MEMSET

#include "stdio.h"

void \*ft\_memset(void \*b, int c, size\_t len)  
//DEVUELVE UN PUNTERO GENÉRICO VOID LLAMADO B

{

size\_t i; // SIZE\_T Garantiza que el valor no sea negativo y se ajuste al rango adecuado- 0 hasta 4,294,967,295 en sistemas de 32 bits  
 unsigned char \*ptr;

// EL UNSIGNED CHAR es para asegurarse de que el valor de PTR se ajuste al rango correcto de un byte (0 a 255) antes de asignarlo al bloque de memoria.

ptr = b;  
// Convertimos el puntero genérico b a un puntero unsigned char

i = 0;  
 while (i < len)

{  
 ptr[i] = (unsigned char)c;  
// asignar el valor de c a la posición de memoria i dentro del bloque de memoria apuntado por ptr

// EL UNSIGNED CHAR es para asegurarse de que el valor de c se ajuste al rango correcto de un byte (0 a 255) antes de asignarlo al bloque de memoria.  
 i++;  
 } return (b); }

BZERO

void ft\_bzero(void \*s, size\_t n)  
// La función ft\_bzero no devuelve ningún valor, es de tipo void. Su propósito es establecer los primeros n bytes del bloque de memoria apuntado por s en cero (es decir, establecerlos en el valor binario de cero).

{

unsigned char \*ptr;

size\_t i;

ptr = (unsigned char \*)s;  
// se utiliza para asignar el valor del puntero s a la variable ptr, pero con un tipo de puntero diferente.

i = 0;

while (i < n)

{

i++;

\*ptr = 0;   
ptr++;

}  
}