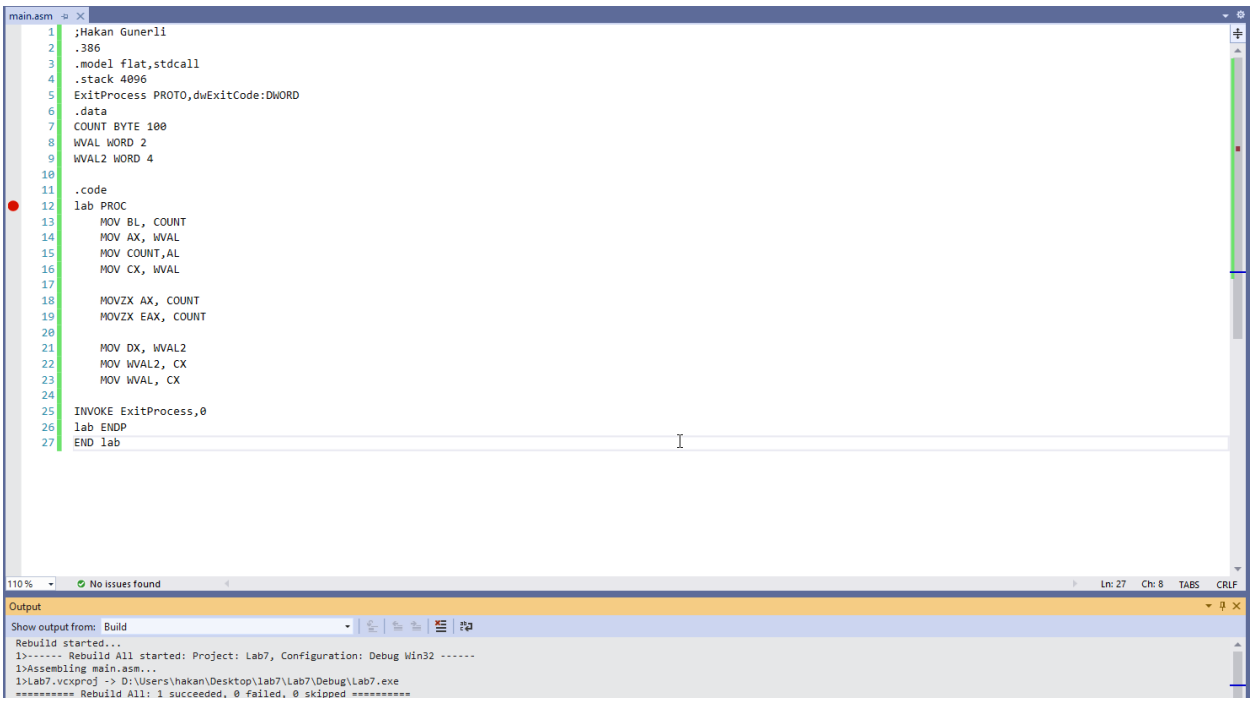
CSC 3210
Computer Organization and Programming
Lab 7
Answer Sheet

Student Name: Hakan Gunerli
Section: 014

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: 9
Instruction: mov bx, 0A69Bh
Register values: EAX = GARBAGE, EBX = A69Bh
Screenshot:

```
Registers
EAX = 006FFCEC EBX = 0040A69B ECX = 000B1005 EDX = 000B1005 ESI = 000B1005 EDI = 000B1005 EIP = 000B1014 ESP = 006FFC98
  EBP = 006FFCA4 EFL = 00000246
```

Explanation: All registers except bx are garbage, 0A69Bh moved to bx.

Line number: 10
Instruction: movzx eax, bx
Register values: EAX = 0000A69Bh, EBX = A69Bh
Screenshot:

```
Registers
EAX = 0000A69B EBX = 0040A69B ECX = 000B1005 EDX = 000B1005 ESI = 000B1005 EDI = 000B1005 EIP = 000B1017 ESP = 006FFC98
  EBP = 006FFCA4 EFL = 00000246
```

Explanation: bx value has been moved to eax. The value in ebx and bx stays the same.

Line number: 11
Instruction: movzx eax, myByte1
Register values: EAX = 0000009Bh, EBX = _ _ _ _ A69Bh
Screenshot:

```
Registers
EAX = 0000009B EBX = 0040A69B ECX = 000B1005 EDX = 000B1005 ESI = 000B1005 EDI = 000B1005 EIP = 000B101E ESP = 006FFC98
  EBP = 006FFCA4 EFL = 00000246
```

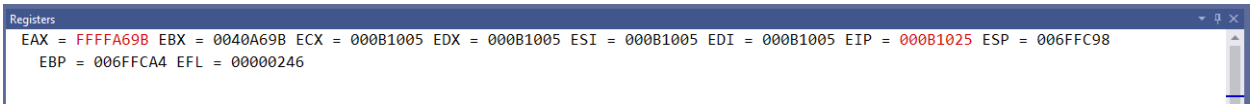
Explanation: The value in myByte1 (9Bh) moved to EAX.

