



LABORATORIO DE COMPUTADORAS

TEMA: PROGRAMACIÓN MASM

TP
09

APELLIDO Y NOMBRE:
CARRERA:

LU:
FECHA:

CONCEPTOS

- Indique ventajas y desventajas del lenguaje ensamblador.
- Grafique los pasos necesarios para ensamblar, enlazar y ejecutar un programa.
- ¿Qué son las directivas simplificadas de segmentos? Haga un esquema-resumen.
- ¿Qué significa que un ensamblador sea de dos pasadas?
- ¿Qué es el listado de referencias cruzadas?
- Resuma los modos de direccionamiento de una arquitectura X86.
- Indique los distintos archivos necesarios y los archivos generados cuando se ensambla y enlaza un programa con el MASM. Explique la finalidad de cada uno.
- ¿Qué significa una dirección cercana y una dirección lejana?
- ¿Cuál es el propósito del BIOS?
- ¿Qué tipo de interrupciones se manejan en arquitectura X86? Describa las principales interrupciones del BIOS y del DOS.
- Describa la instrucción LEA y plantee un ejemplo.
- Explique las finalidades de las directivas PUBLIC, EXTERN y LOCAL.
- ¿Se puede convertir un programa .EXE en un programa .COM? Explique.
- Enumere y describa las directivas que ayudan a la programación de macros.
- Defina las directivas condicionales.

EJERCICIOS

- El siguiente programa permite visualizar la leyenda "LC 2023: Arquitecturas". Codifíquelo y ejecútelo.

```

TITLE ejemplo_1
; ----- Constantes -----
cr equ 13
lf equ 10

;----- Segmento de Pila -----
PILA SEGMENT STACK
    db 64 dup('PILA')
PILA ENDS

;----- Segmento de Datos -----
DATOS SEGMENT
    cartel db cr,lf,'LC2023: Arquitecturas','$'
DATOS ENDS

;----- Segmento de Código -----
CODIGO SEGMENT
    PRINCIPAL PROC FAR
        ASSUME
            DS:DATOS,CS:CODIGO,SS:PILA
        PUSH DS
        SUB AX,AX
        PUSH AX
        MOV AX,DATOS
        MOV DS,AX
        LEA DX,carTEL
        CALL ESCRIBIR
        CALL ESPERAR
        RET
    PRINCIPAL ENDP

    ESCRIBIR PROC
        PUSH AX
        MOV AH,09
        INT 21H
        POP AX
        RET
    ESCRIBIR ENDP

    ESPERAR PROC
        PUSH AX
        MOV AH,00
        INT 16H
        POP AX
        RET
    ESPERAR ENDP
CODIGO ENDS
    END PRINCIPAL
  
```

2. Analice el siguiente programa y determine su propósito (verifique codificándolo).

```

TITLE EJEMPLO
;----- CONSTANTES -----
CR EQU 13
LF EQU 10

;----- SEGMENTO DE PILA -----
PILA SEGMENT STACK
    DB 128 DUP(' ')
PILA ENDS

;----- SEGMENTO DE DATOS -----
DATOS SEGMENT
    MENSAJE DB CR,LF,'SALIDA POR PANTALLA: ','$'
    ESP DB ','','$'
    SALIDA DB 0,'$'
DATOS ENDS

;----- SEGMENTO DE CODIGO -----
CODIGO SEGMENT
INICIO PROC FAR
    ASSUME CS:CODIGO,DS:DATOS,SS:PILA
    PUSH DS
    SUB AX,AX
    PUSH AX
    MOV AX,DATOS
    MOV DS,AX

    LEA DX,MENSAJE
    CALL ESCRIBIR
    MOV SALIDA,7AH
OTRO: LEA DX,SALIDA
    CALL ESCRIBIR
    LEA DX,ESP
    CALL ESCRIBIR
    DEC SALIDA
    CMP SALIDA,61H
    JGE OTRO

    CALL ESPERAR
    RET
INICIO ENDP

ESCRIBIR PROC
    PUSH AX
    MOV AH,09
    INT 21H
    POP AX
    RET
ESCRIBIR ENDP

ESPERAR PROC
    PUSH AX
    MOV AH,00
    INT 16H
    POP AX
    RET
ESPERAR ENDP

CODIGO ENDS
END INICIO
    
```

3. Escriba un programa que muestre, 12₁₀ veces, la leyenda “Creando un bucle ☺”. Codifique 2 versiones usando: a) la instrucción LOOP y b) cualquier otro salto. Tenga en cuenta que ☺ tiene código ASCII 1.

4. Escriba un programa que permita ingresar un carácter y mostrarlo, de acuerdo al siguiente formato:

```

Ingrese un carácter:      F
El carácter F fue ingresado por el usuario.
    
```

5. Modifique el programa anterior de modo que se presente alguno de los siguientes mensajes según el carácter ingresado.: “ES UNA MAYÚSCULA”, “ES UNA MINÚSCULA”, “ES UN SÍMBOLO/DÍGITO”

6. Escriba un programa que permita ingresar 3 caracteres y determinar si éstos son iguales o todos ellos son distintos. Si se detectan 2 caracteres iguales debe presentarse el mensaje “ALERTA!!! EL BICHO” Por ejemplo,

```

Ingrese primer carácter:  @
Ingrese segundo carácter: m
Ingrese tercer carácter:  9
LOS CARACTERES SON DISTINTOS
    
```

7. Escriba un programa que, dados 3 caracteres ingresados por el usuario, muestre éstos en orden creciente. Por ejemplo:

```

Ingrese primer carácter:  A
Ingrese segundo carácter: b
Ingrese tercer carácter:  C
Salida: A C b
    
```

