CSC 3210

Computer Organization and Programming

Lab 7

Answer Sheet

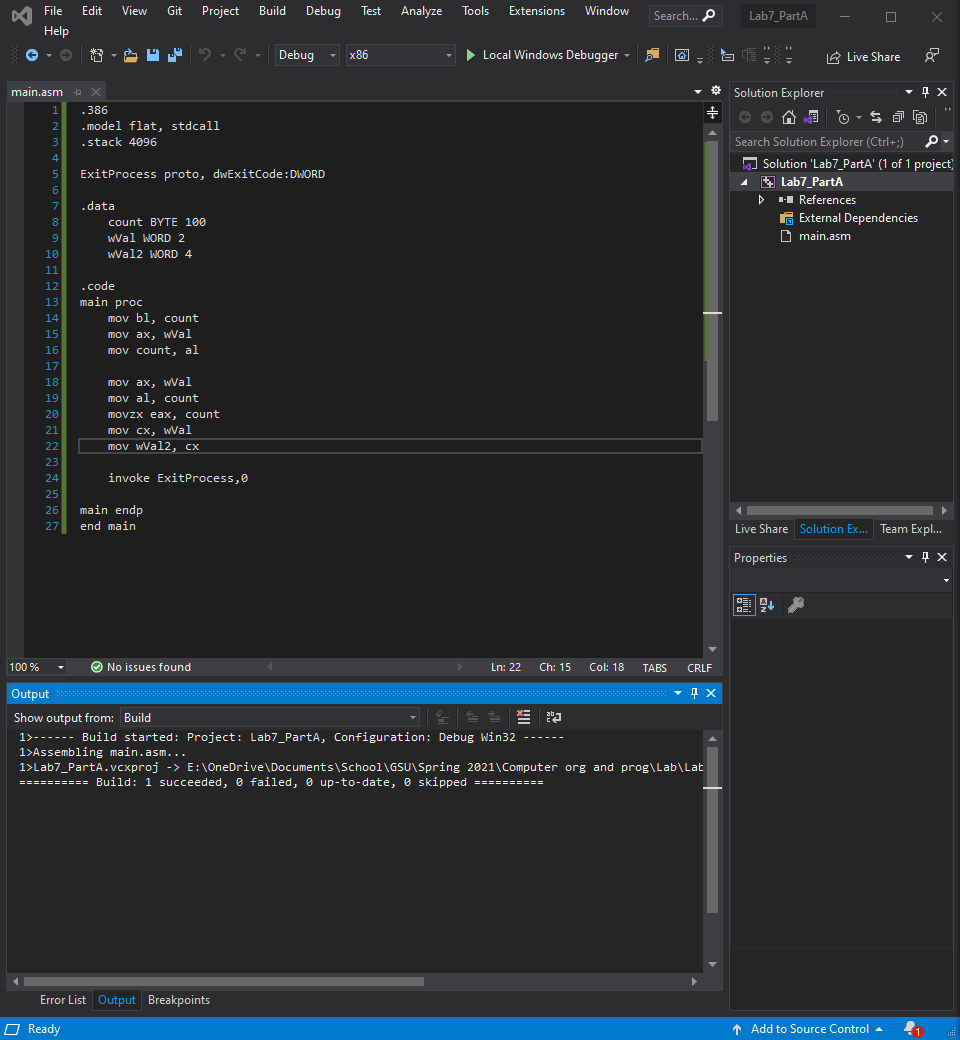
Student Name: Tracy Michaels

Section: 018

Lab 7(a)

Fix the errors in the provided code.

Build and Attach screenshot showing the code and “build succeeded” message.



Lab 7(b)

Debug through each line of instructions.

Take screenshot that includes code and register window.

