



UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA ORIENTAL
INGENIERIA DE SISTEMAS INFORMÁTICOS

PRIMER LABORATORIO

NOMBRE: **Marlon Adonay Alemán González**

CARNET: **AG22011**

1. Conteste las siguientes preguntas:

a. ¿Qué es Lenguaje Ensamblador?

Es un lenguaje de programación de bajo nivel que utiliza códigos mnemotécnicos (según entendí, que son fáciles de memorizar e interpretar para los humanos) para interpretar las instrucciones de máquina

b. ¿Qué es Lenguaje Máquina?

Es el sistema de códigos directamente interpretable por un circuito programable, es decir que son digitales y trabajan solo con 2 valores (1 y 0), es un lenguaje mas complicado de interpretar a simple vista que el lenguaje ensamblador.

2. Explique dos (2) ventajas del Lenguaje Ensamblador.

➤ El lenguaje ensamblador es más amigable (legible) a la interpretación humana que el lenguaje máquina

➤ Usa menos recursos y es útil para microcontroladores y sistemas antiguos pequeños o con poca capacidad de procesamiento. Da directamente instrucciones específicas al procesador, lo cual lo hace mas rentable a la hora del rendimiento.

3. ¿Cuáles son las dos (2) clases de memoria de la PC y cuáles son sus principales usos?

1. Memoria RAM: es la que almacena datos e instrucciones de los programas que se requieren en un momento determinado y dicha información es utilizada en tiempo real por la CPU.

Es la que permite que los programas se inicien, carguen y ejecuten.

2. Memoria ROM: sus siglas significan “Read Only Memory” que quiere decir “memoria de solo lectura”. En dicha memoria los datos se leen y utilizan, pero no se modifican. En ella están los datos principales para que funcione una máquina (BIOS) y sigue almacenando información, aunque el equipo este apagado o se quede momentáneamente sin energía

4. Determine la representación de su nombre completo.
En ascii

Nombre en ascii : “Marlon Adonay Alemán González” , 0

SU NOMBRE COMPLETO EN BINARIO

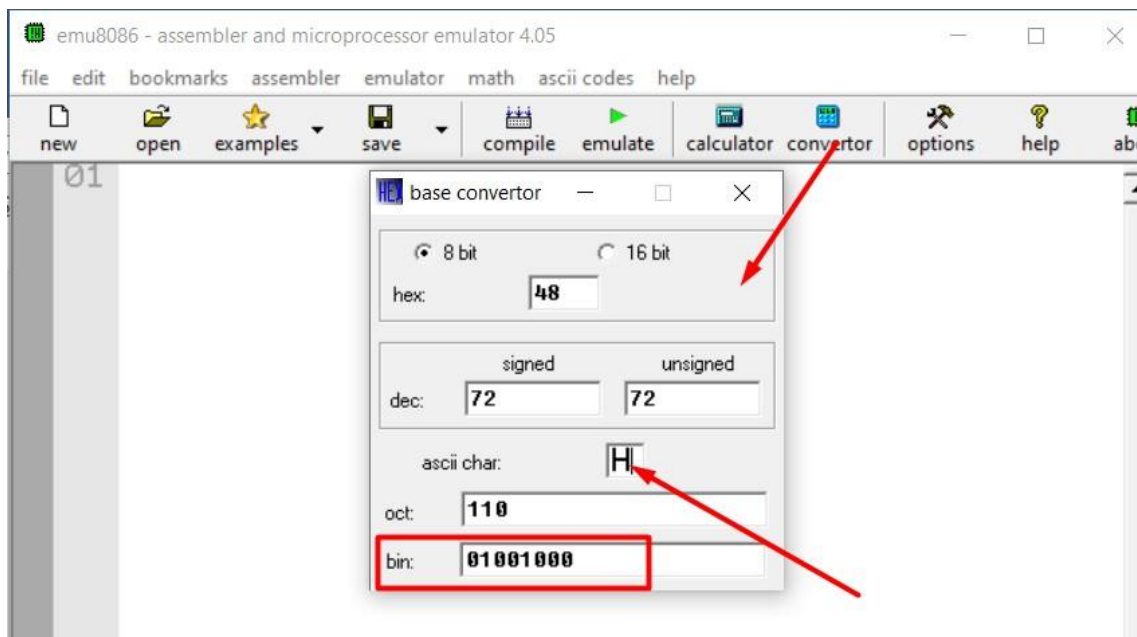
M	a	r	l	o	n
1001101	1100001	1110010	1101100	1101111	1101110

