CSC 3210

Computer Organization and Programming

Lab 7

Answer Sheet

Student Name: **Asrar Syed**

Section: **022 / CRN: 17915**

**Lab 7(a)**

* Create a new application to run the following program.
* The data segment is provided:

|  |
| --- |
| **.data**  Val1 SWORD 23  Val2 SWORD -35  Val3 SDWORD 4 |

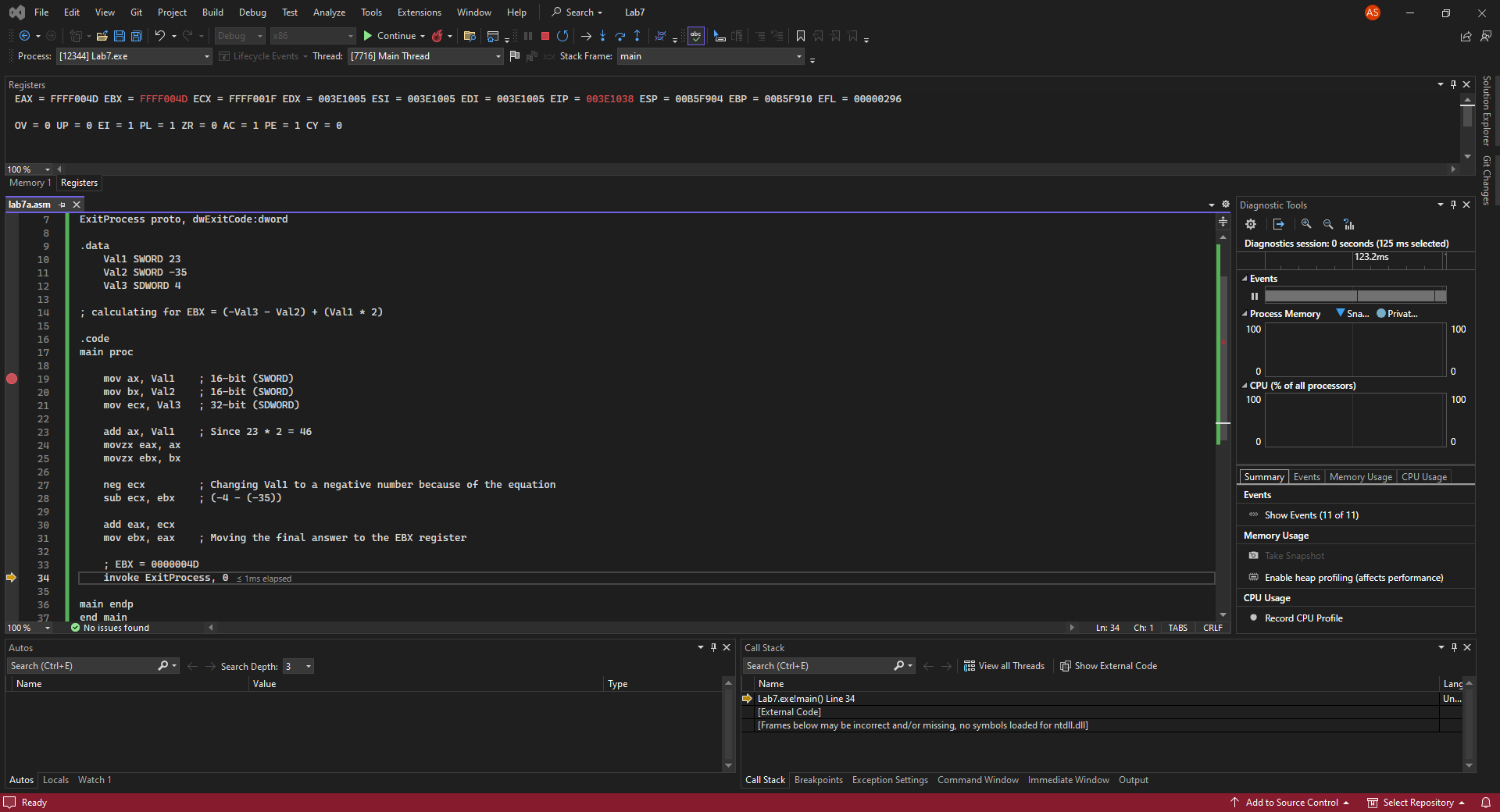
Evaluate the following expression: EBX = (-Val3 - val2) + (val1\*2)

You can only use Mov, Movsx, Movzx, Add, Sub instructions.

* Build and run the program using the debugger
* Examine the content of the registers

Take ONE screenshot showing your codes and the final result is (EBX = 0000004D)

My Result is EBX = FFFF004D



**Lab 7(b)**

Debug through each line of instructions.

Record the register content.

and explain the register contents.