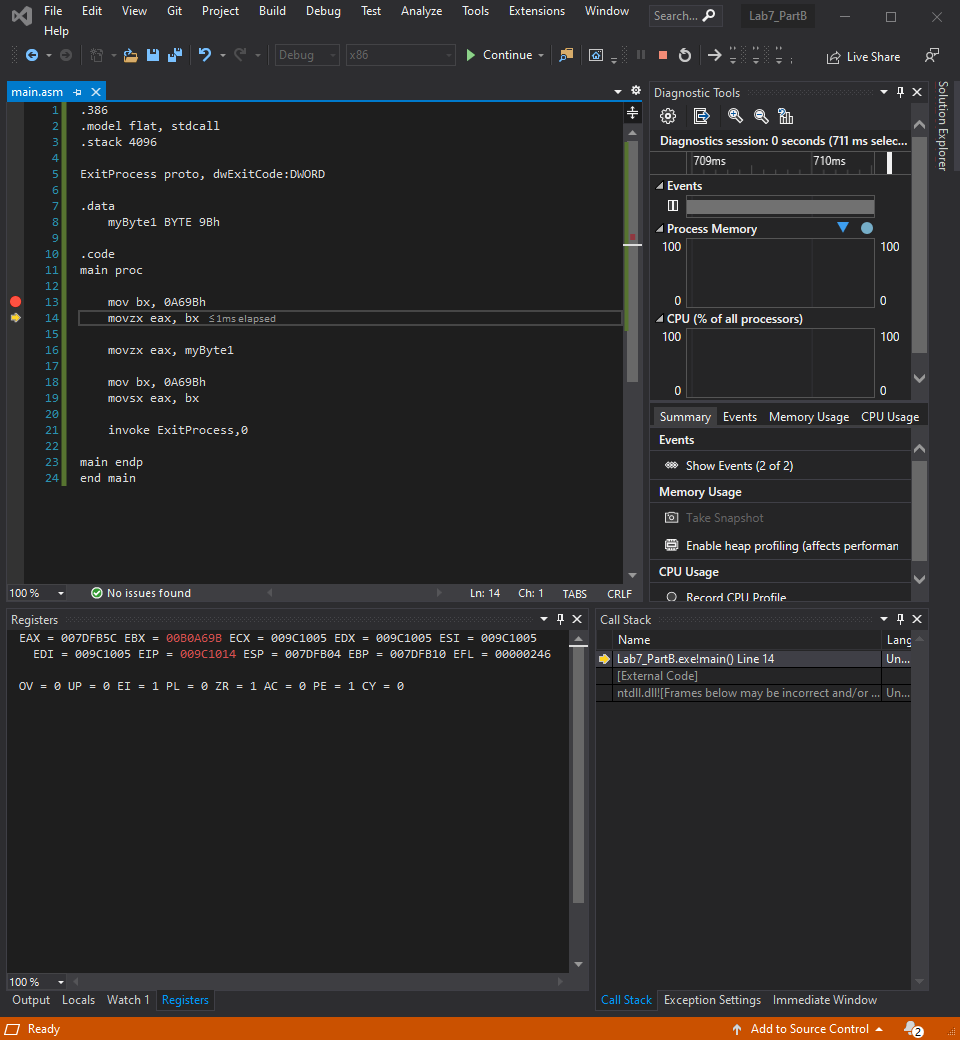
Record the register content.

and explain the register contents.

Line number: 13

Instruction: mov bx, 0A69Bh

Register values: EBX = ----A69B

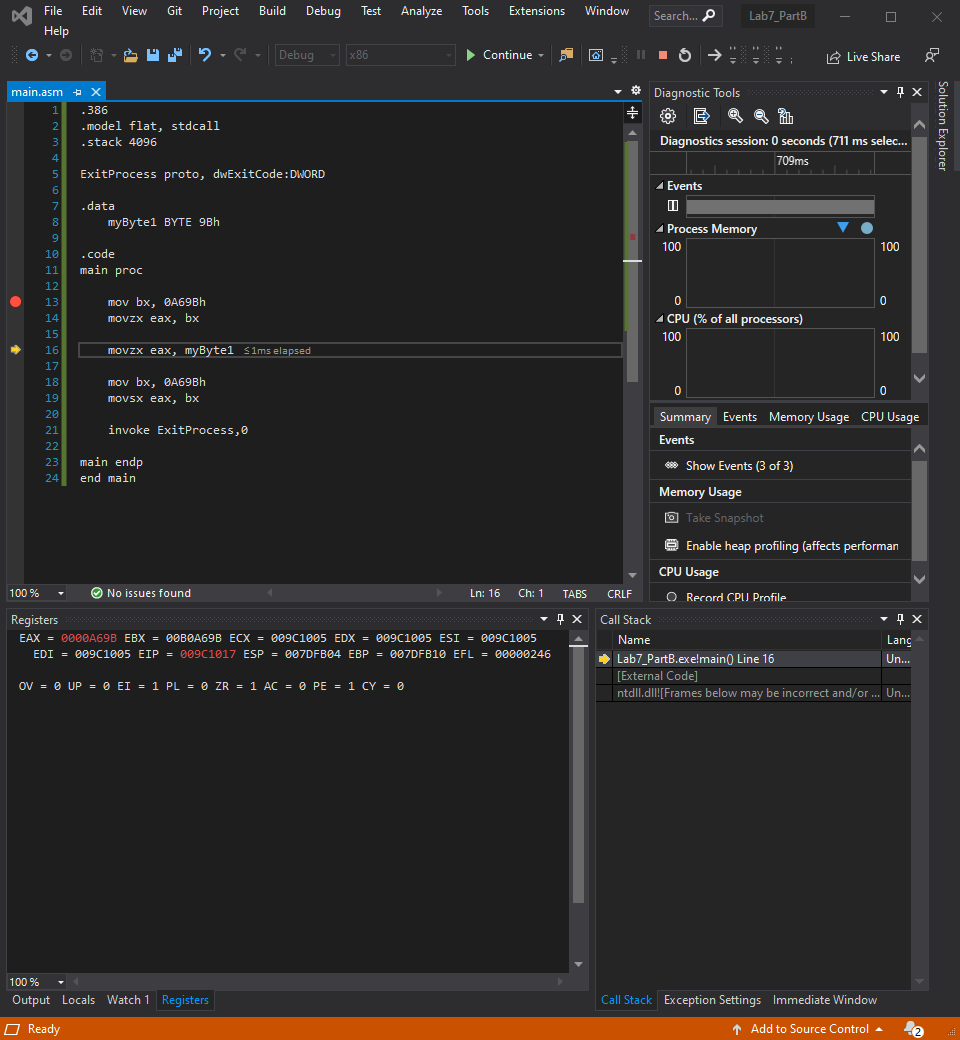
Screenshot: 