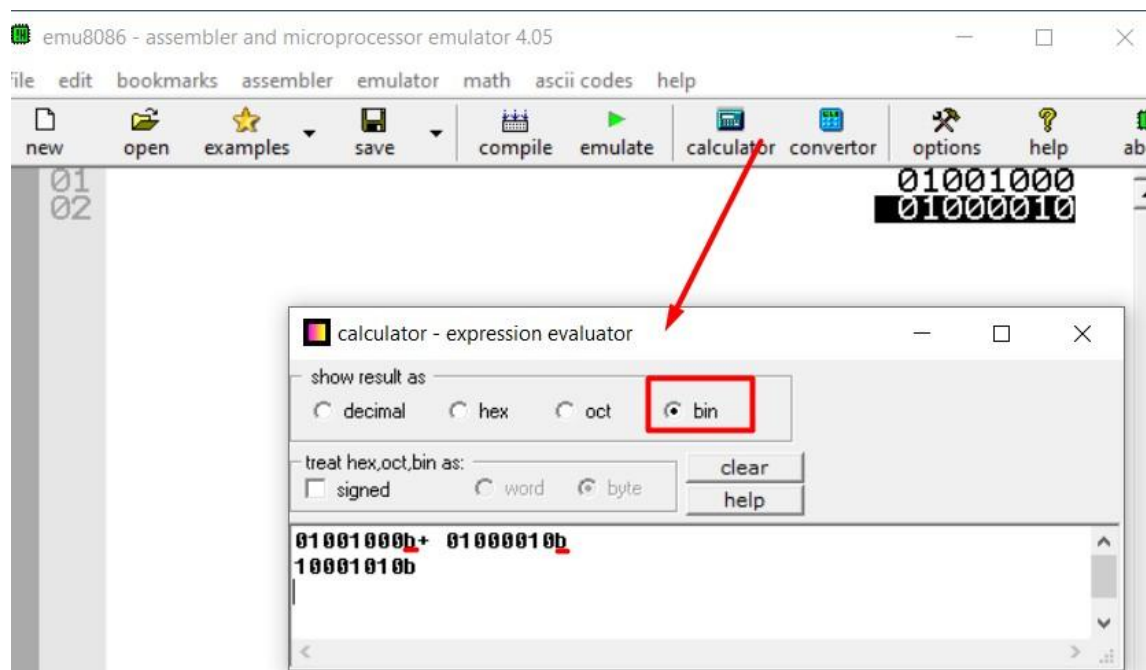
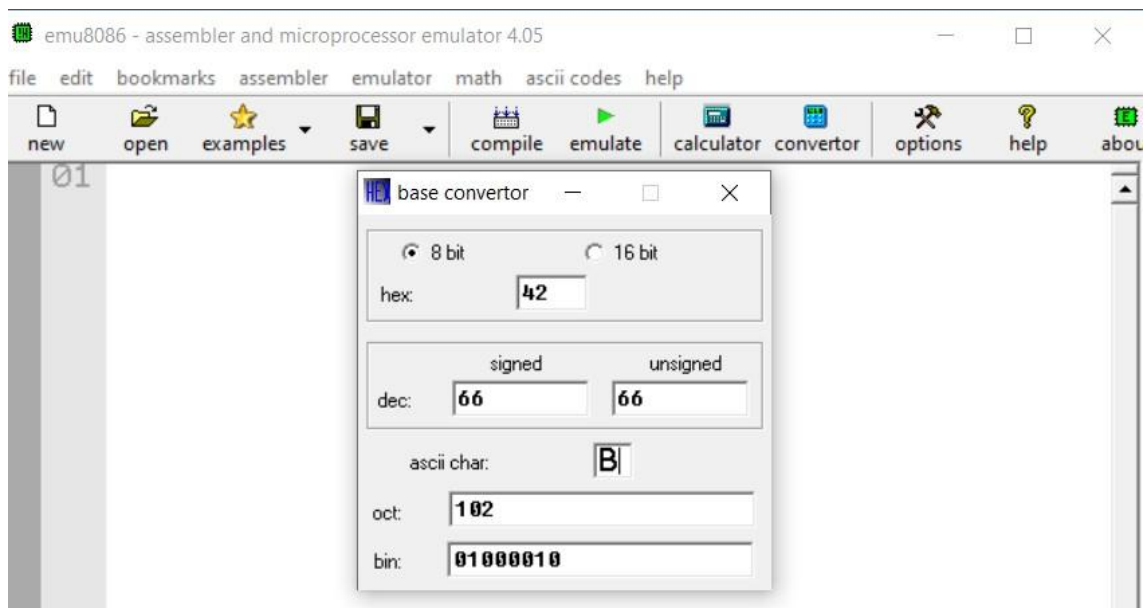
A	d	o	n	a	y
1000001	1100100	1101111	1101110	1100001	1111001

