
 GOBIERNO DEL ESTADO DE MÉXICO	<h1 style="text-align: center;">RETO</h1> <p style="text-align: center;">Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales</p>	 TECNOLOGICO DE ESTUDIOS SUPERIORES JILOTEPEC
---	--	--

5

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

Nombre: Fabiola Castañeda Mondragón

Grupo: 3501

Objetivo: Desplegar la suma de dos números con dos dígitos y su resultado sea también de dos dígitos también.

Elabora un programa en ensamblador que realice la suma de dos números con dos dígitos y el resultado también sea de dos dígitos. Explica lo que realiza cada renglón. Escribe las instrucciones y captura de pantalla que demuestre que el programa si corrió:

```



1  .MODEL SMALL                ; Define el modelo de memoria como pequeño
2  .STACK 100h                ; Reserva 256 bytes para la pila
3
4  .DATA                      ; Inicio del segmento de datos
5      num1 db 0               ; Variable para almacenar el primer número
6      num2 db 0               ; Variable para almacenar el segundo número
7      resultado db 0          ; Variable para almacenar el resultado de la suma
8      mensaje db 10, 13, "Primer número (dos dígitos): $"; Mensaje para solicitar el primer número
9      mensaje2 db 10, 13, "Segundo número (dos dígitos): $"; Mensaje para solicitar el segundo número
10     mensaje3 db 10, 13, "La suma es: $"; Mensaje para mostrar la suma
11     nueva_linea db 10, 13, "$"; Nueva línea para mejorar la visualización
12
13 .CODE                      ; Inicio del segmento de código
14 main proc                  ; Inicio del procedimiento principal
15     mov ax, @data           ; Cargar la dirección del segmento de datos en AX
16     mov ds, ax              ; Mover el valor de AX a DS para acceder a las variables
17
18     ; Solicitar el primer número de dos dígitos
19     mov ah, 09h              ; Función 09H para imprimir cadenas
20     lea dx, mensaje          ; Cargar la dirección del mensaje en DX
21     int 21h                 ; Llamar a la interrupción para mostrar el mensaje
22
23     ; Leer el primer dígito del primer número
24     mov ah, 01h              ; Función 01H para leer un carácter del teclado
25     int 21h                 ; Interrupción para obtener el carácter
26     sub al, 30h              ; Convertir carácter ASCII a número (0-9)
27     mov bl, al               ; Guardar el primer dígito en BL
28
29     ; Leer el segundo dígito del primer número
30     mov ah, 01h              ; Función 01H para leer el siguiente carácter
31     int 21h                 ; Interrupción para obtener el carácter
32     sub al, 30h              ; Convertir carácter ASCII a número
33     mov bh, al               ; Guardar el segundo dígito en BH
34
35     ; Calcular num1 = (primer dígito * 10) + segundo dígito
36     mov al, bl               ; Mover el primer dígito a AL
37     mov ah, 0                ; Limpiar AH para que AX tenga solo el primer dígito
38     mov cl, 10               ; Cargar 10 en CL para multiplicar
39     mul cl                   ; Multiplicar el primer dígito por 10
40     add al, bh                ; Sumar el segundo dígito
41     mov num1, al             ; Guardar el número final en num1
42
43     ; Solicitar el segundo número de dos dígitos