Explanation: this instruction moves the hexadecimal value A69B into the BX portion of the EBX register, the upper 4 bytes of EBX are garbage values

Line number: 14

Instruction: movzx eax, bx

Register values: EAX = 0000A69B

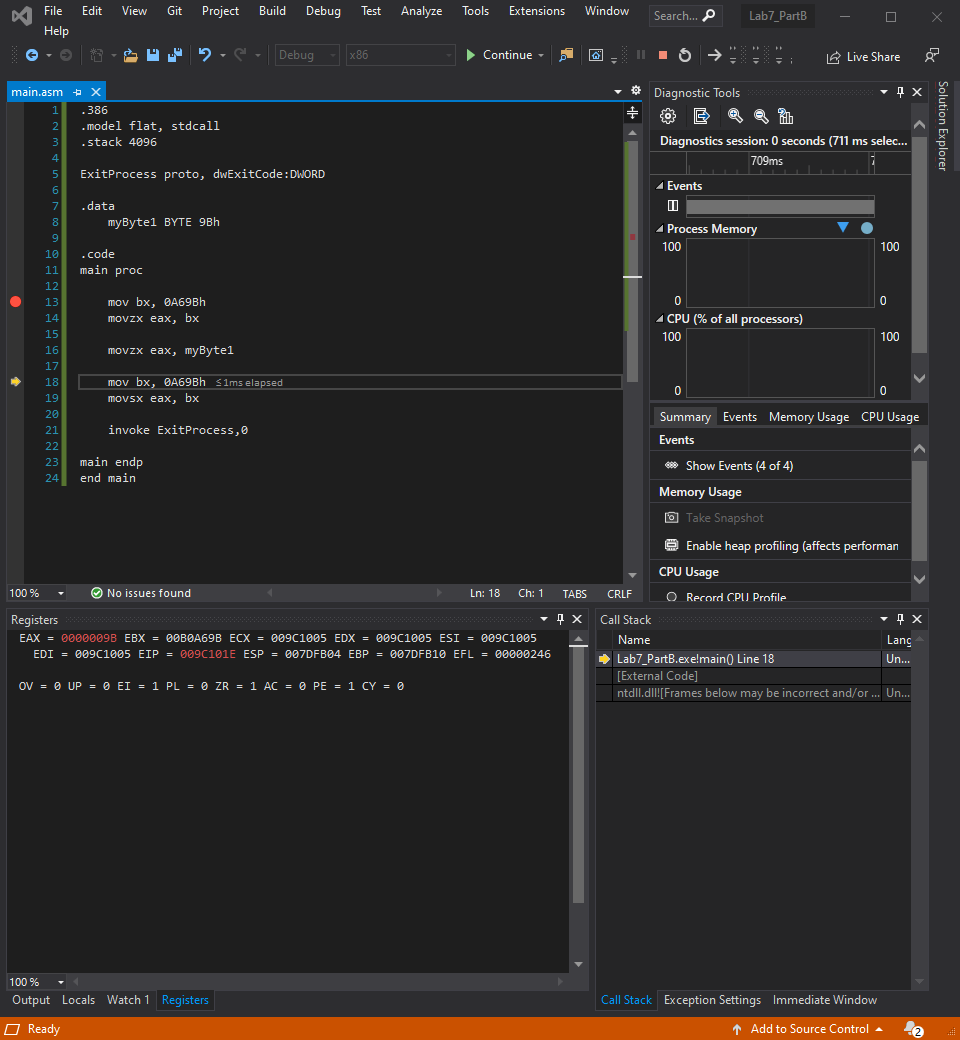
Screenshot: 

