Assignment 7

**6.11.2 -**

5. Boolean Calculator

.data

oprTable BYTE ‘1’

DWORD Process\_1

EntrySize = ($-oprTable)

BYTE ‘2’

DWORD Process\_2

BYTE ‘3’

DWORD Process\_3

BYTE ‘4’

DWORD Process\_4

BYTE ‘5’

DWORD Process\_5

CountEntries = ($-oprTable) / EntrySize

str1 BYTE “1. xANDy”, 0

str2 BYTE “2. xORy”, 0

str3 BYTE “3. NOTx”, 0

str4 BYTE “4. xXORy”, 0

str5 BYTE “5. Exit Program”, 0

strCh BYTE “Enter your choice: “, 0

strA BYTE “AND operation”, 0

strO BYTE “OR operation”, 0

strN BYTE “NOT operation”, 0

strX BYTE “XOR operation”, 0

.code

main PROC

call Clrscr

call DisplayMenu

exit

main ENDP

DisplayMenu PROC USES edx

mov ed x, OFFSET str1

call WriteString

call Crlf

move dx, OFFSET str2

call WriteString

call Crlf

mov ed x, OFFSET str3

call WriteString

call Crlf

mov ed x, OFFSET str4

call WriteString

call Crlf

mov ed x, OFFSET str5

call WriteString

call Crlf

mov ed x, OFFSET strCh

call WriteString

call ReadChar

mov esi,OFFSET oprTable

move cx,CountEntrie

call DisplayOpr

call Crlf

ret

DisplayMenu ENDP

DisplayOpr PROC USES eax esi ecx edx

L1:

cmp al,[esi]

jne L2

call NEAR PTR [esi+1]

call WriteString

call Crlf

jmp L3

L2:

add esi,CountEntries

loop L1

L3:

Ret

DisplayOpr ENDP

Process\_1 PROC USES edx

mov edx,OFFSET strA

ret

Process\_1 ENDP

Process\_2 PROC USES edx

mov edx,OFFSET strO

ret

Process\_2 ENDP

Process\_3 PROC USES edx

mov edx,OFFSET strN

ret

Process\_3 ENDP

Process\_4 PROC USES edx

mov edx,OFFSET strN

ret

Process\_4 ENDP

Process\_5 PROC USES edx

ret

Process\_5 ENDP

END main

6. Boolean Calculator 2

.data

choice BYTE ?

str1 BYTE “1. xANDy” , 0

str2 BYTE “2. xORy” , 0

str3 BYTE “3. NOTx” , 0

str4 BYTE “4. xXORy” , 0

str5 BYTE “5. Exit program” , 0

strCh BYTE “Enter your choice: ” , 0

strA BYTE “AND operation” , 0

strO BYTE “OR operation” , 0

strN BYTE “NOT operation” , 0

strX BYTE “XOR operation” , 0

strOp1 BYTE “Enter one hexadecimal integer: ” ,0

strOp2 BYTE “Enter two hexadecimal integers: ” ,0

strRes BYTE “The result of operation is: ” ,0

.code

main PROC

call Clrscr

call DisplayMenu

call DisplayOpr

exit

mainENDP

DisplayMenu PROC USES edx

mov ed x, OFFSET str1

call WriteString

call Crlf

mov ed x, OFFSET str2

call WriteString

call Crlf

mov ed x, OFFSET str3

call WriteString

call Crlf

mov ed x, OFFSET str4

call WriteString

call Crlf

mov ed x, OFFSET str5

call WriteString

call Crlf

mov ed x, OFFSET strCh

call WriteString

call ReadInt

mov choice,eax

call Crlf

ret

DisplayMenu ENDP

DisplayOpr PROC USES edx

mov eax,choice

cmp eax, 1

jne Label1

mov ed x, OFFSET strA

call WriteString

call Crlf

call AND\_op

jmp LblEnd

Label1:

cmp eax,2

jne Label2

mov ed x, OFFSET strO

call WriteString

call Crlf

call OR\_op

jmp LblEnd

Label2:

cmp eax,3

jne Label3

mov ed x, OFFSET strN

call WriteString

call Crlf

call NOT\_op

jmp LblEnd

Label3:

cmp eax,4

jne LabelEnd

mov ed x, OFFSET strX

call WriteString

call Crlf

call XOR\_op

LblEnd:

ret

DisplayOpr ENDP

AND\_op PROC USES ebx edx

mov ed x, OFFSET strOp2

call WriteString

call Crlf

call ReadHex

mov ebx,eax

call ReadHex

AND ebx,eax

mov eax,ebx

mov ed x, OFFSET strRes

call WriteString

call WriteHex

call Crlf

ret

AND\_op ENDP

OR\_op PROC USES ebx edx

mov ed x, OFFSET strOp2

call WriteString

call Crlf

call ReadHex

mov ebx,eax

call ReadHex

OR ebx,eax

mov eax,ebx

mov ed x,OFFSET strRes

call WriteString

call WriteHex

call Crlf

ret

OR\_op ENDP

NOT\_op PROC USES edx

mov ed x, OFFSET strOp1

call WriteString

call Crlf

call ReadHex

NOT eax

mov ed x, OFFSET strRes

call WriteString

call WriteHex

call Crlf

ret

NOT\_op ENDP

XOR\_op PROC USES ebx edx

mov ed x, OFSSET strOp2

call WriteString

call Crlf

call ReadHex

mov ebx,eax

call ReadHex

XOR ebx,eax

mov eax,ebx

mov ed x, OFFSET strRes

call WriteString

call WriteHex

call Crlf

ret

XOR\_op ENDP

END\_main