```

```

44 mov ah, 09h          ; Funci?n 09H para imprimir cadenas
45 lea dx, mensaje2     ; Cargar la direcci?n del mensaje2 en DX
46 int 21h              ; Llamar a la interrupci?n para mostrar el mensaje
47
48 ; Leer el primer d?gito del segundo n?mero
49 mov ah, 01h          ; Funci?n 01H para leer un car?cter
50 int 21h              ; Interrupci?n para obtener el car?cter
51 sub al, 30h          ; Convertir car?cter ASCII a n?mero
52 mov bl, al            ; Guardar el primer d?gito en BL
53
54 ; Leer el segundo d?gito del segundo n?mero
55 mov ah, 01h          ; Funci?n 01H para leer el siguiente car?cter
56 int 21h              ; Interrupci?n para obtener el car?cter
57 sub al, 30h          ; Convertir car?cter ASCII a n?mero
58 mov bh, al            ; Guardar el segundo d?gito en BH
59
60 ; Calcular num2 = (primer d?gito * 10) + segundo d?gito
61 mov al, bl            ; Mover el primer d?gito a AL
62 mov ah, 0             ; Limpiar AH
63 mul cl                ; Multiplicar por 10
64 add al, bh            ; Sumar el segundo d?gito
65 mov num2, al          ; Guardar el n?mero final en num2
66
67 ; Sumar num1 y num2
68 mov al, num1          ; Cargar num1 en AL
69 add al, num2          ; Sumar num2 a AL
70 mov resultado, al     ; Guardar la suma en resultado
71
72 ; Mostrar el mensaje de la suma
73 mov ah, 09h          ; Funci?n 09H para imprimir cadenas
74 lea dx, mensaje3     ; Cargar la direcci?n del mensaje de resultado en DX
75 int 21h              ; Mostrar el mensaje en pantalla
76
77 ; Preparar para mostrar el resultado
78 mov al, resultado     ; Cargar resultado en AL
79 xor ah, ah            ; Limpiar AH para que AX tenga solo el resultado
80 cmp al, 0             ; Verificar si el resultado es cero
81 je mostrar_cero       ; Si es cero, ir a mostrar_cero
82
83 ; Convertir a caracteres ASCII
84 mov bx, 10            ; Divisor
85 xor cx, cx            ; Limpiar CX para contar los d?gitos
86
87 convertir:
88 xor dx, dx            ; Limpiar DX antes de dividir
89 div bx                ; Dividir AX entre 10
90 push dx               ; Guardar el residuo (d?gito)
91 inc cx                ; Incrementar el contador de d?gitos
92 test ax, ax           ; Verificar si AX es cero
93 jnz convertir         ; Repetir hasta que AX sea cero
94
95 ; Mostrar los d?gitos en orden
96 mostrar_digitos:
97 pop dx                ; Obtener el d?gito
98 add dl, 30h           ; Convertir a ASCII
99 mov ah, 02h          ; Funci?n 02H para mostrar un car?cter
100 int 21h              ; Mostrar d?gito
101 loop mostrar_digitos  ; Repetir para todos los d?gitos
102
103 jmp fin               ; Ir a la secci?n de finalizaci?n
104
105 mostrar_cero:
106 ; Mostrar "0"
107 mov dl, '0'           ; Cargar el car?cter '0'
108 mov ah, 02h          ; Funci?n 02H para mostrar un car?cter
109 int 21h              ; Mostrar el car?cter '0'
110
111 fin:
112 ; Nueva l?nea
113 mov ah, 09h          ; Funci?n 09H para imprimir cadenas
114 lea dx, nueva_linea  ; Cargar la direcci?n de nueva l?nea en DX
115 int 21h              ; Mostrar nueva l?nea
116
117 ; Terminar el programa
118 mov ax, 4C00h         ; Funci?n 4CH para salir
119 int 21h              ; Llamar a la interrupci?n para salir
120
121 main endp             ; Fin del procedimiento principal
122 end main              ; Fin del programa
123

```

 GOBIERNO DEL ESTADO DE MÉXICO	<h1>RETO</h1> <p>Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales</p>	 TESI TECNOLOGICO DE ESTUDIOS SUPERIORES JILOTEPEC
---	--	--

Código ejecutado:

```

GUI Turbo Assembler x64
Run the executable (Alt+R)

Primer n?mero (dos d?gitos): 44
Segundo n?mero (dos d?gitos): 22
La suma es: 66

Press any key to exit...

```

## CONCLUSIONES:

Implementar la suma de dos números de dos dígitos en ensamblador no solo refuerza los conceptos de aritmética básica, sino que también proporciona una valiosa experiencia en el manejo de registros, operaciones de entrada/salida y control de flujo. Esto ayuda a los programadores a desarrollar habilidades críticas para abordar problemas más complejos en sistemas embebidos y desarrollo de software a bajo nivel. Además, el aprendizaje de ensamblador fomenta una mejor comprensión de cómo se ejecutan las operaciones a nivel de máquina, lo que es esencial para optimizar el rendimiento de las aplicaciones.