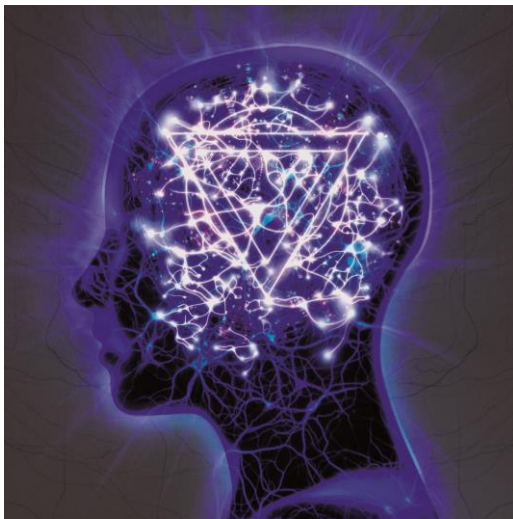
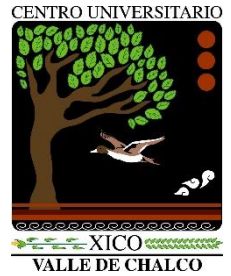




# UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO



## PALÍNDROMO

Ensambladores

### Reporte [10-SEP-2019].

Reporte de la práctica 01 dejada para el día 10 de septiembre en donde se tiene que hacer un código que reciba una cadena y sea impresa a la inversa usando lenguaje ensamblador.

**Javier Enrique Trejo**

**Rodriguez**

ICO 0-6 Vespertino

# Índice

- Introducción
- Objetivo
- Programa
- Desarrollo
- Código
- Conclusiones

## Introducción

El reporte que está a punto de leer a continuación trata del programa de la práctica sobre las cadenas, su forma de leerlas e imprimirlas en el lenguaje de ensamblador, en lo teórico se plantea el uso del pseudocódigo y se explica por qué se emplea su uso y como se va desarrollando, en la práctica, es transcrito en lenguaje pertinente y compilado en DosBox para mostrar que este realmente funciona y cumple con lo establecido.

Se espera que sea fácil de entender, aún me queda mucho que aprender para tener una mejor manera de explicar todo sobre cualquier programa

## Objetivo

El alumno estudiante de la materia de Ensambladores por medio de esta práctica y demás ejercicios aprenderá a manejar mejor el lenguaje en su generalidad, así como usa las cadenas y los distintos registros y segmentos del lenguaje de programación Ensamblador de una forma teórica, pero sobre todo y mejor, práctica, para que le ayude en el ámbito laboral en un futuro cuando se dedique a esto de la programación.

## Programa.

## Desarrollo

Realizar un programa en ensamblador que lea una cadena de n número de caracteres y que este la imprima otra vez, pero hacia el lado contrario, es decir, de diestra a siniestra y otro que lo lea de forma regular y se comparen los caracteres que van saliendo de cada cadena. El programa debe cumplir con los siguientes requisitos:

- a) Estar escrito en lenguaje ensamblador.
- b) Recibir una cadena y guardarla después de pulsar enter.
- c) Imprimirla hacia el lado contrario en que se recibió.

## Código

Primero se pone el tamaño que va a ser el código y posterior a eso la librería .data y .stack para que funcione bien el programa. Después declaramos un segmento donde van a estar todos nuestros datos que se van a imprimir durante el transcurso de todo el código.

BUF es el arreglo que, gracias a la función de dup(duplicar) es que duplicamos 255 veces cada variable para guardar valores indefinidos o en este caso un máximo de 255 valores. Todo esto se muestra en la imagen 01.

```
1  .model small
2  .data
3  .stack
4  .code
5
6  datos segment
7      msj00 db "Alumno: Trejo Rodriguez
8      msj01 db "Introduce una palabra HI
9      msj02 db "la cadena laida fue: $"
10     msj03 db "la cadena escrita fue:
11     nuevalinea db 13,10, "$"
12     contador dw 0
13 datos ends
14
15 buf db 255 dup(0)
```

(Img01. Segmento de datos y librerías)

En la siguiente imagen (Img02) se muestra en la primera parte que se usa la función ASSUME que sirve para que el código entienda o *asuma* que registro le corresponde cada segmento de datos.

