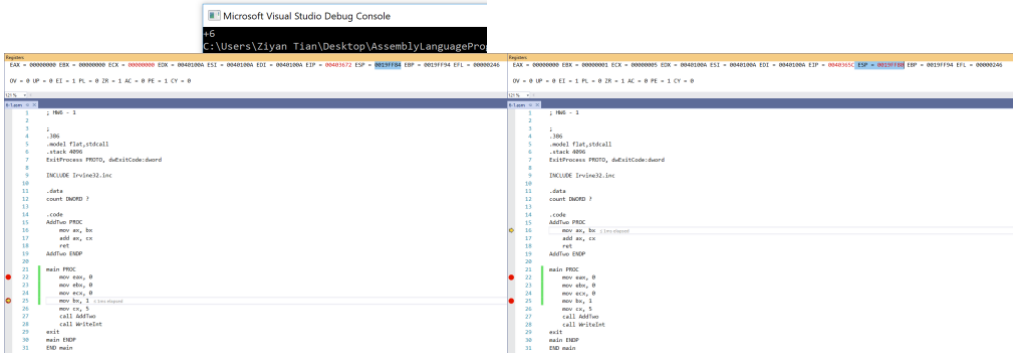
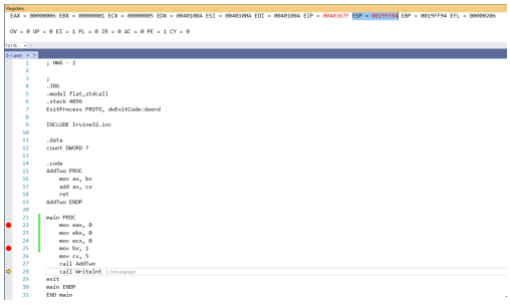
Comp 3350: Computer Organization & Assembly Language
HW # 6: Theme: Stacks and Procedures
All main questions carry equal weight.
(Credit awarded to only those answers for which work has been shown.)

1. [Procedures] Write a main program which sets the registers BX and CX and calls a procedure *Add-Two*. The procedure *Add-Two* adds the values in registers BX and CX and returns the output (which is the sum) in AX. Single step through the program, displaying the value of the stack pointer so that you understand how the call and return are implemented.
   code 18 points; set the registers, call Add-Two, add the values in BX and CX, return output in AX, 4* 4.5 points
   the pointer value 3 * 5 points, it needs at least 3 steps to show how the call and return are implemented

```
; HW6 - 1

;
.386
.model flat,stdcall
.stack 4096
ExitProcess PROTO, dwExitCode:dword

INCLUDE Irvine32.inc

.data
count DWORD ?

.code
AddTwo PROC
    mov ax, bx
    add ax, cx
    ret
AddTwo ENDP

main PROC
    mov eax, 0
    mov ebx, 0
    mov ecx, 0
    mov bx, 1
    mov cx, 5
    call AddTwo
    call WriteInt
exit
main ENDP
END main
```





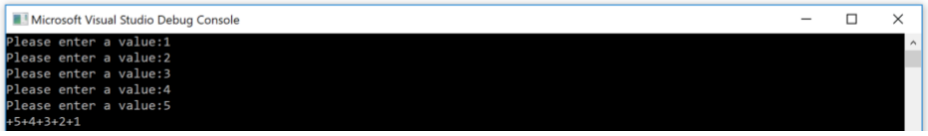
2. [Arrays] Write a program that:

1) Prompts the user for integer input 5 times — 8 points
2) Stores these inputs in a stack using the Push instruction — 8 points
3) After the storing is complete in the Step 2, pop the stored values and display them on the screen using WriteInt (not DumpRegs). — 8 points

In your submission, please embed the full program (.asm and .lst file) and one screen shot with at least one positive and one negative input value. Use the following: files 6 points, screenshot 3 points

```
.data
PromptUser BYTE "Please enter a value:", 0
```

```
.data
PromptUser BYTE "Please enter a value:", 0

.code
main PROC

    ; 1) Prompt the user for integer 5 times
    mov ecx, 5
L1:
    mov edx, OFFSET PromptUser
    call WriteString

    ; 2) Store the input in a stack
    call ReadInt
    push eax
    loop L1

    ; 3) Displays the stored values
    mov ecx, 5
L2: pop eax
    call WriteInt
    loop L2


exit
main ENDP
END main
```



3. [Compares, Procedures] Write a procedure, *Search* which searches the stack for the value that you provide in register AX and returns its index, assuming the first value is stored in index *0*. Write a main program that fills the stack with positive values, sets AX and calls *Search* and prints the index at which the value was found.

For example, if the inputs are: 5, 6, 1, 10, 44, 79
and AX is set in the main to be 1, then the expected output of your code is:

```
The target value is 1, and is located at index:2
```

In cases where more than one element has the same value, you only have to output one of them. If the value is not found, print -1.

Use the following:

```
.data
prompt      BYTE "Please input a value: ", 0
spacing     BYTE ", ",0;
String2     BYTE "The target value is," 0
String2     BYTE "and is located at index: ",0
```

In your submission, please embed the full program (.asm and .lst file) and one screen shot showing the values found. Please test several sets of positive and negative values

code 24 points, files 6 points, screen shot 3points

```
.code
main PROC

    ; 1) Prompt the user for integer 6 times
    mov ecx, 6
L1: mov edx, OFFSET prompt
    call WriteString

    ; 2) Store the input in a stack
    call ReadInt
    push eax
    loop L1

    ; AX = 1,
    mov eax, 1

; Search
    mov ecx, 6
L2: pop ebx
    cmp eax, ebx
    jz L3
    loop L2

L3: cmp ecx, 0
    jnz L5
L4: cmp eax, ebx
    jnz L5
    add ecx, 1
L5: mov edx, OFFSET String1
    call WriteString
    call WriteInt
    mov edx, OFFSET spacing
    call WriteString
    mov edx, OFFSET String2
    call WriteString
    sub ecx, 1
    mov eax, ecx
    call WriteInt

exit
main ENDP
END main
```