Explanation: this moves the contents in BX to the AX part of EAX while filling the upper part of the EAX register with 0’s

Line number: 16

Instruction: movzx eax, myByte1

Register values: EAX = 00000009B

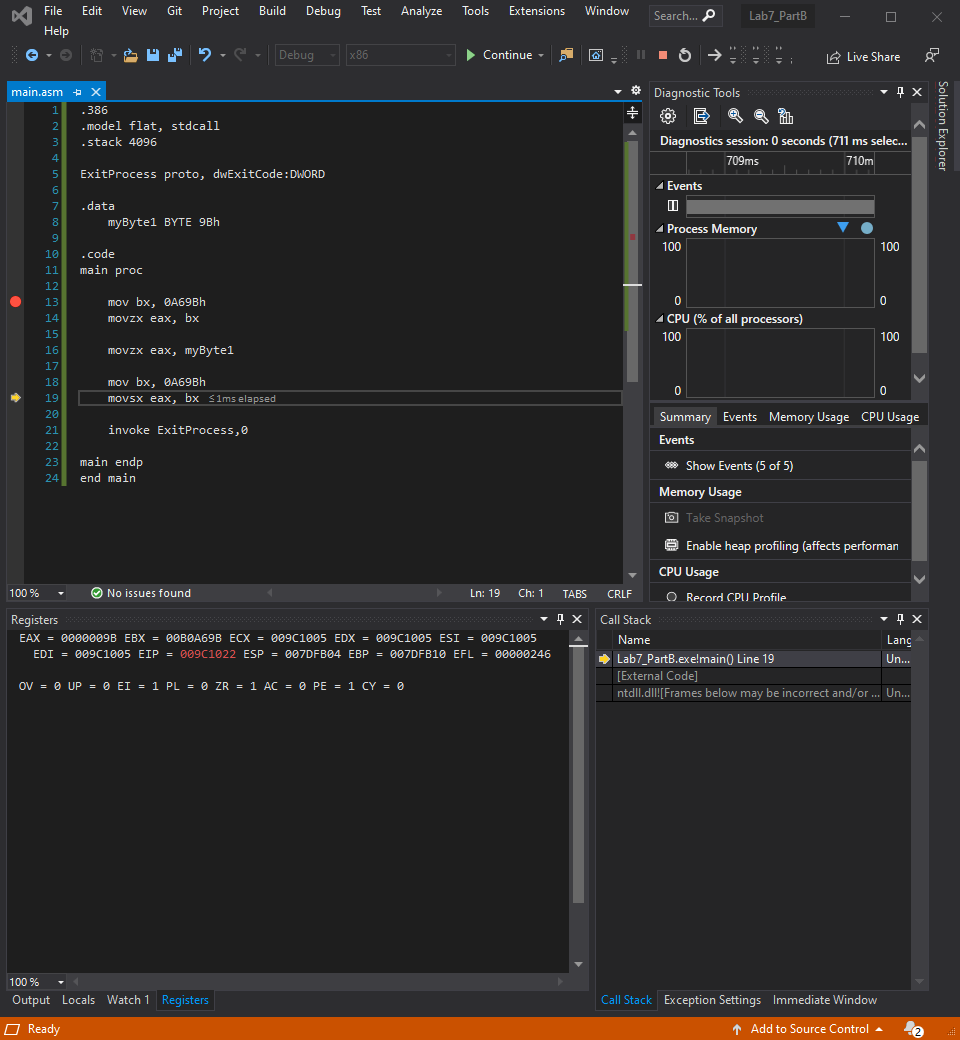
Screenshot: 

Explanation: this moves the hexadecimal value 9B that is stored in the variable myByte1 to the AL part of the EAX register while filling the rest of the EAX register with 0’s

