

5

NOMBRE DE LA PRÁCTICA	Operaciones Aritméticas			No.	UNIDAD 2
ASIGNATURA:	LENGUAJE INTERFAZ	CARRER A:	ISIC	PLAN:	ISIC-2010-204

Nombre: Jesús Navarrete Martínez

Grupo: 3501

Objetivo: Realizar un programa que realice operaciones aritméticas de suma, resta, multiplicación, división y módulo.

1. Realiza un programa en ensamblador que efectúe la suma de dos números con un dígito.

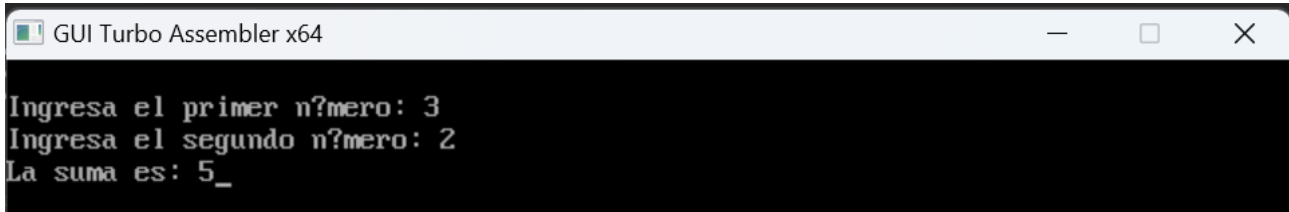
Código del programa del programa:

```

1  .model small
2  .stack 64
3  .data
4  numero1 db 0
5  numero2 db 0
6  suma db 0
7  mensaje1 db 10,13, "Ingresa el primer número: $"
8  mensaje2 db 10,13, "Ingresa el segundo número: $"
9  mensaje3 db 10,13, "La suma es: $"
10 .code
11 begin proc far
12     mov ax, @data
13     mov ds, ax
14     ; Primer número
15     mov ah, 09h
16     lea dx, mensaje1
17     int 21h
18     mov ah, 01
19     int 21h
20     sub al, 30h
21     mov numero1, al
22     ; Segundo número
23     mov ah, 09h
24     lea dx, mensaje2
25     int 21h
26     mov ah, 01
27     int 21h
28     sub al, 30h
29     mov numero2, al
30     ; Suma
31     mov al, numero1
32     add al, numero2
33     mov suma, al
34     ; Mostrar suma
35     mov ah, 09h
36     lea dx, mensaje3
37     int 21h
38     mov al, suma
39     add al, 30h
40     mov dl, al
41     mov ah, 02
42     int 21h
43     ret
44 begin endp
45 end begin
46

```

Ejecución del programa:



- Realiza un programa en ensamblador que efectúe la resta de dos números con un dígito

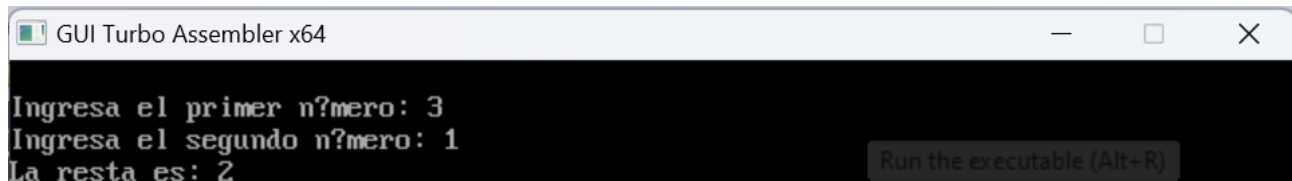
Código del programa:

```

1  .model small
2  .stack 64
3  .data
4  numero1 db 0
5  numero2 db 0
6  resta db 0
7  mensaje1 db 10,13, "Ingresa el primer número: $"
8  mensaje2 db 10,13, "Ingresa el segundo número: $"
9  mensaje4 db 10,13, "La resta es: $"
10 .code
11 begin proc far
12     mov ax, @data
13     mov ds, ax
14
15     ; Primer número
16     mov ah, 09h
17     lea dx, mensaje1
18     int 21h
19     mov ah, 01
20     int 21h
21     sub al, 30h
22     mov numero1, al
23
24     ; Segundo número
25     mov ah, 09h
26     lea dx, mensaje2
27     int 21h
28     mov ah, 01
29     int 21h
30     sub al, 30h
31     mov numero2, al
32
33     ; Resta
34     mov al, numero1
35     sub al, numero2
36     mov resta, al
37
38     ; Mostrar resta
39     mov ah, 09h
40     lea dx, mensaje4
41     int 21h
42     mov al, resta
43     add al, 30h
44     mov dl, al
45     mov ah, 02
46     int 21h
47
48     ret
49 begin endp
50 end begin

