Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

1. [Memory Map] Fill in the following memory diagram with the data provided below. Please assume that the data segment begins at 0x0045B700.

.data

Rose BYTE 0A4h

Magnolia WORD 0CE21h

Cannas DWORD 8CB12F19h, 0A2B2h

Variable	*Address*	*Data*
Rose	0x0045B700	*0A4h*
Magnolia	0x0045B701	*OCE21h*
Cannas	0x0045B703	*8CB12F19h*
Cannas(2)	0x0045B707	*0A2B2h*

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

2. [Addressing Modes] Copy the following code into your assembly development environment and single-step through it. For each single step execution, submit the screenshot. For those instructions referencing memory, do the linear address computation by hand and typewrite it.

; Assume that memory segment begins at 0x0000 4020

```
TITLE Addressing Modes          (main.asm)


INCLUDE Irvine32.inc
.data
 alpha        DWORD      0A1B1C1D1h, 87654321h
 beta         DWORD       67EED9FCh, 21A220C2h
 gamma        DWORD      0BCB1D44Fh
.code
main PROC
        mov eax, 1C2Fh;        Immediate                  ; EAX = 0x????1C2F
        mov ecx, eax;          Register to Register       ; ECX = 0x1C2F
        mov edi, OFFSET beta;  Immediate                  ; edi = 0x0000 4028
        mov [gamma], eax;      Direct                     ; EAX = 0x0000 1C2F
        mov esi, gamma;        Direct                     ; esi = 0x00001C2F
        mov esi, 4;            Immediate                  ; esi = 00000004
        mov eax, beta[esi];    Indirect-offset            ; EAX = C2 (beta(esi + 4))
        mov ebx, OFFSET alpha; Immediate                  ; EBX = 0x0000 4020
        mov eax, 4[ebx];       Indirect-displacement      ; EAX = 0x8765 4321
```

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

```
    mov eax, [ebx];        Indirect                              ; EAX = A1B1C1D1

mov eax,4[ebx][esi]; Base-Indirect-displacement ; EAX = EBX + 8, 0x67EED9Fch

exit

main ENDP

END main
```

Location	**Value**	**Memory Address**
alpha	D1	0x0000 4020
alpha + 1	C1	0x0000 4021
alpha + 2	B1	0x0000 4022
alpha + 3	A1	0x000 4023
alpha + 4	21	0x0000 4024
alpha + 5	43	0x0000 4025
alpha + 6	65	0x0000 4026
alpha + 7	87	0x0000 4027
beta	FC	0x0000 4028
beta + 1	D9	0x0000 4029
beta + 2	EE	0x0000 402A
beta + 3	67	0x0000 402B
beta + 4	C2	0x0000 402C
beta + 5	2D	0x0000 402D

