



# UNIVERSIDAD AUTONOMA DE GUADALAJARA

## Microcomputadoras

### Práctica 1

Escribir las siguientes funciones en C utilizando el compilador del procesador seleccionado:

1) void *vfnMemCpy* (unsigned char\* bpSource, unsigned char\* bpDest, unsigned short wSize)

-Copia los datos del bloque apuntado por bpSource al bloque apuntado por bpDest.

2) void *vfnMemSet*(unsigned char\* bpDest, unsigned char bByteToFill, unsigned short wSize)

-Rellena el bloque apuntado por bpDest con el dato indicado por bByteToFill.

3) unsigned char *bfnFindMax*(unsigned char\* bpDest, unsigned short wSize)

-Devuelve el número sin signo mayor encontrado en el bloque de memoria apuntado por bpDest

4) unsigned char *bfnFindMin*(unsigned char\* bpDest, unsigned short wSize)

-Devuelve el número sin signo menor encontrado en el bloque de memoria apuntado por bpDest

5) unsigned char *bfnStrCmp*(unsigned char\* bpSource, unsigned char\* bpDest, unsigned short wSize)

-Compara las cadenas apuntadas por bpSource y bpDest, devolviendo un 1 si las cadenas son iguales o un 0 si son distintas

6) unsigned char\* *bfnByteAddress*(unsigned char\* bpString, unsigned char bCharToFind, unsigned short wSize)

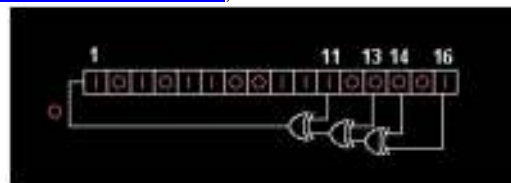
-Entrega la dirección donde se encuentra el valor bCharToFind en la cadena bpString. En caso de no encontrar datos, regresa un apuntador a NULL.

7) unsigned short *wfnStrLen* (unsigned char\* bpString)

-Devuelve el tamaño de la cadena terminada en 0 apuntada por bpString.

8) unsigned short *wfnRand*(unsigned short wSeed)

- Devuelve un número de 16 bits por medio de un generador de números pseudo aleatorio basado en la semilla recibida en wSeed. Utilice el algoritmo de Fibonacci en registros de corrimiento con retroalimentación lineal (LFSR) que muestra la figura ([http://en.wikipedia.org/wiki/Linear\\_feedback\\_shift\\_register](http://en.wikipedia.org/wiki/Linear_feedback_shift_register)):



9) unsigned char *bfnLRC*(unsigned char\* bpSrc, unsigned short wSize)

- Devuelve el número de 8 bits que resulta de realizar un xor entre todos los datos contenidos en el espacio de memoria apuntado por bpSrc con tamaño especificado por wSize. ([http://en.wikipedia.org/wiki/Longitudinal\\_redundancy\\_check](http://en.wikipedia.org/wiki/Longitudinal_redundancy_check))

10) unsigned short *wfnOccurrence*(unsigned char bSymbol, unsigned char\* bpSrc, unsigned short wSize)

- Devuelve el número de 16 bits que representa la cantidad de veces que se repite el elemento bSymbol en el bloque indicado por bpSrc de tamaño wSize ([http://en.wikipedia.org/wiki/Run-length\\_encoding](http://en.wikipedia.org/wiki/Run-length_encoding))

11) void *vfnSort* (unsigned char\* bpString, unsigned short wStringSize)

-Ordena de menor a mayor, en el mismo sitio, los datos contenidos en el buffer apuntado por bpString.

12) unsigned short *wfnSprintf*(char\* string, const char\* fmt, ...)

-Formatea en la cadena string los datos siguiendo el formato. Devuelve el tamaño de la cadena generada

%c – imprime el carácter

%d – imprime en 3 dígitos el número de 8 bits

%s – imprime como cadena los datos actuales

%x – imprime en 2 dígitos el número hexadecimal

%b – imprime 8 dígitos representando el binario de 8 bits

\*\*Las opciones de %d, %d y %b deben soportar la supresión de los ceros precedentes a menos que se utilice %0b, %0d o %0x donde se especifica que se dejarán los ceros iniciales.

NOTA: NO puede usarse NINGUNA de las librerías preexistentes de C. Toda función utilizada, deberá ser creada por el alumno.

Como entregables, se esperan:

- Diagrama de flujo de las mismas

- Comentarios del alumno sobre el trabajo realizado

- Programa de prueba con 3 casos de prueba por función

(Según se especifica en el formato de reporte de práctica)