Midterm: Assembly Programming

Programming Part

1- You are only allowed to use instructions covered in chapters 1, 2, 3, and 4 of the textbook. Write your solution in a text editor (Microsoft Word is not a text editor)

2- A general-purpose program means that the program works with any data and not only with sample data.

Prob 1: (20 points) We have an 8 bytes width number, so we save the lower bytes in *EAX* and higher bytes in *EDX*: **for example** number *1234567812131415h* will be saved like *EAX = 12131415h, EDX = 12345678h.* Write a ***general-purpose program*** that is able to reverses any number 8 bytes width number that its least significant bytes are in *EAX* and its most significant bytes are saved in *EDX* . *Note:* Reverse means that our sample number becomes: *EAX=78563412h* and *EDX = 15141312h.*
Consider this ***sample*** call:

.data
EAX: 12131415h
EDX: 12345678h

Prob 2: (10 points) Write a ***general-purpose program*** that is able to add two 8 bytes length numbers. Numbers are saved in *EBX: EAX* and *EDX: ECX*
Consider this ***sample*** call:
Number1 = 1234567898765432h
Number2 = 1234567898765432h

Prob 3: (20 points) Write a ***general-purpose program*** with loop and ***indexed addressing*** that adds *12h* to 0th, *3rd* , *7th* , *11th* ,*15th* ,*19th* , … elements of a DWORD array. **For example**, in array:
Array1 DWORD *12h, 13h, 14h,15h, 16h, 17h, 18h, 19h, 1ah, 1bh, 1ch, 1dh, 1eh, 1fh*
becomes:
Array1 : *24h, 13h, 14h, 27h,16h,17h,18h, 2bh, 1ah, 1bh, 1ch, 2f, 1eh, 1fh*

Prob 4: (20 points) Use the following variable definitions:
.data

var1 SBYTE	*-20, -1, 1, 29*
var2 WORD	*0FE00h, 0C900h, 9100h, 2F00h*
var3 SWORD	*-16, -27*
var4 DWORD	*-15,14,13,12,11*

What will be the value of the destination operand after each of the following instructions?

Midterm: Assembly Programming

Programming Part

Show your answers in Hexadecimal.

execute in sequence:

mov edx, var4 ;	*a:*
movzx edx, [var2+6] ;	*b:*
mov edx, [var4+12] ;	*c:*
movsx edx, var1 ;	*d:*