```

Ejecución del programa:



- Realiza un programa en ensamblador que efectúe la multiplicación, división y modulo de dos números con un dígito.

Código del programa:

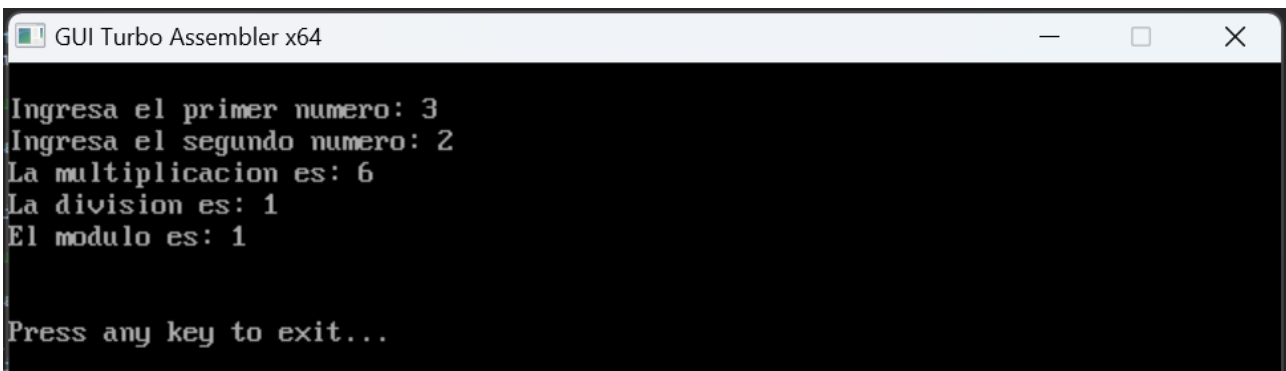
```

1  .model small
2  .stack 64
3  .data
4  numero1 db 0
5  numero2 db 0
6  mult dw 0 ; Para multiplicaci?n de 16 bits
7  divi db 0
8  modu db 0
9  mensaje1 db 10,13, "Ingresa el primer numero: $"
10 mensaje2 db 10,13, "Ingresa el segundo numero: $"
11 mensaje3 db 10,13, "La multiplicacion es: $"
12 mensaje4 db 10,13, "La division es: $"
13 mensaje5 db 10,13, "El modulo es: $"
14 buffer db 6, '$' ; Buffer para imprimir los n?meros
15 .code
16 start:
17 ; Inicializaci?n del segmento de datos
18 mov ax, @data
19 mov ds, ax
20 ; Mostrar mensaje1
21 mov ah, 09h
22 lea dx, mensaje1
23 int 21h
24 ; Leer el primer n?mero
25 mov ah, 01h
26 int 21h
27 sub al, 30h ; Convertir car?cter ASCII a n?mero
28 mov numero1, al
29 ; Mostrar mensaje2
30 mov ah, 09h
31 lea dx, mensaje2
32 int 21h
33 ; Leer el segundo n?mero
34 mov ah, 01h
35 int 21h
36 sub al, 30h ; Convertir car?cter ASCII a n?mero
37 mov numero2, al
38 ; Multiplicaci?n
39 mov al, numero1
40 mov bl, numero2
41 mul bl ; Multiplicaci?n de AL por BL, resultado en AX
42 mov [mult], ax ; Guardar el resultado de la multiplicaci?n
43 ; Divisi?n
44 mov al, numero1
45 xor ah, ah ; Limpiar AH para la divisi?n
46 mov bl, numero2
47 div bl ; Divisi?n AL/BL, cociente en AL, residuo en AH
48 mov divi, al ; Guardar el cociente
49 mov modu, ah ; Guardar el residuo (m?dulo)
50 ; Mostrar mensaje3 (Multiplicaci?n)
51 mov ah, 09h
52 lea dx, mensaje3
53 int 21h
54 mov ax, [mult]
55 call print_number
56

