# ASSEMBLER

PSP en Emu8086

nombreArchivo agr1 agr2
nombreArchivo agr1

nombre 3 lista

```
xor cx,cx
Mov cl,es:[128]
mov t_cadena, cl
```

len → 8

Aceder al los args

```
mov si, 130
Mov DL,ES:[si]
```

Ejemplo:

nombre 2 hola

```
xor cx,cx
Mov cl,es:[128]
mov t_cadena, cl
```

En t_cadena:7
La longitud de la cadena es t_cadena-1

nombre 2 hola

130 131 132

```
mov si, 130
Mov DL,ES:[si]
```

Ejercicio

Contar la cantidad de carracteres

| | | |
|---|---|---|
| w:\>Nombre 3<br>1 | w:\>Nombre 2sd<br>3 | w:\>Nombre 0<br>1 |

Realizar un programa aplicando PSP para introducir un número menor 0 igual a 9 como argumento en la línea de comandos, ejemplo si el argumento del programa es 5, entonces imprimir los siguientes caracteres:

1
12
123
1234
12345

```
; multi-segment executable file template.

data segment
    enter db 10,13,"$"
    n db 0
    i db 0
ends

stack segment
    dw   128  dup(0)
ends

code segment
start:
; set segment registers:
    mov ax, data
    mov ds, ax

    mov si, 130
    mov dl,es:[si]
    sub dl,30h
    mov n , dl

    mov i, 1
    for:
     cmp n, 0
     jz fin
      mov cl, i
      mov ch,0
      mov dl, 31h
      for1:
        mov ah , 2
        int 21h
        inc dl
      loop for1
      lea dx,enter
      mov ah, 9
      int 21h
      inc i
    dec n
   jmp for
    fin:
    mov ax, 4c00h ; exit to operating system.
    int 21h
ends

end start ; set entry point and stop the assembler.

```

2. Ejemplo, si el número es 6, entonces mostrar en la pantalla:
   0
   1
   10
   101
   1010
   10101
   101010

```
01 data segment
02     n db 0
03     i db  0
04     enter db 10,13,"$"
05 ends
06
07 stack segment
08     dw   128  dup(0)
09 ends
10 code segment
11 start:
12     mov ax, data
13     mov ds, ax
14     mov cx,0
15     mov dl,es:[130]
16     sub dl,30h
17     mov n, dl
18     cmp n , 0
19     je fin
20     mov dl,30h
21     mov ah,2
22     int 21h
23     lea dx,enter
24     mov ah,9
25     int 21h
26     dec n
27     mov i,1
28     ciclo:
29       cmp n, 0
30       je fin
31
32       mov bl, i
33       ciclo2:
34         mov dl, 31h
35         mov ah,2
36         int 21h
37        dec bl
38        cmp bl, 0
39        je finCiclo2
40          mov dl, 30h
41          mov ah,2
42          int 21h
43         dec bl
44         cmp bl, 0
45         je finCiclo2
46       jmp ciclo2
47        finCiclo2:
48        lea dx,enter
49        mov ah,9
50        int 21h
```

```
51          inc i
52        dec n
53      jmp ciclo
54      fin:
55      mov ax, 4c00h ; exit to operating system.
56      int 21h
57  ends
58
59  end start ; set entry point and stop the assembler.
60
```

## Manejo de si,di,bx

```
01  ; multi-segment executable file template.
02
03  data segment
04      ; add your data here!
05      pkey db "press any key...$"
06      hola db "Hola MUndo en Assbler$"
07  ends
08
09  stack segment
10      dw   128  dup(0)
11  ends
12
13  code segment
14  start:
15  ; set segment registers:
16      mov ax, data
17      mov ds, ax
18      mov es, ax
19
20
21      mov di,0
22
23      mov cx, 3
24      for:
25          mov dl,hola[di]
26          mov ah , 2
27          int 21h
28          inc di
29      loop for
30
31      mov ax, 4c00h ; exit to operating system.
32      int 21h
33  ends
34
35  end start ; set entry point and stop the assembler.
36
```