



Procedimientos Estadísticos

—

Hector Josue Orozco Salazar

201314296

MIA - A

Proyecto 2

Manual Tecnico

INTERRUPTORES

El programa cuenta con el uso de interruptores para realizar dichas funciones sobre los data del sistema las cuales se usaron los interrumpes y subfunciones del interrumpes 21h

```
mov ah,08  
int 21h
```

```
; ingreso de numero y verif  
lea dx, mensaje_ingreso  
mov ah, 09h  
int 21h  
mov bx, 0
```

```
;----- impresion por UM,C,D,U  
mov ah,02h  
mov dx,unidadesMil_factorial  
add dx,30h  
int 21h
```

```
mov ah,3fh  
mov bx,handler  
mov dx,offset fragmento  
mov cx,1  
int 21h
```

MACROS

El programa tendrá la implementación de macros los cuales fueron ubicadas hasta en la parte de arriba antes de la programación, las macros que se implementaron para la impresión de las variables o cadenas que se usan a lo largo del programa para ahorro de código

```
; ----- macro para impresion de  
11 references  
PRINT macro cadena  
    mov ax,@data  
    mov ds,ax  
    mov ah,09  
    mov dx,offset cadena  
    int 21h  
2 references  
endm  
  
0 references  
IMPRIMIR MACRO msg  
    push ax  
    push dx  
    mov ah,09H  
    mov dx,offset msg  
    int 21H  
    pop dx  
    pop ax  
ENDM
```

Las siguientes macros son macros para la limpieza de la pantalla y su reinicio del cursor para impresiones futuras necesarias y de igual manera se implementó una macro la cual se usa muy pocas veces para hacer pausas para hacer espera al usuario en pantalla para ciertas acciones

```
; ----- macro para limpiar la pantalla
2 references
LIMPIAR_PANTALLA macro
    mov ah,00h
    mov al,03h
    int 10h
2 references
endm

; ----- macro para poner una espera en pantalla
3 references
PAUSA_PANTALLA macro
    mov ah,08
    int 21h
2 references
endm
```

macro para la impresión de caracter cualquiera cuando se necesitó algún símbolo específico por pantalla

```
; -----
1 reference
PRINT_CARACTER macro caracter
    mov ah,02h
    mov dl,caracter
    int 21h
3 references
endm
```

macros para utilidades en los vectores insertar, obtener, imprimir, actualizar

```

169
170 ; ===== macro para guardar numero dentro de un vector
171 3 references
172 INSERT_VECTOR macro vector,posicion,caracter
173     mov bx,0           ; borrador lo que tenga bx para usarlo
174     mov bl,posicion
175     mov si,bx          ; contador se mueve a si
176     mov bl,caracter    ; muevo el caracter
177     mov vector[si],bl ; guardo el caracter en la posicion que resive
178 27 references
179 endm
180
181 ; - - - - -
182
183 ; ===== macro para guardar numero dentro de un vector
184 1 reference
185 GET_NUMBER_VECTOR macro vector,posicion
186     mov bx,0           ; borrador lo que tenga bx para usarlo
187     mov bl,posicion
188     mov si,bx          ; contador se mueve a si
189     mov bl,vector[si] ; obtiene el caracter del vector en posicion si y guarda en bl
190 27 references
191 endm
192
193 ; - - - - -
194
195 ; ===== obtener el numero binario del vector
196 8 references
197 GET_NUMBER_BINARY macro vector,posicion,variable
198     mov bx,0           ; borrador lo que tenga bx para usarlo
199     mov bl,posicion
200     mov si,bx          ; contador se mueve a si
201     mov bl,vector[si] ; obtiene el caracter del vector en posicion si y guarda en bl
202     mov variable,bl
203 27 references
204 endm
205
206 ; - - - - -
207
208 ; ===== insertar en un vector binario
209 5 references
210 SET_VECTOR_BINARY macro vector,posicion,variable
211     ;mov si,0
212     mov bx,0           ; borrador lo que tenga bx para usarlo
213     mov bl,posicion
214     mov si,bx          ; contador se mueve a si
215     mov bl,variable
216     mov vector[si],bl
217 27 references
218 endm
219
220 ; - - - - -
221

