## NAME

**memset** - write a byte to byte string

## LIBRARY

Standard C Library (libc, -lc)

## SYNOPSIS

**#include <string.h>**

*void **
**memset**(*void *dest, int c, size_t len*);

**#define __STDC_WANT_LIB_EXT1__ 1**

*errno_t*
**memset_s**(*void *dest, rsize_t destsz, int c, rsize_t len*);

## DESCRIPTION

The **memset**() function writes *len* bytes of value *c* (converted to an *unsigned char*) to the string *dest*. Undefined behaviour from **memset**(), resulting from storage overflow, will occur if *len* is greater than the length of the *dest* buffer. The behaviour is also undefined if *dest* is an invalid pointer.

The **memset_s**() function behaves the same as **memset**() except that an error is returned and the currently registered runtime-constraint handler is called if *dest* is a null pointer, *destsz* or *len* is greater than RSIZE_MAX, or *len* is greater than *destsz* (buffer overflow would occur). The runtime-constraint handler is called first and may not return. If it does return, an error is returned to the caller. Like explicit_bzero(3), **memset_s**() is not removed through Dead Store Elimination (DSE), making it useful for clearing sensitive data. In contrast **memset**() function may be optimized away if the object modified by the function is not accessed again. To clear memory that will not subsequently be accessed it is advised to use **memset_s**() instead of **memset**(). For instance, a buffer containing a password should be cleared with **memset_s**() before free(3).

## RETURN VALUES

The **memset**() function returns its first argument. The **memset_s**() function returns zero on success, non-zero on error.

## SEE ALSO

bzero(3), explicit_bzero(3), set_constraint_handler_s(3), swab(3), wmemset(3)

## STANDARDS

The **memset**() function conforms to ISO/IEC 9899:1990 ("ISO C90"). **memset_s**() conforms to ISO/IEC 9899:2011 ("ISO C11") K.3.7.4.1.