

PRACTICA 2 (algunos ejercicios resueltos)

Interrupciones

3) Escribir un programa que muestre en pantalla las letras del abecedario, sin espacios, intercalando mayúsculas y minúsculas (AaBb...), sin incluir texto en la memoria de datos del programa. Tener en cuenta que el código de "A" es 41H, el de "a" es 61H y que el resto de los códigos son correlativos según el abecedario.

```
ORG 1000H
MAY DB 41H ; "A"
MIN DB 61H ; "a"
        ; La letra "Z" (mayuscula) tiene el codigo 5A

ORG 2000H
MOV AL, 2 ; Se imprime en pantalla de a 2 caracteres
MOV BX, OFFSET MAY ; a partir de la direccion de MAY
PROX: INT 7
      INC MIN ; Paso al siguiente caracter
      INC MAY ; Paso al siguiente caracter
      CMP MAY, 5BH ; comparo con el caracter siguiente al "Z", que es el ultimo valido
      JNZ PROX ; Si aun no procesamos "Z", continua con el siguiente caracter
      INT 0
END
```

6) Escribir un programa que solicite el ingreso de un número (de un dígito) por teclado y muestre en pantalla dicho número expresado en letras. Luego que solicite el ingreso de otro y así sucesivamente. Se debe finalizar la ejecución al ingresarse en dos vueltas consecutivas el número cero.

```

CERO    ORG 1000H
        DB "CERO  " ; Todos los nombres tienen 6 caracteres para
        DB "UNO  "  ; facilitar posicionarnos al imprimir el nombre del numero
        DB "DOS  "
        DB "TRES "
        DB "CUATRO"
        DB "CINCO "
        DB "SEIS  "
        DB "SIETE "
        DB "OCHO  "
        DB "NUEVE "
MSJ      DB "INGRESE UN NUMERO:"
FIN      DB ?

        ORG 1500H
NUM      DB ?

        ORG 2000H
        MOV CL, 0 ; Contador de veces que ingresa el valor 0 de forma consecutiva
OTRO:    MOV BX, OFFSET MSJ
        MOV AL, OFFSET FIN-OFFSET MSJ
        INT 7 ; Imprimo mensaje en pantalla pidiendo el ingreso de un numero
        MOV BX, OFFSET NUM
        INT 6 ; Leo un caracter y queda guardado en NUM
        CMP NUM, 30H
        JNZ NO_CERO
        INC CL ; Si vino un valor 0, incremento el contador
        JMP SEGUIR
NO_CERO: MOV CL, 0 ; Como no vino un valor 0, reinicializo CL
SEGUIR:  MOV BX, OFFSET CERO ; La direccion BASE sera la del primer mensaje ("CERO")
        ; Luego se posicionara al inicio del mensaje adecuado
        ; Se va a imprimir 6 caracteres, todos tienen el mismo largo
        MOV AL, 6
LOOP:    CMP NUM, 30H
        JZ IMPRIME ; Si es el valor adecuado, imprimo en pantalla el nombre del numero
        ADD BX, 6 ; Si no es el valor adecuado, me posiciono en el siguiente nombre
        DEC NUM ; Al llegar NUM a 0 estara posicionado en el nombre que corresponde
        JMP LOOP
IMPRIME: INT 7
        CMP CL, 2
        JNZ OTRO ; Si no se ingreso dos veces seguidas el numero 0, sigue procesando
        INT 0 ; Se ingreso dos veces seguidas 0, por lo que el programa termina
END

