LABORATORIO DE COMPUTADORAS

TEMA: PROGRAMACIÓN MASM. MACROS

TP 11

2023

APELLIDO Y NOMBRE: LU:

CARRERA: FECHA:

CONCEPTOS

A. Investigue sobre las directivas condicionales. Ejemplifique.

B. ¿Bajo qué condiciones recomendaría el uso de procedimientos?

C. ¿Qué significa un procedimiento intrasegmento y un procedimiento intersegmento?

D. ¿Bajo qué condiciones recomendaría el uso de macros?

E. Describa los principales operadores y directivas útiles para macros.

F. ¿Qué significa Programación Híbrida?

G. Describa y ejemplifique las partes de un procedimiento.

H. Codifique la primera y la última línea de una macro sencilla llamada SETUP.

EJERCICIOS

1. El siguiente programa utiliza la macro **ESCRIBE** para visualizar una cadena de caracteres. Observe que la macro utiliza como parámetro la cadena de caracteres a visualizar.

```
title 'escribe_cadena'
cr equ 13
lf equ 10

escribe macro palabra
        lea dx,palabra
        call escribir
endm

datos segment
     texto1 db cr,lf,'Ingrese cadena:','$'
     texto2 db cr,lf,'Cadena ingresada:','$'
     cadena db 10 dup(' '),'$'
datos ends

pila segment stack
        db 64 dup('pila')
pila ends

codigo segment
inicio proc far
        assume ds:datos,ss:pila,cs:codigo
        push ds
        sub ax,ax
        push ax

        mov ax,datos
        mov ds,ax

        escribe texto1
        sub si,si
        mov ah,01
ingreso:int 21h
        cmp al,13
        je fingreso
        mov cadena[si],al
        inc si
        cmp si,09
        jle ingreso
fingreso:mov cadena[si],'$'
        escribe texto2
        escribe cadena
        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
```

El archivo LST generado durante la creación del programa ejecutable correspondiente al ejemplo anterior, contendrá las referencias a la macro (invocación) y el código de ésta con los parámetros utilizados en cada caso. Esto se denomina expansión de la macro, y derivará en un programa de mayor tamaño que el original ya que se agregan al código las operaciones especificadas en la macro.

```
                                        . . .
0004  B8 ---- R                   mov ax,datos
0007  8E D8                       mov ds,ax

                                escribe texto1
0009  8D 16 0000 R                +               lea dx,texto1
000D  E8 003B R                   +               call escribir
0010  2B F6                       sub si,si
0012  B4 01                       mov ah,01
                                        . . .
0024  C6 84 0026 R 24           fingreso:mov cadena[si],'$'
                                escribe texto2
0029  8D 16 0012 R                +               lea dx,texto2
002D  E8 003B R                   +               call escribir
                                escribe cadena
0030  8D 16 0026 R                +               lea dx,cadena
0034  E8 003B R                   +               call escribir
0037  E8 0042 R                   call esperar
003A  CB                          ret
                                        . . .
```

Puede observarse cómo el código de la macro sustituye al llamado o invocación de ésta, insertando sus instrucciones.

2. Escriba una macro llamada LEE que permita almacenar una cadena de caracteres ingresada por el usuario. Considere que la macro utiliza como parámetros la cadena y la longitud máxima de ésta.
3. Modifique la macro anterior de modo que sólo permita almacenar letras mayúsculas, minúsculas y dígitos. Considere que la macro se llamará ENTRADA.
4. Escriba una macro llamada MAYMIN que convierta una cadena ingresada por el usuario a mayúsculas o minúsculas de acuerdo al valor un parámetro de opción (‘M’, mayúsculas; ‘m’ minúsculas). Tenga en cuenta que la cadena introducida puede contener letras (en mayúsculas y/o minúsculas), símbolos y/o dígitos.
5. Escriba una macro llamada MAXMIN que determine los caracteres máximo y mínimo de una cadena ingresada por el usuario. Considere que la macro utiliza como parámetros la cadena ingresada y las variables que almacenarán los valores máximo y mínimo obtenidos.
6. Escriba una macro llamada ORDENA que ordene los caracteres de una cadena de caracteres. Considere que la macro utiliza como parámetros la cadena a ordenar y el criterio de ordenación (0 creciente o 1 decreciente).
7. Analice el código de la siguiente macro, identifique sus parámetros y determine el propósito de la macro:

```
misterio macro n, m, p
        push ax
        push bx
        sub ax,ax
        sub bx,bx
        mov al,n
        mov bl,m
        cmp al,bl
        jge salto
        xchg al,bl
  salto:sub al,bl
        aaa
        add ax,3030h
        xchg ah,al
        mov p,ax
        pop bx
        pop ax
endm
```

