

5

NOMBRE DE LA PRÁCTICA	Instrucciones de salto Jxx			No.	UNIDAD 2
ASIGNATUR A:	LENGUAJE INTERFAZ	CARR ERA:	ISIC	PLAN:	ISIC-2010-204

Nombre: Vanesa Hernández Martínez

Grupo: 3501

Objetivo: Realizar un programa que utilice instrucciones de salto que permita determinar si un número es mayor o menor

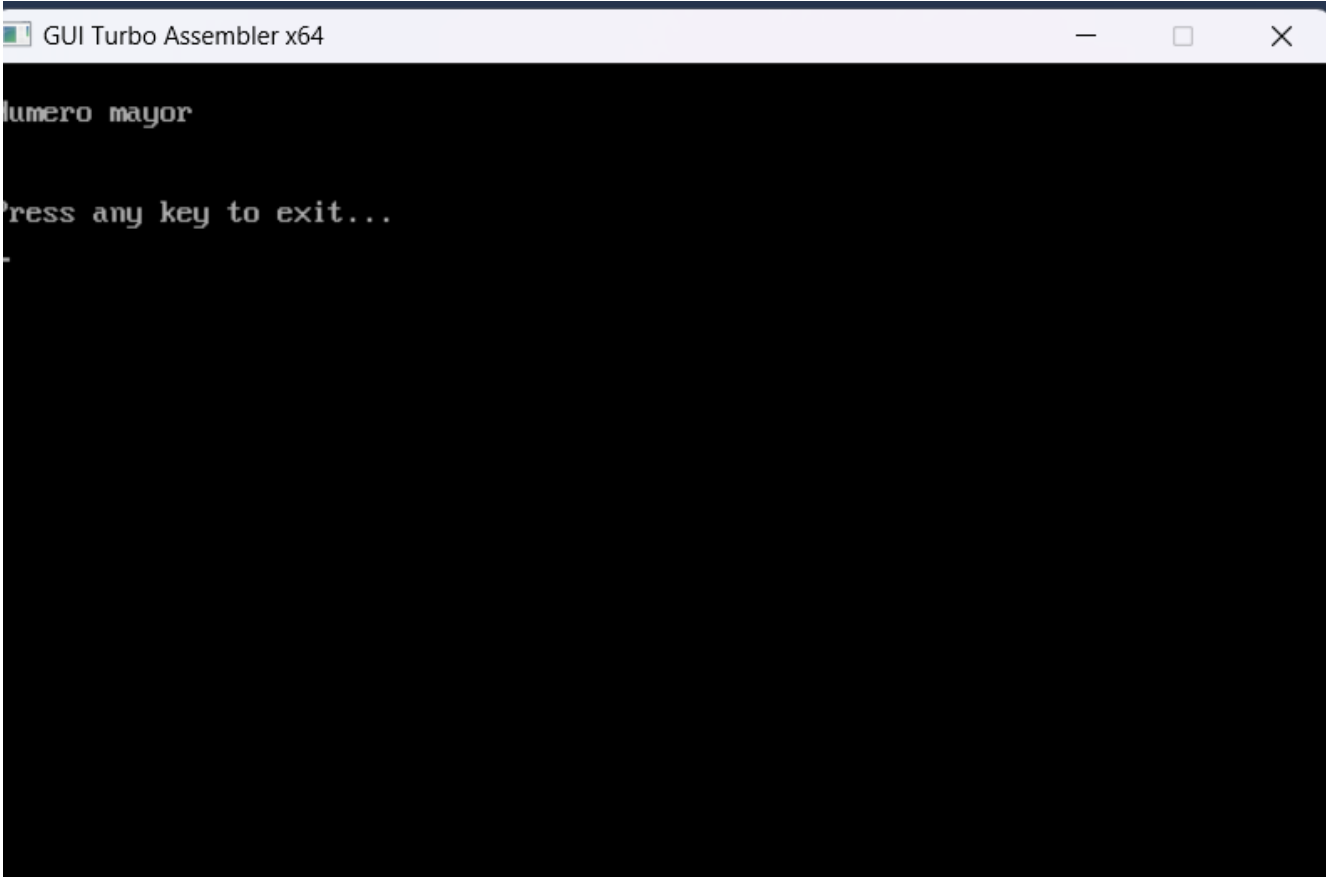
1. Realiza un programa en ensamblador que utilice la instrucción de salto apropiada para indicar que numero es el mayor. Deberán capturarse los dos números y el programa mostrar el mayor.

Anexa la captura de pantalla del código y la corrida del programa:

```

1  .model small ; se asigna el tamaño de la memoria
2  .stack ; modelo de pila
3  .data ;inicio de datos
4  x db 10
5  y db 5
6
7  mensaje db 10,13,7, "Numero menor", "$"
8  mensaje2 db 10,13,7, "Numero mayor", "$"
9  .code ; inicio del codigo
10 main proc; inicia el proceso principal
11
12 mov al,x
13 mov bl,y
14
15 cmp al,bl
16 jg mayor
17 mayor:
18 mov ax , SEG @data
19 mov ds,ax
20
21 mov ah,09h
22 lea dx,mensaje2
23 int 21h
24 jmp fin
25 menor:
26 mov ax ,SEG @data
27 mov ds,ax
28
29 mov ah,09h
30 lea dx,mensaje
31 int 21h
32
33 jmp fin
34 fin:
35 mov ax,4c00h
36 int 21h
37 main endp
38
39 end main