Line number: 18

Instruction: mov bx, 0A69bh

Register values: EBX = ----A69B

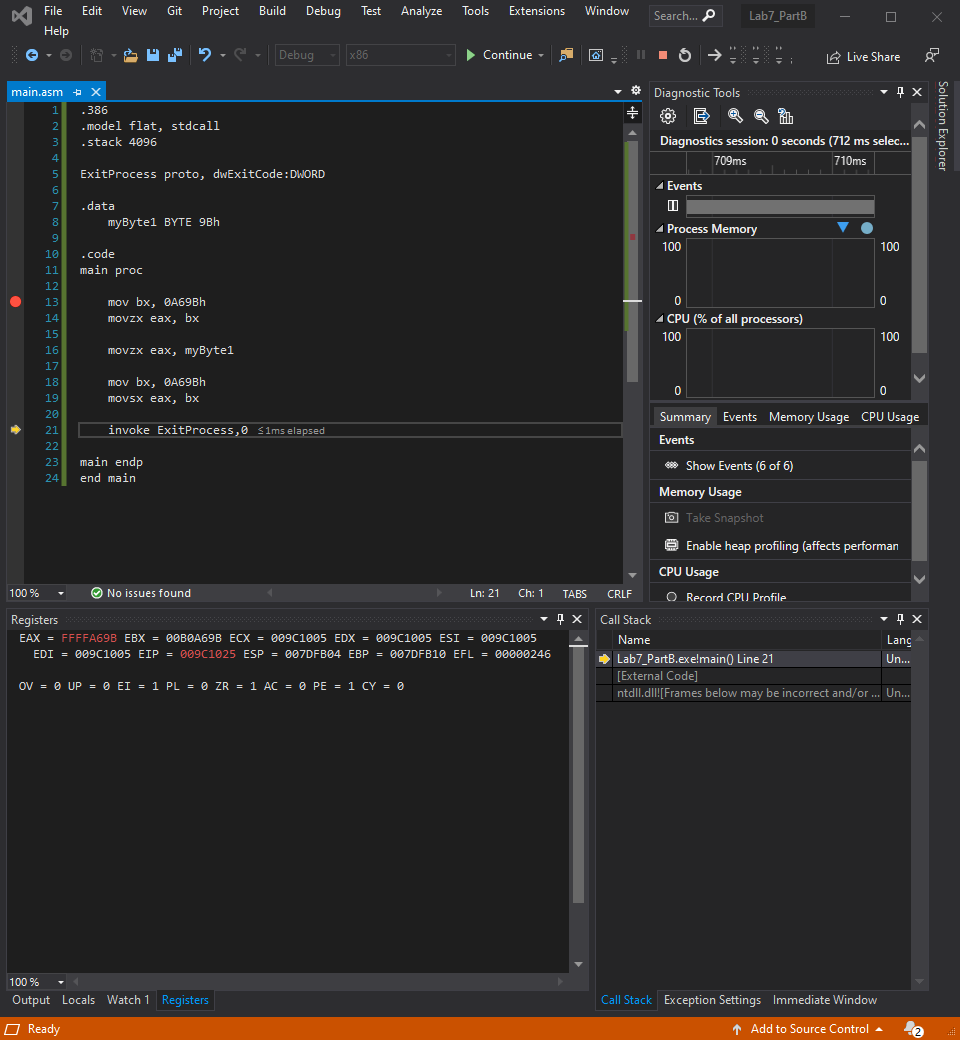
Screenshot: 

Explanation: this makes no changes to the EBX register since it already had this value stored in it. The upper part is still garbage values and the BX part of the EBX register is still the hexadecimal value A69B

Line number: 19

Instruction: movsx eax, bx

Register values: EAX = FFFFA69B

Screenshot: 

Explanation: This moves the hexadecimal value A69B that is stored in the BX part of the EBX register and places it into the AX portion of the EAX register while filling the rest of the upper portion of the EAX register with F’s

Lab 7(c)

Debug through each line of instructions.

Take screenshot that includes code and register window.

