Copying Functions

```
void *memcpy(void * restrict s1,
             const void * restrict s2, size_t n);
void *memmove(void *s1, const void *s2, size_t n);
char *strcpy(char * restrict s1,
             const char * restrict s2);
char *strncpy(char * restrict s1,
              const char * restrict s2, size_t n);
```

Q&A The functions in this category copy characters (bytes) from one place in memory (the "source") to another (the "destination"). Each function requires that the first argument point to the destination and the second point to the source. All copying functions return the first argument (a pointer to the destination).

memcpy memmove memcpy copies n characters from the source to the destination, where n is the function's third argument. If the source and destination overlap, the behavior of memcpy is undefined. memmove is the same as memcpy, except that it works correctly when the source and destination overlap.

strcpy strncpy strcpy copies a null-terminated string from the source to the destination. strncpy is similar to strcpy, but it won't copy more than n characters, where n is the function's third argument. (If n is too small, strncpy won't be able to copy a terminating null character.) If it encounters a null character in the source, strncpy adds null characters to the destination until it has written a total of n characters. strcpy and strncpy, like memcpy, aren't guaranteed to work if the source and destination overlap.

The following examples illustrate the copying functions; the comments show which characters are copied.

```
char source[] = {'h', 'o', 't', '\0', 't', 'e', 'a'};
char dest[7];

memcpy(dest, source, 3);    /* h, o, t                   */
memcpy(dest, source, 4);    /* h, o, t, \0               */
memcpy(dest, source, 7);    /* h, o, t, \0, t, e, a      */

memmove(dest, source, 3);   /* h, o, t                   */
memmove(dest, source, 4);   /* h, o, t, \0               */
memmove(dest, source, 7);   /* h, o, t, \0, t, e, a      */

strcpy(dest, source);       /* h, o, t, \0               */

strncpy(dest, source, 3);   /* h, o, t                   */
strncpy(dest, source, 4);   /* h, o, t, \0               */
strncpy(dest, source, 7);   /* h, o, t, \0, \0, \0, \0 */
```

Note that memcpy, memmove, and strncpy don't require a null-terminated string; they work just as well with any block of memory. The strcpy function, on the other hand, doesn't stop copying until it reaches a null character, so it works only with null-terminated strings.