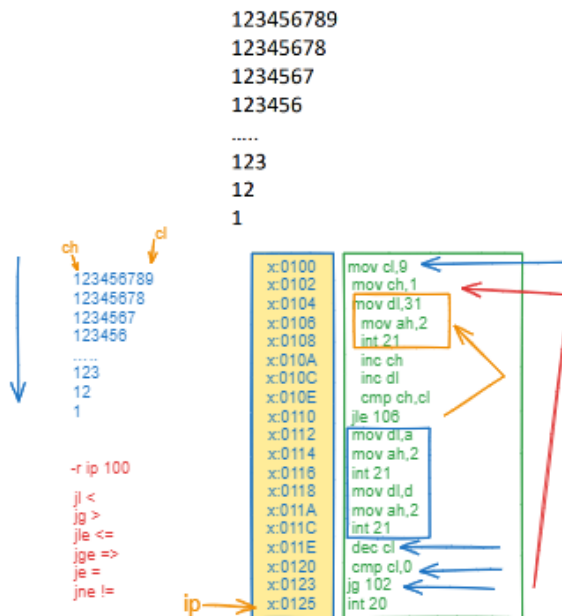


ASSEMBLER

1.- Realice un programa .COM. para mostrar en la pantalla caracteres numéricos de la siguiente manera:



Ingresar N y generar la siguiente serie

N:2
123456789
12345678

N:6
123456789
12345678
1234567
123456
12345
1234

```
01 data segment
02     msg db "N: $"
03     n db 0
04     enter db 10,13,"$"
05 ends
06
07 stack segment
08     dw 128 dup(0)
09 ends
10
11 code segment
12 start:
13     mov ax, data
14     mov ds, ax
15     mov es, ax
16
17     lea dx, msg
18     mov ah, 9
19     int 21h
20
21     mov ah, 1
22     int 21h
23     sub al, 30h
24     mov n, al
25
26     lea dx, enter
27     mov ah, 9
28     int 21h
29
30     mov cl, 9
31     iterando:
32         mov ch, 1
33         mov dl, 31h
34         desplegando:
35             mov ah, 2
36             int 21h
37             inc ch
38             inc dl
39             cmp ch, cl
40             jle desplegando
41
42         mov dl, 0ah
43         mov ah, 2
44         int 21h
45
46         mov dl, 0dh
47         mov ah, 2
48         int 21h
49
50         dec cl
51         dec n
52         cmp n, 0
53         jg iterando
54
55         mov ax, 4c00h
56         int 21h
57 ends
58 end start
59
```

PSP en Emu8086

nombreArchivo agr1 agr2
nombreArchivo agr1

nombre 3 lista

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

len → 8

Acceder 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

Imprimir la siguiente serie

```
w:\>Nombre 3
abc
```

```
w:\>Nombre 2
ab
```

```
w:\>Nombre 0
```

```
w:\>Nombre
```

```
01 PILA SEGMENT PARA STACK "STACK"
02 DW 32 DUP(0)
03 PILA ENDS
04 DATOS SEGMENT PARA "DATA"
05     t_cadena db 10
06     n db 0
07 DATOS ENDS
08 CODIGO SEGMENT PARA "CODE"
09 BEGIN PROC FAR
10
11     ASSUME SS:PILA,DS:DATOS,CS:CODIGO
12     Mov AX,DATOS
13     Mov DS,AX
14
15     ; xor cx,cx ; cx=0
16     mov cx,0
17     Mov cl,es:[128]
18     mov t_cadena, cl
19     ; or cx,cx ; cmp cx, 0
20     cmp t_cadena,0
21     jz fin
22
23     mov si, 130
24     Mov DL,ES:[si]
25     sub dl,30h
26     mov n , dl
27     cmp n, 0
28     jz fin
29
30 ;Generando la serie
31 mov cx, 0
32 mov cl,n
33 mov dl,61h ; a
34 for:
35     mov ah,2
36     int 21h
37     inc dl
38     loop for
39 fin:
40
41     Mov AX,4C00H
42     Int 21H
43
44 BEGIN ENDP |
45 CODIGO ENDS
46 End Begin
```

CX=11010101

CX=11010101

CX=00000000

1. (9 Pts.) Escriba un programa en el emulador 8086, empleando el Prefijo de Segmento de Programa (PSP) para introducir por teclado un número de un dígito y una cadena, luego mostrar en la pantalla la cadena el número de veces del dígito introducido. En la parte inferior del programa corrido, debe mostrar en nombre completo y su cedula de identidad de estudiante.

Ejemplo:

Si ejecutamos el programa con los siguientes datos:

D:/> NomProg 4 Primavera

El corrido del programa debe mostrar en pantalla lo siguiente:

1. Primavera
 2. Primavera
 3. Primavera
 4. Primavera
- Carlos Mullisaca Choque
C.I. 3341228 L.P.

```

01 PILA SEGMENT PARA STACK "STACK"
02 DW 32 DUP(0)
03 PILA ENDS
04
05 DATOS SEGMENT PARA "DATA"
06 t_cadena db 0
07 cant_repetir db 0
08 cont db 31h
09 enter db 0ah,0dh,"$"
10 DATOS ENDS
11
12 CODIGO SEGMENT PARA "CODE"
13
14 BEGIN PROC FAR
15 ASSUME SS:PILA,DS:DATOS,CS:CODIGO
16
17 Mov AX,DATOS
18 Mov DS,AX
19
20 mov ch,0
21 Mov cl,es:[128]
22 Mov t_cadena,cl; guardamos el tamaño de la cadena leida
23 cmp cl,0
24 jz fin
25
26 Mov DL,ES:[130]
27 sub dl,30h
28 mov cant_repetir,dl
29
30 ;->Solucion
31 while:
32 mov cx,0
33 mov cl,t_cadena
34 sub cx,3
35
36 mov ah,2
37 mov dl,cont
38 int 21h
39 mov dl," "
40 int 21h
41
42 inc cont
43
44 mov si,132; del 3ro caracter
45 mostrando:
46 mov dl,ES:[SI]
47 int 21h
48 inc SI
49 Loop mostrando
50
51 mov ah,9
52 mov dx,offset enter
53 int 21h
54
55 dec cant_repetir
56 cmp cant_repetir,0
57 jg while
58
59 fin:
60
61 Mov AX,4C00H
62 int 21h
63 BEGIN ENDP
64 CODIGO ENDS
65 End Begin

```

To move canvas, hold mouse wheel or spacebar while dragging, or u

nombre 2 hola

si 132