```

```
57 ; Mostrar mensaje4 (Divisi?n)
58 mov ah, 09h
59 lea dx, mensaje4
60 int 21h
61 mov al, divi
62 call print_digit
63
64 ; Mostrar mensaje5 (M?dulo)
65 mov ah, 09h
66 lea dx, mensaje5
67 int 21h
68 mov al, modu
69 call print_digit
70
71 ; Terminar el programa
72 mov ax, 4C00h
73 int 21h
74
75 ; Rutina para imprimir un n?mero de 16 bits (almacenado en AX)
76 print_number:
77 ; Conversi?n de n?mero a caracteres ASCII
78 mov bx, 10 ; Divisor
79 xor cx, cx ; Reiniciar contador de d?gitos
80 next_digit:
81 xor dx, dx ; Limpiar DX para la divisi?n
82 div bx ; AX = AX / BX, residuo en DX, cociente en AX
83 add dl, 30h ; Convertir el residuo a ASCII
84 push dx ; Guardar el car?cter en la pila
85 inc cx ; Incrementar el contador de d?gitos
86 test ax, ax ; Si AX = 0, terminar
87 jnz next_digit
88
89 print_loop:
90 pop dx ; Recuperar el car?cter
91 mov ah, 02h ; Funci?n para mostrar car?cter
92 int 21h ; Imprimir car?cter
93 loop print_loop ; Repetir para todos los d?gitos
94 ret
95
96 ; Rutina para imprimir un n?mero de 1 d?gito (almacenado en AL)
97 print_digit:
98 add al, 30h ; Convertir a ASCII
99 mov ah, 02h ; Funci?n para mostrar car?cter
100 mov dl, al
101 int 21h ; Imprimir car?cter
102 ret
103
104 end start
```

Ejecuci3n del programa:



```
GUI Turbo Assembler x64

Ingresa el primer numero: 3
Ingresa el segundo numero: 2
La multiplicacion es: 6
La division es: 1
El modulo es: 1

Press any key to exit...
```

RETO: Realiza un programa con las operaciones anteriores, pero para dos números de dos dígitos.

Programa de suma y resta con números de dos dígitos

Código del programa:

```

1  .model small
2  .stack 64
3  .data
4
5  numerol_tens db 0
6  numerol_units db 0
7  numero2_tens db 0
8  numero2_units db 0
9  suma_tens db 0
10 suma_units db 0
11 resta_tens db 0
12 resta_units db 0
13
14 mensaje1 db 10,13, "Ingresa el primer n?mero (dos d?gitos): ", "$"
15 mensaje2 db 10,13, "Ingresa el segundo n?mero (dos d?gitos): ", "$"
16 mensaje3 db 10,13, "La suma es: ", "$"
17 mensaje4 db 10,13, "La resta es: ", "$"
18
19 .code
20 begin proc far
21
22 mov ax,@data
23 mov ds, ax
24
25 ; Primer n?mero
26 mov ah,09h
27 lea dx,mensaje1
28 int 21h
29 mov ah,01
30 int 21h
31 sub al, 30h
32 mov numerol_tens, al
33 mov ah,01
34 int 21h
35 sub al, 30h
36 mov numerol_units, al
37
38 ; Segundo n?mero
39 mov ah,09h
40 lea dx,mensaje2
41 int 21h
42 mov ah,01
43 int 21h
44 sub al, 30h
45 mov numero2_tens, al
46 mov ah,01
47 int 21h
48 sub al, 30h
49 mov numero2_units, al
50
51 ; Suma
52 mov al, numerol_tens
53 add al, numero2_tens
54 daa ; adjust for carry
55 mov suma_tens, al
56 mov al, numerol_units
57 add al, numero2_units
58 mov suma_units, al
59
60 ; Resta
61 mov al, numerol_tens
62 sub al, numero2_tens
63 jnb nc ; no carry
64 mov nc, 10h
65 mov suma_tens, al
66 mov al, numerol_units
67 sub al, numero2_units
68 jnb nc
69 mov nc, 10h
70 mov suma_units, al
71
72 ; Mensaje de resultado
73 lea dx,mensaje3
74 int 21h
75
76 ; Mensaje de resultado
77 lea dx,mensaje4
78 int 21h
79
80 end proc
81 end

