ASSEMBLER

Ejercicio

Ingresar un numero generar la siguiente serie

N: 1 1	N: 5 1 22 333 4444 55555	N: 0	N:3 1 22 333	N:8 1 22 333 4444 55555 666666 7777777 88888888

Ejercicio

Ingresar una cadena y mostrar la cantidad de vocales

Hola mundo 4	python 1	Java 2	dwqft 0	Informatica 3

a=61h, e= 65h , i=69h , o=6fh , u=75h

Ejercicio 1

```
; multi-segment executable file template.

data segment
    nro db 0
    i db 1
    enter db 10,13,"$"
ends

stack segment
    dw   128  dup(0)
ends

code segment
start:
    mov ax, data
    mov ds, ax
    mov es, ax


    mov ah, 1
    int 21h
    cmp al , 48
    je finPrograma

    mov nro , al
    sub nro ,48

      lea dx, enter
      mov ah , 9
      int 21h
    ; Mostrando nro-1>0
    mostrando:

      mov ch,0
      mov cl , i
      ;Desplegando  cx
      desplegando:

        mov dl , i
        add dl , 48
        mov ah , 2
        int 21h

      loop desplegando
      inc i; incrementamos i

      lea dx, enter
      mov ah , 9
      int 21h

     dec nro
     cmp nro, 0
     jg mostrando

     finPrograma:

    mov ax, 4c00h
    int 21h
ends

end start
```

Ejercicio 2

```
data segment
    enter db 10,13,"$"
    cadena db ?
ends
stack segment
    dw   128  dup(0)
ends
code segment
start:
    mov ax, data
    mov ds, ax
    mov es, ax

    lea si, cadena
    leer:
      mov ah, 1
      int 21h
      cmp al , 0dh
      je FinLectura
      mov [si],al
      inc si
      jmp leer
    FinLectura:

    dec si
    mov cx, 0
    buscando:;Se esta verificando si son vocales
        cmp [si],61h
        je contar
        cmp [si],65h
        je contar

        cmp [si],69h
        je contar

        cmp [si],6fh
        je contar

        cmp [si],75h
        je contar

        jmp noContar

        contar:
        inc cx

        noContar:
        dec si
        cmp si, 0
        jge buscando

    lea dx , enter
    mov ah,9
    int 21h

    mov dl,cl
    add dl,30h
    mov ah,2
    int 21h

    mov ax, 4c00h ; exit to operating system.
    int 21h

ends
```

Ejer de lic

```
data segment
    ; add your data here!
    msg1 db 10,13,"Numero 1 $"
    msg2 db 10,13,"Numero 2 $"
    sol db 10,13,"El numero mayor  $"
    a db 0
    b db 0
ends

stack segment
    dw   128  dup(0)
ends

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

    lea dx, msg1
    mov ah, 9
    int 21h        ; output string at ds:dx
    mov ah, 1
    int 21h
    mov a,al

    lea dx, msg2
    mov ah, 9
    int 21h        ; output string at ds:dx
    mov ah, 1
    int 21h
    mov b,al

    cmp al,a; cmp b ,a
    jg mayor

    lea dx,sol
    mov ah,9
    int 21h

    mov dl,a
    mov ah,2
    int 21h

    mov ax, 4c00h ; exit to operating system.
    int 21h

    mayor:
    lea dx,sol
    mov ah,9
    int 21h

    mov dl,b
    mov ah,2
    int 21h

    mov ax, 4c00h ; exit to operating system.
    int 21h
ends
end start ; set entry point and stop the assembler.

```