

5

| NOMBRE DE LA PRÁCTICA | SUMA DE DOS NÚMEROS CON UN DÍGITO |           |      | No.   | UNIDAD 1      |
|-----------------------|-----------------------------------|-----------|------|-------|---------------|
| ASIGNATURA:           | LENGUAJE INTERFAZ                 | CARRER A: | ISIC | PLAN: | ISIC-2010-204 |

Nombre: Vanesa Hernández Martínez

Grupo: 3501

Objetivo: Desplegar la suma de dos números con un dígito y su resultado sea también de un solo dígito.

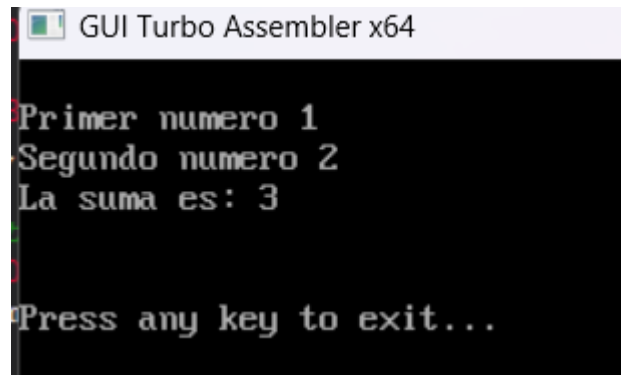
Utilizando los registros acumuladores y de datos, elabora un programa en ensamblador que permita desplegar la suma de dos números de un dígito cada uno y el resultado también sea de un solo dígito. Indica lo que realiza cada renglón.

```

1 .model small           ; Define el modelo de memoria como "small", que permite que el código y los datos se almacenen en un único segmento.
2 .stack                ; Define la pila para almacenar direcciones y datos temporales.
3 .data                ; Inicia la sección de datos para definir variables y mensajes.
4     num1 db 0          ; Define un byte (db) para almacenar el primer número ingresado.
5     num2 db 0          ; Define un byte para almacenar el segundo número ingresado.
6     num3 db 0          ; Define un byte para almacenar el resultado de la suma.
7
8     ; Mensajes que se mostrarán en la pantalla.
9     mensaje db 10,13,7,"Primer número ","$" ; Mensaje que se muestra para solicitar el primer número.
10    mensaje2 db 10,13,7,"Segundo número ","$" ; Mensaje para solicitar el segundo número.
11    mensaje3 db 10,13,7,"La suma es: ","$" ; Mensaje que se muestra antes de mostrar el resultado de la suma.
12
13 .code                ; Inicia la sección de código.
14 main proc           ; Define el procedimiento principal.
15     mov ax, SEG 8data ; Carga la dirección del segmento de datos en AX.
16     mov ds, ax       ; Mueve el valor de AX al registro DS para establecer el segmento de datos.
17
18     ; Solicitar primer número
19     mov ah, 09h       ; Función DOS para mostrar una cadena (interrupción 21h, función 09h).
20     lea dx, mensaje   ; Carga la dirección del mensaje en el registro DX.
21     int 21h           ; Llamada a la interrupción 21h para mostrar el mensaje.
22
23     ; Leer el primer número ingresado por el usuario
24     mov ah, 01h       ; Función DOS para leer un carácter del teclado (interrupción 21h, función 01h).
25     int 21h           ; Llamada a la interrupción 21h para leer el carácter.
26     sub al, 30h       ; Convierte el carácter ASCII ingresado (que está en AL) a un valor numérico restando 30h (ASCII de '0').
27     mov num1, al      ; Guarda el número ingresado en la variable 'num1'.
28
29     ; Solicitar segundo número
30     mov ah, 09h       ; Función DOS para mostrar una cadena.
31     lea dx, mensaje2  ; Carga la dirección del mensaje2 en DX.
32     int 21h           ; Llamada a la interrupción 21h para mostrar el mensaje.
33
34     ; Leer el segundo número ingresado por el usuario
35     mov ah, 01h       ; Función DOS para leer un carácter del teclado.
36     int 21h           ; Llamada a la interrupción 21h para leer el carácter.
37     sub al, 30h       ; Convierte el carácter ASCII a número restando 30h.
38     mov num2, al      ; Guarda el número ingresado en la variable 'num2'.
39
40     ; Sumar los dos números
41     mov al, num1       ; Cargar el primer número en AL.
42     add al, num2       ; Sumar el segundo número (num2) a AL.
43     add al, 30h        ; Convertir el resultado numérico de nuevo a ASCII para poder mostrarlo.
44     mov num3, al       ; Guardar el resultado (en formato ASCII) en 'num3'.
45
46     ; Mostrar el mensaje de suma
47     mov ah, 09h       ; Función DOS para mostrar una cadena.
48     lea dx, mensaje3  ; Carga la dirección del mensaje3 en DX.
49     int 21h           ; Llamada a la interrupción 21h para mostrar el mensaje.
50
51     ; Mostrar el resultado de la suma
52     mov ah, 02h       ; Función DOS para mostrar un solo carácter (interrupción 21h, función 02h).
53     mov dl, num3       ; Cargar el resultado de la suma (num3) en DL.
54     int 21h           ; Llamada a la interrupción 21h para mostrar el carácter (el resultado de la suma).
55
56     ; Terminar el programa
57     mov ax, 4c00h     ; Función DOS para terminar el programa (interrupción 21h, función 4Ch).
58     int 21h           ; Llamada a la interrupción 21h para salir del programa.
59
60 main endp            ; Fin del procedimiento principal.
61 end main              ; Fin del programa.

