

TP Nº 3: Introducción al lenguaje de bajo nivel. Intel 8086

1. Escriba las instrucciones en Assembler que almacene los siguientes valores en los registros indicados:

- a. AX = 00000000b
- b. BX = 01010101b
- c. CX = 00001111b
- d. DX = 11001100b

Para asignar los valores a los registros utilice notación hexadecimal.

2. Escriba un programa en Assembler que copie el contenido del registro CX en los registros AX, BX y DX (al final del programa, los cuatro registros deben contener el mismo valor que tenía CX al comenzar).
3. Realice un programa en Assembler que sume los contenidos de los cuatro registros de propósito general y guarde el resultado en AX. (En un lenguaje de alto nivel, sería algo tan sencillo como $AX = AX + BX + CX + DX$).
4. Dada la edad de una persona almacenada en el registro AX y el año actual almacenado en CX, realice un programa que calcule y almacene en el registro DX el año en que dicha persona nació.
5. Indique qué valores hexadecimales tienen los registros AX, BX, CX y DX al finalizar la ejecución del siguiente fragmento de código Assembler. Para ello, "ejecute" paso a paso el programa, simulando la tarea que realiza el microprocesador.

```
Mov    ax,    500
Mov    bx,    1Ah
Sub    ax,    bx
Mov    cx,    ax
Dec    cx
Mov    dx,    111100011111b
Inc    dx
```

6. Indique cuál es el contenido del segmento de datos, dada la siguiente declaración de utilice notación hexadecimal.

```
numeroPrimo    DB 7
primoGrande     DW 9973
impares         DB 1, 3, 7, 9
arreglo_5      DB 5 DUP (0Fh)
nombre         DB "Luis"
```

7. Tomando como base las declaraciones del ejercicio anterior, indique cómo realizar las operaciones:
 - a. Copiar en AX el valor de primoGrande.
 - b. Copiar en AL el tercer elemento (número 7) del arreglo impares.
 - c. Asignar a los elementos 0, 1 y 2 de arreglo_5 el valor de los registros BL, CL y DL.
 - d. Cambiar el contenido de la variable nombre para que almacene el apellido "Ruiz".
8. Realice la declaración de las siguientes variables. Si no se indica, utilice el tamaño que considere adecuado.
 - a. Variable cuenta, de tamaño word, inicializada en 0.
 - b. Variable lista, un arreglo de 50 elementos de tamaño byte, inicializados todos en 0Ah.
 - c. Variable suma que permita almacenar un número entre 0 y 3000, sin inicializar.
 - d. Variable DNI que permita almacenar como string un DNI (incluidos los puntos), inicializada con su DNI.

9. Escriba un programa en Assembler que tenga dos variables, nombre y apellido, inicializadas con su nombre y su apellido. El programa debe copiar sus iniciales en los registros AL (inicial del nombre) y AH (inicial del apellido). Nota: si tiene más de un nombre o apellido, elija sólo uno de forma de tener sólo dos iniciales.
10. Dado el siguiente contenido de memoria y estado de los registros, indicar para la expresión (para los accesos a memoria, considere un operando de 8 bits).

Dir. de memoria	Contenido	Registros	Variables
139Eh	05h	AX = 001Ah BX = 139Fh SI = 0002h DI = 0060h	arreglo -> 139Eh lista -> 005Ah
139Fh	AAh		
13A0h	03h		
13A1h	00h		
13A2h	0Dh		
13A3h	FFh		
13A4h	9Eh		
13A5h	13h		
13A6h	02h		
13A7h	10h		

- al
- ds:[bx]
- ds:[13A3h]
- ds:[bx + si + 2]
- arreglo[si]
- lista[13A7h]

11. ¿Cuál es el contenido del registro AX al terminar la ejecución del siguiente código Assembler?

```

mov cx, 30h
mov ds:[A600h], cx
mov bx, A600h
mov ax, ds:[bx]

```

12. En la dirección de memoria 3000h comienza un arreglo de bytes que almacena, en posiciones consecutivas de memoria, los primeros diez números pares. ¿Qué hace el siguiente fragmento de código?

```

mov ax, 0
mov si, 0
mov cx, 10
siguiente: add al, ds:[3000h + si]
inc si
loop siguiente