```

macro para hacer una copia de un vector

```
; ===== macro para hacer una copi de un vector
2 references
COPY_PASTE_VECTOR macro vector_fuente,vector_destino,tamano_vector,size_copia

    ;-- etiquetas locales de la macro
    local for,fin_for

    ;--- variables utilizadas en el ciclo
    mov inicio_for_copia,0d ;variable para el for
    mov numero_a_copiar,0d ;
    mov variable_copia,0d ;variable para copiar el numero

    mov bx,0
    mov bl,tamano_vector
    mov si,0d

    ;-> incio del ciclo for
    5 references
    for:
        ;-> condicion de salida del ciclo for (mayor o igual)
        cmp inicio_for_copia,bl
        jge fin_for

        ;-> obtener el numero en cierta poscion del vector original
        GET_NUMBER_BINARY vector_fuente,inicio_for_copia,variable_copia

        ;->depositar el numero en el vector destino (copia)
        SET_VECTOR_BINARY vector_destino,inicio_for_copia,variable_copia

        ;-> incremento en el for
        inc inicio_for_copia

        ;-> incremeo el tamaño del vector copia
        inc size_copia

        ;-> para comprara con bl en el for por si se pierde el dato en bl
        mov bl,tamano_vector

        ;-> regreos a la etiqueta for
        jmp for

    ;-> termina el cilo for
    2 references
    fin_for:
```

macro para escribir los numeros de la frecuencia en el archivo del reporte

```
; ===== macro para escribir numero en el reporte txt
9 references
NUMBERS_WRITE_IN_FILE macro numero,contador

    local DOWHILE2,ESCRIBIR_ASCII_TXT2,FIN_ESCRIBIR_ASCII_TXT2

    ;->limpieza
    mov contador,0D
    mov dx,0d
    mov ax,0d

    mov ax,numero

    2 references
    DOWHILE2:

        MOV DX,0D
        MOV CX,10D
        DIV CX

        ;-> empuje a la pila y aumento numero en pila
        PUSH DX
        INC contador

        ;-> comparamo el resultado del cociente
        CMP AL,0
        JNLE DOWHILE2

    ;-> scando de la pila para escirbir en el archivo de txt
    MOV AX,0

    2 references
    ESCRIBIR_ASCII_TXT2:

        ;-> condicion de salida
        ;-> salta si el contador es igual a 0
        CMP contador,0
        JE FIN_ESCRIBIR_ASCII_TXT2

        POP DX
        MOV numero_ingresador_en_pila,dx
        ADD numero_ingresador_en_pila,30h
        DEC contador

        ;->ESCRIBE ENEL ARCHIVO DE TEXTO
```

macro para escribir texto dentro del archivo que se genera reporte

```
; ===== macro para escribir en archivo de reporte
```

```
33 references
```

```
WRITE_IN_FILE macro cadena,tamano_cadena
```

```
    mov ah,40h
```

```
    mov bx,handler2
```

```
    mov cx,tamano_cadena
```

```
    mov dx,0
```

```
    mov dx,offset cadena
```

```
    int 21h
```

```
16 references
```

```
endm
```

```
; - - - - -
```


