MANUAL TECNICO

EN ESTE MANUAL ENCONTRAREMOS COMO FUE PROGRAMADA NUESTRO JUEGO

```
include macros.asm ;archivo con los macros a utilizar
include archivos.asm ; incluimos macros para manipulacion de archivos
ImprimirTablero macro vector
LOCAL Mientras, FinMientras, ImprimirSalto
push si
push di


xor si,si
mov si,1
xor di,di

        Mientras:
                cmp si,82
                je FinMientras              ; while(si<=82){}

                        mov al, vector[si]
                        mov aux, al                     ; print(arreglo[si])
                        print aux

                        cmp di,8
                        je ImprimirSalto         ; if(di == 8){ Imprimir salto}

                        mov aux,32                  ; else{print(" ")
                        print aux

                        inc di                              ;di++
                        inc si                          ; si++}
                jmp Mientras
        ImprimirSalto:
                xor di,di                        ; di = 0
                print salto                       ;print("/n")
                inc si                          ; si++
                jmp Mientras
        FinMientras:
pop di
pop si
endm


AnalizarComando macro com ; A1:B2 arreglo1 = [A][1]; arreglo2 = [B][2]
```

```
mov al,com[0]
mov posicionInicial[0],al ; arreglo1[0] = comando[0]


mov al,com[1]
mov posicionInicial[1],al ; arreglo1[1] = comando[1]


mov al,com[3]
mov posicionFinal[0],al ;  arreglo2[0] = comando[3]


mov al,com[4]
mov posicionFinal[1],al ;  arreglo2[1] = comando[4]

;-------------------------------------------------------------------------------
------
ConversionCoordenadas posicionInicial ;convierte la coordenada y la guarda en al

xor si,si ;si tiene el indicie inicial
mov si,ax

ConversionCoordenadas posicionFinal ;convierte la coordenada y la guarda en al

xor di,di ;di tiene el indice inicial
mov di,ax
;-------------------------------------------------------------------------------
--------

;aqui van validaciones

xor ax,ax

mov al, tablero[si] ;al = arreglo[si]
mov tablero[si],95 ;arreglo[si] = '_'

mov tablero[di],al ;arreglo[di] = arreglo[si]

endm



ConversionCoordenadas macro coordenada ; A1 -> 1 -> (columna) + (fila-1)*4
; ADD valor1, valor2 -> valor1 = valor1 + valor2
; MUL valor -> al = al * valor
```

```
; SUB valor1, valor2 -> valor1 = valor1 - valor2
; DIV valor -> al = al / valor -> ah tiene el residuo


; RECORDAR QUE AHORA HAY UN DESFACE DE LOS NUMEROS DADO A QUE HAY MAS NUMEROS, Y
HAY ALGUNOS USADOS PARA LAS PAREDES
; A1-> 1 -> (COLUMNA) + (FILA - 1)*9
; A1-> 1 -> (2) + (2-1)*9
; A1 = 11

mov al, coordenada[0] ; al = A = 65
mov columna, al        ; columna = 65
ConversionColumna columna ; columna convertida


mov al, coordenada[1] ; al = 1 = 49
mov fila,al                    ; fila =  49


ConversionFila fila


xor ax,ax
xor bx,bx


mov al,fila ;fila - 1
SUB al,1


mov bl,9
MUL bl  ; (fila-1)*4 -> al


xor bx,bx
mov bl,columna


ADD al,bl ;(columna) + (fila-1)*4 = al


;la conversion del resultado se guarda en al
;IntToString ax,numero
;print numero
;print salto
```

```
endm


ConversionColumna macro valor ; valor = valor - 64

; LE RESTAMOS SOLO 63 PARA QUE CONCUERDE CON EL DESFACE DE NUMEROS QUE TENEMOS PO
R LAS PAREDES
mov al,valor ; al = valor
sub al,63   ; al = al - 64
mov valor,al ; valor = al


endm


ConversionFila macro valor ; valor = valor - 48

; LE RESTAMOS SOLO 47 PARA QUE CONCUERDE CON EL DESFACE DE NUMEROS QUE TENEMOS PO
R LAS PAREDES
mov al,valor ; al = valor
sub al,47   ; al = al - 48
mov valor,al ; valor = al


endm
;-------------------------------------------------------------------------------
-------------------------------------------------
AnalizarRepo macro txtrepo
;print txtrepo[0]
LOCAL exi,letra2, letra3, exito
mov al,txtrepo[0]
mov repaux, al
print repaux
cmp repaux, 114 ; r
JE letra2
jmp exi
letra2:
mov al,txtrepo[1]
mov repaux, al
print repaux
cmp repaux, 101 ; e
JE exito
;desde aca mandamos a la etiqueta exito, por alguna razon no llega bien el valor
de la p
jmp exi
letra3:
```

```
xor ax,ax
mov al,txtrepo[2]
mov repaux, al
print repaux
cmp repaux,112 ; p
JE exito
print alert
jmp exi
exito:
print salto

call CrearHtml
print reporte
print salto
;salirr:
;print reporte2

exi:
endm
;-------------------------------------------------------------------------------