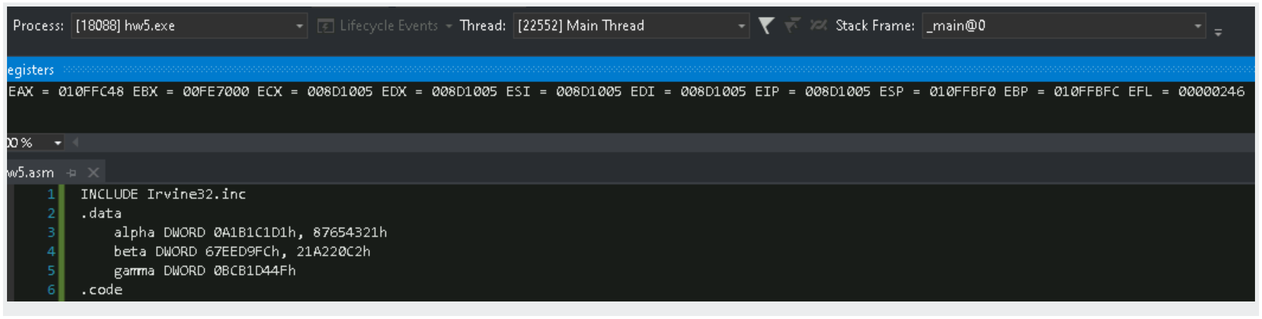
Noah Jones

Dr. Baskiyar

COMP 3350

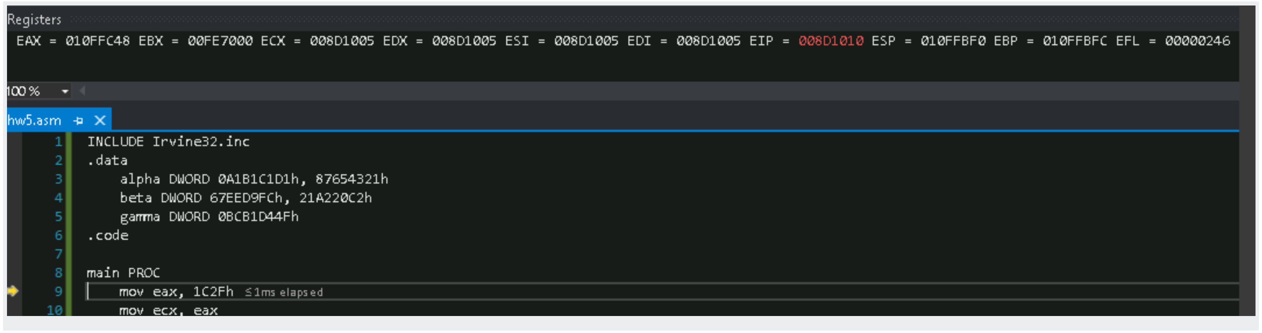
HW #5

beta + 6	A2	0x0000 402E
beta + 7	21	0x0000 402F
gamma	4F	0x0000 4030
gamma + 1	D4	0x0000 4031
gamma + 2	B1	0x0000 4032
gamma + 3	BC	0x0000 4033



```
EAX = 010FFC48 EBX = 00FE7000 ECX = 008D1005 EDX = 008D1005 ESI = 008D1005 EDI = 008D1005 EIP = 008D1005 ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246
```

```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code
```



```
EAX = 010FFC48 EBX = 00FE7000 ECX = 008D1005 EDX = 008D1005 ESI = 008D1005 EDI = 008D1005 EIP = 008D1010 ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246
```

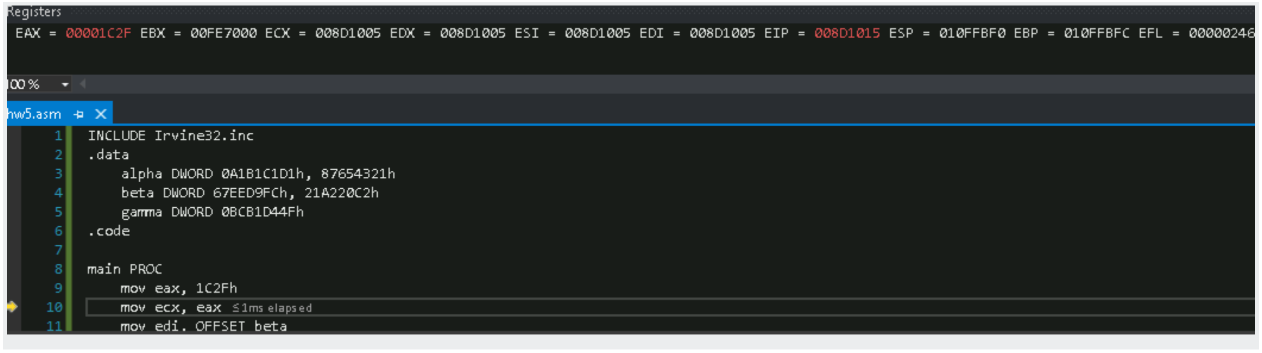
```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh  ≤1ms elapsed
    mov ecx, eax
```

