1. What is the value (in hexadecimal) of AL, AH, and AX given the following hexadecimal values in the EAX register? (1) 37E11449 (2) 8A29713D

(1) AH: 14 AL: 49 8 bits

AX: 1449

(2) AH: 71 AL: 3D

AX: 713D

2. Calculating the Size of a Word Array

List WORD 1000h, 2000h, 3000h, 4000h

ListSize = ($ - List) / 2

3. True (T) / False (F)

.data

cout BYTE 100

Val WORD 2

.code WORD: 2 bytes -> 16 bits

3.1 mov al, Val (T/F)

3.2 mov ax, count (T/F)

movzx -> BYTE -> 8 bits

16 bits

(not necessarily correct if register isn't clear)

4. Implement Arithmetic Expressions: R = -X + (Y - Z)

```
.data
R DWORD ?
X DWORD 26
Y DWORD 30
Z DWORD 40
.code
mov  eax, X          ; copy X to a proper register

     neg  eax        ; set it to a negative X

mov  ebx, Y          ; copy Y to a proper register

     sub ebx, z      ; Calculate Y –Z

     add  eax, ebx   ; Calculate –X + (Y –Z)

     mov R, eax      ; copy the result to R
```

5. Please use proper direct-offset operands to implement a correct value in the comments

```
.data
 arrayW WORD 1000h, 2000h, 3000h
 arrayD DWORD 1, 2, 3, 4

.code
 mov ax, [arrayW + 2]    ; AX = 2000h

 mov ax, [arrayW + 4]    ; AX = 3000h

 mov ax, [arrayD + 8]    ; EAX = 00000003h
```

6. Write a program that does the following:

(1) Set the value of EAX to the hexadecimal value F00D.

(2) Add BEEF to EAX.

What is the value of EAX?

```
; AddTwo.asm - adds two 32-bit integers.
; Chapter 3 example

.386
.model flat, stdcall
.stack 4096
ExitProcess proto,dwExitCode:dword

.code
   main proc

movzx/   mov eax, ØF00Dh

         add eax, ØBEEFh

         invoke ExitProcess,0
   main endp
end main
```

7. Create two variables. The first one should be 1 with BYTE data type, the second one should be 2 with DWORD data type. Sum the two and put the answer in the EBX register.

```
.386
.model flat,stdcall
.stack 4096
ExitProcess proto,dwExitCode:dword

.data
        var1 BYTE  1


        var2 DWORD 2


.code
   main proc

        mov eax, 0

        mov ebx, 0

        mov al, var1

        mov ebx, var2

        add ebx, eax

        invoke ExitProcess,0
   main endp
end main
```