<https://youtu.be/qKTMWSb9SAw>

; Program Template (labw16.asm)

; Program Description: boolean calculator for 32-bit integers

; Author: Timothy Bryant

; Creation Date: 4/30/2021

; Revisions:

; Date:

; Modified by:

.386

.model flat,stdcall

.stack 4096

ExitProcess PROTO, dwExitCode:DWORD

INCLUDE Irvine32.inc

.data

; declare variables here

displayMenu BYTE "Boolean Calculator",0dh,0ah

BYTE 0dh,0ah

BYTE "1. x AND y" ,0dh,0ah

BYTE "2. x OR y" ,0dh,0ah

BYTE "3. NOT x" ,0dh,0ah

BYTE "4. x XOR y" ,0dh,0ah

BYTE "5. Exit program",0dh,0ah,0dh,0ah

BYTE "Enter number from menu: ",0

displayError BYTE "INVALID INPUT",0

displayAND BYTE "Boolean AND",0

displayOR BYTE "Boolean OR",0

displayNOT BYTE "Boolean NOT",0

displayXOR BYTE "Boolean XOR",0

caseTable BYTE '1' ; lookup value

DWORD AND\_op ; address of procedure

InputSize = ($ - caseTable )

BYTE '2'

DWORD OR\_op

BYTE '3'

DWORD NOT\_op

BYTE '4'

DWORD XOR\_op

BYTE '5'

DWORD ExitProgram

NumberOfInputs = ($ - caseTable) / InputSize

displayInt1 BYTE "Enter the first 32-bit hexadecimal integer: ",0

displayInt2 BYTE "Enter the second 32-bit hexadecimal integer: ",0

displayResult BYTE "The 32-bit hexadecimal result is: ",0

.code

main PROC

;write your code here

call Clrscr ; clear console window

Menu:

mov edx, OFFSET displayMenu ; menu choices

call WriteString ; display menu

call Crlf ; go to next output line

L1:

call ReadChar ; wait for input and return char

cmp al, '5' ; is selection valid (1-5)?

ja L2 ; jump if above 5, go back

cmp al, '1'

jb L2 ; jump if below 1, go back

call Crlf

call ChooseProcedure

jc quit ; jump if carry = 1, exit

call Crlf

jmp Menu ; display menu again

L2:

call Crlf

mov edx, OFFSET displayError ; error message

call WriteString ; display error

call Crlf ; go to next output line

jmp L1

quit:

INVOKE ExitProcess, 0

main ENDP

;---------------------------------------------------------------------------------

ChooseProcedure PROC

;

; Reads the user input and decides the procedure to call.

; Receives: al, the user input for the menu procedure.

; Returns: nothing

; Requires: valid input on the menu from the user

;---------------------------------------------------------------------------------

push ebx ; push EBX onto stack

push ecx ; push ECX onto stack

mov ebx, OFFSET caseTable ; pointer to the table

mov ecx, NumberOfInputs ; loop counter

L1:

cmp al, [ebx] ; match found?

jne L2 ; if no, continue

call NEAR PTR [ebx + 1] ; if yes, call procedure

jmp L3

L2:

add ebx, InputSize ; point to the next entry

loop L1 ; repeat until ECX = 0

L3:

pop ecx ; remove ECX from stack

pop ebx ; remove EBX from stack

ret ; return from procedure

ChooseProcedure ENDP

;---------------------------------------------------------------------------------

AND\_op PROC

;

; Prompt the user for two hexadecimal integers. AND them together and display the result in hexadecimal.

; Receives: nothing

; Returns: nothing

; Requires: the user to input '1'

;---------------------------------------------------------------------------------

pushad ; push all registers onto stack

mov edx, OFFSET displayAND ; name of the operation

call WriteString ; display message

call Crlf

call Crlf

mov edx, OFFSET displayInt1 ; ask for first integer

call WriteString

call ReadHex ; get hex integer

mov ebx, eax ; move first integer to EBX

mov edx, OFFSET displayInt2 ; ask for second integer

call WriteString

call ReadHex ; get second hex integer

and eax, ebx ; integer1 AND integer2

mov edx, OFFSET displayResult ; result

call WriteString ; display result

call WriteHex ; display hex to window

call Crlf

popad ; save and restore registers

ret ; return from procedure

AND\_op ENDP

;---------------------------------------------------------------------------------

OR\_op PROC

;

; Prompt the user for two hexadecimal integers. OR them together and display the result in hexadecimal.

; Receives: nothing

; Returns: nothing

; Requires: the user to input '2'

;---------------------------------------------------------------------------------

pushad ; push all registers onto stack

mov edx, OFFSET displayOR ; name of the operation

call WriteString ; display message

call Crlf

call Crlf

mov edx, OFFSET displayInt1 ; ask for first integer

call WriteString

call ReadHex ; get hexadecimal integer

mov ebx, eax ; move first integer to EBX

mov edx, OFFSET displayInt2 ; ask for second integer

call WriteString

call ReadHex ; get hex integer

or eax, ebx ; integer1 OR integer2

mov edx, OFFSET displayResult ; result of operation

call WriteString

call WriteHex ; display hex to window

call Crlf

popad ; restore registers

ret ; return from procedure

OR\_op ENDP

;---------------------------------------------------------------------------------

NOT\_op PROC

;

; Prompt the user for a hexadecimal integer. NOT the integer and display the result in hexadecimal.

; Receives: nothing

; Returns: nothing

; Requires: the user to input '3'

;---------------------------------------------------------------------------------

pushad ; push all registers onto stack

mov edx, OFFSET displayNOT ; name of the operation

call WriteString ; display message

call Crlf

call Crlf

mov edx, OFFSET displayInt1 ; ask for integer

call WriteString

call ReadHex ; get hex integer

not eax ; NOT operand

mov edx, OFFSET displayResult ; result of operation

call WriteString

call WriteHex ; EAX = result

call Crlf

popad ; restore registers

ret ; return from procedure

NOT\_op ENDP

;---------------------------------------------------------------------------------

XOR\_op PROC

;

; Prompt the user for two hexadecimal integers. Exclusive-OR them together and display the result in hexadecimal

; Receives: nothing

; Returns: nothing

; Requires: the user to input '4'

;---------------------------------------------------------------------------------

pushad ; push all registers onto stack

mov edx, OFFSET displayXOR ; name of the operation

call WriteString ; display message

call Crlf

call Crlf

mov edx, OFFSET displayInt1 ; ask for first operand

call WriteString

call ReadHex ; get hexadecimal integer

mov ebx, eax ; move first operand to EBX

mov edx, OFFSET displayInt2 ; ask for second operand

call WriteString

call ReadHex ; get hex integer

xor eax, ebx ; integer1 XOR integer2

mov edx, OFFSET displayResult ; result of operation

call WriteString

call WriteHex ; display hex to window

call Crlf

popad ; save and restore registers

ret ; return from procedure

XOR\_op ENDP

;---------------------------------------------------------------------------------

ExitProgram PROC

;

; Sets the carry flag to 1.

; Receives: nothing

; Returns: The carry flag to exit the program.

; Requires: the user to input '5'

;---------------------------------------------------------------------------------

stc ; set the carry flag to 1

ret ; return from procedure

ExitProgram ENDP

END main