HW7: Chap. 6: Theme: Conditionals, Booleans, Loops

1. Draft a program that scans an array testing each index for a negative value. If a negative value is found, the program should quit and show some indication that a negative value has been found. This can be as simple a "call DumpRegs." The `.data` section is provided below.

```
TITLE Scanning array            (main.asm)
; Description: Scans an array for a negative value.
; Author: Matthew J Swann
; Version 1.0, 2012-08-02

INCLUDE Irvine32.inc
.data
	myArray SWORD 3,6,1,10,-10,-30,-40,-4
	sentinel SWORD 0

.code

main PROC
	MOV esi, OFFSET myArray
	MOV ecx, LENGTHOF myArray

	L1:
		test WORD PTR [esi], 8000h
		jnz FOUND
		ADD esi, TYPE myArray
	LOOP L1

	jmp NOTFOUND

	FOUND:
		MOV eax, 0
		MOVSX eax, WORD PTR [esi]
		call WriteInt
	NOTFOUND:

exit
main ENDP
END main
```

2. In the following instruction sequence, show the changed value of AL where indicated, in hexadecimal. ***Answers without work shown, will not receive credit.***

mov al, 9Bh : 1001 1011
not al ; a. flip all bits in al → 0110 0100 → **64h**

mov al, 4Dh : 0010 1101
and al, 76h ; b. al AND (0111 0110) → 0010 0100 → **44h**

mov al, 9Ch :1001 1100
or al, 45h ; c. al OR (0010 0011) → 1011 1111 → **BFh**

mov al, 73h : 0111 0011
xor al, BDh ; d. al XOR (1011 1101) → 1100 1110 → **CEh**

3. Implement the following pseudo-code in assembly language (assume signed numbers):

A. if (bx = > cx AND bx != val1)
X =1;
Else
X =2;

```
TITLE  Logic Statement A    (main.asm)
; Description: Performs comparisons.
; Author: Matthew J Swann
; Version 1.0, 2012-08-02

INCLUDE Irvine32.inc
.data
	val1 WORD 5
	x BYTE ?
.code
main PROC
	MOV ebx, 6			;sample value for ebx
	MOV ecx, 1			;sample value for ecx

	cmp bx, cx			;compare bx to cx
	jl FAIL				;if bx < cx jump to FAIL

	CHECKNE:			;if it gets here, bx >= cx
		PUSH bx
		SUB bx, val1
		POP bx
		jnz PASS		;if bx == val1 jump to PASS
		jz FAIL			;else jump to FAIL

	PASS:
		MOV x, 1
		exit
	FAIL:
		MOV x, 2
exit
main ENDP
END main
```

B. if(ax < dx OR cx == val1)
X = 3
Else
X = 4

```
TITLE  Logic Statement B    (main.asm)
; Description: Performs comparisons.
; Author: Matthew J Swann
; Version 1.0, 2012-08-02

INCLUDE Irvine32.inc
.data
	val1 WORD 5
	x BYTE ?
.code
main PROC
	MOV eax, 0			;sample value for eax
	MOV ecx, 5			;sample value for ecx
	MOV edx, 5			;sample value for edx

	cmp ax, dx			;compare bx to cx
	jl PASS				;if ax < dx jump to PASS

	CHECKNE:			;if it gets here, bx >= cx
		PUSH cx
		SUB cx, val1
		POP cx
		jz PASS			;if cx == val1 jump to PASS
		jnz FAIL		;else jump to FAIL

	PASS:
		MOV x, 3
		exit
	FAIL:
		MOV x, 4
exit
main ENDP
END main
```