```

13. Dado dos valores ubicados en la posición DS:00H y la posición DS:10H respectivamente, determinar cuál de los dos es mayor, indicando tal situación colocando un uno en la posición original del valor mayor, de ser iguales colocar cero en ambas posiciones.
14. Dado un valor ubicado en la posición DS:1FH, determinar si su bit 5 = 0, de ser así guardar en la posición el valor original incrementado en 2EH.
15. Determinar si el valor ubicado en DS:100H es par y mayor a 1FH, en el caso de ser así calcular la suma de dicho valor con 15H, almacenarlo en DS:200H y terminar, sino volver al principio y leer la posición nuevamente.
16. Indicar para un valor almacenado en DS:10H, como es la relación entre el bit 3 y 7, de ser iguales guardar en la misma posición un cero sino un uno.
17. Indicar para un valor almacenado en DS:00, si el bit2 y el bit 6 son iguales a 1, de ser así calcular el OR de dicho valor con el de la posición DS:10 y almacenarlo en DS:15
18. Indicar para un valor almacenado en DS:A1H, si el bit2 = 0 y el bit 5 = 1, de ser así calcular el XOR de dicho valor con el de la posición DS:35H y almacenarlo en DS:15H
19. Indicar par un valor almacenado en DS:10H, si el bit 6 = bit 3, en el caso de ser así calcular la suma de dicho valor con 55H, almacenarlo en DS:20H y terminar, sino volver al principio y leer la posición nuevamente.
20. Para valores ubicados en un área de memoria que va desde la posición DS:00H a la DS:10H determinar la cantidad de valores pares, almacenar el resultado en DS:15.
21. Para valores ubicados en un área de memoria que va desde la posición DS:00H a la DS:10H determinar la cantidad de valores pares mayores a 2EH, almacenar el resultado en DS:15H.
22. Para un conjunto de valores ubicados en un área de memoria que va desde DS:10H a la DS:00H, se pide determinar el mínimo de dichos valores, almacenar resultado en DS:15H.
23. Para un conjunto de valores ubicados en un área de memoria que va desde DS:10H a la DS:100H, se pide determinar el máximo de dichos valores menores a D1H, almacenar resultado en DS:150H.
24. Para un conjunto de valores ubicados en un área de memoria que va desde DS:00H a la DS:20H, se pide determinar el mínimo de dichos valores y de existir más de uno, indicar la cantidad de veces que aparece dicho valor. Almacenar el valor mínimo en DS:22H y la cantidad de veces DS:23H.
25. Para los valores almacenados entre la posición DS:1FH y DS:2FH, determinar qué cantidad de valores cumplen con la siguiente condición: bit 7=1, bit 4 =0 y bit 2=1, almacenar resultado en DS:30H.

26. Para un dato ubicado en DS:20 se pide:

bit 7	bit 4	bit 1	
1	1	0	sumar al dato original 3H, almacenarlo en DS:22 y terminar
0	0	1	realizar una xor con el valor 13H, almacenarlo en DS:23 y terminar
1	0	1	realizar una or con el valor 4AH, almacenarlo en DS:24 y terminar
x	x	x	Volver a leer las posición DS:20 creando un ciclo

27. Para un conjunto de valores ubicados en un área de memoria que va desde DS:12 a la DS:22, se pide trasladar aquellos valores que cumplan la siguiente condición: bit5 = bit2 a partir de la posición DS:25.
28. Para el ejercicio anterior determinar el valor mínimo de los datos trasladados, almacenar dicho valor en la posición DS:40
29. Se tienen valores que se ubican desde DS:00 hasta DS:30, determinar el promedio de aquellos valores que cumplan con ser mayores de 1AH, almacenar el promedio en DS:37.
30. Realice los programas de los ejercicios 28, 29, 30 utilizando una lógica que involucre procedimientos (utilizar las sentencias del macro assembler "proc" y "endp").
31. La instrucción CALL utilizada en el programa anterior necesita guardar en algún lugar la dirección de retorno, para que el procesador pueda continuar ejecutando el programa una vez que vuelva de la subrutina. ¿Cuáles es el lugar donde guarda la información y qué registros intervienen en esta tarea?