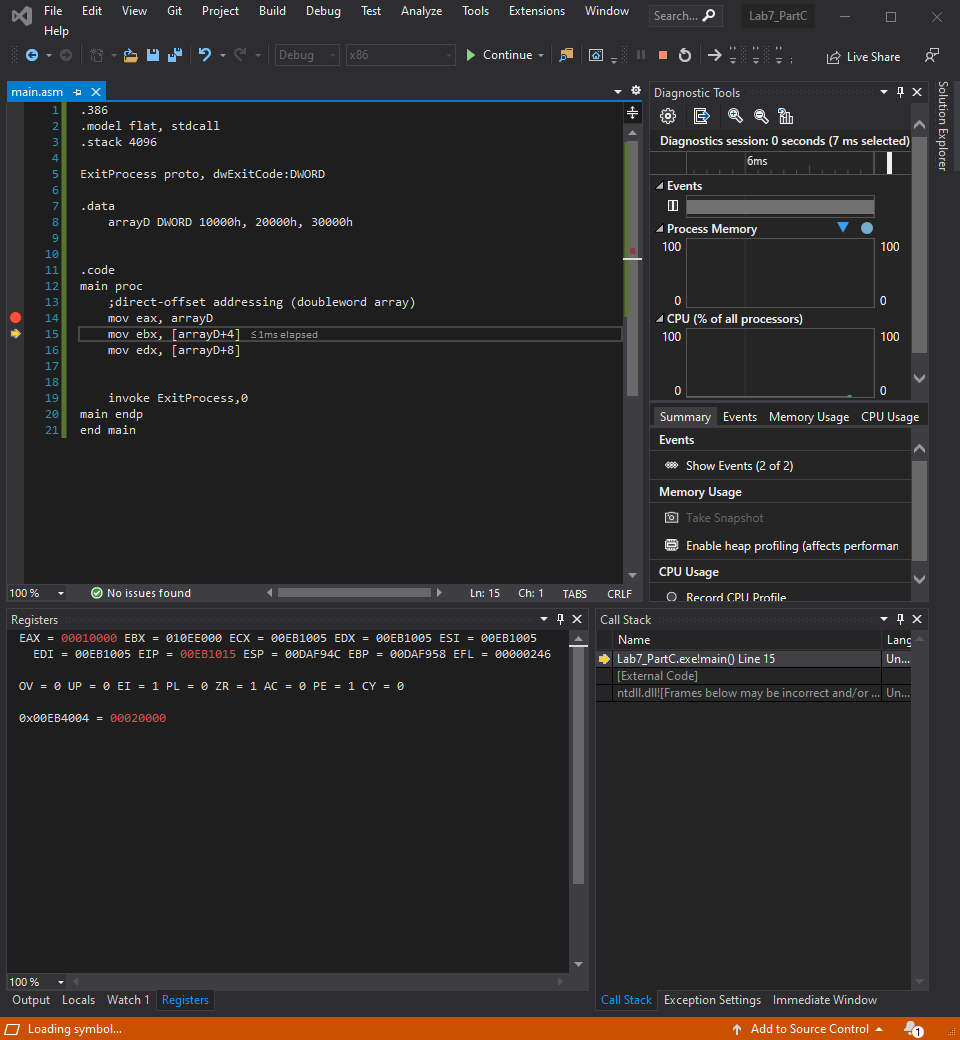
Record the register content.

and explain the register contents.

Line number: 14

Instruction: mov eax, arrayD

Register values: EAX = 00010000

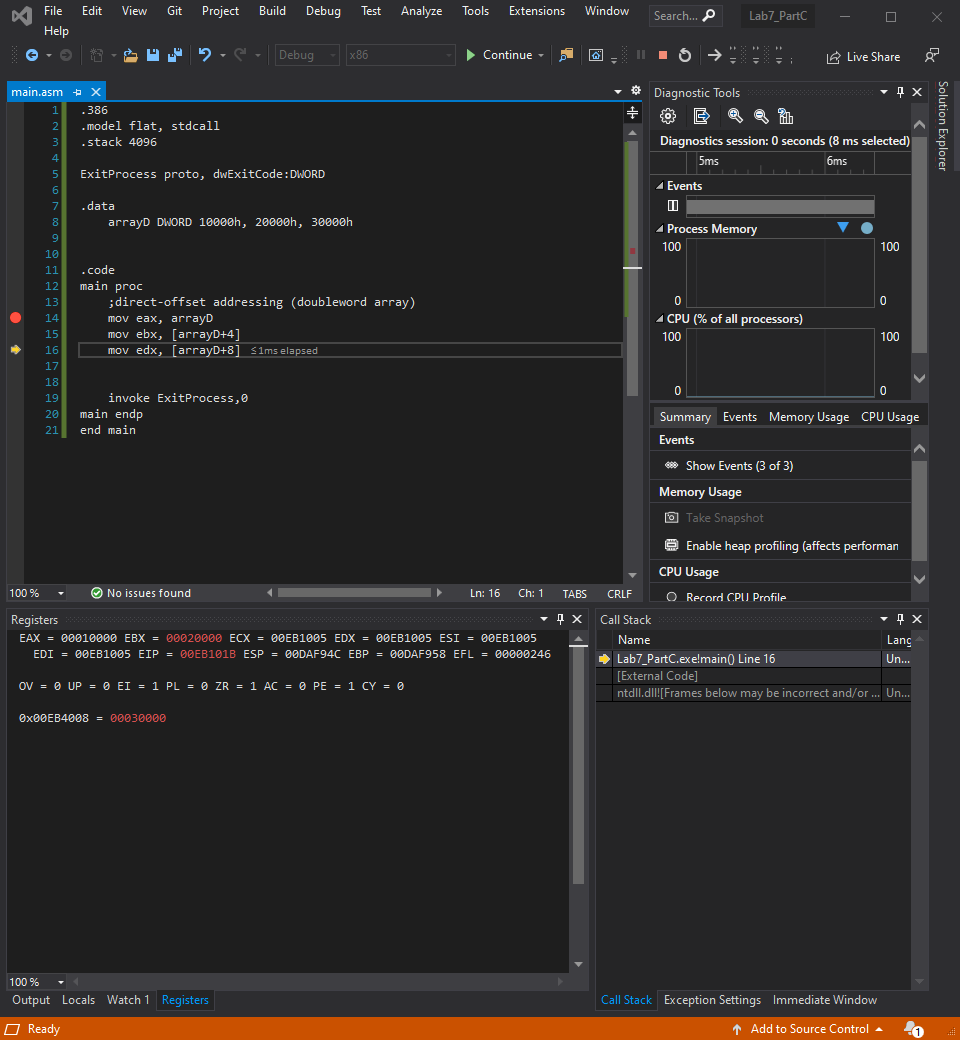
Screenshot: 