```

7) Escribir un programa que efectúe la suma de dos números (de un dígito cada uno) ingresados por teclado y muestre el resultado en la pantalla de comandos. Recordar que el código de cada caracter ingresado no coincide con el número que representa y que el resultado puede necesitar ser expresado con 2 dígitos.

```
ORG 1000H
MSJ DB "INGRESE UN NUMERO:"
FIN DB ?

ORG 1500H
NUM1 DB ?
NUM2 DB ?
RES_D DB "0" ; Decena del resultado.
RES_U DB ? ; Unidad del resultado.
; Por ej. si se suma "6" + "7", la decena del resultado sera "1" y la unidad "3"

ORG 2000H
MOV BX, OFFSET MSJ
MOV AL, OFFSET FIN-OFFSET MSJ
INT 7 ; Imprimo mensaje en pantalla pidiendo el ingreso de un numero
MOV BX, OFFSET NUM1
INT 6 ; Leo un caracter y queda guardado en NUM1
MOV BX, OFFSET MSJ
INT 7 ; Imprimo mensaje en pantalla pidiendo el ingreso de un numero
MOV BX, OFFSET NUM2
INT 6 ; Leo un caracter y queda guardado en NUM2
MOV AL, NUM2 ; Copio el segundo caracter leido en AL
SUB AL, 30H ; Le resto 30H, para quedarme con el valor del numero
ADD AL, NUM1 ; Le sumo el primer caracter leido
CMP AL, 3AH ; Si quedo un valor entre 30H y 39H, la suma no supero 9
; Entonces la unidad esta lista
; Y la decena tambien, ya que comienza con valor "0"

JS NUM_OK
SUB AL, 10 ; Si quedo un valor mayor a 39H
; entonces se le resta 10 para obtener la unidad
; Se suma 1 a la decena (pasa de ser el caracter "0" a "1"
NUM_OK: MOV RES_U, AL ; Copio el valor de la unidad a RES_U
MOV BX, OFFSET RES_D ; A partir de la dir. de RES_D, se imprime 2 caracteres
MOV AL, 2
INT 7
INT 0

END
```

14) Implementar un reloj similar al utilizado en los partidos de básquet, que arranque y detenga su marcha al presionar sucesivas veces la tecla F10 y que finalice el conteo al alcanzar los 30 segundos.

```

TIMER EQU 10H
PIC EQU 20H
EOI EQU 20H
N_CLK EQU 10
N_F10 EQU 20

IP_CLK DW
IP_F10 DW

SEG DB 30H ; Decena
DB 30H ; Unidad
FIN DB ?

RUT_CLK:
    ORG 40
    RUT_CLK
    ORG 80
    RUT_F10
    ORG 1000H
    PUSH AX ; Se guarda el valor de AX, porque se va a usar el registro
    INC SEG+1
    CMP SEG+1, 3AH
    JNZ RESET
    MOV SEG+1, 30H
    INC SEG
    CMP SEG, 33H
    JNZ RESET
    MOV DL, 1 ; Pongo en TRUE el flag de finalizacion
    MOV AL, 0FFH ; Deshabilito interrupciones en IMR
    OUT PIC+1, AL
RESET:
    MOV AL, 2 ; El contador tiene 2 caracteres
    INT 7 ; Se imprime el valor actual
    MOV AL, 0 ; Se vuelve a cero el contador del TIMER
    OUT TIMER, AL
    MOV AL, EOI ; Se finaliza la atencion de la interrupcion
    OUT PIC, AL
    POP AX ; Se recupera el valor que contenia AX al entrar en la rutina
    IRET

RUT_F10:
    ORG 3500H
    PUSH AX ; Se guarda el valor de AX, porque se va a usar el registro
    IN AL, PIC+1 ; Recupero el valor actual del IMR
    XOR AL, 00000010B ; Y cambio la linea correspondiente al TIMER
    OUT PIC+1, AL
    MOV AL, EOI ; Se finaliza la atencion de la interrupcion
    OUT PIC, AL
    POP AX ; Se recupera el valor que contenia AX al entrar en la rutina
    IRET

    ORG 2000H
    CLI
    MOV AL, 0FEH
    OUT PIC+1, AL ; PIC: registro IMR
    MOV AL, N_F10
    OUT PIC+4, AL ; PIC: registro INT0, F10
    MOV AL, N_CLK
    OUT PIC+5, AL ; PIC: registro INT1, TIMER
    MOV AL, 1
    OUT TIMER+1, AL ; TIMER: registro COMP
    MOV AL, 0
    OUT TIMER, AL ; TIMER: registro CONT
    MOV BX, OFFSET SEG ; Direccion del contador
    MOV DL, 0
    STI
LAZO:
    CMP DL, 0
    JZ LAZO
    INT 0

END

```