Noah Jones

Dr. Baskiyar

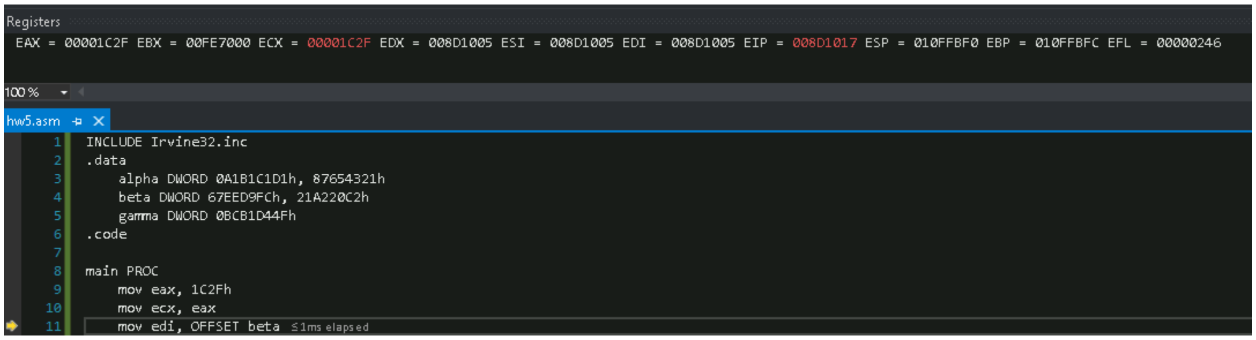
COMP 3350

HW #5



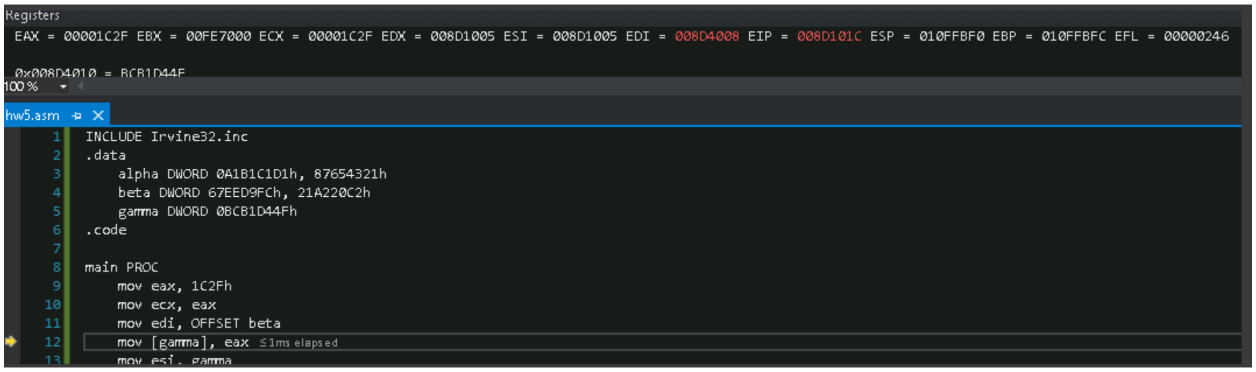
```
Registers
 EAX = 00001C2F EBX = 00FE7000 ECX = 008D1005 EDX = 008D1005 ESI = 008D1005 EDI = 008D1005 EIP = 008D1015 ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246
```

```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh
    mov ecx, eax  ≤1ms elapsed
    mov edi, OFFSET beta
```