```

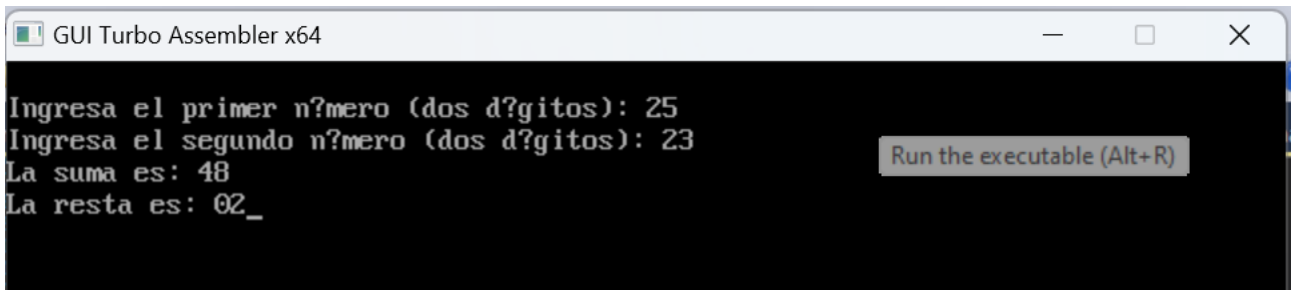
PRÁCTICA 8

Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales

5

```
57 mov al, numero1_units
58 add al, numero2_units
59 daa ; adjust for carry
60 mov suma_units, al
61 ; Resta
62 mov al, numero1_tens
63 sub al, numero2_tens
64 das ; adjust for borrow
65 mov resta_tens, al
66 mov al, numero1_units
67 sub al, numero2_units
68 das ; adjust for borrow
69 mov resta_units, al
70
71 ; Mostrar suma
72 mov ah, 09h
73 lea dx, mensaje3
74 int 21h
75 mov al, suma_tens
76 add al, 30h
77 mov dl, al
78 mov ah, 02
79 int 21h
80 mov al, suma_units
81 add al, 30h
82 mov dl, al
83 mov ah, 02
84 int 21h
85
86 ; Mostrar resta
87 mov ah, 09h
88 lea dx, mensaje4
89 int 21h
90 mov al, resta_tens
91 add al, 30h
92 mov dl, al
93 mov ah, 02
94 int 21h
95 mov al, resta_units
96 add al, 30h
97 mov dl, al
98 mov ah, 02
99 int 21h
100
101 begin endp
102 end begin
```

Ejecución del programa:



Programa de multiplicación, división y modulo con números de dos dígitos

Código del programa:



