CSC 3210 – Assignment #2

Spring 2022

Objective: Learn memory organization/layout, data transfer concepts and instructions, direct memory access, memory allocation.

Requirements:

1. **(5 points) Implement the following expression in assembly language:**

   EDX = (val3 + val4) - (val2 - val1) - (5/3)*7

   - Assume that val1, val2, val3 and val4 are 16-bit integer variables

   - You need to implement the expression the way it is provided, you cannot do any reduction on the expression while implementing it.

   - Initialize val1 with 120 (hexadecimal), val2 with 39 (hexadecimal), val3 with 20 (hexadecimal) and val4 with 27 (hexadecimal)

   - You are NOT allowed to update the values of any variables.

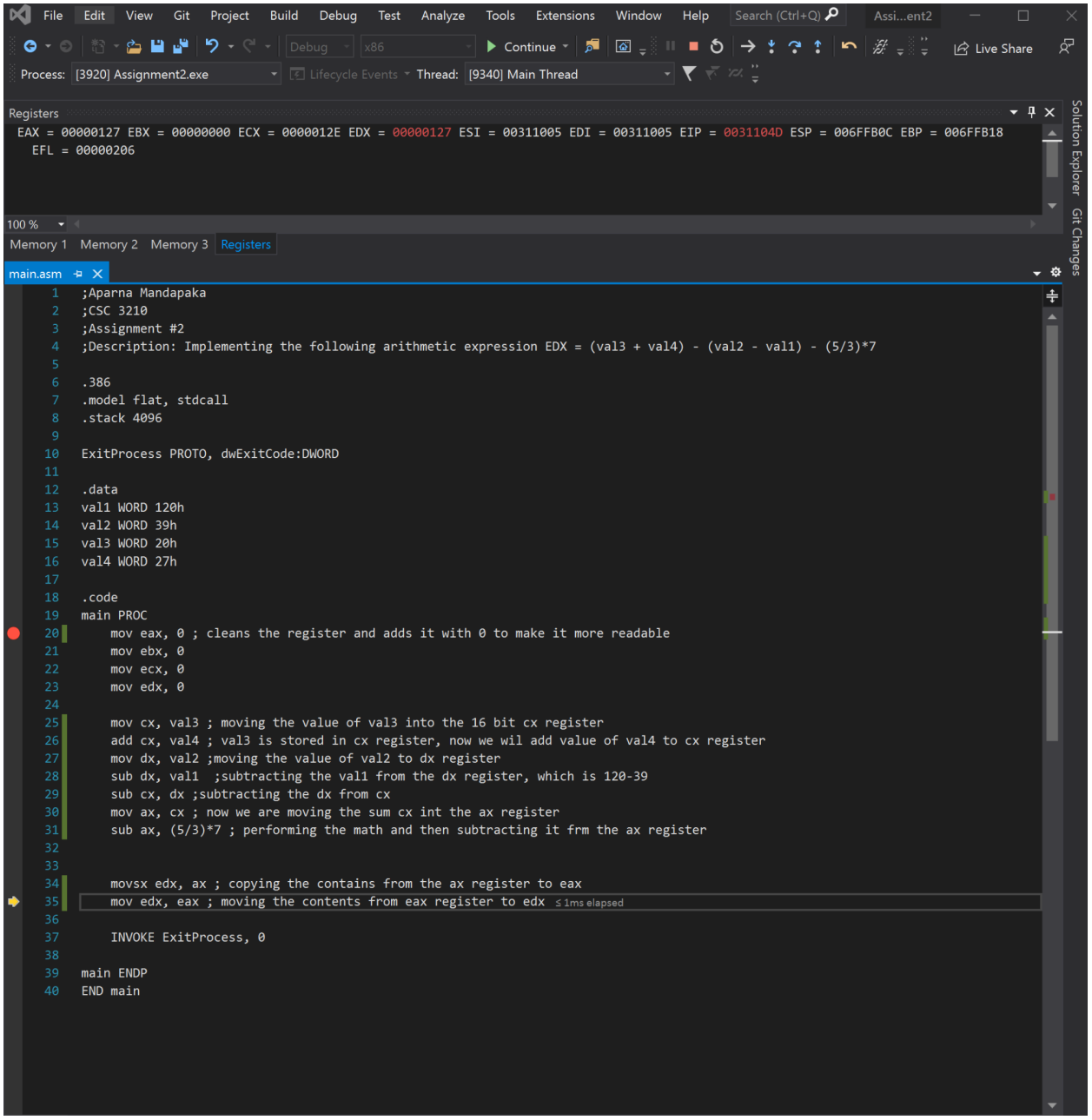
   - Use ONLY **mov, add, sub, movzx, movsx, or neg** instructions whenever needed.

   - Use the debugger to verify your answer.

   ○ **Submit the following:**

   § Save your source code using your last name, Lastname1.asm and upload the Lastname1.asm

   § Screenshot (showing the code and register window) of EDX register contains the correct result.



