

1. Completar, para cada símbolo del siguiente programa la tabla de símbolos. En reubicable indicar si, no o indefinido.

| Símbolo | Valor | Reubicable |
|---------|-------|------------|
|         |       |            |

```
        .org 100h
        mov AL, 00h
        mov [201h], AL
        mov AL, [200h]
        mov CX, 0008h
rotar:   rol AL, 1
        jnc no_suma
        inc [201h]
no_suma: loop rotar
        end
```

2. Completar, para cada símbolo del siguiente programa la tabla de símbolos. En reubicable indicar si, no o indefinido.

; this is a program in 8086 assembly language that  
; accepts a character string from the keyboard and  
; stores it in the string array. the program then converts  
; all the lower case characters of the string to upper case.  
; if the string is empty (null), it doesn't do anything.

name "upper"

org 100h

jmp start

; first byte is buffer size,

; second byte will hold number

; of used bytes for string,

; all other bytes are for characters:

string db 20, 22 dup('?')

new\_line db 0Dh, 0Ah, '\$' ; new line code.

start:

; int 21h / ah=0ah - input of a string to ds:dx,

; first byte is buffer size, second byte is number

; of chars actually read. does not add '\$' in the

; end of string. to print using int 21h / ah=09h

; you must set dollar sign at the end of it and

; start printing from address ds:dx + 2.

lea dx, string

mov ah, 0ah

int 21h

```

mov bx, dx
mov ah, 0
mov al, ds:[bx+1]
add bx, ax ; point to end of string.
mov byte ptr [bx+2], '$' ; put dollar to the end.
; int 21h / ah=09h - output of a string at ds:dx.
; string must be terminated by '$' sign.
lea dx, new_line
mov ah, 09h
int 21h
lea bx, string
mov ch, 0
mov cl, [bx+1] ; get string size.
jcxz null ; is string is empty?
add bx, 2 ; skip control chars.
upper_case:
; check if it's a lower case letter:
cmp byte ptr [bx], 'a'
jb ok
cmp byte ptr [bx], 'z'
ja ok
; convert to uppercase:
; upper case letter do not have
; third bit set, for example:
; 'a'      : 01100001b
; 'a'      : 01000001b
; upper case mask : 11011111b
; clear third bit:
and byte ptr [bx], 11011111b
ok:
inc bx ; next char.
loop upper_case
; int 21h / ah=09h - output of a string at ds:dx.
; string must be terminated by '$' sign.
lea dx, string+2
mov ah, 09h
int 21h
; wait for any key press....
mov ah, 0
int 16h
null:
ret ; return to operating system.

```

3. Generar el código objeto del siguiente programa como si fuera el resultado de la segunda pasada de un ensamblador de dos pasadas.

```

      .
      .
      .
add    CL, 05h
mov    AX, [BX]
add    AL, -1
mov    [BX], AX
      .
      .
      .

```

4. Desensamblar un programa es leer su código objeto y volver a obtener el lenguaje simbólico. Dado el siguiente código objeto, desensamblar para obtener las instrucciones simbólicas de assembler.

```

=====
LOC: MACHINE CODE
=====
0100: B0 05
0102: B4 02
0104: 02 C4
0106: C3

```

5. Escribir un programa con una macroinstrucción que sume los dos números almacenados en las direcciones 200h y 201h de la memoria y que al resultado lo multiplique por un tercer número almacenado en la dirección 203h. Los tres números son enteros positivos.
6. a. Escriba una macroinstrucción llamada subn similar a la instrucción sub, que le reste al primer parámetro el segundo complementado. Por ejemplo subn num1, num2 debe restar num1 menos el número que surja de invertir "1" por "0" y "0" por "1" a cada bit de num 2.  
b. Verifique el funcionamiento del programa después de cortar y pegar esa macro al principio del archivo emu8086.inc que se encuentra en la carpeta ..\emu8086\inc\ e incluir ese archivo en el programa.
7. Cuando se invoca a un macro, ¿las instrucciones CALL y RET se insertan en forma automática en el programa ensamblado?
8. Indicar si es verdadero o falso que al sustituir un procedimiento extenso con un macro que contenga el código de ese procedimiento, por lo general, se incrementa el tamaño del código compilado de un programa si el macro se invoca varias veces.
9. ¿Cuál es el propósito de la directiva LOCAL en una macro?
10. Indique si es verdadero o falso que un macro no puede contener definiciones de datos.