macro para ordenar la tabla de frecuencia

```

; ===== macro para ordenar la frecuencia descendientemente
; reference
FRECUENCIA_DES macro vector_frecuencia,vector_valor,size_vector

    local for_frecuencia_des,intercambio_frecuencia_des,for_frecuencia_j_des,fin_fre_j_des,fin_fre_j_des

    ;----- limpieza y copia de variables
    mov i,0d
    mov j,0d
    mov temporal,0d
    mov valor_en_posicion_j,0d
    mov valor_en_posicion_j_masUno,0d
    mov valor_en_posicion_i, 0d
    mov siguiente_j,1d

    mov ax,0
    mov ax,size_vector
    mov size_copia2,ax
    dec size_copia2
    mov cx,0
    mov dx,0
    mov bx,0
    mov ax,0
    mov cx,size_vector
    mov dx,size_copia2

;-----
; 2 references
for_frecuencia_des:

    ;-- condicion de salida
    ;-- salta si i es mayor a dx
    mov cx,size_vector
    cmp i,cx
    jnle fin_frecuencia_des

    ;---- for interno
    ; 3 references
    for_frecuencia_j_des:

        ;---- condicion de salida
        mov dx,size_copia2
        cmp j,dx
        jnl fin_fre_j_des

        ;--- if vector[j] > vector[j+1]
        GET_NUMBER_BINARY vector_frecuencia,j,valor_en_posicion_j
        GET_NUMBER_BINARY vector_frecuencia,siguiente_j,valor_en_posicion_j_masUno

        ;----valor de referencia de la frecuencia de j+1
        GET_NUMBER_BINARY vector_valor,siguiente_j,numero_en_posicion_j_masUno

        mov ax,valor_en_posicion_j
        mov bx,valor_en_posicion_j_masUno

        ;--- si el numero vector[j] < a vector[j+1]
        cmp valor_en_posicion_j,bx
        jng intercambio_frecuencia_des

        ;--- regresa
        inc j
        inc siguiente_j
        jmp for_frecuencia_j_des

    ;-- intercambio de posiciones relativos en la frecuencia

```

DATA

El apartado de la data el cual consta con todas las variables, arreglos los cuales son utilizados a lo largo en cada uno de los modos del programa

```
; --> para imprimir un salto de linea por consola
salto_linea db ' ',10,13,'$'

; --> comando para mostara por consola
comando_consola db 'consolap2> ','$'

; --> vector para guardar el comando por entrada
comando_entrada db 100 dup('$')
contador db 0

; --> palabras reservadas
tk_cprom db 'cprom','$';5
tk_cmediana db 'cmediana','$';8
tk_cmoda db 'cmoda','$';5
tk_cmax db 'cmax','$';4
tk_cmin db 'cmin','$';4
tk_gbarra_asc db 'gbarra_asc','$';10
tk_gbarra_desc db 'gbarra_desc','$';11
tk_ghist db 'ghist','$';5
tk_glinea db 'glinea','$';6
tk_limpiar db 'limpiar','$';
tk_reporte db 'reporte','$';7
tk_info db 'info','$';4
tk_salir db 'salir','$';5
tk_abrir db 'abrir','$';6

; --> informacion para mostrar por pantalla
msg_informacion db 'ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1',13,10
                db 'SECCION A',13,10
                db 'PRIMER SEMESTRE 2021',10,13
                db 'HECTOR JOSUE OROZCO SALAZAR',10,13
                db '201314296',10,13
                db 'PROYECTO 2 ASSEMBLER',10,13,'$'

msg_diferentes db 'El comando no se reconoce',10,13,'$'
msg_iguales db 'Los comando son iguales',10,13,'$'

;----- utilidades para la apertura del archivo
msg_apertura_archivo db 'Ingrese la ruta del archivo: ','$'
posicion_en_comando db 6d
caracter db 0
nombre_archivo db 100 dup('$')
posicion_nombre_archivo db 0

handler dw ?
msg_error1 db 'Error no se puede abrir el archvo',10,13,'$'
msg_error2 db 'Error no se puede leer el archivo',10,13,'$'
fragmento db 2 dup('$')
contador_numeros_entrada db 0

; ----- utilidades para guardar los archivos de entrada
vector_entrada dw 500 dup('$')
posicion_vector_entrada dw 0
aux_unidades db 0
respaldo_registro_ax dw 0
respaldo_registro_cx dw 0
size_vector dw 0
inicio_vector dw 0
```

```

; ----- utilidades para imprimir los numeros del vector binario
cantidad_numero_en_pila dw 0
resultado_division dw 0
numero_aux_entrada dw 0
contImpVector dw 0
auxRegAX dw 0 ;respaldos ax
auxRegCX dw 0 ; respaldo de cx

; ----- utilidades para el ordenamiento burbuja
i dw 0
j dw 0
temporal dw 0
size_copia2 dw 0
valor_en_posicion_j dw 0
valor_en_posicion_j_masUno dw 0
valor_en_posicion_i dw 0
siguiente_j dw 1

; ----- utilidades para saber el menor numero
valor_minimo dw 0

; ----- utilidades para saber el mayor numero
valor_maximo dw 0

; ----- utilidades para la impresion del valor de una variable
contador_de_decimales db 0
numero_en_pila dw 0

; ----- utilidades para encontrar el promedio
numero_a_sumar dw 0
posicion_valor dw 0
suma_total dw 0
decimales dw 0
copia_size dw 0

aux_unidades2 dw 0

contador_decimales dw 0
parte_entera dw 0
residuo dw 0
decimal dw 0

; ----- utilidades para la tabla de frecuencias
vector_frecuencia dw 500 dup('$')
numero_frecuencia dw 500 dup('$')
tamano_vector_frecuencia dw 0D

cantidad_frecuencia dw 1D
contador_frecuencia dw 0d
valor_para_frecuencia dw 0d
valor_para_frecuencia_siguiente dw 0d
posicion_valor_vector_entrada dw 0d
posicion_vector_frecuencia dw 0d
copia_size2 dw 0d

;----- variables extras para el orden DES de la frecuencia
numero_en_posicion_j_masUno dw 0

