

### LABORATORIO 3b

Para el presente laboratorio se requiere Contar con Turbo Assembler y la utilidad de DOS debug.

El presente laboratorio es una compilación de los archivos demo, presentados en la parte teórica, y están relacionados a varios capítulos que se verán progresivamente en el curso:

Parte I:

1. Ejemplo de uso de la instrucción MOV:

```
%TITLE "Demostracion del mov"

IDEAL
MODEL small
STACK 256

DATASEG
    excode      DB    0
    velocidad   DB    99

CODESEG
inicio:
    mov ax, @data
    mov ds,ax
    mov ax,1
    mov bx,2
    mov cx,3
    mov dx,4

    mov ah, [velocidad]
    mov si, offset velocidad

salir:
    mov ah, 04Ch
    mov al, [excode]
    int 21h

    END inicio
```

2. Compilalo usando Turbo assembler y enlázalo. Verifica no tener errores.
3. Explica que hace el programa.

Parte II:

1. Ejemplo de uso de las instrucciones ADD, SUB INC Y DEC:

```
%TITLE "ADD, SUB, INC, DEC DEMO"

IDEAL
MODEL small
STACK 256

DATASEG
    excode      DB    0
    count       DW    1
```

```
CODESEG
inicio:
    mov ax, @data
    mov ds,ax
    mov ax,4
    mov bx,2
    add ax, bx
    mov cx,8
    add cx, [count]
    add [count], cx

    inc [count]
    dec [count]
    inc ax
    dec ax

salir:
    mov ah, 04Ch
    mov al, [excode]
    int 21h

    END inicio
```

2. Compilarlo usando Turbo assembler y enlázalo. Verifica no tener errores.
3. Explica que hace el programa.

### Parte III:

1. Ejemplo de uso de las instrucciones AND, OR y XOR :

```
%TITLE "AND OR XOR DEMO"

IDEAL
MODEL small
STACK 256

DATASEG
    excode          DB    0
    sourceword      DW    0ABh
    wordmask        DW    0CFh

CODESEG
inicio:
    mov ax, @data
    mov ds,ax

    mov ax, [sourceword]
    mov bx, ax
    mov cx, ax
    mov dx, ax

    and ax, [wordmask]
    or  bx, [wordmask]
    xor cx, [wordmask]
    xor dx,dx
```

```
salir:
    mov ah, 04Ch
    mov al, [excode]
    int 21h

    END inicio
```

2. Compilarlo usando Turbo assembler y enlázalo. Verifica no tener errores.
3. Explica que hace el programa.

Parte IV:

1. Ejemplo de uso de la instrucción SHIFT:

```
%TITLE "SHIFT DEMO"

IDEAL
MODEL small
STACK 256

DATASEG
    excode      DB    0
    operand     DB    0AAh

CODESEG
inicio:
    mov ax, @data
    mov ds,ax

    shl [operand], 1
    shr [operand], 1
    rol [operand], 1
    ror [operand], 1
    rcl [operand], 1
    rcr [operand], 1
    sal [operand], 1
    sar [operand], 1

salir:
    mov ah, 04Ch
    mov al, [excode]
    int 21h

    END inicio
```

2. Compilarlo usando Turbo assembler y enlázalo. Verifica no tener errores.
3. Explica que hace el programa.

Parte V:

1. Ejemplo de uso de las instrucciones MUL y DIV:

```
%TITLE "MUL DIV IMUL IDIV DEMO"

IDEAL
MODEL small
STACK 256
```

```
DATASEG
    excode      DB    0
    opbyte      DB    8
    opword      DW   100
    sourcebyte  DB   64
    sourceword  DW  4000

CODESEG
inicio:
    mov ax, @data
    mov ds,ax

    mov al, [opbyte]
    mul [sourcebyte]

    mov ax, [opword]
    mul [sourceword]

    mov ax, [opword]
    mul ax

    mov ax, [opword]
    div [sourcebyte]

    mov ax, [opword]
    mov dx, [opword]
    div [sourceword]

salir:
    mov ah, 04Ch
    mov al, [excode]
    int 21h

    END inicio
```

2. Compilarlo usando Turbo assembler y enlázalo. Verifica no tener errores.
3. Explica que hace el programa.

#### Parte VI:

1. Ejemplo de uso de las instrucciones PUSH y POP:

```
%TITLE "Push Pop DEMO"

IDEAL
MODEL small
STACK 256

DATASEG
    excode      DB    0

CODESEG
inicio:
    mov ax, @data
    mov ds,ax

    push ax
    push bx
```

```
    mov ax,-1
    mov bx,-2
    mov cx, 0
    mov dx, 0

    push ax
    push bx
    pop cx
    pop dx

    pop bx
    pop ax

salir:
    mov ah, 04Ch
    mov al, [excode]
    int 21h

    END inicio
```

2. Compilarlo usando Turbo assembler y enlázalo. Verifica no tener errores.
3. Explica que hace el programa.

#### Parte VII:

1. Ejemplo de uso de subrutinas (procedures):

```
%TITLE "subrutinas DEMO"

IDEAL
MODEL small
STACK 256

DATASEG
    excode      DB      0

CODESEG

inicio:
    mov ax, @data
    mov ds,ax

    mov al, 1
    mov bl, 2
    mov cl, 3
    mov dl, 4
    call sumregs
    call sumregs
    call sumregs

salir:
    mov ah, 04Ch
    mov al, [excode]
    int 21h
```

```
-----  
; Procedimiento :Sumregs  
; entrada: al,bl,cl,dl  
; salida : ax= al+bl+cl+dl  
-----  
PROC sumregs  
    xor ah, ah  
    xor bh, bh  
    xor ch, ch  
    xor dh, dh  
    add ax, bx  
    adc ax, cx  
    adc ax, dx  
    ret  
ENDP sumregs  
    END inicio
```

2. Compilarlo usando Turbo assembler y enlázalo. Verifica no tener errores.
3. Explica que hace el programa.

**NOTA:** Los programas presentados no muestran ningún resultado en pantalla, por lo que si se quiere observar el proceso de ejecución instrucción por instrucción, se debe emplear DEBUG o TURBO DEBUGGER.