Line number: 12
Instruction: mov bx, 0A69Bh
Register values: EAX = 0000009Bh, EBX = A69Bh
Screenshot:

```
Registers
EAX = 0000009B EBX = 0040A69B ECX = 000B1005 EDX = 000B1005 ESI = 000B1005 EDI = 000B1005 EIP = 000B1022 ESP = 006FFC98
  EBP = 006FFCA4 EFL = 00000246
```

Explanation: The value 0A69Bh moved to the bx register, even though the value already exists in it. There is no difference in values of any register.

Line number: 13
Instruction: movsx eax, myByte1
Register values: EAX = FFFFA69Bh, EBX = _ _ _ _ A69Bh
Screenshot:

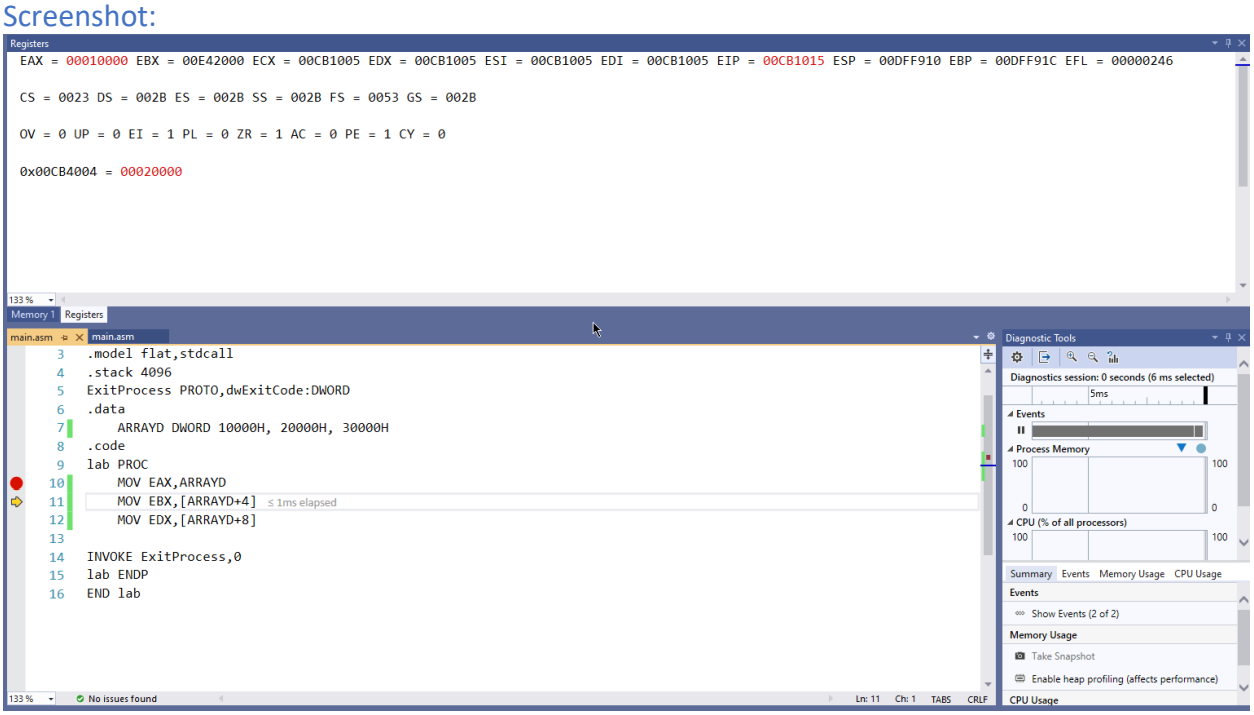


Explanation: value in the bx registers moved to eax after it got flashed with Fs (it is the signed).

Lab 7(c)

