

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: Jesús Navarrete 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.

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

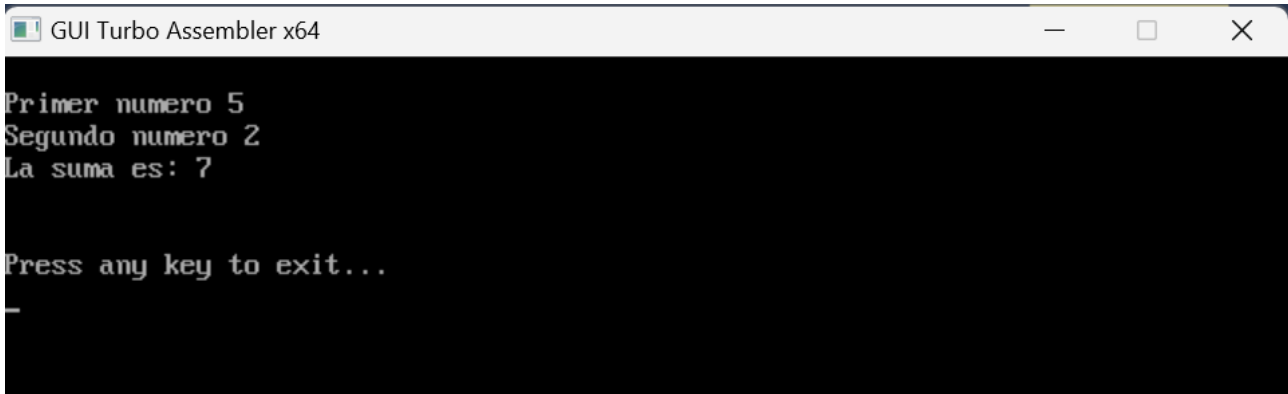
Código Fuente Escrito

```

1  .model small          ; Define un modelo de memoria pequeño
2  .stack               ; Define una pila (stack) para el programa
3  .data               ; Sección de datos del programa
4  num1 db 0           ; Variable para almacenar el primer número ingresado
5  num2 db 0           ; Variable para almacenar el segundo número ingresado
6  num3 db 0           ; Variable para almacenar el resultado de la suma
7
8  mensaje db 10,13,7, "Primer numero ", "$" ; Mensaje para solicitar el primer número
9  mensaje2 db 10,13,7, "Segundo numero ", "$" ; Mensaje para solicitar el segundo número
10 mensaje3 db 10,13,7, "La suma es: ", "$" ; Mensaje para mostrar el resultado de la suma
11
12 .code               ; Sección de código
13 main proc
14     mov ax, SEG @data ; Mueve el segmento de datos a AX
15     mov ds, ax        ; Carga el segmento de datos en DS para acceder a las variables
16
17     ; Solicitar primer número
18     mov ah, 09h       ; Función 09h de la interrupción 21h: mostrar cadena de caracteres
19     lea dx, mensaje   ; Carga la dirección del mensaje en DX
20     int 21h           ; Llamada a la interrupción para mostrar el mensaje
21
22     mov ah, 01h       ; Función 01h de la interrupción 21h: leer un carácter
23     int 21h           ; Leer el carácter ingresado por el usuario
24     sub al, 30h       ; Convierte el valor ASCII a su valor numérico restando 30h
25     mov num1, al      ; Almacena el primer número en la variable 'num1'
26
27     ; Solicitar segundo número
28     mov ah, 09h       ; Función 09h para mostrar el segundo mensaje
29     lea dx, mensaje2  ; Carga la dirección del segundo mensaje en DX
30     int 21h           ; Llamada a la interrupción para mostrar el segundo mensaje
31
32     mov ah, 01h       ; Función 01h para leer el segundo carácter ingresado
33     int 21h           ; Leer el segundo carácter ingresado por el usuario
34     sub al, 30h       ; Convierte el valor ASCII a su valor numérico
35     mov num2, al      ; Almacena el segundo número en la variable 'num2'
36
37     ; Sumar los dos números
38     mov al, num1      ; Carga el primer número en AL
39     add al, num2       ; Suma el segundo número al valor en AL
40     add al, 30h       ; Convierte el resultado de nuevo a su valor ASCII
41     mov num3, al      ; Almacena el resultado en la variable 'num3'
42
43     ; Mostrar el mensaje de suma
44     mov ah, 09h       ; Función 09h para mostrar el mensaje de la suma
45     lea dx, mensaje3  ; Carga la dirección del mensaje en DX
46     int 21h           ; Llamada a la interrupción para mostrar el mensaje
47     ; Mostrar el resultado
48     mov ah, 02h       ; Función 02h para desplegar un carácter en pantalla
49     mov dl, num3       ; Carga el resultado de la suma en DL
50     int 21h           ; Llamada a la interrupción para mostrar el resultado
51
52     ; Terminar el programa
53     mov ax, 4c00h     ; Función 4Ch de la interrupción 21h: terminar el programa
54     int 21h           ; Llamada a la interrupción para terminar la ejecución
55 main endp
56 end main

