**PONTIFICIA UNIVERSIDAD CATOLICA MADRE Y MAESTRA**

****

**Nombre:**

Félix Alejandro Guzmán           2014 - 0565

**Materia:**

Arquitectura Computacional

**Profesor:**

Alvaro A. Reyes P.

**Practica sobre:**

Práctica #2

**Fecha de Entrega:**

martes, 24 de octubre de 2017

# Código

.model small

.stack 256

.data

;========================Variables declaradas aquí===========================

mensaje db 'Digite 100 caracteres o menos:','$' ;string

mensajebusqueda db 'Digite la letra o suma de letras para buscar:','$' ;string

caracteres db 101 dup (0) ; buffer para 100 bytes de caracteres

caracterbusqueda db 3 dup (0) ;buffer para caracteres de búsqueda

resultado db 'Hay: ','$' ;string

caracb1 db 1 ;buffer para caracteres de búsqueda

caracb2 db 1 ;buffer para caracteres de búsqueda

caracb3 db 1 ;buffer para caracteres de búsqueda

count dw 1

;============================================================================

.code

main:

mov ax,@data

mov ds,ax

mov es, ax ;set segment register

and sp, not 3 ;align stack to avoid AC fault

;====================================Codigo==================================

principio:

call clrscr

call inicio

mov ah,02 ;para imprimir en pantalla un carácter.

mov dl, 0ah ; cambio de línea.

int 21h ; llamada al sistema operativo.

mov bx,-1 ;índice en la posición 0

ok:

mov ah,1 ;capturar de pantalla.

int 21h ;llamada al sistema operativo.

cmp al,1Bh ;revisar si ESC fue presionado

jz salir ;función para terminar el programa

inc bx

mov caracteres[bx],al

cmp al,0Dh ;revisar si ENTER fue presionado

jz buscar

cmp bx,100 ;reviso si he tomado los 100 caracteres.

jne ok ;sigo tomando caracteres hasta tener 100.

buscar:

;mov dl, 0ah ;cambio de línea.

;int 21h ;llamada al sistema operativo.

call mensajebu

mov si,0 ;índice en la posición 0

b:

mov ah,1 ;capturar de pantalla.

int 21h ;llamada al sistema operativo.

cmp al,1Bh ;revisar si ESC fue presionado

jz salir ;función para terminar el programa

mov caracb1,al

mov ah,1 ;capturar de pantalla.

int 21h ;llamada al sistema operativo.

cmp al,1Bh ;revisar si ESC fue presionado

jz salir ;función para terminar el programa

cmp al,2Bh ;revisar si el carácter fue el +

jne calculateOne ;calcular solo el primer carácter

cmp al,0Dh ;revisar si el carácter fue el +

jz calculateOne ;calcular solo el primer carácter

mov caracb2,al

mov ah,1 ;capturar de pantalla.

int 21h ;llamada al sistema operativo.

cmp al,1Bh ;revisar si ESC fue presionado

jz salir ;función para terminar el programa

mov caracb3,al ;guardar tercer carácter en otro buffer

jmp calculateTwo ;calcular el primer y segundo carácter

;cmp si,0 ;revisar si es el primer carácter a buscar

;jz introducir1 ;introducir carácter

;cmp si,1 ;revisas si es el segundo carácter

;jz introducir2 ;introducir carácter

;cmp si,2 ;revisar si es el tercer carácter

;jz introducir3 ;introducir carácter

;============================================================================

.exit

;================================Funciones aquí==============================

inicio:

mov ah,09 ;imprimir un string.

mov dx, offset mensaje ;paso al sistema la ubicación del letrero.

int 21h ;llamada al sistema operativo.

ret

clrscr proc ;limpiar pantalla

mov ax,0003h

int 10h

ret

endp

salir:

mov ah, 4CH ;terminar programa

int 21h

ret

calculateOne:

mov di,-1 ;índice en la posición -1

mov dl,caracb1 ;tomar primer carácter

mov count,-1 ;inicio de contador en -1

incremento:

inc count ;incrementar contador

inc di ;apuntar índice a la siguiente posición

cant:

cmp caracteres[di],dl ;comparar 100 dígitos con el primero a buscar

je incremento

cmp caracteres[di],0dh ;comparar si el carácter actual es el enter

jz sigue

inc di ;apuntar índice a la siguiente posición

jmp cant

ret

sigue:

call texto3 ;texto de cantidad de letras

mov ax, count ;mover cantidad a AX para hacer la división

call decimalprint ;imprimir resultado

mov ah,1 ;capturar de pantalla.

int 21h ;llamada al sistema operativo.

jmp principio ;reiniciar

ret

calculateTwo:

mov di,-1 ;índice en la posición -1

mov count,-1 ;inicio de contador en -1

mov bl,caracb1 ;guardar primer carácter en un registro

mov dl,caracb3 ;guardar segundo carácter en un registro

incrementoa:

inc count ;incrementar contador

inc di ;apuntar índice a la siguiente posición

cant12:

cmp caracteres[di],bl ;comparar 100 dígitos con el primero a buscar

je incrementoa ;incrementar cantidad de letras

cmp caracteres[di],dl ;comparar si el carácter actual es el enter

je incrementoa ;incrementar cantidad de letras

cmp caracteres[di],0dh ;comparar si el carácter actual es el enter

jz sigue

inc di ;apuntar índice a la siguiente posición

jmp cant12

ret

mensajebu:

mov ah,09

mov dx, offset mensajebusqueda ;paso al sistema la ubicación del letrero.

int 21h ;llamada al sistema operativo.

ret

texto3:

mov ah,09

mov dx, offset resultado ;paso al sistema la ubicación del letrero.

int 21h ;llamada al sistema operativo.

ret

decimalprint proc

mov BX, 10 ;número para dividir

mov DX, 0000H ;iniciar DX en null

mov CX, 0000H ;iniciar CX en null

pushtoStack:

mov DX, 0000H ;poner DX en null

div BX ;dividir count entre 10

push DX ;guardo residuo

inc CX

cmp AX, 0 ;pregunto si la división es 0

jne pushtoStack ;mientras no sea 0 sigo dividiendo

popandPrint:

pop DX ;saco el residuo

add DX, 30H ;sumo el residuo con el código asci 30H para llevarlo a número decimal

mov AH, 02H ;imprimo en pantalla

int 21H

loop popandPrint

ret

ENDP

;============================================================================

end main