they complete a valid multibyte character. If so, the multibyte character is converted into a wide character. If `pwc` isn't a null pointer, the wide character is stored in the object pointed to by `pwc`. The value of `ps` should be a pointer to an object of type `mbstate_t` that contains the current conversion state. If `ps` is a null pointer, `mbrtowc` uses an internal object to store the conversion state. If the result of the conversion is the null wide character, the `mbstate_t` object used during the call is left in the initial conversion state.

Returns 0 if the conversion produces a null wide character. Returns a number between 1 and `n` if the conversion produces a wide character other than null, where the value returned is the number of bytes used to complete the multibyte character. Returns `(size_t)(-2)` if the `n` bytes pointed to by `s` weren't enough to complete a multibyte character. Returns `(size_t)(-1)` and stores `EILSEQ` in `errno` if an encoding error occurs. *25.5*

---

mbsinit — *Test for Initial Conversion State (C99)* — <wchar.h>

```
int mbsinit(const mbstate_t *ps);
```

Returns A nonzero value if `ps` is a null pointer or it points to an `mbstate_t` object that describes an initial conversion state; otherwise, returns zero. *25.5*

---

mbsrtowcs — *Convert Multibyte String to Wide String – Restartable (C99)* — <wchar.h>

```
size_t mbsrtowcs(wchar_t * restrict dst,
                 const char ** restrict src,
                 size_t len, mbstate_t * restrict ps);
```

Converts a sequence of multibyte characters from the array indirectly pointed to by `src` into a sequence of corresponding wide characters. `ps` should point to an object of type `mbstate_t` that contains the current conversion state. If the argument corresponding to `ps` is a null pointer, `mbsrtowcs` uses an internal object to store the conversion state. If `dst` isn't a null pointer, the converted characters are stored in the array that it points to. Conversion continues up to and including a terminating null character, which is also stored. Conversion stops earlier if a sequence of bytes is encountered that doesn't form a valid multibyte character or—if `dst` isn't a null pointer—when `len` wide characters have been stored in the array. If `dst` isn't a null pointer, the object pointed to by `src` is assigned either a null pointer (if a terminating null character was reached) or the address just past the last multibyte character converted (if any). If the conversion ends at a null character and if `dst` isn't a null pointer, the resulting state is the initial conversion state.

Returns Number of multibyte characters successfully converted, not including any terminating null character. Returns `(size_t)(-1)` and stores `EILSEQ` in `errno` if an invalid multibyte character is encountered. *25.5*

---

mbstowcs — *Convert Multibyte String to Wide String* — <stdlib.h>

```
size_t mbstowcs(wchar_t * restrict pwcs,
                const char * restrict s, size_t n);
```