

05) Verificación de carácter numérico

```

ORG 1000H
ERROR DB "CARACTER NO VALIDO"
MSJ DB "INGRESE UN NUMERO:"
FIN DB ?

```

```

ORG 1500H
NUM DB ?

```

```

ORG 3000H
; Precondición: en BX está la dirección del carácter ingresado.
; Precondición: existe un mensaje de error definido. Retorno en AH
; Optimización: un dígito es aquel que se encuentra entre 30H y 39H

```

```

ES_NUM:    PUSH BX
            MOV AL, [BX]
            CMP AL, 30H      ; 1º condición: AL mayor o igual a "0"
            JS FALSE
            CMP AL, 40H      ; 2º condición: AL menor o igual a "9"
            JS TRUE
FALSE:      MOV BX, OFFSET ERROR
            MOV AL, OFFSET MSJ-OFFSET ERROR
            INT 7
            MOV AH, 0        ; Caso falso, se devuelve 0 en AH
            JMP FIN1
TRUE:       MOV AH, 0FFH     ; Caso verdadero
FIN1:       POP BX
            RET

```

```

ORG 2000H
MOV BX, OFFSET MSJ
MOV AL, OFFSET FIN-OFFSET MSJ
INT 7
MOV BX, OFFSET NUM
INT 6
CALL ES_NUM
MOV AL, 1
INT 7
MOV CL, NUM
INT 0
END

```

06) Impresión en letras de dígitos ingresados

; Nota: cada dígito en palabras ocupará 6 bytes en memoria.

```

ORG 1000H
PAL DB "CERO UNO  DOS  TRES  CUATROCINCO SEIS SIETE OCHO  NUEVE "
ERROR DB "CARACTER NO VALIDO"
MSJ DB "INGRESE UN NUMERO:"
FIN DB ?

```

```

ORG 1500H
NUM DB ?
ANTERIOR DB 0

```

```

ORG 3000H
ES_NUM: ...      ; Importado de ejercicio anterior

```

```

ORG 3500H      ; Precondición: BX = OFFSET NUM; se definió PAL.

```

```

CAST:      MOV AH, [BX]      ; Guardo dígito en registro
              MOV BX, OFFSET PAL
              SUB AH, 30H      ; Casteo a Entero
              JZ FIN2        ; Si es "0", no desplazar
LOOP2:    ADD BX, 6        ; "Avance" a siguiente palabra
              DEC AH        ; Corrimientos restantes
              JNZ LOOP2
FIN2:      RET            ; Se devuelve dirección en BX

```

```

ORG 2000H
BUCLE:    MOV BX, OFFSET MSJ
              MOV AL, OFFSET FIN-OFFSET MSJ
              INT 7
              MOV BX, OFFSET NUM
              INT 6
              CALL ES_NUM    ; Subrutina de verificación
              CMP AH, 0FFH   ; Si no es dígito, reintentar
              JNZ BUCLE
              CMP WORD PTR [BX], 3030H
              JZ TERMINAR   ; Se volvió a ingresar "0"
              MOV AL, 6     ; Se imprimirán 6 letras máximo
              CALL CAST     ; Retorna dirección de palabra
              INT 7
              MOV AL, NUM
              MOV ANTERIOR, AL
              JMP BUCLE

```

```

TERMINAR: INT 0
              END

```

07) Impresión de la suma de dos dígitos ingresados

```
ORG 1000H
ERROR DB "CARACTER NO VALIDO"
MSJ DB "INGRESE UN NUMERO:"
FIN DB ?
```

```
ORG 1500H
NUM DB ?
RES DW ?
```

```
ORG 3000H
ES_NUM: ... ; Importado de ejercicio 5
```

```
ORG 2000H ; Para simular Big Endian en memoria, invertimos CX
MOV CX, 20H ; Por defecto, CL (decena) tendrá un "blanco"
```

```
BUCLE:  MOV BX, OFFSET MSJ
        MOV AL, OFFSET FIN-OFFSET MSJ
        INT 7
        MOV BX, OFFSET NUM
        INT 6
        CALL ES_NUM ; Subrutina de verificación
        CMP AH, 0FFH ; Si no es dígito, reintentar
        JNZ BUCLE
        CMP CH, 0 ; ¿Es el primer operando?
        JNZ OPERAR
        MOV CH, NUM ; Guardar en registro
        JMP BUCLE
```

```
OPERAR: ADD CH, NUM ; Suma de caracteres
        SUB CH, 30H ; Reconvertir a dígito
        CMP CH, 40H ; ¿Es menor o igual a "9"?
        JS PRINT
        MOV CL, 31H ; Caso decena: "1"
        SUB CH, 10 ; Resto 10 unidades
```

```
PRINT:  MOV AL, 2
        MOV BX, OFFSET RES
        MOV [BX], CX
        INT 7
        INT 0
        END
```

08) Impresión de la resta de dos dígitos ingresados

ORG 1000H

ERROR DB "CARACTER NO VALIDO"

MSJ DB "INGRESE UN NUMERO:"

FIN DB ?

ORG 1500H

NUM DB ?

RES DW ?

ORG 3000H

ES_NUM: ... ; Importado de ejercicio 5

ORG 2000H ; Para simular Big Endian en memoria, invertimos CX

MOV CX, 20H ; Por defecto, CL (signo) tendrá un "blanco"

BUCLE: MOV BX, OFFSET MSJ
 MOV AL, OFFSET FIN-OFFSET MSJ
 INT 7
 MOV BX, OFFSET NUM
 INT 6
 CALL ES_NUM ; Subrutina de verificación
 CMP AH, 0FFH ; Si no es dígito, reintentar
 JNZ BUCLE
 CMP CH, 0 ; ¿Es el primer operando?
 JNZ OPERAR
 MOV CH, NUM ; Guardar en registro
 JMP BUCLE

OPERAR: **SUB CH, NUM** ; Resta de caracteres
 JNS PRINT
 MOV CL, 2DH ; Se colocará signo "-"
 DEC CH ; MSX88 tiene doble cero
 NEG CH ; Obtengo el módulo
 PRINT: **ADD CH, 30H** ; Reconvertir a dígito

MOV AL, 2
 MOV BX, OFFSET RES
 MOV [BX], CX
 INT 7
 INT 0
 END