```

```

mediana dw 0
mediana_siguiete dw 0

cociente_mediana dw 0
residuo_mediana dw 0
copia_size3 dw 0d

posicion_de_mediana dw 0
posicion_de_mediana_siguiete dw 0
suma_mediana dw 0

decimales_mediana dw 0
decimal_mediana dw 0

; ----- utilidades para crear el reporte txt
archivo_reporte db 'reporte.txt',0
handler2 dw ?
contador_pila dw 0
numero_ingresador_en_pila dw 0

dia db 0
mes db 0
anio dw 0
txt_anio_2021 db '2021','$';4
txt_separador_fecha db '-','$';1

hora db 0
minutos db 0
segundos db 0
txt_separador_tiempo db ':','$';1

;separador par decimales
txt_punto_decimal db '.','$';1

; PARA AGREGAR DECIMALES AL ARCHIVO DE TEXTO
decimal_txt dw 0

;promedio
promedio_parte_entera dw 0
promedio_residuo dw 0

;mediana
mediana_parte_entera dw 0
mediana_parte_decimal dw 0

;CONTADOR PARA ESCRIBIR LA FRECUENCIA
contador_escritura_frecuencia dw 0
numero_frecuencia_a_escribir dw 0
valor_de_frecuencia_a_escribir dw 0

txt_separador_frecuencia db '<--->','$';6

txt_nueva_linea db ' ',13,10,'$';2

txt_encabezado db "Universidad de San Carlos de Guatemala",13,10;39
db "Facultad de Ingenieria",13,10;23
db "Arquitectura de Computadores y Ensambladores 1",13,10;47
db "Primer Semestre 2021",13,10;21
db "Seccion A",13,10;10
db "Hector Josue Orozco Salazar",13,10;28
db "201314296",13,10,'$';10

txt_promedio db "Promedio: ",'$';10
txt_mediana db "Mediana: ",'$';10

```

ETIQUETAS

El programa hace uso de etiquetas para la programación de trozos de códigos y poder tener los flujos de cada modo del programa por separado y mantener un orden, para el cambio del flujo del programa del cual hace uso de los jmp para hacer salto hacia dichas etiquetas de cada modo

