

NAME

wmemchr, **wmemcmp**, **wmemcpy**, **wmemmove**, **wmempcpy**, **wmemset**, **wcpcpy**, **wcpncpy**, **wscasecmp**, **wscat**, **wchr**, **wscmp**, **wscopy**, **wscspn**, **wcsdup**, **wslcat**, **wslcpy**, **wslen**, **wcncasecmp**, **wcncat**, **wcncmp**, **wcncpy**, **wcslen**, **wcspbrk**, **wcsrchr**, **wcsspn**, **wcsstr** - wide character string manipulation operations

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <wchar.h>
```

wchar_t *

```
wmemchr(const wchar_t *s, wchar_t c, size_t n);
```

int

```
wmemcmp(const wchar_t *s1, const wchar_t *s2, size_t n);
```

wchar_t *

```
wmemcpy(wchar_t * restrict s1, const wchar_t * restrict s2, size_t n);
```

wchar_t *

```
wmemmove(wchar_t *s1, const wchar_t *s2, size_t n);
```

wchar_t *

```
wmempcpy(wchar_t * restrict s1, const wchar_t * restrict s2, size_t n);
```

wchar_t *

```
wmemset(wchar_t *s, wchar_t c, size_t n);
```

wchar_t *

```
wcpcpy(wchar_t * restrict s1, const wchar_t * restrict s2);
```

wchar_t *

```
wcpncpy(wchar_t * restrict s1, const wchar_t * restrict s2, size_t n);
```

int

```
wcscasecmp(const wchar_t *s1, const wchar_t *s2);
```

wchar_t *

wscat(*wchar_t* * restrict *s1*, const *wchar_t* * restrict *s2*);

wchar_t *

wchr(const *wchar_t* **s*, *wchar_t* *c*);

int

wscmp(const *wchar_t* **s1*, const *wchar_t* **s2*);

wchar_t *

wscpy(*wchar_t* * restrict *s1*, const *wchar_t* * restrict *s2*);

size_t

wscspn(const *wchar_t* **s1*, const *wchar_t* **s2*);

wchar_t *

wcscdup(const *wchar_t* **s*);

size_t

wcslcat(*wchar_t* **s1*, const *wchar_t* **s2*, *size_t* *n*);

size_t

wcslcpy(*wchar_t* **s1*, const *wchar_t* **s2*, *size_t* *n*);

size_t

wcslen(const *wchar_t* **s*);

int

wcncasecmp(const *wchar_t* **s1*, const *wchar_t* **s2*, *size_t* *n*);

wchar_t *

wcncat(*wchar_t* * restrict *s1*, const *wchar_t* * restrict *s2*, *size_t* *n*);

int

wcncmp(const *wchar_t* **s1*, const *wchar_t* * *s2*, *size_t* *n*);

wchar_t *

wcncpy(*wchar_t* * restrict *s1*, const *wchar_t* * restrict *s2*, *size_t* *n*);

size_t

wcslen(const *wchar_t* **s*, *size_t* *maxlen*);

wchar_t *

wcspbrk(*const wchar_t *s1, const wchar_t *s2*);

wchar_t *

wcsrchr(*const wchar_t *s, wchar_t c*);

size_t

wcsspn(*const wchar_t *s1, const wchar_t *s2*);

wchar_t *

wcsstr(*const wchar_t * restrict s1, const wchar_t * restrict s2*);

DESCRIPTION

The functions implement string manipulation operations over wide character strings. For a detailed description, refer to documents for the respective single-byte counterpart, such as `memchr(3)`.

SEE ALSO

`memchr(3)`, `memcmp(3)`, `memcpy(3)`, `memmove(3)`, `memset(3)`, `stpcpy(3)`, `stpncpy(3)`, `strcasemp(3)`, `strcat(3)`, `strchr(3)`, `strcmp(3)`, `strcpy(3)`, `strcspn(3)`, `strdup(3)`, `strlcat(3)`, `strncpy(3)`, `strlen(3)`, `strncat(3)`, `strncmp(3)`, `strncpy(3)`, `strnlen(3)`, `strpbrk(3)`, `strrchr(3)`, `strspn(3)`, `strstr(3)`

STANDARDS

These functions conform to ISO/IEC 9899:1999 ("ISO C99"), with the exception of **wcpcpy()**, **wcpncpy()**, **wscasecmp()**, **wcsdup()**, **wcsncasecmp()**, and **wcsnlen()**, which conform to IEEE Std 1003.1-2008 ("POSIX.1"); and **wslcat()**, **wslcpy()**, and **wmempcpy()**, which are extensions.