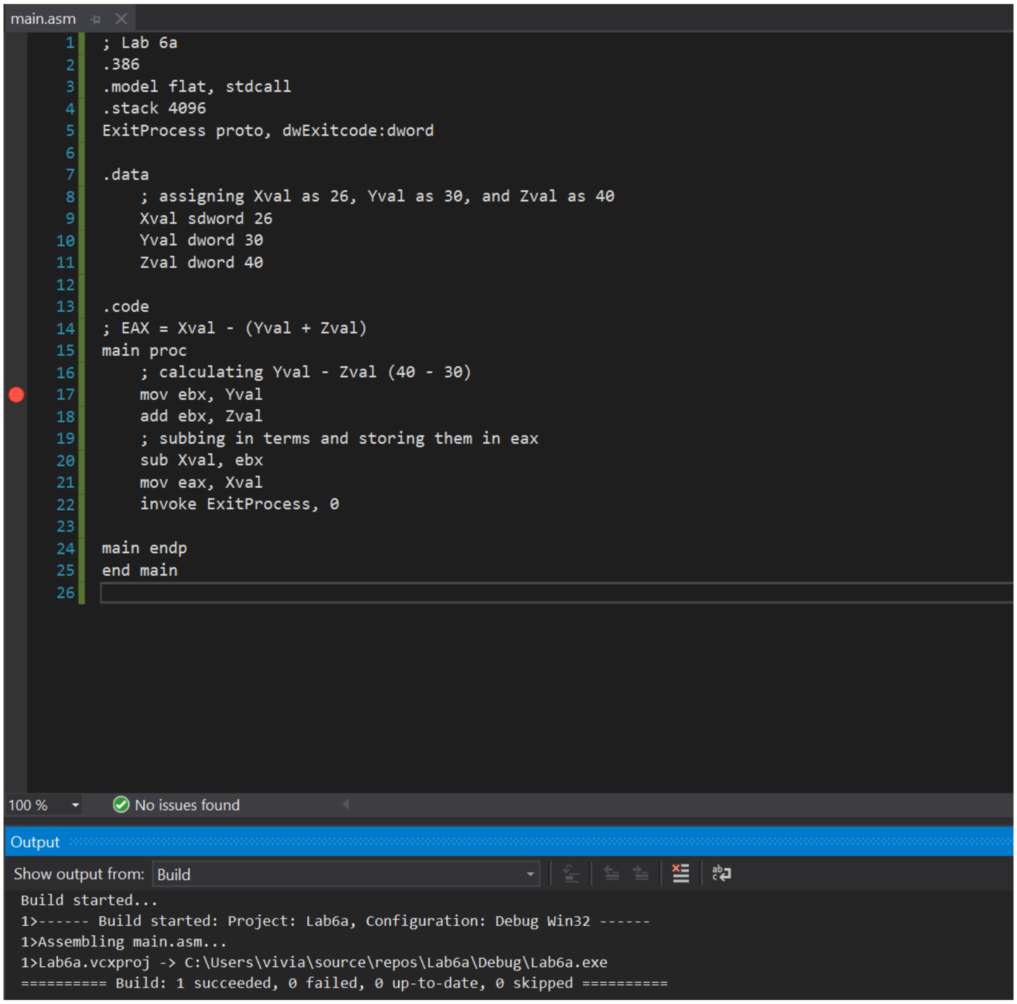
CSC 3210
Computer Organization and Programming
Lab 6
Answer Sheet

Student Name: Vivian Do
Section: CRN 90913; 11:00-12:40
Oct 01 2021

Lab 6(a)

Debug through each line of instructions.
Take screenshot that includes code and register window.
Record the register content.
and explain the register contents. (4 points)

screenshot of code where build was successful:



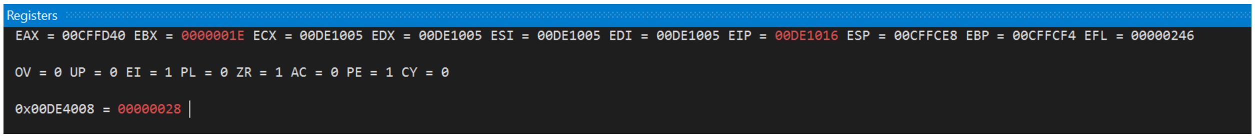
```
 1  ; Lab 6a
 2  .386
 3  .model flat, stdcall
 4  .stack 4096
 5  ExitProcess proto, dwExitcode:dword
 6
 7  .data
 8      ; assigning Xval as 26, Yval as 30, and Zval as 40
 9      Xval sdword 26
10      Yval dword 30
11      Zval dword 40
12
13  .code
14  ; EAX = Xval - (Yval + Zval)
15  main proc
16      ; calculating Yval - Zval (40 - 30)
17      mov ebx, Yval
18      add ebx, Zval
19      ; subbing in terms and storing them in eax
20      sub Xval, ebx
21      mov eax, Xval
22      invoke ExitProcess, 0
23
24  main endp
25  end main
26
```