Explanation: this instruction moves the first element of arrayD which is the hexadecimal number 10000 into the EAX register

Line number:15

Instruction: mov ebx, [arrayD+4]

Register values: EBX = 00020000

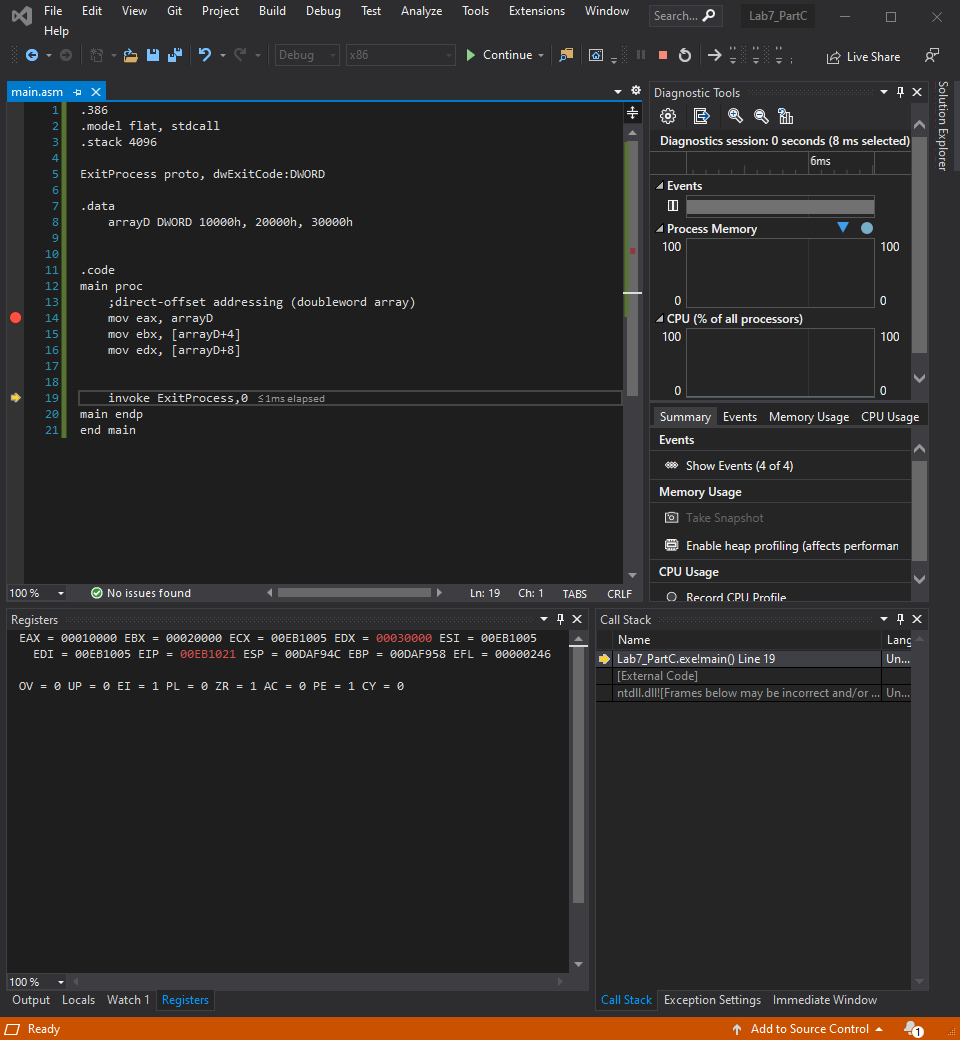
Screenshot: 