-------------------------------------------------------

.model small


;----------------SEGMENTO DE PILA---------------------

.stack 100h


;----------------SEGMENTO DE DATO---------------------

.data
msjEntrada db 0ah, 0dh, 'Universidad de San Carlos de Guatemala',0ah, 0dh,'Arquit
ectura de Ensambladores y Computadores 1' , 0ah, 0dh, 'CESAR LEONEL CHAMALE SICAN
      201700634' , 0ah, 0dh, 'Practica 3',0ah, 0dh, 'Inrese x si desea cerrar el
programa' , '$'
jp1 db 0ah, 0dh, 'Jugador 1 ingrese su nombre:', '$'
jp2 db 0ah, 0dh, 'Jugador 2 ingrese su nombre:', '$'
reporte db 0ah, 0dh, 'generando reportes ', '$'
alert db 0ah, 0dh, 'llego despues de je ', '$'
msjRepo db 0ah, 0dh, 'Ingrese rep para generar reporte: ', '$'
msjTurno db 0ah, 0dh, 'Turno del jugador:', '$'
finJuego db 0ah, 0dh, 'Ingrese x para finalizar el juego', '$'
juegaB db 0ah, 0dh, 'Juega blancas', '$'
```

```
textoComando db 0ah, 0dh, 'Ingrese su comando:', '$'
juegaN db 0ah, 0dh, 'Juega negras', '$'
punteo db 0ah, 0dh, 'Punteo actual:', '$'
salto db 0ah,0dh, '$' ,'$'
nombre1 db 10 dup('$'), '$'
nombre2 db 10 dup('$'), '$'
posicionInicial db 2 dup('$'), '$'
posicionFinal db 2 dup('$'), '$'
valorInicial db 0, '$'
valorFinal db 0, '$'
aux db 0, '$'
columna db 0, '$'
repo db 0, '$'
repaux db 0, '$'
auxPunteo1 db 0, '0'
auxPunteo2 db 0, '0'
resultado db 0, '$'
tablero db 82 dup('$'), '$'
numero db 2 dup('$'), '$'
comando db 5 dup('$'), '$' ; A1:B2
fila db 0, '$'

iniHtml db "<html><body>"
finHtml db "</body></html>"

htmlh1o db "<h1>"
htmlh1c db "</h1>"
htmlh2o db "<h2>"
htmlh2c db "</h2>"
htmlSalto db "<br/>"
; -----------para html de jugador 1-----------------
htmlJugador1 db "nombre Jugador 1: "
htmlPunteo1 db "punteo jugador 1: "
; -----------para html de jugador 2 ----------------
htmlJugador2 db "nombre Jugador 2: "
htmlPunteo2 db "punteo jugador 2: "
;---------------TABLERO ------------------------
htmlTablero db "TABLERO DE JUEGO ACTUAL: "

htmlrepo db "REPORTE DE JUEGO DE DAMAS: "

   ;=======Archivos=====
input db "REPORTE.htm",00h;Nombre
contenedor db 200 dup("$"),"$";Guardar Lectyra
handle dw ?
```

```
;----------------SEGMENTO DE CODIGO--------------------

.code

CrearHtml proc
createFile input,handle
OpenFile input,handle
WriteFile handle,iniHtml,12

;-------------- IMPRESIONES EN HTML ----------------------
WriteFile handle,htmlh1o,4
WriteFile handle,htmlrepo, 27
WriteFile handle,htmlh1c,5
WriteFile handle,htmlSalto,5
WriteFile handle,htmlSalto,5
WriteFile handle,htmlh2o,4
WriteFile handle,htmlJugador1,18
WriteFile handle,htmlSalto,5
WriteFile handle,nombre1,5
WriteFile handle,htmlSalto,5
WriteFile handle,htmlPunteo1,18
WriteFile handle,htmlSalto,5
WriteFile handle,auxPunteo1,2
WriteFile handle,htmlSalto,5
WriteFile handle,htmlSalto,5
WriteFile handle,htmlJugador2,18
WriteFile handle,htmlSalto,5
WriteFile handle,nombre2,5
WriteFile handle,htmlSalto,5
WriteFile handle,htmlPunteo2,18
WriteFile handle,htmlSalto,5
WriteFile handle,auxPunteo2,2
WriteFile handle,htmlSalto,5
WriteFile handle,htmlh2c,5
WriteFile handle,htmlh1o,4
WriteFile handle,htmlTablero,25
WriteFile handle,htmlh1c,5
;----------IMPRESION DE TABLERO
```

```
WriteFile handle,htmlh2o,4
WriteFile handle,tablero,9
WriteFile handle,htmlSalto,5
WriteFile handle,tablero[10],9
WriteFile handle,htmlSalto,5
WriteFile handle,tablero[19],9
WriteFile handle,htmlSalto,5
WriteFile handle,tablero[28],9
WriteFile handle,htmlSalto,5
WriteFile handle,tablero[37],9
WriteFile handle,htmlSalto,5
WriteFile handle,tablero[46],9
WriteFile handle,htmlSalto,5
WriteFile handle,tablero[55],9
WriteFile handle,htmlSalto,5
WriteFile handle,tablero[64],9
WriteFile handle,htmlSalto,5
WriteFile handle,tablero[73],9
WriteFile handle,htmlSalto,5
WriteFile handle,htmlh2c,5
;-------------------------------------------------------
WriteFile handle,finHtml,14
CloseFile handle
CrearHtml endp
InicializarTablero proc
                mov tablero[0], 0 ; tablero[0] = 0
                mov tablero[1], 32
                mov tablero[2], 65
                mov tablero[3], 66
                mov tablero[4], 67
                mov tablero[5], 68
                mov tablero[6], 69
                mov tablero[7], 70
                mov tablero[8], 71
                mov tablero[9], 72

                mov tablero[10], 49 ;NUMERO 1
                mov tablero[11], 78
                mov tablero[12], 95