GOBIERNO DEL
ESTADO DE MÉXICO

PRÁCTICA 8

Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales



TECNOLOGICO DE ESTUDIOS SUPERIORES
JILOTEPEC

```
1  .model small
2  .stack 64
3  .data
4
5  numero1_tens db 0
6  numero1_units db 0
7  numero2_tens db 0
8  numero2_units db 0
9  numero1 dw 0      ; Almacenar el primer n?mero completo (16 bits)
10 numero2 dw 0      ; Almacenar el segundo n?mero completo (16 bits)
11 mult dw 0         ; Resultado de la multiplicaci?n (16 bits)
12 divi db 0         ; Cociente de la divisi?n (8 bits)
13 modu db 0         ; Residuo de la divisi?n (m?dulo, 8 bits)
14
15 mensaje1 db 10,13, "Ingresa el primer numero (dos d?gitos): $"
16 mensaje2 db 10,13, "Ingresa el segundo numero (dos d?gitos): $"
17 mensaje3 db 10,13, "La multiplicacion es: $"
18 mensaje4 db 10,13, "La division es: $"
19 mensaje5 db 10,13, "El modulo es: $"
20
21 buffer db 6, '$'   ; Buffer para imprimir los n?meros
22
23 .code
24 start:
25     ; Inicializaci?n del segmento de datos
26     mov ax, @data
27     mov ds, ax
28
29     ; Mostrar mensaje1
30     mov ah, 09h
31     lea dx, mensaje1
32     int 21h
33
34     ; Leer el primer n?mero (dos d?gitos)
35     mov ah, 01h
36     int 21h
37     sub al, 30h      ; Convertir decenas de ASCII a n?mero
38     mov numero1_tens, al
39     mov ah, 01h
40     int 21h
41     sub al, 30h      ; Convertir unidades de ASCII a n?mero
42     mov numero1_units, al
43
44     ; Mostrar mensaje2
45     mov ah, 09h
46     lea dx, mensaje2
47     int 21h
48
49     ; Leer el segundo n?mero (dos d?gitos)
50     mov ah, 01h
51     int 21h
52     sub al, 30h      ; Convertir decenas de ASCII a n?mero
53     mov numero2_tens, al
54     mov ah, 01h
55     int 21h
56     sub al, 30h      ; Convertir unidades de ASCII a n?mero
57
```