Explanation: this moves the contents of the second element in arrayD, which is the hexadecimal value 20000 into the EBX register

Line number: 16

Instruction: mov edx, [arrayD+8]

Register values: EDX = 00030000

Screenshot: 

Explanation: this moves the contents of the 3rd element in arrayD which is the hexadecimal value 30000 into the EDX register

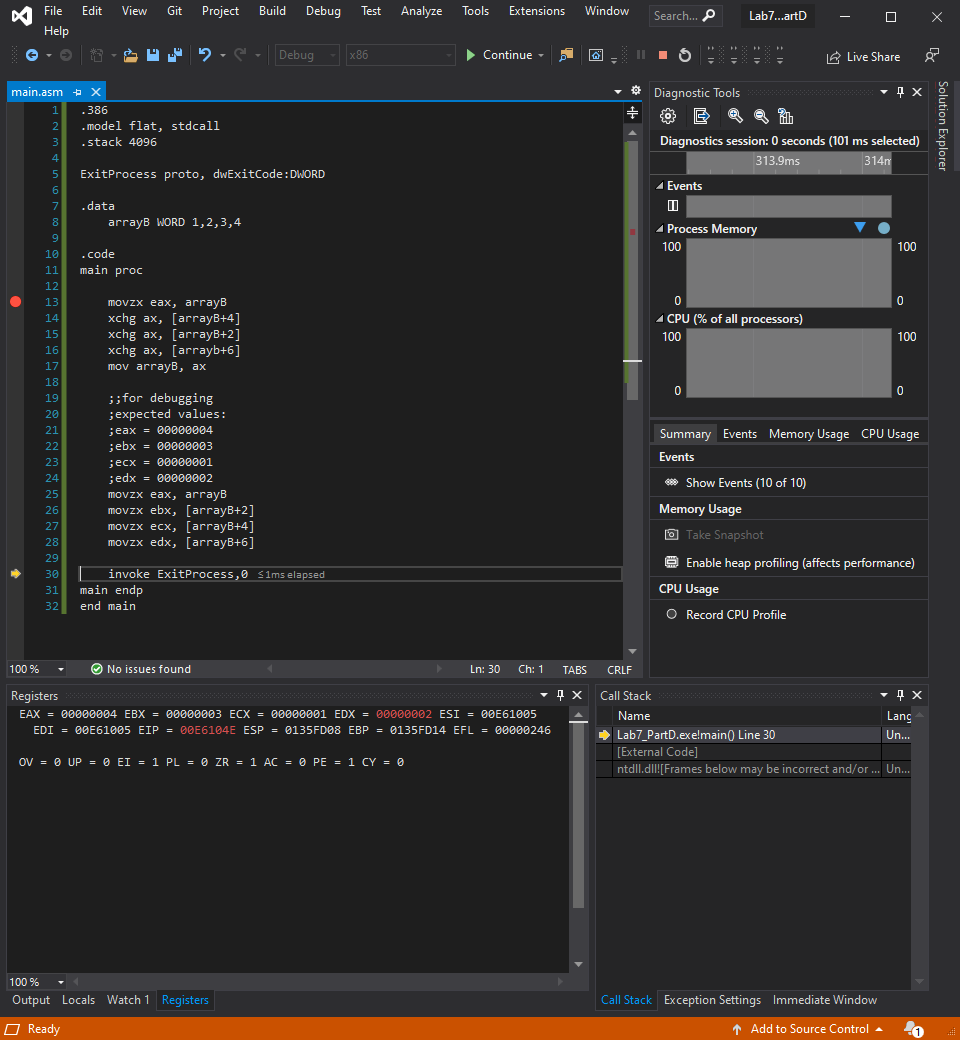
Lab 7(d)

Create a new project to run the following program.

Declare an array in the data segment: arrayB WORD 1,2,3,4

Write code to Rearrange the array as follows: 4,3,1,2

Add the screenshot of your code here.



Lab 7(e)

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 = (-Val1 + val2) + (val3\*3)

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

Build and run the program using the debugger

Debug the code until you reach “INVOKE ExitProcess, 0” and attach a screenshot of your code and EBX register content.