```

Ejecución y compilación del código



```
GUI Turbo Assembler x64

Primer numero 5
Segundo numero 2
La suma es: 7

Press any key to exit...
```

CUESTIONARIO:

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

En el programa de suma, se utilizan los registros acumuladores y de datos para manejar los números y realizar la operación:

- **AL:** El registro acumulador AL se usa para almacenar los números ingresados (primero num1 y luego num2) y realizar las operaciones aritméticas. Se cargan los números en AL para realizar la suma y luego se convierten de vuelta a su valor ASCII para mostrar el resultado.
- **DL:** El registro de datos DL se usa para mostrar el resultado final. Después de convertir el valor numérico a ASCII, se carga el carácter resultante en DL y se utiliza la interrupción int 21h para desplegarlo.

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

Esto ocurre porque el programa está diseñado para trabajar con números de un solo dígito, y solo se despliega un carácter en pantalla. Cuando la suma de dos dígitos excede 9 (por ejemplo, $7 + 5 = 12$), el resultado es un valor que corresponde a un solo carácter ASCII, por lo que solo se despliega el último dígito (en este caso, el '2'). El programa no está preparado para manejar la descomposición del resultado en decenas y unidades ni para mostrar múltiples caracteres en ese caso.

3. ¿Si se requieren más dígitos, como realizarías la suma? Explica.

Si se requieren más dígitos, deberías modificar el programa para descomponer el resultado en decenas y unidades. El enfoque sería el siguiente:

- Almacenar la suma completa en un registro (por ejemplo, AX).
- Separar las decenas y unidades:
 - Dividir el resultado entre 10 para obtener las decenas. Esto se puede hacer usando la instrucción DIV, que dividirá el valor por 10 y guardará el cociente (decenas) en AH y el residuo (unidades) en AL.
- Mostrar las decenas y unidades por separado:
 - Primero, se agrega 30h a AH (las decenas) para convertirlo a ASCII y mostrarlo.
 - Después, se hace lo mismo con AL (las unidades) para desplegar ambos dígitos.

CONCLUSIONES:

En esta práctica de programación en ensamblador, se ha demostrado cómo realizar la suma de dos números de un dígito utilizando los registros acumuladores y de datos, y cómo manipular valores en su formato ASCII para mostrar el resultado en pantalla. Este ejercicio resalta la importancia del control detallado que se tiene en ensamblador al trabajar directamente con los registros del procesador, lo que permite una manipulación precisa de los datos.

A pesar de la simplicidad del programa, se evidencia que en ensamblador cada paso debe ser explícitamente definido, desde la conversión de caracteres ASCII a sus valores numéricos, hasta la gestión de la entrada y salida de datos mediante interrupciones. Además, se identificó la limitación de manejar resultados de un solo dígito, lo que muestra la necesidad de implementar técnicas adicionales para trabajar con operaciones que produzcan resultados de más de un dígito.

Esta práctica no solo fortalece la comprensión del uso de registros en operaciones aritméticas básicas, sino que también fomenta una visión más profunda de cómo extender el programa para manejar situaciones más complejas, como la suma de números de múltiples dígitos.