

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

Nombre: Jesus Navarrete Martinez
Grupo: 3501

Objetivo: Desplegar números que contengan dos dígitos

Utilizando los registros acumuladores y de datos, elabora un programa en ensamblador que permita desplegar dos números de dos dígitos cada uno. 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
2 .stack
3 .data
4
5 u db 0          ; Variable para almacenar la unidad del número ingresado
6 d db 0          ; Variable para almacenar la decena del número ingresado
7 n db 0          ; Variable para almacenar el número total calculado
8
9 mensaje db 10,13,7,"Ingrese un numero: ","$" ; Mensaje que solicite un número
10 mensaje2 db 10,13,1,"numero ingresado: ","$" ; Mensaje para mostrar el número ingresado
11
12 .code
13
14 main proc
15     mov ax, SEG @data ; Mueve el segmento de datos a AX
16     mov ds, ax        ; Mueve el segmento de datos a DS para acceder a las variables
17
18     mov ah, 09h       ; Cargar la función 09h de la interrupción 21h para mostrar el mensaje
19     lea dx, mensaje    ; Cargar la dirección del mensaje en DX
20     int 21h           ; Llamada a la interrupción para mostrar el mensaje
21
22     mov ah, 01h       ; Cargar la función 01h de la interrupción 21h para leer un carácter
23     int 21h           ; Leer el primer carácter ingresado (decenas)
24     sub al, 30h        ; Restar 30h para convertir de ASCII a su valor numérico
25     mov d, al          ; Almacenar el valor en la variable de decenas
26
27     mov ah, 01h       ; Leer el segundo carácter ingresado (unidades)
28     int 21h           ; Llamada a la interrupción para leer el carácter
29     sub al, 30h        ; Restar 30h para convertir de ASCII a su valor numérico
30     mov u, al          ; Almacenar el valor en la variable de unidades
31
32     mov al, d          ; Mover el valor de las decenas a AL
33     mov bl, 10         ; Cargar 10 en BL (para multiplicar las decenas por 10)
34     mul bl             ; Multiplicar AL (decenas) por 10
35     add al, u           ; Sumar el valor de las unidades a AL (completando el número)
36     mov n, al          ; Almacenar el número completo en la variable 'n'
37
38     mov ah, 09h       ; Cargar la función 09h para mostrar el segundo mensaje
39     lea dx, mensaje2   ; Cargar la dirección del mensaje en DX
40     int 21h           ; Llamada a la interrupción para mostrar el mensaje
41
42     mov al, n          ; Cargar el número completo en AL
43     AAM               ; Ajuste decimal: divide el valor de AL entre 10 y almacena los resultados en AH y AL (decenas en AH, unidades en AL)
44     mov bx, ax         ; Mover el valor de AX (decenas y unidades) a BX
45
46     mov ah, 02h       ; Cargar la función 02h para desplegar un carácter
47     mov dl, bh         ; Mover la decena (almacenada en BH) a DL
48     add dl, 30h        ; Convertir la decena a su valor ASCII
49     int 21h           ; Desplegar el carácter de la decena
50
51     mov ah, 02h       ; Cargar nuevamente la función 02h
52     mov dl, bl         ; Mover la unidad (almacenada en BL) a DL
53     add dl, 30h        ; Convertir la unidad a su valor ASCII
54     int 21h           ; Desplegar el carácter de la unidad
55
56 main endp
57 end main

```

Ejecución y compilación del código



```
GUI Turbo Assembler x64
Run the executable (Alt+R)
Ingrese un numero: 12
numero ingresado: 12_
```

 GOBIERNO DEL ESTADO DE MÉXICO	<h2 style="text-align: center;">PRÁCTICA 2</h2> <p style="text-align: center;">Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales</p>	 TECNOLOGICO DE ESTUDIOS SUPERIORES JILOTEPEC
---	--	--

CUESTIONARIO:

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

- **Acumulador (AL):** Se utiliza para almacenar temporalmente los dígitos ingresados por el usuario. Al recibir el número a través de int 21h, se resta 30h (valor ASCII de '0') para convertir el carácter en su valor numérico correspondiente. Luego, estos valores se multiplican y suman para obtener el número final en base decimal.
- **Registros auxiliares (BL y BX):** El registro BL almacena el valor 10, que se usa para convertir el dígito de las decenas. Posteriormente, BX se utiliza para almacenar el resultado de la instrucción AAM, que divide el número en decenas y unidades, separándolos para ser desplegados.



2. ¿Por qué no se pueden escribir como en un lenguaje de alto nivel?

En ensamblador, se trabaja directamente con el hardware y las instrucciones de bajo nivel del procesador. A diferencia de los lenguajes de alto nivel, como C o Python, donde se puede usar una sola línea para mostrar números o manejar variables, en ensamblador se requiere un control más granular. Las operaciones aritméticas, la gestión de memoria y la conversión de datos entre diferentes representaciones (como ASCII y binario) deben realizarse de manera explícita, usando registros y llamadas a interrupciones.

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

Si se requieren más dígitos, se podría extender el código de la siguiente manera:

- Para números de más de dos dígitos, sería necesario recibir y almacenar cada dígito en variables adicionales. Luego, al reconstruir el número completo, se podrían aplicar multiplicaciones sucesivas para formar la cantidad deseada. Por ejemplo, para un número de tres dígitos, el primer dígito se multiplicaría por 100, el segundo por 10, y el último se sumaría directamente.
- Posteriormente, al desplegar el número, se usaría una secuencia similar a la de AAM, pero dividiendo sucesivamente entre 10 para extraer cada dígito del número y convertirlo a su representación ASCII.

 GOBIERNO DEL ESTADO DE MÉXICO	<h2 style="text-align: center;">PRÁCTICA 2</h2> <p style="text-align: center;">Ing. y Esp. Rodolfo Guadalupe Alcántara Rosales</p>	 TECNOLÓGICO DE ESTUDIOS SUPERIORES JILOTEPEC
---	--	--

CONCLUSIONES:

En esta práctica de programación en ensamblador, se ha demostrado cómo utilizar los registros acumuladores y de datos para manipular y desplegar dos números de dos dígitos ingresados por el usuario. A través del uso de interrupciones y operaciones aritméticas básicas, se han realizado conversiones entre la representación ASCII y los valores numéricos, permitiendo reconstruir el número final y mostrarlo en pantalla.

Este ejercicio resalta la importancia del manejo preciso de los registros y las operaciones de bajo nivel que son necesarias en ensamblador, ofreciendo un control detallado sobre el flujo y los datos del programa. Aunque lenguajes de alto nivel abstraen estos detalles, la experiencia con ensamblador proporciona una comprensión más profunda del funcionamiento interno del procesador, las operaciones de la CPU y la manipulación de datos en memoria.