2. (5 points) Implement the following expression in assembly language:

ECX = –(val3 + val1) + (-val4 – val2) + 3

- Assume that val1 is 8-bit variable, val2 is 16-bit variable, val3 is 32-bit variable, and val4 is 8-bit variable.

- You need to implement the expression the way it is provided, you cannot do any reduction on the expression while implementing it.

- Initialize val1 with 12 (decimal), val2 with 9 (decimal), val3 with 2 (decimal), val4 with 20 (decimal),

- **You are NOT allowed to update the values** stored in val1, val2, val3 and val4

- Use mov, add, sub, movsx, movzx, or neg instructions whenever needed.

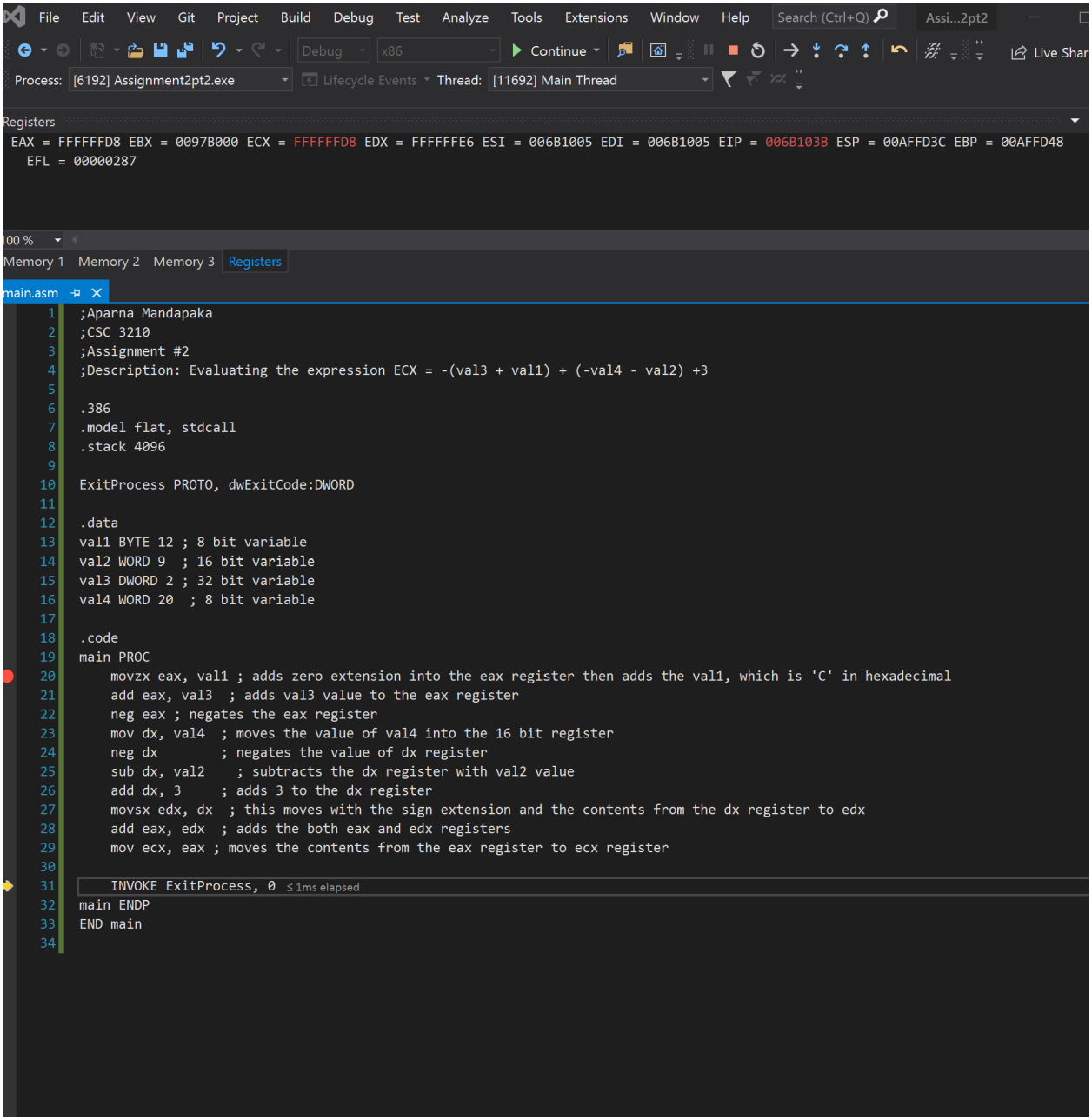
- Use the debugger to verify your answer.

-

- **Submit the following:**

  § Save your source code using your last name, Lastname2.asm and upload the Lastname2.asm

  § Screenshot (showing the code and register window) of ECX register contains the correct result.


