strxfrm returns the length of the transformed string. As a result, it's typically called twice: once to determine the length of the transformed string and once to perform the transformation. Here's an example:

```
size_t len;
char *transformed;

len = strxfrm(NULL, original, 0);
transformed = malloc(len + 1);
strxfrm(transformed, original, len);
```

Search Functions

```
void *memchr(const void *s, int c, size_t n);
char *strchr(const char *s, int c);
size_t strcspn(const char *s1, const char *s2);
char *strpbrk(const char *s1, const char *s2);
char *strrchr(const char *s, int c);
size_t strspn(const char *s1, const char *s2);
char *strstr(const char *s1, const char *s2);
char *strtok(char * restrict s1,
             const char * restrict s2);
```

strchr The strchr function searches a string for a particular character. The following example shows how we might use strchr to search a string for the letter f.

```
char *p, str[] = "Form follows function.";

p = strchr(str, 'f');   /* finds first 'f' */
```

strchr returns a pointer to the first occurrence of f in str (the one in the word follows). Locating multiple occurrences of a character is easy; for example, the call

```
p = strchr(p + 1, 'f');   /* finds next 'f' */
```

finds the second f in str (the one in function). If it can't locate the desired character, strchr returns a null pointer.

memchr memchr is similar to strchr, but it stops searching after a set number of characters instead of stopping at the first null character. memchr's third argument limits the number of characters it can examine—a useful capability if we don't want to search an entire string or if we're searching a block of memory that's not terminated by a null character. The following example uses memchr to search an array of characters that lacks a null character at the end:

```
char *p, str[22] = "Form follows function.";

p = memchr(str, 'f', sizeof(str));
```