Michael Gil

**6.11.2 Programming Exercises Q.5 and Q.6**

INCLUDE Irvine32.inc

.data

msgBoolCalcMenu BYTE "bool 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. Quit Program",0

msgANDSECTION BYTE " AND",0

msgORSECTION BYTE " OR",0

msgNOTSECTION BYTE " NOT",0

msgXORSECTION BYTE " XOR",0

msgInteger1 BYTE "input 32 bit hex integer: ",0

msgInteger2 BYTE "input the next 32-bit hex integer: ",0

msgEnteredHex BYTE "The 32 bit hex number input is: ",0

caseTable BYTE '1'

DWORD AND\_op

EntrySize = ($ - caseTable )

BYTE '2'

DWORD OR\_op

BYTE '3'

DWORD NOT\_op

BYTE '4'

DWORD XOR\_op

BYTE '5'

DWORD ExitProgram

EnteredNumberAmount = ($ - caseTable) / EntrySize

.code

main PROC

call Clrscr

Menu:

mov edx, OFFSET msgBoolCalcMenu

call WriteString ; menu

call Crlf

L1:

call ReadChar

cmp al, '5'

ja L1

cmp al, '1'

jb L1

call Crlf

call ProcedureChoice

jc quit

call Crlf

jmp Menu

quit: exit

main ENDP

ProcedureChoice PROC

push ebx

push ecx

mov ebx, OFFSET caseTable

mov ecx, EnteredNumberAmount

L1: cmp al, [ebx]

jne L2

call NEAR PTR [ebx + 1]

jmp L3

L2: add ebx, EntrySize

loop L1

L3: pop ecx

pop ebx

ret

ProcedureChoice ENDP

AND\_op PROC

pushad

mov edx, OFFSET msgANDSECTION

call WriteString

call Crlf

call Crlf

mov edx, OFFSET msgInteger1

call WriteString

call ReadHex

mov ebx, eax

mov edx, OFFSET msgInteger2

call WriteString

call ReadHex

and eax, ebx

mov edx, OFFSET msgEnteredHex

call WriteString

call WriteHex

call Crlf

popad

ret

AND\_op ENDP

OR\_op PROC

pushad

mov edx, OFFSET msgORSECTION

call WriteString ; display message

call Crlf

call Crlf

mov edx, OFFSET msgInteger1

call WriteString

call ReadHex

mov ebx, eax

mov edx, OFFSET msgInteger2

call WriteString

call ReadHex

or eax, ebx

mov edx, OFFSET msgEnteredHex

call WriteString

call WriteHex

call Crlf

popad

ret

OR\_op ENDP

NOT\_op PROC

pushad

mov edx, OFFSET msgNOTSECTION

call WriteString

call Crlf

call Crlf

mov edx, OFFSET msgInteger1

call WriteString

call ReadHex

not eax

mov edx, OFFSET msgEnteredHex

call WriteString

call WriteHex

call Crlf

popad

ret

NOT\_op ENDP

XOR\_op PROC

pushad

mov edx, OFFSET msgXORSECTION

call WriteString

call Crlf

call Crlf

mov edx, OFFSET msgInteger1

call WriteString

call ReadHex

mov ebx, eax

mov edx, OFFSET msgInteger2

call WriteString

call ReadHex

xor eax, ebx

mov edx, OFFSET msgEnteredHex

call WriteString

call WriteHex

call Crlf

popad

ret

XOR\_op ENDP

ExitProgram PROC

stc

ret

ExitProgram ENDP

END main