



Práctica #5: Ciclos numéricos y captura básica de cadenas

OBJETIVOS:

- Implementar ciclos numéricos en ensamblador.
- Realizar capturas básicas de cadena en EMU8086 y DosBOX.

Temas Cubiertos:

- Ciclos numéricos.
- Captura básica de cadenas.

Materiales Necesarios:

- emu8086
- DOSBox con MASM
- Documentation for emu8086

DESARROLLO:

1 – CICLOS NUMÉRICOS

Instrucciones: Correr los siguientes códigos en EMU8086.

Ejercicio 1.

```
org 100h
mov cx, 10      ; Inicializar el contador
mov ax, 0       ; Inicializar el acumulador
start_loop:
    add ax, cx   ; Sumar CX a AX
    dec cx       ; Decrementar CX
    jnz start_loop ; Saltar a start_loop si CX no es cero
mov ah, 4Ch     ; Terminar el programa
int 21h
```

Preguntas:

- Describir cada acción realiza cada línea de código.
- ¿Qué valor tendrá AX al final del ciclo y por qué?

Ejercicio 2.

```
org 100h
mov cx, 10      ; Inicializar el contador
mov ax, 0       ; Inicializar el acumulador
```

```

start_loop:
    add ax, cx    ; Sumar CX a AX
    dec cx       ; Decrementar CX
    jnz start_loop ; Saltar a start_loop si CX no es cero
    mov ah, 4Ch   ; Terminar el programa
    int 21h

```

Preguntas:

- Describir cada acción realiza cada línea de código.
- ¿Qué instrucción se usa para la multiplicación en ensamblador?
- ¿Qué valor tendrá AX al final del ciclo si se multiplica del 5 al 1?

2 – CAPTURA BÁSICA DE CADENAS

Ejercicio 1: En EMU8086 y DosBOX realizar la captura una cadena de caracteres desde el teclado y la almacena en la memoria.

```

.model small
.stack 100h
.data
    buffer db 20, 0, 20 dup('$')
.code
main:
    mov ax, @data
    mov ds, ax
    lea dx, buffer
    mov ah, 0Ah
    int 21h
    mov ah, 4Ch
    int 21h
end main

```

- Indicar que realiza cada línea en el código.
- ¿Qué función de la interrupción 21h se utiliza para capturar una cadena de caracteres?
- ¿Cómo se define el buffer para capturar la cadena?

Ejercicio 2: Completar el siguiente código para que: en EMU8086 y DosBOX realizar la captura de una cadena y desplegado de la misma.

```

.model small
.data
    mensaje db 10,13, "ingresa cadena: $"
    crlf db 10,13, "$"
    cadena db 50,?,10 dup(' ')

```

```

.code
mov ax, @data
mov ds, ax
mov ah, 09h
lea dx, mensaje
int 21h
mov dx,
mov ah, 0ah
int 21h
mov ah, 09h
lea dx, crlf
int 21h
mov bl, cadena[1]
mov cadena[bx+2], '$'
mov dx,
mov ah, 09h
int 21h
.exit
end

```

- ¿Qué elemento hacía falta para completar el código y por qué?
- Describir cada acción realiza cada línea de código.
- ¿Por qué es necesario el uso de incrementos en el índice en: *mov cadena[bx+2]*?