GOBIERNO DEL
ESTADO DE MÉXICO

PRÁCTICA 8

Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales



TECNOLOGICO DE ESTUDIOS SUPERIORES
JILOTEPEC

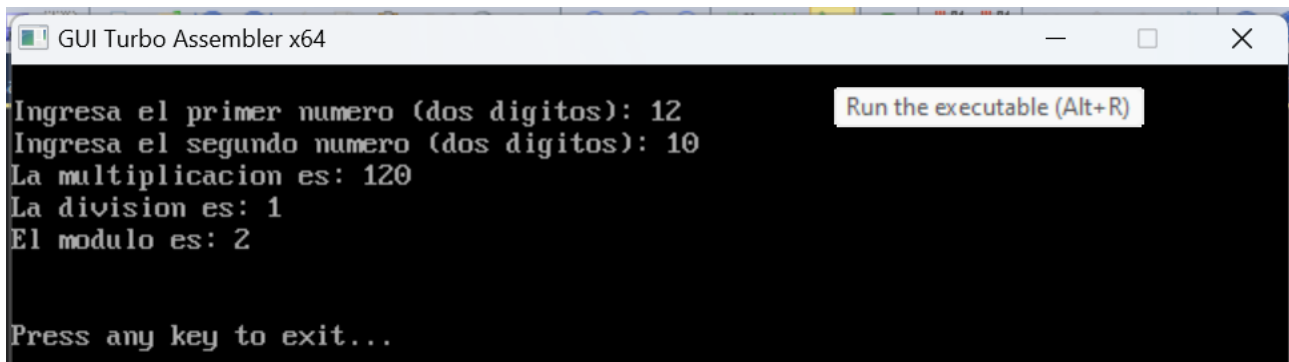
```
57     mov numero2_units, al
58
59     ; Combinar decenas y unidades para formar n?meros de 2 d?gitos
60     mov al, numero1_tens
61     mov ah, 0
62     mov bl, 10
63     mul bl           ; Multiplicar por 10
64     add al, numero1_units
65     mov numero1, ax   ; Guardar el n?mero completo en numero1
66
67     mov al, numero2_tens
68     mov ah, 0
69     mul bl           ; Multiplicar por 10
70     add al, numero2_units
71     mov numero2, ax   ; Guardar el n?mero completo en numero2
72
73     ; Multiplicaci?n
74     mov ax, numero1
75     mov bx, numero2
76     mul bx           ; Multiplicar AX por BX, resultado en DX:AX
77     mov mult, ax      ; Guardar el resultado en mult
78
79     ; Divisi?n y m?dulo
80     mov ax, numero1   ; Cargar el primer n?mero en AX
81     xor dx, dx         ; Limpiar DX para la divisi?n
82     div bx             ; AX / BX -> Cociente en AX, residuo en DX
83     mov divi, al       ; Guardar el cociente en divi
84     mov modu, dl       ; Guardar el residuo (m?dulo) en modu
85
86     ; Mostrar mensaje3 (Multiplicaci?n)
87     mov ah, 09h
88     lea dx, mensaje3
89     int 21h
90     mov ax, [mult]
91     call print_number
92
93     ; Mostrar mensaje4 (Divisi?n)
94     mov ah, 09h
95     lea dx, mensaje4
96     int 21h
97     mov al, divi
98     call print_digit
99
100    ; Mostrar mensaje5 (M?dulo)
101    mov ah, 09h
102    lea dx, mensaje5
103    int 21h
104    mov al, modu
105    call print_digit
106
107    ; Terminar el programa
108    mov ax, 4C00h
109    int 21h
110
111    ; Rutina para imprimir un n?mero de 16 bits (almacenado en AX)
112    print_number:
```

```

113 ; Conversi?n de n?mero a caracteres ASCII
114 mov bx, 10 ; Divisor
115 xor cx, cx ; Reiniciar contador de d?gitos
116 next_digit:
117 xor dx, dx ; Limpiar DX para la divisi?n
118 div bx ; AX = AX / BX, residuo en DX, cociente en AX
119 add dl, 30h ; Convertir el residuo a ASCII
120 push dx ; Guardar el car?cter en la pila
121 inc cx ; Incrementar el contador de d?gitos
122 test ax, ax ; Si AX = 0, terminar
123 jnz next_digit
124
125 print_loop:
126 pop dx ; Recuperar el car?cter
127 mov ah, 02h ; Funci?n para mostrar car?cter
128 int 21h ; Imprimir car?cter
129 loop print_loop ; Repetir para todos los d?gitos
130 ret
131
132 ; Rutina para imprimir un n?mero de 1 d?gito (almacenado en AL)
133 print_digit:
134 add al, 30h ; Convertir a ASCII
135 mov ah, 02h ; Funci?n para mostrar car?cter
136 mov dl, al
137 int 21h ; Imprimir car?cter
138 ret
139
140 end start

```

Ejecuci3n del programa:



```

GUI Turbo Assembler x64
Ingresa el primer numero (dos digitos): 12
Ingresa el segundo numero (dos digitos): 10
La multiplicacion es: 120
La division es: 1
El modulo es: 2
Press any key to exit...
Run the executable (Alt+R)

```

Conclusiones:

En estas prácticas, se desarrollaron programas en ensamblador que realizan operaciones aritméticas fundamentales sobre números de uno y dos dígitos. La implementación de la suma, resta, multiplicación, división y módulo en lenguaje ensamblador permitió profundizar en el manejo de registros y en el uso de interrupciones para interactuar con el usuario. Este enfoque ofrece un entendimiento detallado de la arquitectura del sistema y el procesamiento de datos a bajo nivel.

Cada programa siguió un flujo de trabajo estructurado, incluyendo:

1. Solicitud de entrada al usuario para ambos operandos.
2. Ejecución de la operación aritmética correspondiente.
3. Manejo de los resultados y su salida en pantalla, facilitando la interpretación de cada cálculo.

Por último, el reto de extender las operaciones a números de dos dígitos implicó un desafío adicional en el manejo de las entradas y cálculos, reforzando conceptos clave de ensamblador como el almacenamiento en registros y la manipulación de datos de varios bytes. Estas prácticas mejoraron la comprensión del control a nivel de bits y bytes en un entorno de bajo nivel, una habilidad crucial para el desarrollo y optimización en sistemas de propósito específico o embebidos.