METODO MAIN - INICIO DEL PROGRAMA

el método main el cual es el flujo principal del programa la cual es el que hace el llamado a los demás flujos principales del programa según se el caso del ingreso de usuario en el menú principal

```
16 references
ingreso_comando:

    PRINT comando_consola

    mov cx,100
    mov si,0

    ; inicio de la lectura del comando
    1 reference
    lectura:

        ; verificacion si el contador es igual que al tama
        cmp si,cx
        jge fin_lectura

        ;ingreso de caracter por consola
        mov ah,01
        int 21h

        ; verificacion de la tecla enter
        cmp al,13d
        je fin_lectura

        ; guardar el caracter leído en el vector path
        mov comando_entrada[si],al
        inc si
        jmp lectura

    ; fin de la lectura del comando
    2 references
    fin_lectura:
        mov comando_entrada[si],0

    ; --> analisis de la entrada
    PRINT salto_linea
    ;PRINT comando_entrada
    ;PRINT salto_linea

    ; == swith case para comparar el comando de entrada
    ; == con las palabras reservadas aceptadas por el pro
```

Etiqueta para calcular el promedio

```
; -> se a ingresado el comando cprom
1 reference
comando_cprom:

    ; reinicio de variables
    MOV suma_total,0D
    MOV posicion_valor,0D
    MOV decimales,0D
    MOV contador_decimales,0D

    mov ax,0
    mov bx,0
    mov cx,0
    mov cx,size_vector
    mov copia_size,cx

    ;PAUSA_PANTALLA

    ;-> recorrer el vector y ir sumando sus valores
    1 reference
    FOR_PROMEDIO:

        ;->condicion de salida
        cmp copia_size,0
        je FIN_FOR_PROMEDIO

        ;-> obtener el valor que se va a sumar
        GET_NUMBER_BINARY vector_entrada,posicion_valor,numero_a_sumar

        ;-> se suma el valor en una variable total
        mov ax,numero_a_sumar
        add suma_total,ax

        ;-> pasa a la siguiente posicion del vector
        inc posicion_valor

        ;-> decrementa cantidad numeros en el vector
        dec copia_size

        ;-> repite el ciclo
        jmp FOR_PROMEDIO

    1 reference
    FIN_FOR_PROMEDIO:
```

Etiqueta para calcular la Mediana

```

; -> se a ingresado el comando cmediana
1 reference
comando_cmediana:

; -> se ordena de menor a mayor
; -> ordena de menor a mayor se obtiene el primer valor
; -> prueba para ver el ordenamiento burbuja ascendente
ORDENAMIENTO_BURBUJA_ASC vector_entrada,size_vector

; PAUSA_PANTALLA

; -> limpiamos registro para trabajar
mov ax,0
mov bx,0
mov cx,0
mov dx,0

; hacemos una copia del tamaño del vector de entrada
mov dx,size_vector

mov copia_size3,dx

; limpiamos
mov dx,0

; dividimos el tamaño del vector en dos para saber si es par o impar
mov ax,copia_size3
mov bx,2
div bx

; comparamos el residuo para saber si el número es par o impar
; dx = 0 es par
; dx != 0 es impar

; si dx =
cmp dx,0D
je MEDIANA_PAR

jmp MEDIANA_IMPAR

; -> CUANDO EL NUMERO SEA PAR
1 reference
MEDIANA_PAR:
; ->

