

## C Library - <string.h>

The **string.h** header defines one variable type, one macro, and various functions for manipulating arrays of characters.

### Library Variables

Following is the variable type defined in the header string.h –

Sr.No.	Variable & Description
1	<b>size_t</b> This is the unsigned integral type and is the result of the <b>sizeof</b> keyword.

### Library Macros

Following is the macro defined in the header string.h –

Sr.No.	Macro & Description
1	<b>NULL</b> This macro is the value of a null pointer constant.

### Library Functions

Following are the functions defined in the header string.h –

Sr.No.	Function & Description
1	void *memchr(const void *str, int c, size_t n) Searches for the first occurrence of the character c (an unsigned char) in the first n bytes of the string pointed to, by the argument <i>str</i> .
2	int memcmp(const void *str1, const void *str2, size_t n) Compares the first n bytes of <i>str1</i> and <i>str2</i> .
3	void *memcpy(void *dest, const void *src, size_t n) Copies n characters from src to <i>dest</i> .
4	void *memmove(void *dest, const void *src, size_t n) Another function to copy n characters from <i>str2</i> to <i>str1</i> .
5	void *memset(void *str, int c, size_t n) Copies the character c (an unsigned char) to the first n characters of the string pointed to, by the argument <i>str</i> .
6	char *strcat(char *dest, const char *src) Appends the string pointed to, by <i>src</i> to the end of the string pointed to by <i>dest</i> .
7	char *strncat(char *dest, const char *src, size_t n) Appends the string pointed to, by <i>src</i> to the end of the string pointed to, by <i>dest</i> up to n characters long.
8	char *strchr(const char *str, int c) Searches for the first occurrence of the character c (an unsigned char) in the string pointed to, by the argument <i>str</i> .
9	int strcmp(const char *str1, const char *str2) Compares the string pointed to, by <i>str1</i> to the string pointed to by <i>str2</i> .
10	int strncmp(const char *str1, const char *str2, size_t n) Compares at most the first n bytes of <i>str1</i> and <i>str2</i> .
11	int strcoll(const char *str1, const char *str2) Compares string <i>str1</i> to <i>str2</i> . The result is dependent on the LC_COLLATE setting of the location.

12	<code>char *strcpy(char *dest, const char *src)</code> Copies the string pointed to, by <i>src</i> to <i>dest</i> .
13	<code>char *strncpy(char *dest, const char *src, size_t n)</code> Copies up to <i>n</i> characters from the string pointed to, by <i>src</i> to <i>dest</i> .
14	<code>size_t strcspn(const char *str1, const char *str2)</code> Calculates the length of the initial segment of <i>str1</i> which consists entirely of characters not in <i>str2</i> .
15	<code>char *strerror(int errnum)</code> Searches an internal array for the error number <i>errnum</i> and returns a pointer to an error message string.
16	<code>size_t strlen(const char *str)</code> Computes the length of the string <i>str</i> up to but not including the terminating null character.
17	<code>char *strpbrk(const char *str1, const char *str2)</code> Finds the first character in the string <i>str1</i> that matches any character specified in <i>str2</i> .
18	<code>char *strrchr(const char *str, int c)</code> Searches for the last occurrence of the character <i>c</i> (an unsigned char) in the string pointed to by the argument <i>str</i> .
19	<code>size_t strspn(const char *str1, const char *str2)</code> Calculates the length of the initial segment of <i>str1</i> which consists entirely of characters in <i>str2</i> .
20	<code>char *strstr(const char *haystack, const char *needle)</code> Finds the first occurrence of the entire string <i>needle</i> (not including the terminating null character) which appears in the string <i>haystack</i> .
21	<code>char *strtok(char *str, const char *delim)</code> Breaks string <i>str</i> into a series of tokens separated by <i>delim</i> .
22	<code>size_t strxfrm(char *dest, const char *src, size_t n)</code> Transforms the first <i>n</i> characters of the string <i>src</i> into current locale and places them in the string <i>dest</i> .