En la segunda parte de la imagen se muestra que empieza el código con el main, después se pasan todos los datos al registro acumulador AX y este mismo los pasa al *segmento* de datos para almacenarlos al comienzo del segmento. Lo que le sigue a esa parte solo son las impresiones de los datos del alumno y después el mensaje 01 que dice que ingreses una cadena por teclado.

```
17 main:
18     mov cx, @data
19     mov ds, cx
20
21     lea dx, msj00
22     mov ah, 09h
23     int 21h
```

(Img02. Impresiones de datos)

En la siguiente imagen (Img03) se muestra la parte de código donde se piden los datos, es decir, la que va recibiendo los caracteres y esta misma parte los va comparando cada uno con el "Enter", en dado caso que si llega a ingresar el enter, entonces el programa pasa a la orden de JMP

En la segunda parte es donde empieza una etiqueta de "Meter" que lo que hace es primero mover SI al registro AX, después incrementa en 2 el puntero que apunta al arreglo, ahora después se mueve AX a SI, como el primer paso, pero a la inversa. Después de todo este movimiento, se mete el carácter a la pila usando PUSH y se incrementa el contador para que después salte otra vez a Pedir y se ingrese otro carácter.

```
Pedir:
    lea si,buf
    Pedir01:
    mov ah,01h
    int 21h
    mov ah,00h
    cmp ax,13h
    jz Sacar
    jnz Meter

Meter:
    mov ax, si
    add ax,02h
    mov si,ax
    push ax
    mov ax,contador
    mov ax,01h
    mov contador,ax
    jmp Pedir01
```

(Img03. Pide los datos y los mete caracter por caracter tanto a la pila como al arreglo)

```
45
46 Sacar:
47     lea di,arr1
48     mov cx,contador
49
50     Sacar01:
51         mov dx,[di]
52         pop ax
53         mov ah,02h
54         int 21h
55         mov ax,02h
56         mov di,ax
57         cmp dx, ax
58         jz Incrementar
59         Incrementar:
60             inc cx
61             cmp cx, contador
62             jnz Palindromo
63     loop Sacar01
64
65     Palindromo:
66         lea dx,msg04
67         mov ah,09h
68         int 21h
```

En la siguiente imagen (Img04) se muestra como va la etiqueta Sacar que, lo que hace es primero posicionarse en el contador de puntero del arreglo y desde ahí, empezar a sacar carácter por carácter del propio arreglo, todo eso lo hace cuando entra al loop de Sacar01 que este a su vez también saca los caracteres de la pila haciendo uso de la palabra reservada POP.

Al final solo compara la cadena que sacó del arreglo con la que sacó de la pila y así te dice si esta palabra/cadena es un palíndromo o no.

(Img04. Compara cadenas e imprime el resultado)

## Resultados

Después de que el programa hace todo lo previamente explicado, los resultados quedan en DosBox quedan de la siguiente manera como se muestra en las siguientes imágenes: (Img05. El caso donde si es un palíndromo y la Img06. El caso donde no lo es).

```
C:\SOFTWARE>p05
Alumno: Trejo Rodriguez Javier Enrique?
Grupo: ICO 0-6
Materia: Ensambladores
Introduce una palabra:
anitalavalatina
la cadena laida fue:
anitalavalatina
la cadena escrita fue:  anitalavalatina
SI es un PALINDROMO:
C:\SOFTWARE>
```

*(Img05.Resultados positivos del programa)*

```
C:\SOFTWARE>p05
Alumno: Trejo Rodriguez Javier Enrique?
Grupo: ICO 0-6
Materia: Ensambladores
Introduce una palabra:
holaquehace
la cadena laida fue:
holaquehace
la cadena escrita fue:  ecaheugaloh
NO es un PALINDROMO:
C:\SOFTWARE>
```

*(Img06.Resultados negativos del programa)*

## **Conclusiones**

Se concluye que la práctica recién hecha usando el lenguaje ensamblador para invertir una cadena y sacara haciendo uso de arreglos y pilas será de gran ayuda para el estudiante y le servirá en parte, para mejorar y seguir estudiando el manejo de pilas, arreglos, registros y segmentos y en la programación en general pues, estos requieren su propia lógica y tienen su cierto grado de dificultad, por eso se tiene que seguir estudiando todos estos aspectos de la materia, se hicieron uso de imágenes en las cuales se muestra el código en Notepad++ para mejor comprensión del estudiante y del profesor pero también se usaron imágenes para ilustrar los resultados del programa y verificar que los resultados sean correctos en estos mismos.