	0	≠0	≠0	0	0	0	0	0	0	0	0
10–13	\xA-\xD	\12-\15	whitespaces (\n, \v, \f, \r)	≠0	0	≠0	0	0	0	0	0	0	0	0	0
14–31	\xE-\x1F	\16-\37	control codes	≠0	0	0	0	0	0	0	0	0	0	0	0
32	\x20	\40	space	0	≠0	≠0	≠0	0	0	0	0	0	0	0	0
33–47	\x21-\x2F	\41-\57	!"#$%&'()*+,-./	0	≠0	0	0	≠0	≠0	0	0	0	0	0	0
48–57	\x30-\x39	\60-\71	0123456789	0	≠0	0	0	≠0	0	≠0	0	0	0	≠0	≠0
58–64	\x3A-\x40	\72-\100	:;<=>?@	0	≠0	0	0	≠0	≠0	0	0	0	0	0	0
65–70	\x41-\x46	\101-\106	ABCDEF	0	≠0	0	0	≠0	0	≠0	≠0	≠0	0	0	≠0
71–90	\x47-\x5A	\107-\132	GHIJKLMNOP QRSTUVWXYZ	0	≠0	0	0	≠0	0	≠0	≠0	≠0	0	0	0
91–96	\x5B-\x60	\133-\140	[\]^_`	0	≠0	0	0	≠0	≠0	0	0	0	0	0	0
97–102	\x61-\x66	\141-\146	abcdef	0	≠0	0	0	≠0	0	≠0	≠0	0	≠0	0	≠0
103–122	\x67-\x7A	\147-\172	ghijklmnop qrstuvwxyz	0	≠0	0	0	≠0	0	≠0	≠0	0	≠0	0	0
123–126	\x7B-\x7E	\172-\176	{|}~	0	≠0	0	0	≠0	≠0	0	0	0	0	0	0
127	\x7F	\177	backspace character (DEL)	≠0	0	0	0	0	0	0	0	0	0	0	0

Conversions to numeric formats

Defined in header <stdlib.h>

atof	converts a byte string to a floating-point value (function)

atoi **atol** **atoll** (C99)	converts a byte string to an integer value (function)
strtol **strtoll** (C99)	converts a byte string to an integer value (function)
strtoul **strtoull** (C99)	converts a byte string to an unsigned integer value (function)
strtof (C99) **strtod** **strtold** (C99)	converts a byte string to a floating point value (function)
Defined in header <inttypes.h>	
strtoimax (C99) **strtoumax** (C99)	converts a byte string to `intmax_t` or `uintmax_t` (function)

String manipulation

Defined in header <string.h>	
strcpy **strcpy_s** (C11)	copies one string to another (function)
strncpy **strncpy_s** (C11)	copies a certain amount of characters from one string to another (function)
strcat **strcat_s** (C11)	concatenates two strings (function)
strncat **strncat_s** (C11)	concatenates a certain amount of characters of two strings (function)
strxfrm	transform a string so that strcmp would produce the same result as strcoll (function)
strdup (C23)	allocates a copy of a string (function)
strndup (C23)	allocates a copy of a string of specified size (function)

String examination

Defined in header <string.h>	
strlen **strnlen_s** (C11)	returns the length of a given string (function)
strcmp	compares two strings (function)
strncmp	compares a certain amount of characters of two strings (function)
strcoll	compares two strings in accordance to the current locale (function)
strchr	finds the first occurrence of a character (function)
strrchr	finds the last occurrence of a character (function)
strspn	returns the length of the maximum initial segment that consists of only the characters found in another byte string (function)
strcspn	returns the length of the maximum initial segment that consists of only the characters not found in another byte string (function)
strpbrk	finds the first location of any character in one string, in another string (function)
strstr	finds the first occurrence of a substring of characters (function)
strtok **strtok_s** (C11)	finds the next token in a byte string (function)

Character array manipulation

Defined in header <string.h>	
memchr	searches an array for the first occurrence of a character (function)
memcmp	compares two buffers (function)
memset **memset_s** (C11)	fills a buffer with a character (function)
memcpy **memcpy_s** (C11)	copies one buffer to another (function)
memmove **memmove_s** (C11)	moves one buffer to another (function)
memccpy (C23)	copies one buffer to another, stopping after the specified delimiter (function)

Miscellaneous

Defined in header <string.h>	
	returns a text version of a given error code (function)

strerror
strerror_s (C11)
strerrorlen_s (C11)

References

- C11 standard (ISO/IEC 9899:2011):
  - 7.4 Character handling <ctype.h> (p: 200-204)
  - 7.8 Format conversion of integer types <inttypes.h> (p: 217-220)
  - 7.22 General utilities <stdlib.h> (p: 340-360)
  - 7.24 String handling <string.h> (p: 362-372)
  - 7.31.2 Character handling <ctype.h> (p: 455)
  - 7.31.5 Format conversion of integer types <inttypes.h> (p: 455)
  - 7.31.12 General utilities <stdlib.h> (p: 456)
  - 7.31.13 String handling <string.h> (p: 456)
  - K.3.6 General utilities <stdlib.h> (p: 604=613)
  - K.3.7 String handling <string.h> (p: 614-623)
- C99 standard (ISO/IEC 9899:1999):
  - 7.4 Character handling <ctype.h> (p: 181-185)
  - 7.8 Format conversion of integer types <inttypes.h> (p: 198-201)
  - 7.20 General utilities <stdlib.h> (p: 306-324)
  - 7.21 String handling <string.h> (p: 325-334)
  - 7.26.2 Character handling <ctype.h> (p: 401)
  - 7.26.4 Format conversion of integer types <inttypes.h> (p: 401)
  - 7.26.10 General utilities <stdlib.h> (p: 402)
  - 7.26.11 String handling <string.h> (p: 402)
- C89/C90 standard (ISO/IEC 9899:1990):
  - 4.3 CHARACTER HANDLING <ctype.h>
  - 4.10 GENERAL UTILITIES <stdlib.h>
  - 4.11 STRING HANDLING <string.h>
  - 4.13.2 Character handling <ctype.h>
  - 4.13.7 General utilities <stdlib.h>
  - 4.13.8 String handling <string.h>

See also

C++ documentation for **Null-terminated byte strings**