```



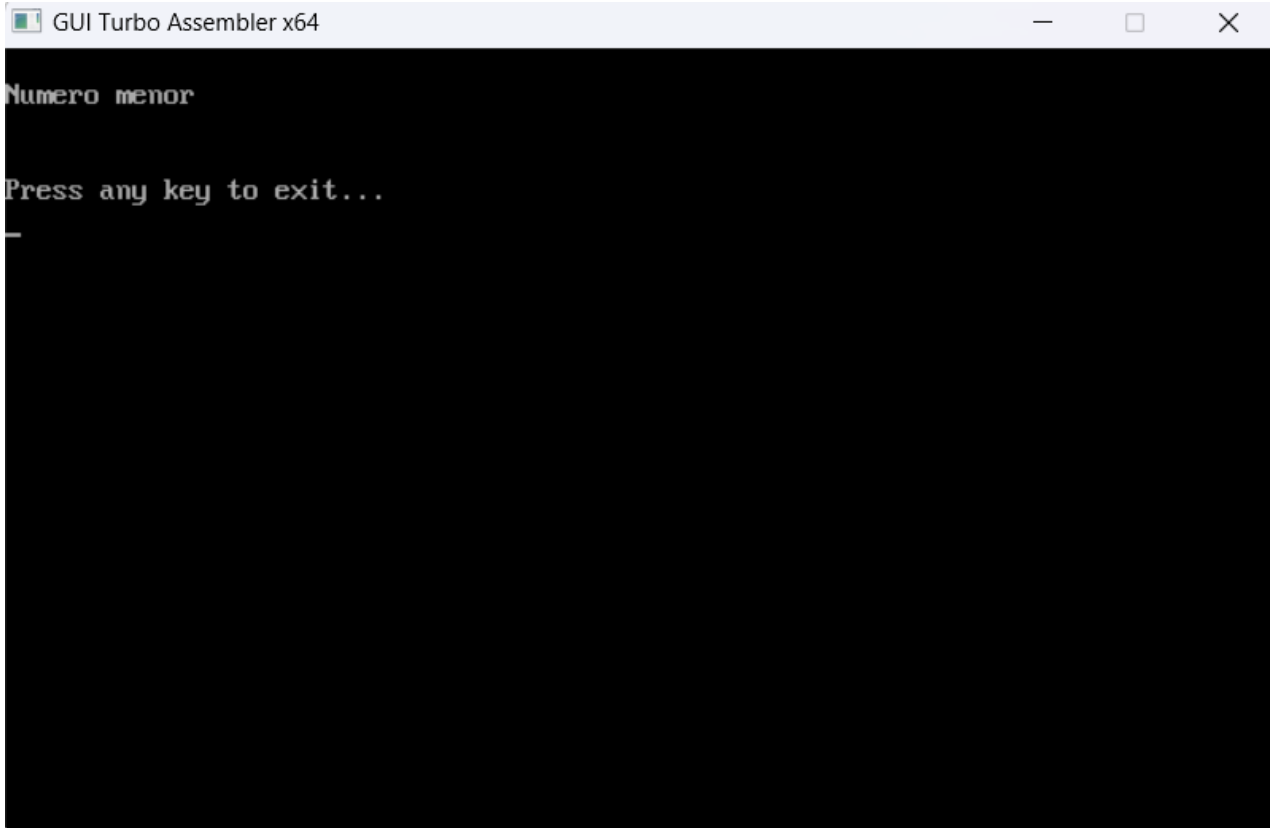
- Realiza un programa en ensamblador que utilice la instrucción de salto apropiada para indicar que número es el menor. Deberán capturarse los dos números y el programa mostrar el menor.

Anexa la captura de pantalla del código y la corrida del programa:

```

1  .model small    ; se asigna el tamaño de la memoria
2  .stack         ; modelo de pila
3  .data         ; inicio de datos
4  x db 10
5  y db 5
6
7  mensaje db 10,13,7, "Numero menor", "$"
8  mensaje2 db 10,13,7, "Numero mayor", "$"
9  .code         ; inicio del código
10 main proc     ; inicia el proceso principal
11
12 mov al,x      ; mueve el valor de x al registro AL
13 mov bl,y      ; mueve el valor de y al registro BL
14
15 cmp al,bl     ; compara AL (x) con BL (y)
16 jl menor      ; si x < y, salta a menor
17 ; Si no es menor, va a la sección mayor
18 mayor:
19 mov ax, SEG @data
20 mov ds,ax
21
22 mov ah,09h    ; muestra el mensaje de "Numero mayor"
23 lea dx,mensaje2
24 int 21h
25 jmp fin       ; salta al final después de mostrar el mensaje
26 menor:
27 mov ax, SEG @data
28 mov ds,ax
29
30 mov ah,09h    ; muestra el mensaje de "Numero menor"
31 lea dx,mensaje
32 int 21h
33
34 jmp fin       ; salta al final
35 fin:
36 mov ax,4c00h  ; termina el programa
37 int 21h
38 main endp
39
40 end main
41

```



CONCLUSIONES

La modificación del código en ensamblador presentada demuestra la importancia de entender y manipular las comparaciones condicionales en ensamblador para controlar el flujo de ejecución. Inicialmente, el código estaba diseñado para imprimir cuál de dos números es mayor; sin embargo, con ajustes simples en las instrucciones de comparación y las etiquetas de salto, el comportamiento se invirtió para identificar el número menor.

Estos cambios consisten principalmente en usar la instrucción `jl` (jump if less) para comparar los registros y dirigir el flujo de ejecución hacia la sección que imprime el mensaje correspondiente al número menor. Esta modificación no solo refuerza la capacidad de implementar comparaciones lógicas en ensamblador, sino que también evidencia la flexibilidad y precisión que ofrece este lenguaje de bajo nivel para controlar el flujo del programa.