

### **Función bit:**

Esta función debe testear el valor del bit número *nb* del *buffer* y devolver su valor booleano (0 o 1) como un entero.

ejemplo:

sea buffer =

buffer	0	1	0	1	1	0	1	0
	b7	b6	b5	b4	b3	b2	b1	b0

si nb = 5, se desea obtener el valor del bit b5

buffer	0	1	0	1	1	0	1	0
	b7	b6	b5	b4	b3	b2	b1	b0

entonces, se debe devolver un entero con valor 0.

### **Función ver\_binario:**

Esta función debe mostrar en terminal los *nb* bits menos significativos de la palabra de entrada *buffer* en forma binaria. No devuelve nada.

Ejemplo:

sea buffer =

buffer	0	...	1	1	1	0	0	0	1	0	1	1	0	1	0
	b31		b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

si nb = 4, se desea mostrar en pantalla los primeros 4 bits (b3, b2, b1, b0), esto es:

1	0	1	0
b3	b2	b1	b0

Vale aclarar que la idea no es guardar cada dígito en una variable formando un arreglo para luego desplegarlo, sino que la idea es desplegar en pantalla cada dígito a medida que se haya.

### **Función set\_bit:**

Esta función debe setear el bit nb del buffer a 0 o a 1 según sea el valor del parámetro val y devolver buffer con ese bit seteado al valor correspondiente, como caracter sin signo.

Ejemplo:

sea buffer =

buffer	0	1	0	1	1	0	1	0
	b7	b6	b5	b4	b3	b2	b1	b0

si nb = 5 y val = 1, se desea obtener el valor del bit b5 y darle valor 1:

buffer	0	1	1	1	1	0	1	0
	b7	b6	b5	b4	b3	b2	b1	b0

entonces, se debe devolver el nuevo buffer.

### **Función concatena:**

Esta función debe concatenar buffer con código. Para ello debe primero correr nb a la izquierda al contenido de buffer y colocar en su parte baja la palabra código (de tamaño nb bits), y devolver la nueva palabra así creada como caracter sin signo.

Ejemplo:

sea buffer =

buffer	0	1	0	1	1	0	1	0
	b7	b6	b5	b4	b3	b2	b1	b0

sea código =

código	0	0	0	0	1	0	0	1
	b7	b6	b5	b4	b3	b2	b1	b0

En este caso el valor de nb es 4 (largo de código, 1001), por lo que se debe correr buffer a la izquierda para dejar los 4 bits menos significativos en cero:

buffer	1	0	1	0	0	0	0	0
	b7	b6	b5	b4	b3	b2	b1	b0

concatenando, se debe devolver buffer como:

buffer	1	0	1	0	1	0	0	1
	b7	b6	b5	b4	b3	b2	b1	b0

### **Función crear\_mascara:**

Esta función debe crear una máscara cuyos bits valgan 0 con excepción de los que se encuentren entre el *bit\_min* y el *bit\_max* (incluyéndolos), que deben valer 1. Devuelve la máscara como carácter sin signo.

Ejemplo:

Sea *bit\_max* = 5 (b5) y *bit\_min* = 2 (b2), entonces se debe devolver:

máscara	0	0	1	1	1	1	0	0
	b7	b6	b5	b4	b3	b2	b1	b0

### **Función espejar:**

Esta función debe tomar los *nb* bits menos significativos de la palabra de entrada *in* y espejarlos, es decir que el bit más significativo de ese conjunto (el bit nb-1) debe aparecer ahora en la posición menos significativa (en la posición 0), el bit siguiente (bit nb-2) ir a la posición 1 y del mismo modo el resto de los bits. Se debe devolver los nb bits menos significativos de la palabra espejada como carácter sin signo.

Ejemplo:

Sea in =

in	0	...	1	1	1	0	0	0	1	0	0	1	0	1	1
	b31		b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

si nb = 6, entonces se debe espejar los bits b5, b4, b3, b2, b1, b0, es decir:

in				0	0	1	0	1	1
				b5	b4	b3	b2	b1	b0

y debe devolver:

espejado				1	1	0	1	0	0
				b5	b4	b3	b2	b1	b0

como caracter sin signo se debe devolver:

out	0	0	1	1	0	1	0	0
	b7	b6	b5	b4	b3	b2	b1	b0

### **Función paridad:**

Esta función debe evaluar la paridad de la palabra de entrada *in* y retornar el valor 1 si el número de bits 1 de in es par y 0 si ese número es impar. El valor de retorno es un entero.

Ejemplo:

sea in =

in	0	...	1	1	1	0	0	0	1	0	0	1	0	1	1
	b31		b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

supongamos que los bits b13 a b31 valen 0, entonces en este caso se debería retornar el valor 0, dado que la cantidad de 1 en in es impar.