                mov tablero[13], 110
                mov tablero[14], 95
                mov tablero[15], 110
                mov tablero[16], 95
                mov tablero[17], 110
                mov tablero[18], 95
```

```
mov tablero[19], 50 ; NUMERO 2
mov tablero[20], 95
mov tablero[21], 110
mov tablero[22], 95
mov tablero[23], 110
mov tablero[24], 95
mov tablero[25], 110
mov tablero[26], 95
mov tablero[27], 110

mov tablero[28], 51 ; NUMERO 3
mov tablero[29], 110
mov tablero[30], 95
mov tablero[31], 110
mov tablero[32], 95 ; tablero[0] = 0
mov tablero[33], 110
mov tablero[34], 95
mov tablero[35], 110
mov tablero[36], 95

mov tablero[37], 52 ; NUMERO 4
mov tablero[38], 95
mov tablero[39], 95
mov tablero[40], 95
mov tablero[41], 95
mov tablero[42], 95
mov tablero[43], 95
mov tablero[44], 95
mov tablero[45], 95

mov tablero[46], 53 ;NUMERO 5
mov tablero[47], 95
mov tablero[48], 95
mov tablero[49], 95
mov tablero[50], 95
mov tablero[51], 95
mov tablero[52], 95
mov tablero[53], 95
mov tablero[54], 95

mov tablero[55], 54 ;NUMERO 6
mov tablero[56], 95
mov tablero[57], 119
mov tablero[58], 95
```

```
                mov tablero[59], 119
                mov tablero[60], 95
                mov tablero[61], 119
                mov tablero[62], 95
                mov tablero[63], 119

                mov tablero[64], 55 ;NUMERO 7
                mov tablero[65], 119
                mov tablero[66], 95
                mov tablero[67], 119
                mov tablero[68], 95
                mov tablero[69], 119
                mov tablero[70], 95
                mov tablero[71], 119
                mov tablero[72], 95

                mov tablero[73], 56 ;NUMERO 8
                mov tablero[74], 95
                mov tablero[75], 119
                mov tablero[76], 95
                mov tablero[77], 119
                mov tablero[78], 95
                mov tablero[79], 119
                mov tablero[80], 95
                mov tablero[81], 87
                mov tablero[82], 0
        ret
InicializarTablero endp
XOR_REG proc
                xor ax, ax
                xor bx, bx
                xor cx, cx
                xor dx, dx
                ret
XOR_REG endp
main proc
        mov ax,@data
        mov ds,ax
        Menu:
                print msjEntrada
                print salto
                call InicializarTablero

                getChar ; lee un caracter del teclado y lo guarda en al
```

```
                cmp al, 120 ; if (al == 120){va a brincar a la etiqueta salir}els
e{va a continuar con el programa}
                je Salir
              ;  cmp al, 106
              ;  je

                print jp1
                obtenerTexto nombre1
                print salto

                print jp2
                obtenerTexto nombre2
                print salto



        Juego:


                Turno1:
                print msjTurno
                print nombre1
                print juegaB
                print punteo
                print salto
                ImprimirTablero tablero
                print textoComando
                obtenerTexto comando
                print salto


                AnalizarComando comando


                print finJuego
                getChar ; lee un caracter del teclado y lo guarda en al
                cmp al, 120 ; if (al == 120){va a brincar a la etiqueta salir}els
e{va a continuar con el programa}
                je Menu

                print msjRepo

                obtenerTexto repo
                ;print repo
                AnalizarRepo repo
```

```
                jmp Turno2


                Turno2:
                print msjTurno
                print nombre2
                print juegaN
                print punteo
                print salto
                ImprimirTablero tablero
                print textoComando
                obtenerTexto comando

                AnalizarComando comando

                print finJuego

                getChar ; lee un caracter del teclado y lo guarda en al
                cmp al, 120 ; if (al == 120){va a brincar a la etiqueta salir}els
e{va a continuar con el programa}
                je Menu

                print msjRepo
                obtenerTexto repo
                AnalizarRepo repo

                jmp Turno1

                jmp Menu







        Salir:
                close


main endp
end main
```