ASSEMBLER

PSP en Emu8086

nombreArchivo agr1 agr2 nombreArchivo agr1

nombre 3 lista

Aceder al los args

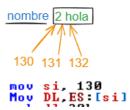
```
Mov Si, 130
Mov DL,ES:[si]
```

Ejemplo:

nombre 2 hola

```
Mov cl.es:[128]
mov t_cadena, cl
```

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



Ejercicio

Contar la cantidad de carracteres

w:\>Nombre 3 1 w:\>Nombre 2sd 3

w:\>Nombre 0 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:

```
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 ch,0
mov dl, 31h
for1:
mov ah, 2
int 21h
inc dl
loop for1
lea dx,enter
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
```

(D) (B) (E) (E

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

```
inc i
dec n
jmp ciclo
fin:
mov ax, 4c00h; exit to operating system.
int 21h
ends
end start; set entry point and stop the assembler.
```

Manejo de si,di,bx

```
01 ; multi-segment executable file template.
02
03 data segment
       ; add your data here!
pkey db "press any key...$"
hola db "Hola MUndo en Assbler$"
04
05
06
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
mov ds, ax
17
18
        mov es, ax
19
20
21
22
23
24
25
26
27
28
29
30
        mov di,0
        mov cx, 3 for:
            mov dl.hola[di]
mov ah . 2
int 21h
            ine di
         loop for
        31
32
33 ends
34
35 end start ; set entry point and stop the assembler.
```