A	l	e	m	a	n
1000001	1101100	1100101	1101101	1100001	1101110

G	o	n	z	a	l	e	z
1000111	1101111	1101110	1111010	1100001	1101100	1100101	1111010

5. Sume la representación en binario de las letras de su apellido





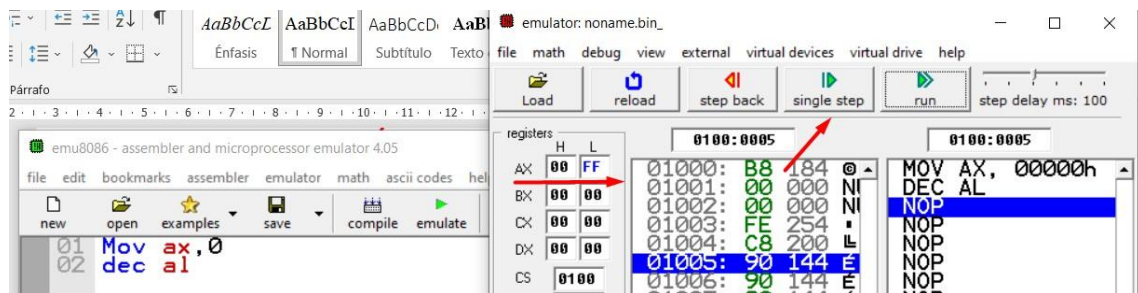
R//100010
10b

A	I	e	m	a	n
1000001	1101100	1100101	1101101	1100001	1101110

R//00000001001001110b

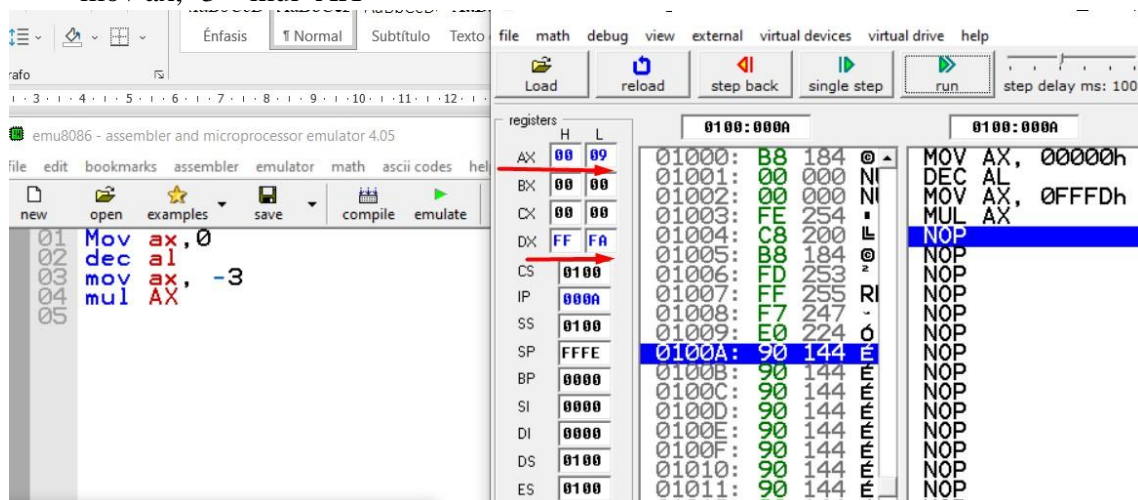
6. Diga cuales son los valores de los registros requeridos después de cada instrucción :

```
mov ax, 0
dec AL
```