Show output from: Build

```
Build started...
1>------ Build started: Project: Lab6a, Configuration: Debug Win32 ------
1>Assembling main.asm...
1>Lab6a.vcxproj -> C:\Users\vivia\source\repos\Lab6a\Debug\Lab6a.exe
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Line number: 17

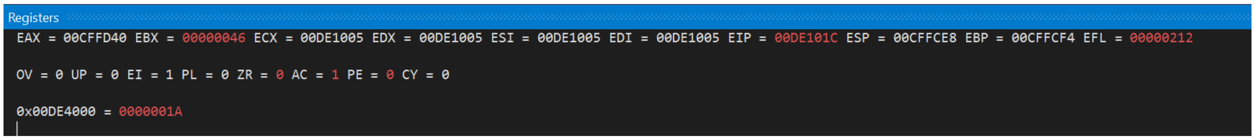
Instruction: mov ebx, Yval
Register Values: EBX = 0000001E
Screenshot:



```
Registers
 EAX = 00CFFD40 EBX = 0000001E ECX = 00DE1005 EDX = 00DE1005 ESI = 00DE1005 EDI = 00DE1005 EIP = 00DE1016 ESP = 00CFFCE8 EBP = 00CFFCF4 EFL = 00000246

 OV = 0 UP = 0 EI = 1 PL = 0 ZR = 1 AC = 0 PE = 1 CY = 0

 0x00DE4008 = 00000028
```

Explanation: EBX register is 32-bit long with an unsigned integer variable. This register is updated by storing the Yval, or 30 (hex = 1E) with *mov ebx, Yval*.

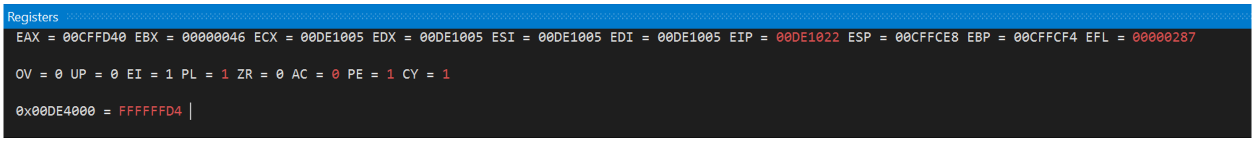
Line number: 18
Instruction: add ebx, Zval
Register Values: EBX = 00000046
Screenshot:



```
Registers
 EAX = 00CFFD40 EBX = 00000046 ECX = 00DE1005 EDX = 00DE1005 ESI = 00DE1005 EDI = 00DE1005 EIP = 00DE101C ESP = 00CFFCE8 EBP = 00CFFCF4 EFL = 00000212

 OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 1 PE = 0 CY = 0

 0x00DE4000 = 0000001A
```

Explanation: Adding the current EBX (30/hex = 1E) with Zval (40/hex = 28). The sum of 1E and 28 is 46.
If 30 + 40 = 70, then 1E + 28 = 46

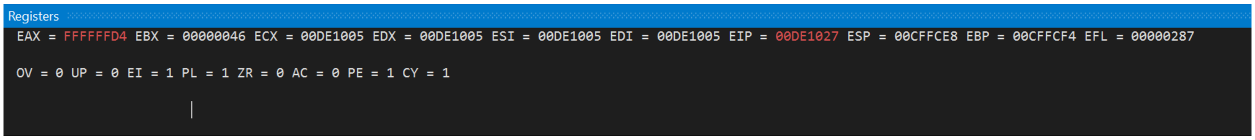
Line number: 20
Instruction: sub Xval, ebx
Register Values: Xval = FFFFFFD4
Screenshot:



```
Registers
 EAX = 00CFFD40 EBX = 00000046 ECX = 00DE1005 EDX = 00DE1005 ESI = 00DE1005 EDI = 00DE1005 EIP = 00DE1022 ESP = 00CFFCE8 EBP = 00CFFCF4 EFL = 00000287

 OV = 0 UP = 0 EI = 1 PL = 1 ZR = 0 AC = 0 PE = 1 CY = 1

 0x00DE4000 = FFFFFFD4
```

Explanation: Subtracting the Xval (26/hex = 1A) with the value of the EBX (70/hex = 46). The difference between 1A and 46 is FFD4.
if 26 - 70 = -44, then 1A - 46 = -2C or FFD4

Line number: 21
Instruction: mov ebx, Yval
Register Values: EAX = FFFFFFD4
Screenshot:



```
Registers
 EAX = FFFFFFD4 EBX = 00000046 ECX = 00DE1005 EDX = 00DE1005 ESI = 00DE1005 EDI = 00DE1005 EIP = 00DE1027 ESP = 00CFFCE8 EBP = 00CFFCF4 EFL = 00000287

 OV = 0 UP = 0 EI = 1 PL = 1 ZR = 0 AC = 0 PE = 1 CY = 1
```

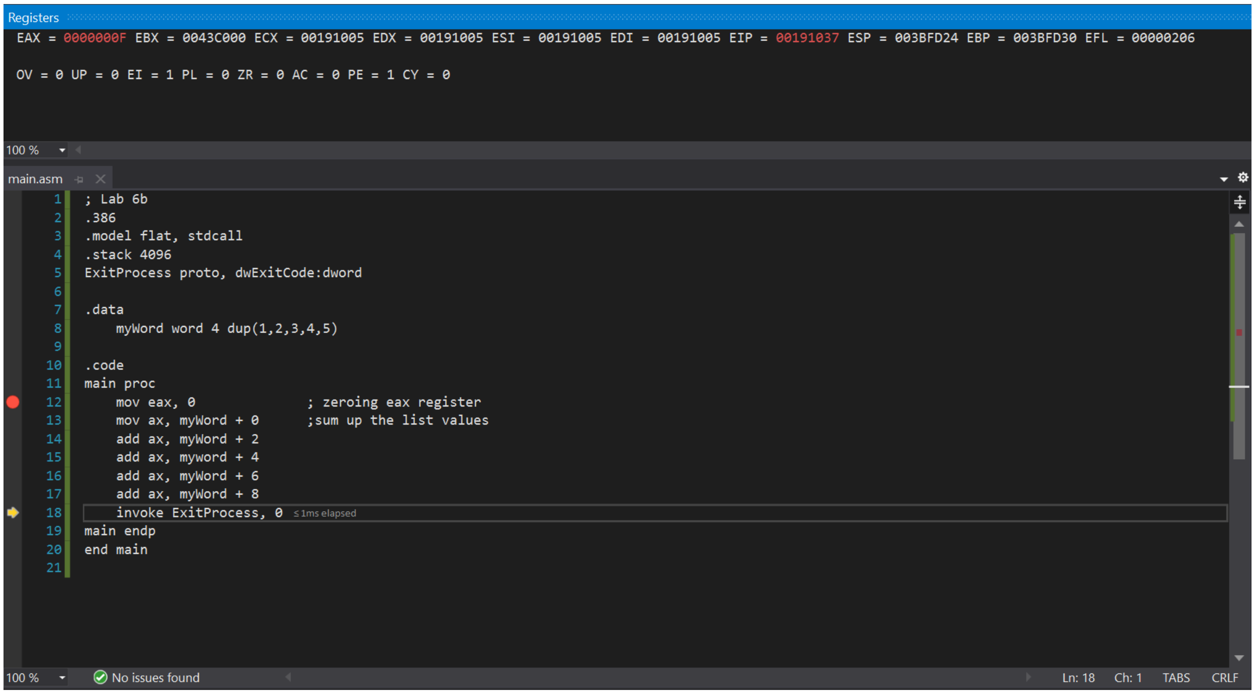
Explanation: EAX register is 32-bit long with a signed integer variable. This register is updated by storing the difference of Xval and EBX.

Lab 6(b)

(1) What is the total size of the myWord array? (1 Point)
The total size of myWord array is 40 bytes

word = 2 bytes
[1, 2, 3, 4, 5] = 2 * 5 = 10
4 * 10 = 40

(2) Debug the code until the 'invoke ExitProcess, 0'. Attach screenshot showing the content of AX register. (2 points)



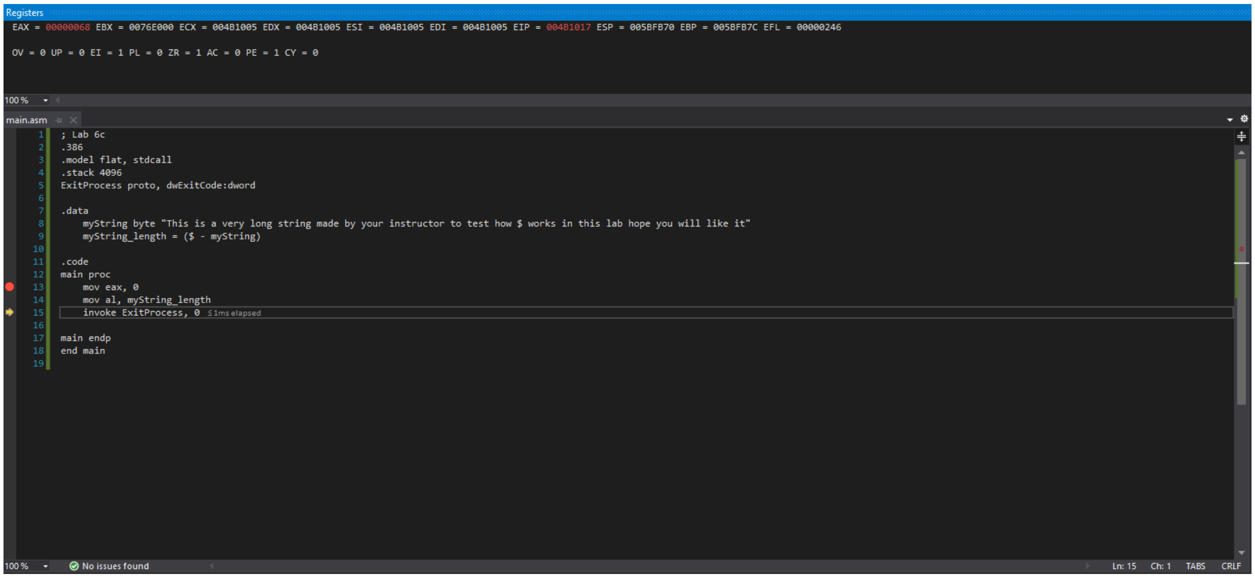
```
EAX = 0000000F EBX = 0043C000 ECX = 00191005 EDX = 00191005 ESI = 00191005 EDI = 00191005 EIP = 00191037 ESP = 003BFD24 EBP = 003BFD30 EFL = 00000206

