

1. (5 Pts.) Realice un programa para introducir por teclado un número con dos dígitos, luego sumar los dígitos del número introducido; mostrar en la pantalla el mensaje: "Si puedes controlar la información, puede controlar a la gente." El número de veces del nuevo número sumado.

En la parte inferior del programa corrido, debe mostrar en nombre completo y su cedula de identidad de estudiante.

Ejemplo:

Si introducimos por teclado 13, entonces la suma de los dígitos es 4; y debe mostrar en pantalla lo siguiente:

1. Si puedes controlar la información, puede controlar a la gente
2. Si puedes controlar la información, puede controlar a la gente
3. Si puedes controlar la información, puede controlar a la gente
4. Si puedes controlar la información, puede controlar a la gente

Carlos Mullisaca Choque

C.I. 3341228 L.P.

```

01 data segment
02 ; add your data here!
03 texto db ". Si puedes controlar la informacion... ",10,13,"$"
04 n db 0
05 enter db 10, 13, "$"
06 res db 5 dup(" "), "$"
07 val10 dw 10
08 ends
09 stack segment
10 dw 128 dup(0)
11 ends
12 macro mostrarText txt
13     lea dx, txt
14     mov ah, 9
15     int 21h
16 endm
17 macro mostraNum num
18     mov dl, num
19     add dl, 30h
20     mov ah, 2
21     int 21h
22 endm
23 code segment
24 start:
25     mov ax, data
26     mov ds, ax
27     mov es, ax
28
29     call lecturaSuma
30     mostrarText enter
31     mov bl, 1
32     for:
33         mov ax, 0
34         mov al, bl
35         call mostrar
36         mostrarText texto
37         inc bl
38     cmp bl, n
39     jle for
40     mov ax, 4c00h ; exit to operating system.
41     int 21h
42 ends
43
44
45 lecturaSuma:
46     mov n, 0
47     reading:
48         mov ah, 1
49         int 21h
50         cmp al, 13
51         je finLeer
52         sub al, 30h
53         add n, al
54     jmp reading
55     finLeer:
56     ret
57

```

```

58 mostrar:
59     ; (ax)
60     mov di, 4
61     descomp:
62         mov dx, 0
63         div val10
64         add dl, 30h
65         mov res[di], dl
66         dec di
67         cmp ax, 0
68         jg descomp
69         inc di
70     mostrarText res
71     ret
72 end start ; set entry point and stop the assembler.
73


```

2. (10 Pts.) Realice un programa para introducir por teclado un número con dos dígitos, luego sumar los dígitos del número introducido y mostrar en la pantalla la generación de la siguiente serie: 0,1,00,11,000,111,0000,1111,...,El número de términos del nuevo número sumado.

En la parte inferior del programa corrido, debe mostrar en nombre completo y su cedula de identidad de estudiante.

Ejemplo:

Si introducimos por teclado 45, entonces la suma de los dígitos es 9; en la pantalla se debe mostrar la serie generada para 9 términos:



0,1,00,11,000
Carlos Mullisaca Choque
C.I. 3341228 L.P.

To move canvas, hold mouse wheel or spacebar while draggin

```

03 data segment
04     MESAJe db "Ingrese N: $"
05     enter db 10,13,"$"
06     n db 0
07     i db 1
08
09 ends
10
11 stack segment
12     dw 128 dup(0)
13 ends
14
15 macro mostrarChar char
16     mov dl, char
17     mov ah, 2
18     int 21h
19 endm
20
21 code segment
22 start:
23     mov ax, data
24     mov ds, ax
25     mov es, ax
26
27     call leerDigSuma
28
29     lea dx, enter
30     mov ah, 9
31     int 21h
32
33     procesando:
34     cmp n, 0
35     je finProcesar
36     mov cx, 0
37     mov cl, i
38     ceros:
39     mostrarChar '0'
40     loop ceros
41
42     dec n
43     cmp n, 0
44     je finProcesar
45
46     mostrarChar ','
47
48     mov cx, 0
49     mov cl, i
50     unos:
51     mostrarChar '1'
52     loop unos
53
54     dec n
55     cmp n, 0
56     je finProcesar
57
58     mostrarChar ','
59     inc i
60     jmp procesando
61 finProcesar:
62     mov ax, 4c00h ; exit to operating system.
63     int 21h
64 ends

```

```

66 leerDigSuma:
67     mov n, 0
68     reading:
69     mov ah, 1
70     int 21h
71     cmp al, 13
72     je fin
73     sub al, 30h
74     add n, al
75     jmp reading
76     fin:
77     ret
78
79 end start ; set entry point and stop the assembler.
80
81

```

1. Realice un programa .COM. , para introducir por teclado una cadena máximo de 10 caracteres, y mostrar en pantalla la cadena en forma vertical.

Ejemplo:

Introducir cadena: ASSEMBLER

R
E
L
B
M
E
S
S
A

si

```
01
02 data segment
03     ; add your data here!
04     Texto db 10 dup("& "), "$"
05     enter db 10,13,'$'
06
07 ends
08
09 stack segment
10     dw 128 dup(0)
11 ends
12
13 macro mostrarTxt txt
14     lea dx, txt
15     mov ah, 9
16     int 21h
17 endm
18
19 code segment
20 start:
21     mov ax, data
22     mov ds, ax
23     mov es, ax
24
25     mov si, 0
26
27     reading:
28     mov ah,1
29     int 21h
30     cmp al, 13
31     je finLectura
32     mov Texto[si], al
33     inc si
34     jmp reading
35
36 finLectura:
37     mostrarTxt enter
38
39     dec si
40     mostrando:
41     mov dl, Texto[si]
42     mov ah, 2
43     int 21h
44
45     mostrarTxt enter
46
47     dec si
48     cmp si, 0
49     jge mostrando
50
51     mov ax, 4c00h ; exit to operating system.
52     int 21h
53 ends
54
55
56 end start ; set entry point and stop the assembler.
57
```

2. Realice un programa .COM. , para introducir por teclado una cadena máximo de 10 caracteres, y mostrar en pantalla la cadena en forma vertical.

Ejemplo:

Introducir cadena: ASSEMBLER → A,10,13,S,10,13,S,...

A
S
S
E
M
B
L
E
R

```
02
03 data segment
04     enter db 10,13,"$"
05     Texto db " "
06 ends
07
08 stack segment
09     dw 128 dup(0)
10 ends
11
12 macro mostrarText txt
13     lea dx, txt
14     mov ah, 9
15     int 21h
16 endm
17
18 code segment
19 start:
20 ; set segment registers:
21     mov ax, data
22     mov ds, ax
23     mov es, ax
24
25     mov si, 0
26
27     reading:
28     mov ah, 1
29     int 21h
30     cmp al,13
31     je finRead
32     mov texto[si], al
33     inc si
34     mov texto[si], 10
35     inc si
36     mov texto[si], 13
37     inc si
38     jmp reading
39 finRead:
40
41     mov texto[si], "$"
42     mostrarText enter
43     mostrarText texto
44
45     mov ax, 4c00h ; exit to operating system.
46     int 21h
47 ends
48 end start ; set entry point and stop the assembler.
49
```