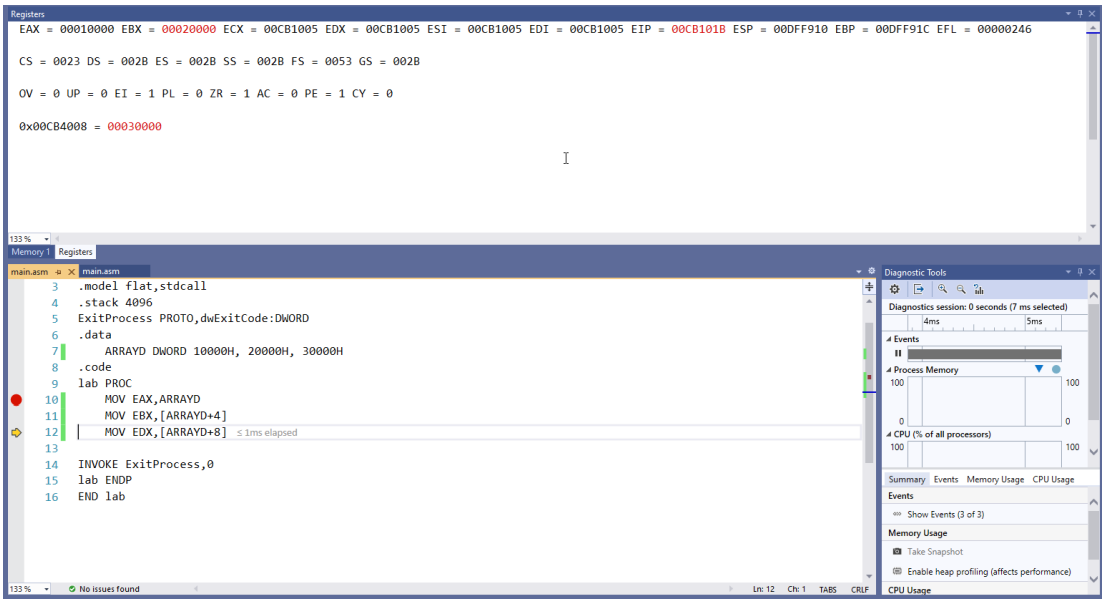
Debug through each line of instructions.
Take screenshot that includes code and register window.
Record the register content.
and explain the register contents.

Line number: 10
Instruction: mov eax, arrayd
Register values: eax= 0010000h
Screenshot:



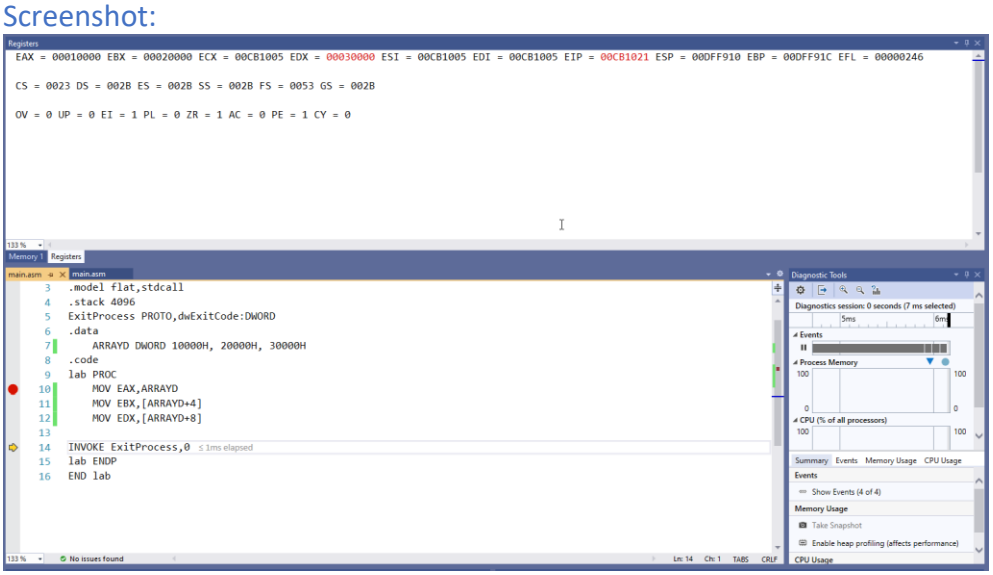
Explanation: first element is moved to eax.

Line number: 11
Instruction: `mov ebx,[arrayD+4]`
Register values: EAX = 0010000h, EBX = 0020000h
Screenshot:



Explanation:
Second element is moved to ebx. Ebx = 0020000h

Line number: 12
Instruction: mov edx [arrayD+8]
Register values: EAX = 0010000h, EBX = 0020000h, EDX = 0030000h
Screenshot:



Explanation:
Third element of the array is moved to edx, edx contains 0030000h.

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.

main.asm main.asm

```
.386
.model flat,stdcall
.stack 4096
ExitProcess PROTO,dwExitCode:DWORD
.data
    ARRAYB WORD 1,2,3,4
.code
lab PROC
MOV EAX, 000000
MOV EBX, 000000
MOV ECX, 000000
MOV EDX, 000000
    MOV AX, [ARRAYB]
    MOV BX,[ARRAYB+2]
    MOV CX, [ARRAYB+4]
    MOV DX, [ARRAYB+6]

    XCHG AX,DX
    XCHG CX,BX
    XCHG CX,DX

    MOV [ARRAYB], AX
    MOV [ARRAYB+2], BX
    MOV [ARRAYB+4],CX
    MOV [ARRAYB+6],DX

INVOKE ExitProcess,0
lab ENDP
END lab
```

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.