8. Escriba un programa que permita ingresar caracteres hasta que se introduzca el símbolo “#” o el símbolo “@”.

9. Modifique el programa anterior de modo que al finalizar el ingreso se muestren valores mínimo y máximo de la secuencia. Tenga en cuenta que los valores de salida no serán considerados para mínimo o máximo.

```

Ingrese secuencia:      M@/uA+#
Carácter Mínimo:      #
Carácter Máximo:      u
    
```

10. Escriba un programa que permita convertir, de mayúsculas a minúsculas o viceversa, un carácter ingresado por el usuario. Por ejemplo,

Ingrese un carácter:	H	O	Ingrese un carácter:	m
Carácter en Minúscula:	h		Carácter en Mayúscula:	M

Cualquier otro carácter generará un mensaje de error y repetirá el ingreso de datos.

11. Escriba un programa que muestre, de acuerdo a un valor de opción, en orden creciente o decreciente los caracteres de un rango especificado por el usuario. Por ejemplo:

Ingrese mínimo:	h
Ingrese máximo:	m
Opción de orden (1-crec., 2-dec.):	1
Salida:	hijklm

12. Analice el siguiente programa y determine su propósito:

title ejemplo	lea dx,texto1	lea dx,texto2
cr equ 13	call escribir	call escribir
lf equ 10	mov ah,01	lea dx,num3
pila segment stack	int 21h	call escribir
db 64 dup(' ')	mov num2,al	ret
pila ends	lea dx,texto1	inicio endp
datos segment	call escribir	escribir proc
texto1 db cr,lf,'Ingreso: ','\$'	mov ah,01	push ax
texto2 db cr,lf,'Salida: ','\$'	int 21h	mov ah,09
num1 db 0,'\$'	mov num3,al	int 21h
num2 db 0,'\$'	mov ah,num1	pop ax
num3 db 0,'\$'	mov al,num2	ret
datos ends	mov bl,num3	escribir endp
codigo segment	bucle: cmp ah,al	esperar proc
inicio proc far	jg etiq1	push ax
assume cs:codigo,ds:datos,ss:pila	cmp al,bl	mov ah,00
push ds	jg etiq2	int 16h
sub ax,ax	jmp etiq3	pop ax
push ax	etiq1: xchg ah,al	ret
mov ax,datos	jmp bucle	esperar endp
mov ds,ax	etiq2: xchg al,bl	codigo ends
lea dx,texto1	jmp bucle	end inicio
call escribir	etiq3: mov num1,ah	
mov ah,01	mov num2,al	
int 21h	mov num3,bl	
mov num1,al		

13. Escriba un programa que muestre los n (1 a 9) caracteres anteriores y n (1 a 9) caracteres siguientes a uno ingresado por el usuario. Por ejemplo,

Ingrese un carácter:	m
Cantidad de caracteres a mostrar	4
Caracteres anteriores:	lkji
Caracteres siguientes:	nopq

14. Escriba un programa que permita ingresar caracteres y determinar si se trata de valores primos o no. Tenga en cuenta que el programa analiza el código ASCII del valor ingresado. El ingreso finaliza con el carácter '#'.

15. Analice el siguiente programa y determine su propósito (verifique codificándolo).

title ejemplo3	minus: cmp dato,122	escribir proc near
cr equ 13	jg especial	push ax
lf equ 10	lea dx,texto2	mov ah,09
	call escribir	int 21h
pila segment stack	jmp jugar	pop ax
db 64 dup(' ')		ret
pila ends	mayus: cmp dato,90	escribir endp
	jg especial	
datos segment	lea dx,texto1	esperar proc near
texto db cr,lf,'Ingrese dato: ','\$'	call escribir	push ax
dato db 0,'\$'	jmp jugar	mov ah,00
texto1 db cr,lf,'Es una letra mayuscula','\$'		int 16h
texto2 db cr,lf,'Es una letra minuscula','\$'	digito: cmp dato,57	pop ax
texto3 db cr,lf,'Es un digito','\$'	jg especial	ret
texto4 db cr,lf,'Es un simbolo especial','\$'	lea dx,texto3	esperar endp
texto5 db cr,lf,'Arriesgas: ','\$'	call escribir	
texto6 db cr,lf,'Ganaste!!! :)','\$'	jmp jugar	codigo ends
texto7 db cr,lf,'Perdiste!!! :(','\$'		end principal
datos ends	especial: lea dx,texto4	
	call escribir	
codigo segment		
principal proc far	jugar: sub cx,cx	
assume cs:codigo,ds:datos,ss:pila	mov cl,3	
push ds	intento: lea dx,texto5	
sub ax,ax	call escribir	
push ax	mov ah,01	
mov ax,datos	int 21h	
mov ds,ax	cmp al,dato	
	je ganador	
lea dx,texto	loop intento	
call escribir		
mov ah,07	lea dx,texto7	
int 21h	call escribir	
mov dato,al	jmp fin	
cmp dato,97	ganador: lea dx,texto6	
jge minus	call escribir	
cmp dato,65	jmp fin	
jge mayus		
cmp dato,48	fin: call esperar	
jge digito	ret	
	principal endp	

Modifique el programa de modo que se permitan 5 intentos y que por cada intento incorrecto se presente un mensaje indicando si el carácter arriesgado es menor o mayor al carácter secreto.

16. Escriba un programa que ingrese 2 valores y calcule su suma. Tenga en cuenta que puede ser necesario realizar algunas operaciones de ajuste.

Ingrese primer valor:	2
Ingrese segundo valor:	6
La suma es:	8