```

```
NUMBER_BINARY_ASCII cociente_mediana
```

```
; -> IMPRESION DEL PUNTO
```

```
PRINT_CARACTER '.'
```

```
; -> DECIMAL DE LA MEDIANA
```

```
0 references
```

```
FOR_DECIMAL_MEDIANA:
```

```
    mov ax, 0D
```

```
    mov ax, residuo_mediana
```

```
; -> CONDICION DE SALIDA DE CICLO
```

```
    cmp decimales_mediana, 1D
```

```
    je FIN_FOR_DECIMAL_MEDIANA
```

```
; -> LIMPIEZA
```

```
    mov bx, 0D
```

```
    mov bx, 10D
```

```
    mul bx
```

```
    mov bx, 0D
```

```
    mov bx, 2D
```

```
    div bx
```

```
    mov decimal_mediana, ax
```

```
; -> guardo la parte decimal para imprimir en el report txt
```

```
    mov mediana_parte_decimal, ax
```

```
NUMBER_BINARY_ASCII decimal_mediana
```

```
; -> FINALIZACION DEL CICLO
```

```
1 reference
```

```
FIN_FOR_DECIMAL_MEDIANA:
```

```
PRINT salto_linea
```

```
PRINT salto_linea
```


Etiqueta para moda

```

;*****
;
; -> se a ingreado el comando cmoda
1 reference
comando_cmoda:

    MOV moda,0D
    MOV moda2,0D
    MOV tamaño_vector_frecuencia,0D

; -> orden de menor a mayor
ORDENAMIENTO_BURBUJA_ASC vector_entrada,size_vector

; -> calculo de la moda se usa para la tabla de frecuencias
    MOV cantidad_frecuencia,1D
    MOV contador_frecuencia,0d
    MOV valor_para_frecuencia,0d
    MOV valor_para_frecuencia_siguiente,0d
    MOV posicion_valor_vector_entrada,0d
    MOV posicion_vector_frecuencia,0d
    MOV copia_size2,0d
    mov ax,0
    mov bx,0
    mov cx,0
    mov dx,0

; -> inicio del for para ir sacando la frecuencia
2 references
FOR_FRECUENCIA:

    ; -> reviso el tamaño del vector de entrada
    mov dx,size_vector

    ; -> condicion de salida para el for de frecuencia
    cmp copia_size2,dx
    jge FIN_FOR_FRECUENCIA

    ; -> obtengo el valor para la frecuencia
    GET_NUMBER_BINARY vector_entrada,posicion_valor_vector_entrada,va

    ; -> aumenta en uno
    inc copia_size2
    inc posicion_valor_vector_entrada

```

Etiqueta para calcular el máximo

```

;*****
;
; -> se a ingreado el comando cmax
1 reference
comando_cmax:

    ;-> ordenamiento del vector descendente
    ORDENAMIENTO_BURBUJA_DES vector_entrada,size_vector
    ;PRINT salto_linea
    ;IMPRIMIRVECTOR vector_entrada,size_vector

    ;->obtiene el valor maximo = primer valor del vector
    GET_NUMBER_BINARY vector_entrada,0,valor_maximo

    ;->impresion del comando de consola
    PRINT comando_consola

    ;-> impresion del valor minimo
    NUMBER_BINARY_ASCII valor_maximo

    PRINT salto_linea
    PRINT salto_linea

    jmp ingreso_comando

```

Etiqueta para calcular el mínimo

```

;*****
;
; -> se a ingresado el comando cmin
1 reference
comando_cmin:

    ;-> ordena de menor a mayor se obtiene el primer valor
    ; -> prueba para ver el ordenamiento burbuja ascendente
    ORDENAMIENTO_BURBUJA_ASC vector_entrada,size_vector
    ;PRINT salto_linea
    ;IMPRIMIRVECTOR vector_entrada,size_vector

    ;-> obtiene el valor minimo = primer valor del vector
    GET_NUMBER_BINARY vector_entrada,0,valor_minimo

    ;-> impresion del comando de consola
    PRINT comando_consola

    ;-> impresion del valor minimo
    NUMBER_BINARY_ASCII valor_minimo

    PRINT salto_linea
    PRINT salto_linea

    jmp ingreso_comando