8. Escriba una macro llamada ENCRIPTA que encripte una cadena de caracteres usando una clave de 4 dígitos. Considere que la macro utiliza como parámetros la cadena a encriptar y la clave (ingresadas por el usuario). El proceso de encriptación se describe en el TP10 (ejercicio 14).
9. Escriba una macro llama CODIGO que obtenga el código ASCII (en decimal) de un carácter ingresado por el usuario. Considere que la macro utiliza como parámetros un carácter y una cadena.
10. Escriba una macro llamada CARACTER que obtenga, a partir del código ASCII ingresado por el usuario, el carácter correspondiente. Considere que la macro utiliza como parámetros el código ASCII (3 dígitos decimales) y la variable en la que se almacenará el carácter obtenido. Escriba una macro adicional que controle que el código ingresado pertenezca al intervalo [0-255], en caso contrario debe presentarse un mensaje de error.
11. Escriba una macro llamada BINADEC que obtenga, a partir de un valor de 8 bits ingresado por el usuario, el número decimal correspondiente. Tenga en cuenta que tanto el valor binario como el número decimal se almacenarán en cadenas de caracteres.
12. Escriba una macro llamada DECABIN que obtenga, a partir de un valor DECIMAL (3 dígitos) ingresado por el usuario, el equivalente binario. Considere que los valores decimal y binario se almacenarán en cadenas de caracteres.
13. Analice el siguiente programa y determine el propósito de las macros primera, segunda y tercera.

```
title ejemplo_macro
cr equ 10
lf equ 13

escribe macro cadena
        lea dx,cadena
        call escribir
endm

leecar macro dato
        push ax
        mov ah,01
        int 21h
        mov dato,al
        pop ax
endm

primera macro cadena,long
        push ax
        push di
otro:   leecar cadena[di]
        cmp cadena[di],13
        je termina
        cmp di,long
        jg termina
        inc di
        jmp otro
termina:mov cadena[di],'$'
        pop di
        pop ax
endm

pila segment stack
        db 64 dup(' ')
pila ends

datos segment
  op1 db cr,lf,'1-Opcion 1','$'
  op2 db cr,lf,'2-Opcion 2','$'
  op3 db cr,lf,'3-Opcion 3','$'
```

```
pausa macro
        escribe cad5
        call esperar
endm

menu macro op
        escribe op1
        escribe op2
        escribe op3
        escribe op4
        escribe texto
        leecar op
endm

segunda macro cadena,valor
        push si
        push ax
        sub si,si
        mov ah,cadena[si]
mas:    inc si
        cmp cadena[si],'$'
        je finseg
        cmp ah,cadena[si]
        jle mas
        mov ah,cadena[si]
        jmp mas
finseg: mov valor,ah
        pop ax
        pop si
endm
```

```
tercera macro cad1,cad2
      push si
      push cx
      sub cx,cx
      sub si,si
reinicia:mov cl,05
continua:dec cl
      cmp cl,00
      je reinicia
      cmp cad1[si],'$'
      je ftercera
      mov ch,cad1[si]
      add ch,cl
      mov cad2[si],ch
      inc si
      jmp continua
ftercera:mov cad2[si],'$'
      pop cx
      pop si
endm
```

```
  op4 db cr,lf,'4-Salir','$'
  texto db cr,lf,'Elija opcion:','$'
  texto1 db cr,lf,'Ingrese: ','$'
  texto2 db cr,lf,'Salida','$'
  cad0 db 10 dup (' '),'$'
  cad1 db 10 dup (' '),'$'
  cad2 db cr,lf,'FIN PROGRAMA','$'
  cad3 db cr,lf,'ERROR DE OPCION','$'
  cad4 db cr,lf,'SALIDA: ','$'
  cad5 db cr,lf,'Pulse una tecla ...','$'
  dato db 0,'$'
  opcion db 0,'$'
datos ends

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

mostrar:call posicionar
    call borrar
    menu opcion
    cmp opcion,'1'
    jne opcion2

    sub di,di
    escribe texto1
    primera cad1,09
    pausa
    jmp mostrar

opcion2:cmp opcion,'2'
    jne opcion3
    segunda cad1,dato
    escribe cad4
    escribe dato
    pausa
    jmp mostrar

opcion3:cmp opcion,'3'
    jne opcion4
    tercera cad1,cad0
    escribe cad4
    escribe cad0
    pausa
    jmp mostrar

opcion4:cmp opcion,'4'
    jne error
    escribe cad2
    pausa
    jmp final

 error: escribe cad3
    pausa
    jmp mostrar

 final: ret

inicio endp

escribir proc near
    push ax
    mov ah,09
    int 21h
    pop ax
    ret
 escribir endp

 esperar proc near
    push ax
    mov ah,00
    int 16h
    pop ax
    ret
 esperar endp

 borrar proc
    push ax
    push bx
    push cx
    mov ax,0600h ; opt limpiar pantalla
    mov bh,07 ; opt color letra=7 y fondo=0
    mov cx,0000  ; p. inicial fila 0, col 0
    mov dx,184fh ; p. final fila 24, col 79
    int 10h ; interrup de BIOS
    pop cx
    pop bx
    pop ax
    ret
 borrar endp

 posicionar proc
    push ax
    push bx
    push dx
    mov ah,2 ; opt posicionar el cursor
    mov bh,0 ; pagina a imprimir
    mov dh,01h ; cursor en fila 1
    mov dl,01h ; cursor en columna 1
    int 10h ; interrup de BIOS
    pop dx
    pop bx
    pop ax
    ret
 posicionar endp

codigo ends
end inicio
```