**Line number: 17**

**Instruction: mov esi, OFFSET myBytes**

**Register values: ESI = 00824000, EIP = 00821015**

**Explanation: This line moves the offset address of the memory location where the myBytes array is stored into the ESI register.**

**Line number: 18**

**Instruction: mov ax, [esi]**

**Register values: EAX = 01182010, EIP = 00821018**

**Explanation: EAX register is 32-bit long with a signed integer variable. The register is updated with 2010 when the first element in ESI is moved to AX.**

**Line number: 19**

**Instruction: mov eax, DWORD PTR myWord**

**Register values: EAX = 003B008A, EIP = 0082101D**

**Explanation: This loads the first DWORD (4 bytes) of the myWords array into the eax register.**

**Line number: 20**

**Instruction: mov esi, OFFSET myWords**

**Register values: ESI = 00824004, EIP = 00821022**

**Explanation: This line prepares ESI to point to the beginning of the myWords array. The myWords array is defined as a sequence of WORDs (16-bit values), and OFFSET myWords calculates the memory address where this array starts within the data segment.**

**Line number: 21**

**Instruction: mov ax, [esi+2]**

**Register values: EAX = 003B003B, EIP = 00821016**

**Explanation: This loads the value at the memory address esi + 2 into the ax register. It will access the third element (72h) of the myWords array.**

**Line number: 22**

**Instruction: mov ax, [esi+6]**

**Register values: EAX = 003B0044, EIP = 0082102A**

**Explanation: This loads the value at the memory address esi + 6 into the ax register. It will access beyond the bounds of the myWords array.**

**Line number: 23**

**Instruction: mov ax, [esi-4]**

**Register values: EAX = 003B2010, EIP = 0082102E**

**Explanation: This loads the value at the memory address esi - 4 into the ax register. It will access the memory before the beginning of the myWords array.**

Take ONE screenshot that showing your codes in the debugging mode with register values.

A screenshot of a computer

Description automatically generated

**Lab 7(c)**

**Line number: 17**

**Instruction: mov ax, WORD PTR [varB+2]**

**Register values: EAX = 012F0502, EIP = 00E21016**

**Explanation: AX in EAX register is updated with 0502.**

**Line number: 18**

**Instruction: mov bl, BYTE PTR varD**

**Register values: EBX = 010D8078, EIP = 00E2101C**

**Explanation: BL in EBX register is updated with the last 2 numbers of varD (12345678h).**

**Line number: 19**

**Instruction: mov bl, BYTE PTR [varW+2]**

**Register values: EBX = 010D8002, EIP = 00E21022**

**Explanation: BL in EBX register is updated with the last two numbers of the second element in varW (1202h).**

**Line number: 20**

**Instruction: mov ax, WORD PTR [varD+2]**

**Register values: EAX = 012F1234, EIP = 00E21018**

**Explanation: AX in EAX register is updated with the first 4 numbers of the second element in varD (12345678).**

**Line number: 21**

**Instruction: mov eax, DWORD PTR varW**

**Register values: EAX = 12026543, EIP = 00E21028**

**Explanation: EAX register is updated with varW, but in descending order. 1202 is the first half, and 6543 is the second half of the register.**

Take ONE screenshot that showing your codes in the debugging mode with register values.

A screenshot of a computer

Description automatically generated

**LAB 7(d)**

Write and run a program to find the values of a memory location and a register:

.data

dVal DWORD ?

.code

mov dVal,12345678h

mov ax,WORD PTR dVal+2

add ax,3

mov WORD PTR dVal,ax ; dVal=

mov eax,dVal ;EAX=

**Line number: 18**

**Instruction: mov WORD PTR dVal, ax**

***Memory dVal* values: 37, 12, 34, 12**

**Explanation: #** **The first two elements of Memory/dVal are updated with AX in Little Endian. The overall value of Memory is no 12341237.**

**Line number: 19**

**Instruction: mov eax, dVal**

***EAX Register* values: 12341237**

**Explanation:**  **EAX register is updated with the value in Memory/dVal.**

Take ONE screenshot that showing your codes in the debugging mode with register values & the dVal variable in memory windows

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**LAB 7(e)**

**Line number: 17**

**Instruction: mov eax, TYPE myBytes**

**EAX Register values: 00000001**

**Explanation: EAX register is updated, returning with the size of myBytes (1 byte).**

**Line number: 18**

**Instruction: mov eax, LENGTHOF myBytes**

**EAX Register values: 00000004**

**Explanation: EAX register is updated, returning the number of elements in myBytes (4).**

**Line number: 19**

**Instruction: mov eax, SIZEOF myBytes**

**EAX Register values: 00000004**

**Explanation: EAX register is updated, returning the size of myBytes (4). In this case, EAX register did not change.**

**Line number: 20**

**Instruction: mov eax, TYPE myWords**

**EAX Register values: 00000002**

**Explanation: EAX register is updated, returning with the size of myWords (2 bytes).**

**Line number: 21**

**Instruction: mov eax, LENGTHOF myWords**

**EAX Register values: 00000004**

**Explanation: EAX register is updated, returning the number of elements in myWords (4).**

**Line number: 22**

**Instruction: mov eax, SIZEOF myWords**

**EAX Register values: 00000008**

**Explanation: EAX register is updated, returning the size of myWords (8).**

**Line number: 23**

**Instruction: mov eax, SIZEOF myString**

**EAX Register values: 00000005**

**Explanation: EAX register is updated, returning the size of myString (5).**

Take ONE screenshot that showing your codes in the debugging mode with register values

A screenshot of a computer

Description automatically generated