```

Etiqueta para abrir un archivo

```

; *****
; -> se a ingresado el comando para abrir un archivo
1 reference
comando_abrir:
    ;REINICIO DE VARIABLES PARA LA LECTURA DE UN NUEVO VECTOR
    MOV posicion_en_comando,6D
    MOV posicion_nombre_archivo,0D
    MOV contador_numeros_entrada,0D
    MOV posicion_vector_entrada,0D
    MOV size_vector,0D

    ;PRINT msg_apertura_archivo
    ;PAUSA_PANTALLA
    ;PRINT salto_linea

; -> extraer el nombre del archivo
1 reference
for_obtener_ruta:

    ;desde la posicion despues del _
    ; parametros: vector_origen, posicion, variable donde depositar
    GET_CHARACTER_VECTOR comando_entrada,posicion_en_comando,caracter

    ; me muevo a la siguiente posicion del comando ingresado
    inc posicion_en_comando

    ; el caracter obtenido del comando
    ;PRINT caracter

    ;parametros: vector origen, posicion, caracter
    SET_CHARACTER_VECTOR nombre_archivo,posicion_nombre_archivo,caracter
    inc posicion_nombre_archivo

    ; condicion de salida
    cmp caracter,0
    je fin_obtener_ruta
    jmp for_obtener_ruta

; ->se termino de obtener la ruta
1 reference
fin_obtener_ruta:
    ;PAUSA_PANTALLA
    PRINT nombre_archivo
    PRINT salto_linea
    mov posicion_en_comando,6d ;incio de donde tiene que estar el nombre del a

```

;-> comieza la lectura uno a uno del archivo

8 references

leer:

```
    mov ah,3fh
    mov bx,handler
    mov dx,offset fragmento
    mov cx,1
    int 21h

    jc error2

    ;EOF del archivo
    ;salta para cerrar un archivo
    cmp ax,0
    jz cerrar_archivo

    ; caracter < ej: <>49<L>
    cmp fragmento,60
    je separador

    ; mayor al rango :
    cmp fragmento,57
    jg no_es_numero

    ; menor al rango /
    cmp fragmento,48
    jl no_es_numero

    ; la lectura es un digito
    jmp es_numero
```

;-> verifica con la lectura es un numero

1 reference

es_numero:

```
    ; PRINT fragmento
    inc contador_numeros_entrada

    ;-> se toma el fragmento y se empuja a la pila
    mov bl,fragmento
    push bx
    ;PAUSA_PANTALLA

    jmp leer
```

```

; -> numero leido del archivo de entrada es un digito
1 reference
una_unidad:
    pop bx
    sub bl,30h
    mov al,1d
    mul bl
    mov aux_unidades,al

    mov respaldo_registro_ax,ax
    mov ax,posicion_vector_entrada
    mov cx,2
    mul cx
    mov si,ax
    mov ax,respaldo_registro_ax
    mov vector_entrada[si],ax

    ; incremento la posicion en el vector de entrada
    inc posicion_vector_entrada

    ; incremento el size del vector
    inc size_vector

    ; se vuelve 0 el contador de numeros leidos en el arch
    mov contador_numeros_entrada,0

    ; salta para seguir leyendo el archivo de entrada
    jmp leer

; -> numero leido del archivo de entrada es dos digitos
1 reference
dos_unidades:

    pop bx
    sub bl,30h
    mov al,1d
    mul bl

    mov aux_unidades,al

    pop bx
    sub bl,30h
    mov al,10d
    mul bl

    ; numero de dos unidades guardado en al

```

REFERENCIAS

<https://www.profesores.frc.utn.edu.ar/electronica/tecnicasdigitalesii/Practicos/TP3.pdf>

https://upload.wikimedia.org/wikipedia/commons/e/eb/MICROCOMPUTADORAS_AL_DETALLE.pdf

<http://miensamblador.blogspot.com/2013/04/ensamblador-basico-hola-todos-en-este.html>