; ax = 00FF

```
mov ax, -3 mul AX
```



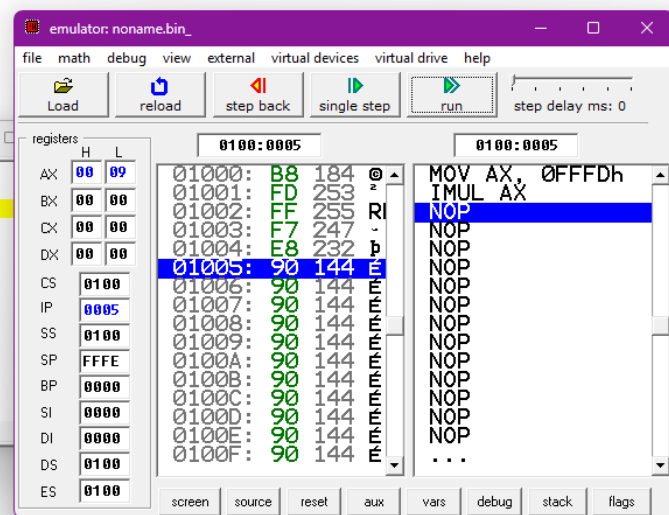
; dx =FFFA , ax = 0009

mov ax, -3

imul AX

```
Mov ax, -3
imul AX
```

```
01
02 Mov ax, -3
03 imul AX
04
05
```



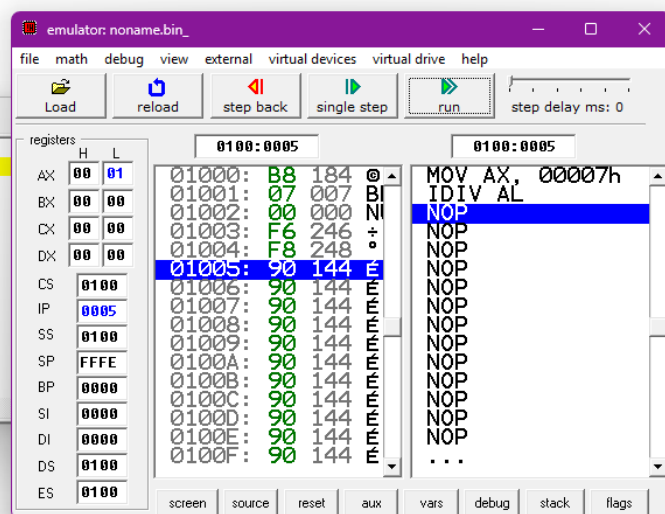
; dx = 0000, ax = 0009

mov ax, 7

idiv AL

```
mov ax, 7
idiv AL
```

```
01 mov ax, 7
02 idiv AL
03
04
05
```

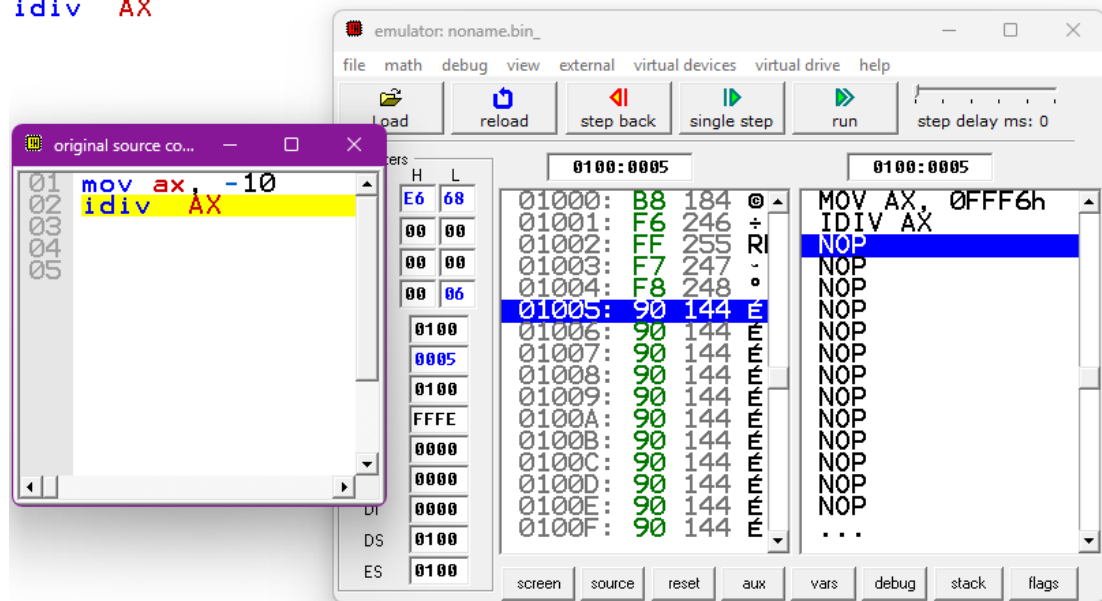


; dx =0000 , ax =0001

mov ax, -10

idiv AX

```
mov ax, -10  
idiv AX
```



; dx =0006, ax = E668

nop