Considerando que las opciones 2 y 3 sólo pueden ejecutarse luego de la opción 1, adicione al programa los controles necesarios que verifiquen esto. ¿Cuál es el propósito de los procedimientos *borrar* y *posicionar*?

14. En ensamblador, la directiva INCLUDE permite agregar código a un programa a partir de otros archivos. Por ejemplo, para el programa del ítem 1 (*primero.asm*) se almacenó el código de la macro ESCRIBE en el archivo *primero.lib* que se incluye en el programa mediante la directiva *INCLUDE primero.lib*. Tenga en cuenta que el archivo *lib* debe almacenarse en la misma carpeta que el *asm* para ser utilizado (también pueden usarse rutas).

primero.asm

```
title 'escribe_cadena'
cr equ 13
lf equ 10

INCLUDE primero.lib

datos segment
     texto1 db cr,lf,'Ingrese cadena:','$'
     texto2 db cr,lf,'Cadena ingresada:','$'
     cadena db 10 dup(' '),'$'
datos ends

pila segment stack
        db 64 dup('pila')
pila ends

codigo segment
inicio proc far
        assume ds:datos,ss:pila,cs:codigo
              ...
        escribe texto1
        lee cadena,09
        escribe texto2
        escribe cadena
              ...
inicio endp
codigo ends
end inicio
```

primero.lib

```
escribe macro palabra
        lea dx,palabra
        call escribir
endm

lee macro cad, long
       ...
Endm
```

Considerando esto, guarde todas las macros desarrolladas en este práctico en el archivo *macros.lib* y sustituya sus definiciones por la correspondiente sentencia INCLUDE.

15. Analice la macro:

```
XPUSH MACRO R1, R2, R3, R4, R4, R5, R6, R7, R8, R9, R10
                  IRP reg; <R1, R2, R3, R4, R4, R5, R6, R7, R8, R9, R10>
                        IFNB <reg>
                                PUSH reg
                        ENDIF
                   ENDM
                  ENDM
```

Considerando que la macro XPUSH se invoca como sigue

```
XPUSH AX, BX, DS, ES, VAR1
```

Determine:

a) Finalidad de la macro.
b) Resultado de la expansión de la macro con el formato de la invocación señalado.
c) ¿Se puede escribir la macro de otra manera y que cumpla el mismo cometido? En caso afirmativo escriba el código de la macro, invocación y expansión.