Comp 3350: Computer Organization & Assembly Language
HW # 7: Theme: Conditionals, Booleans, Loops

1. Draft a program that scans an array testing each index for a positive value. If a positive value is found the program should print "found" and the value. If no positive value is found the program should print "not found."
   Use:

```
.data
      myArray1 SWORD -12,4,1,23,-21,45,12,-2
      sentinel SWORD 0
```

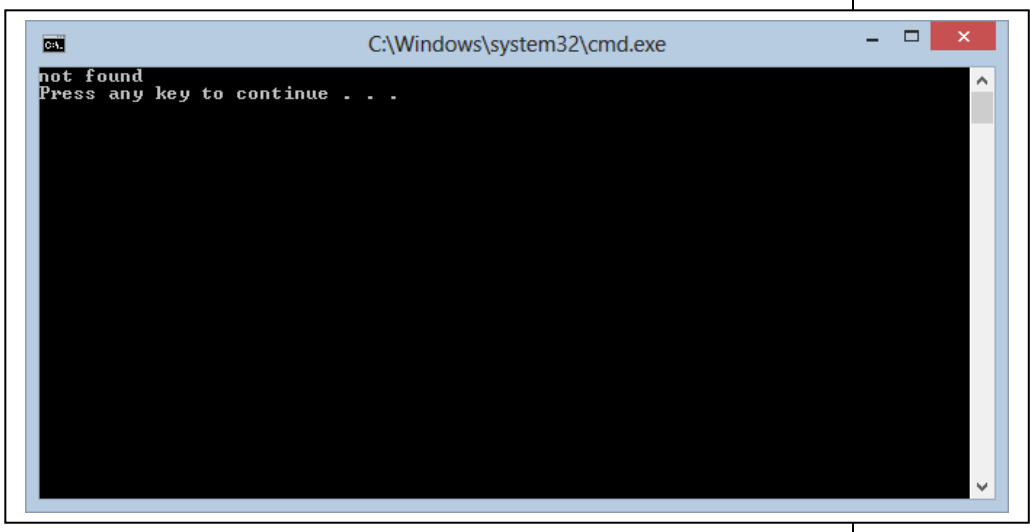
   Next, change the array so that all values are negative and show the run.

```
TITLE POSITIVE VALUE FINDER
Include Irvine32.inc

.data
myArray1 SWORD -12,-4,-23,-21,-45,-12,-2
messageS BYTE "found",0
messageF BYTE "not found",0
.code
main PROC
mov ecx, LENGTHOF myArray1
mov ebx, 0
L1:
mov ax, [myArray1 + ebx]
TEST ax, 8000h
jz MSG1
add ebx, TYPE myArray1
LOOP L1
mov edx, OFFSET messageF
CALL WriteString
CALL Crlf
jmp exitP

MSG1:
mov edx, OFFSET messageS
CALL WriteString
CALL Crlf

exitP:
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, C2h
not al
```

FF
-C2
3D (one's complement)

mov al, D3h
and al, 21h

11010011	01h
/\ 00100001	
00000001	

mov al, F3h
or al, 7Dh

11110011	0FFh
\/ 01111101	
11111111	

mov al, 2Dh
xor al, E5h

00101101	C8h
Xor 11100101	
11001000	

3. Implement the following pseudo-code in assembly language (assume unsigned numbers). Declare Apple and Pear as byte sized variables.

A. if ((cx >= bx) OR (cx != val1))
    Apple = 10;
  Else
    Apple = 20;

```
TITLE 3a
Include Irvine32.inc

.data
apple BYTE ?
val1 WORD 30

.code
main PROC
mov bx, 20h
mov cx, 10h
cmp cx, bx
jae label1
cmp cx, val1
jne label1
mov apple, 10
jmp exitp
label1:
mov apple, 20
exitp:
exit
main ENDP
END main
```

B. if ((bx <= dx) OR (dx = val1))
Pear = 0;
Else
Pear = 1;

```
TITLE 3b
Include Irvine32.inc

.data
pear BYTE ?
val1 WORD 10

.code
main PROC
mov bx, 05h
mov dx, 10h
cmp bx, dx
jbe label1
cmp dx, val1
je label1
mov pear, 0
jmp exitp
label1:
mov pear, 1
exitp:
exit
main ENDP
END main
```