```

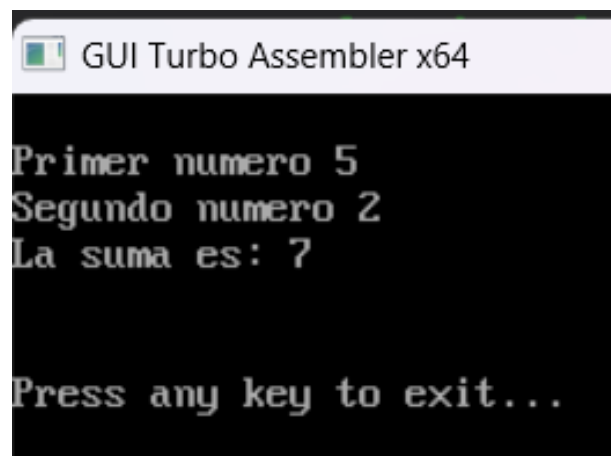
Escribe las instrucciones y captura de pantalla que demuestre que el programa si corrió:



```
GUI Turbo Assembler x64

Primer numero 1
Segundo numero 2
La suma es: 3

Press any key to exit...
```



```
GUI Turbo Assembler x64

Primer numero 5
Segundo numero 2
La suma es: 7

Press any key to exit...
```

### ¿Cómo utilizaste los registros acumuladores y de datos para realizar el programa de suma?

En el programa, se utilizan principalmente dos registros acumuladores para realizar la suma y manejar los datos:

- **Registro AL (parte baja del acumulador AX):** Este registro se utiliza para cargar, manipular y almacenar los valores ingresados por el usuario y para realizar las operaciones aritméticas. Primero se guarda el número ingresado por el usuario (después de convertirlo de carácter ASCII a número) en el registro **AL**, y luego, después de sumar ambos números, el resultado se guarda de nuevo en **AL**.
- **Registro DL (parte baja del registro DX):** Este registro se utiliza para almacenar el carácter (el resultado de la suma) que se va a mostrar al usuario. El valor de **AL** se transfiere a **DL** para que la interrupción DOS (función 02h) pueda mostrarlo en pantalla.

Los registros **AH** y **DX** también se utilizan para las funciones de interrupción de DOS, como la lectura de caracteres del teclado y la visualización de mensajes en la pantalla.

### ¿Por qué no se puede obtener un resultado de dos dígitos cuando la operación rebasa las unidades?

El programa está diseñado para manejar la suma de dos números de un solo dígito y mostrar únicamente resultados que también sean de un solo dígito. Si la suma de ambos números supera el valor de 9 (como  $5 + 6 = 11$ ), el programa no está configurado para descomponer ese valor en dos dígitos (unidades y decenas). El resultado se almacenaría en **AL**, pero no se maneja cómo tratar más de un dígito (un valor mayor que 9) al mostrarlo. Al agregar 30h para convertir el valor numérico a su representación ASCII, solo el dígito de las unidades (entre 0 y 9) es mostrado correctamente. Si el valor resultante es mayor de 9, el programa mostraría caracteres incorrectos porque no está preparado para manejar dígitos adicionales, ni para dividir el valor en dos caracteres (decenas y unidades).

### ¿Si se requieren más dígitos, cómo realizarías la suma? Explica.

Si se requiere que el programa maneje y muestre resultados de más de un dígito (por ejemplo, sumas mayores a 9), se deben realizar las siguientes modificaciones:

1. **Descomposición del resultado en decenas y unidades:** Una vez realizada la suma, si el resultado es mayor que 9, se puede dividir el número en dos partes: el dígito de las decenas y el de las unidades.
  - Dividir el resultado por 10 para obtener el dígito de las decenas.
  - Obtener el residuo de la división para obtener el dígito de las unidades.
2. **Conversión a caracteres ASCII:** Convertir ambos dígitos (decenas y unidades) a sus representaciones ASCII, sumando 30h a cada uno para que puedan ser mostrados correctamente en pantalla.
3. **Mostrar los dígitos en pantalla:** Mostrar los dígitos de las decenas y unidades uno por uno en la pantalla, utilizando interrupciones de DOS.

## **Conclusión**

El programa original, al estar limitado a resultados de un solo dígito, no puede manejar sumas que excedan el valor de 9. Para corregir esto, es necesario dividir el resultado en decenas y unidades. Esto se logra dividiendo el valor por 10, almacenando las decenas en un registro y el residuo en otro. Posteriormente, ambos se convierten a su representación ASCII para ser mostrados. Esta modificación permite que el programa procese y muestre correctamente sumas con resultados de dos dígitos o más, haciendo que el programa sea más robusto y funcional en operaciones aritméticas.