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Assi...2pt2
Debug
x86
Continue
Live Shar
Process: [6192] Assignment2pt2.exe
Lifecycle Events
Thread: [11692] Main Thread
Registers
EAX = FFFFFFD8 EBX = 0097B000 ECX = FFFFFFD8 EDX = FFFFFFE6 ESI = 006B1005 EDI = 006B1005 EIP = 006B103B ESP = 00AFFD3C EBP = 00AFFD48
EFL = 00000287
100 %
Memory 1 Memory 2 Memory 3 Registers
main.asm
1 ;Aparna Mandapaka
2 ;CSC 3210
3 ;Assignment #2
4 ;Description: Evaluating the expression ECX = -(val3 + val1) + (-val4 - val2) +3
5
6 .386
7 .model flat, stdcall
8 .stack 4096
9
10 ExitProcess PROTO, dwExitCode:DWORD
11
12 .data
13 val1 BYTE 12 ; 8 bit variable
14 val2 WORD 9 ; 16 bit variable
15 val3 DWORD 2 ; 32 bit variable
16 val4 WORD 20 ; 8 bit variable
17
18 .code
19 main PROC
20 movzx eax, val1 ; adds zero extension into the eax register then adds the val1, which is 'C' in hexadecimal
21 add eax, val3 ; adds val3 value to the eax register
22 neg eax ; negates the eax register
23 mov dx, val4 ; moves the value of val4 into the 16 bit register
24 neg dx ; negates the value of dx register
25 sub dx, val2 ; subtracts the dx register with val2 value
26 add dx, 3 ; adds 3 to the dx register
27 movsx edx, dx ; this moves with the sign extension and the contents from the dx register to edx
28 add eax, edx ; adds the both eax and edx registers
29 mov ecx, eax ; moves the contents from the eax register to ecx register
30
31 INVOKE ExitProcess, 0 ≤ 1ms elapsed
32 main ENDP
33 END main
34

3. (5 points) Write an assembly program to compute the following expressions

- Create a DWORD array named 'z' of size 3 using DUP operator. Leave the array 'z' uninitialized. You can denote the items in the array as [, where is the first item, is the second item, is the third item

- Update each array item using the following expression

- Where x, y, r are 16-bit integer memory *variables*.

- x = 10, y = 15, r = 4

- Use mov, movzx, movsx, add, sub instructions only.

- (hint: Do not alter the value of x, y and r during the computation. Transfer them to appropriate registers to do computation)

- At the end, open memory window to see the variable z stored in memory (little endian format).

- Use the debugger to verify your answer.

    - **Submit the following:**

        § Rename the asm file using your last name as Lastname1.asm

        § Screenshot of the code and memory window showing the content of the variable z (little endian format).

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Assignment2pt3

Debug x86 Continue

Process: [28308] Assignment2pt3.exe Lifecycle Events Thread: [27156] Main Thread Stack Frame: main

Registers

```
EAX = 0000008C EBX = 00000000 ECX = 00000091 EDX = FFFFFF7D ESI = 00A61005 EDI = 00A61005 EIP = 00A61074 ESP = 006FFE24 EBP = 006FFE30 EFL = 00000287
```

100 %

Memory 1 Memory 2 Memory 3 Registers

main.asm

```
;Description: This expression evaluates
            ; z0 = x + 130
            ;z1 = y - x +z0
            ;z2 = r + x -z1
.386
.model flat, stdcall
.stack 4096

ExitProcess PROTO, dwExitCode: DWORD

.data
z DWORD 3 DUP(?)
x WORD 10
y WORD 15
r WORD 4

.code
main PROC
    mov eax,0 ; cleans the register and adds it with 0 to make it more readable
    mov ebx,0
    mov ecx, 0
    mov edx,0
    movsx eax, x ; moves the value of x into eax register
    add eax, 130 ; adds 130 to the content in the eax register
    mov edx, OFFSET z ;loads the address of z into the edx register
    mov z[0], eax ;moves the value x+130 into the z[0]
    mov cx, y ; loads the y value
    sub cx, x ;subtracts x from the cx register
    movsx ecx, cx ;moves the value of cx into ecx with 0 extension
    add ecx, z[0]
    mov z[4], ecx
    mov dx, r ; moves the r value into the dx register
    add dx, x ;adds x to the dx register
    movsx edx, dx ; moves the dx regiser into edx register
    sub edx, z[4]
    mov z[8], edx ;moves r + x - z1

    INVOKE ExitProcess,0 ≤ 1ms elapsed
main ENDP
END main
```

Note:

§ **Submit** your source code by **only** uploading **.ASM file** using **iCollege** in the respective assignment dropbox:

§ Lastname1.ASM, Lastname2.ASM

§ **Put the following information as Comment header** for .ASM files:

Student: Full name

Class: CSC3210

Assignment#: 2

Description: This program …….

§ Follow the program standards as presented in your book. Pay more attention to code comments and consistent indentation.