OV = 0 UP = 0 EI = 1 PL = 0 ZR = 0 AC = 0 PE = 1 CY = 0
```

```
; Lab 6b
.386
.model flat, stdcall
.stack 4096
ExitProcess proto, dwExitCode:dword

.data
    myWord word 4 dup(1,2,3,4,5)

.code
main proc
    mov eax, 0              ; zeroing eax register
    mov ax, myWord + 0      ;sum up the list values
    add ax, myWord + 2
    add ax, myword + 4
    add ax, myWord + 6
    add ax, myWord + 8
    invoke ExitProcess, 0
main endp
end main
```

Lab 6(c):

(1) What is the difference between symbolic constant and variables? (1 point)
Symbolic constants are represented by a name, and after it is initialized, its value cannot be changed. Variables are constants that are only represented by a name.

(2) Debug the code until 'invoke ExitProcess, 0'. Attach the screenshot showing the content of al register. (2 points)



```
EAX = 00000068 EBX = 0076E000 ECX = 004B1005 EDX = 004B1005 ESI = 004B1005 EDI = 004B1005 EIP = 004B1017 ESP = 005BFB70 EBP = 005BFB7C EFL = 00000246

OV = 0 UP = 0 EI = 1 PL = 0 ZR = 1 AC = 0 PE = 1 CY = 0
```

```
; Lab 6c
.386
.model flat, stdcall
.stack 4096
ExitProcess proto, dwExitCode:dword

.data
    myString byte "This is a very long string made by your instructor to test how $ works in this lab hope you will like it"
    myString_length = ($ - myString)

.code
main proc
    mov eax, 0
    mov al, myString_length
    invoke ExitProcess, 0

main endp
end main
```