

#### TP N° 4: Programación en Assembler.

1. Realice un programa que lea de teclado 10 valores, e indique cuantos son pares en la posición DS:10.
2. Realice un programa que transfiera los 30 valores ingresados por teclados a partir de la dirección DS:10 si son mayores del valor promedios de dichos valores.
3. Realice un programa que utilice módulos de procesos para calcular el valor máximo y mínimo del rango de DS: 20 a DS:5A y los almacena en la dirección DS:100 y DS:120.
4. Repetir el programa anterior considerando que los valores se pueden repetir. Guardar las aparaciones en DS:110 y DS:121 respectivamente.
5. Realice un programa que ordene de menor a mayor un conjunto de valores que se encuentran entre la posición DS:10 y DS:5F.
6. Realice un programa que muestre las 10 primeras letras del alfabeto en minúsculas.
7. Realice un programa que muestre una frase cualquiera por pantalla.
8. Realice un programa que pida ingresar un caracter por teclado, ingresar 10 valores de 1 a 9 [validar la entrada] y mostrar el menor por pantalla
9. Realice un programa que pida ingresar un carácter por teclado valores de 1 a 9 [validar la entrada] hasta que se ingrese un ENTER y que muestre el mayor por pantalla
10. Realice un programa que muestre por pantalla el valor contenido en el registro BX, en binario.
11. Realice un programa que pida ingresar un carácter por teclado 10 valores de dos byte que muestre el mayor por pantalla
12. Realice un programa que sume el valor contenido en el registro AX al valor contenido en el registro BX y que muestre por pantalla el resultado de la suma en binario.
13. Realice un programa que tome valores del puerto 3F8H, hasta que el valor ingresado sea igual a 5FH, para los datos leídos se pide realizar las siguientes acciones, de darse más de una (concurrente), resolver en forma consecutiva,

bit 7	bit 4	bit 1	
1	1	0	sumar al dato original 3H, y escribir en el puerto 2F8H
0	0	1	realizar una or con el valor 13H, almacenarlo puerto 2F9H
1	0	1	realizar una xor con el valor AAH, almacenarlo en el puerto 3F9H

TP N° 4: Programación en Assembler.

14. Realice una subrutina que intercambie los contenidos de dos posiciones de memoria:
  - a) pasando como parámetros los valores en AX y BX,
  - b) pasando como parámetros las direcciones de los valores a intercambiar.Escriba además para cada caso como sería una llamada típica a la subrutina en el main().
15. Realice un programa que llame a una subrutina para convertir un carácter ASCII a decimal, el carácter es pasado a la subrutina en el acumulador AL. Si es un número decimal, colocarlo en AH y sino escribir FFh en AH.
16. Escriba una subrutina que cuente la cantidad de ocurrencias de un carácter, en una tabla de caracteres que finaliza con 00h. El caracter a buscar esta contenido en AL. La dirección de comienzo de la tabla esta en BX. El resultado del conteo quedará almacenado en CX.
17. Escriba una subrutina que cuente el número de “1” en un byte. El byte es pasado en AL y el conteo es devuelto en CX.
18. Escriba una subrutina que cuente el número de unos en una palabra, llamando a la subrutina escrita en el punto anterior. La palabra es pasada en AX y el conteo es devuelto en CX.
19. Escriba nuevamente la subrutina del ejercicio 4, haciendo rotaciones sucesivas para contar el número de unos.
20. Escribir un programa para recorrer una tabla de caracteres (bytes) que comienza en TABLA y cuya longitud está en AX, los pares quedarán inalterados y en los impares los 4 bits menos significativos cambiarán sus posiciones con los 4 bits más significativos.  
Ej. antes=  $b_7b_6b_5b_4b_3b_2b_1b_0$                       después= $b_3b_2b_1b_0b_7b_6b_5b_4$
21. Describa brevemente que hacen las siguientes instrucciones :

Push bx	Pop bx
Push [bx]	Pop Data3
Push DS	Pop CS
22. Escribir una subrutina que encuentre el máximo (mayor) entre números de 16 bits con signo de un arreglo. La dirección de comienzo del arreglo (16 bits) es pasada a la subrutina en el Stack y la longitud en palabras del mismo, es pasada a continuación. El número mayor es devuelto en AX.
23. Describa como es el llamado de una subrutina en Assembler desde un programa en PASCAL, indicando como es el pasaje de parámetros.
24. Escribir una subrutina que calcule el complemento a 10 de dos dígitos decimales almacenados en el registro AL en BCD empaquetado.

TP N° 4: Programación en Assembler.

25. Encontrar la primera ocurrencia en BUFFER (arreglo en memoria) de un carácter pasado en AL. Retornar Z=0 (flag) si no se encuentra el carácter ó con Z=1 y DI apuntando al carácter encontrado. BUFFER tiene una longitud de 200 bytes.
26. Encontrar la primera ocurrencia en BUFFER de un especificado string. La dirección base del string es pasada en DX. Devolver el flag Z=0 si no se encuentra el string ó con Z=1 y ES:DI apuntando al primer carácter del string que se encontró en BUFFER. BUFFER tiene una longitud de 200 bytes. El string ocupa 5 bytes.