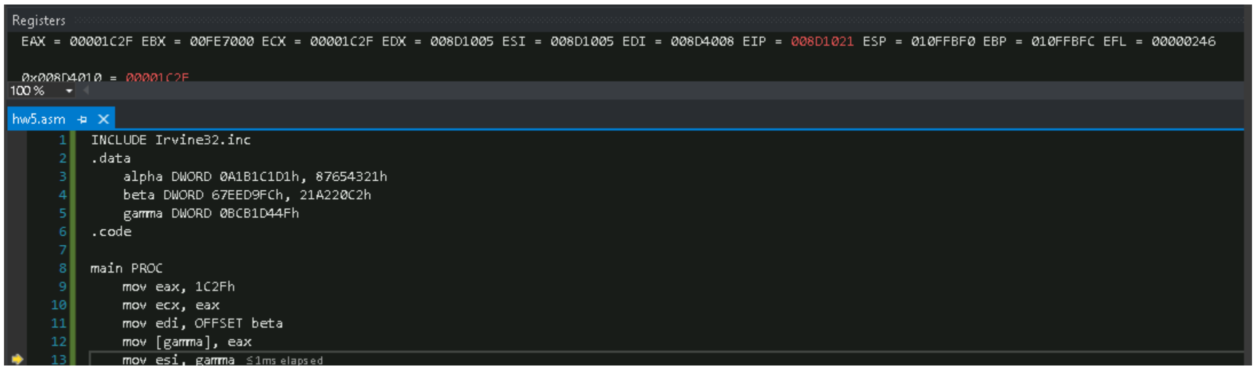
```
Registers
 EAX = 00001C2F EBX = 00FE7000 ECX = 00001C2F EDX = 008D1005 ESI = 008D1005 EDI = 008D1005 EIP = 008D1017 ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246
```

```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh
    mov ecx, eax
    mov edi, OFFSET beta  ≤1ms elapsed
```



```
Registers
 EAX = 00001C2F EBX = 00FE7000 ECX = 00001C2F EDX = 008D1005 ESI = 008D1005 EDI = 008D4008 EIP = 008D101C ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246

 0x008D4010 = BCB1D44F
```

```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh
    mov ecx, eax
    mov edi, OFFSET beta
    mov [gamma], eax  ≤1ms elapsed
    mov esi, gamma
```



```
Registers
 EAX = 00001C2F EBX = 00FE7000 ECX = 00001C2F EDX = 008D1005 ESI = 008D1005 EDI = 008D4008 EIP = 008D1021 ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246

 0x008D4010 = 00001C2F
```

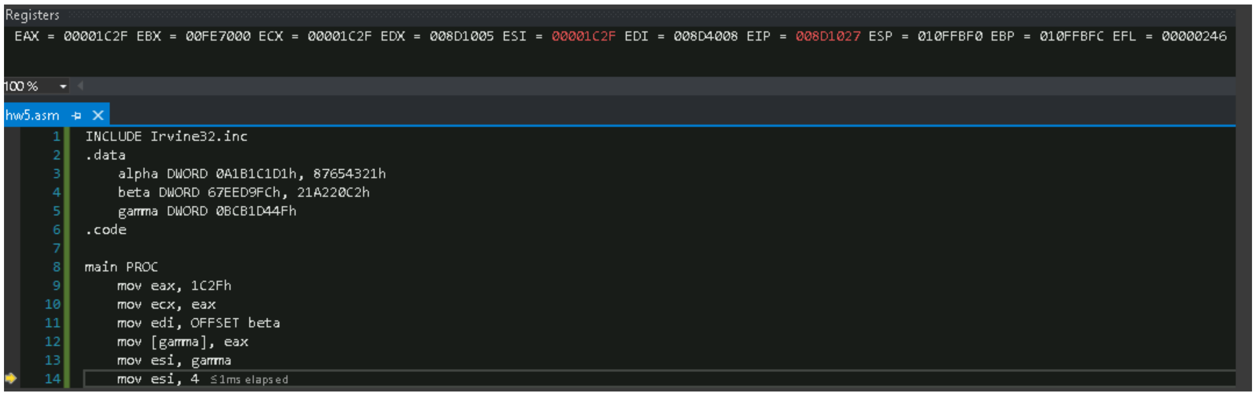
```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh
    mov ecx, eax
    mov edi, OFFSET beta
    mov [gamma], eax
    mov esi, gamma  ≤1ms elapsed
```

Noah Jones

Dr. Baskiyar

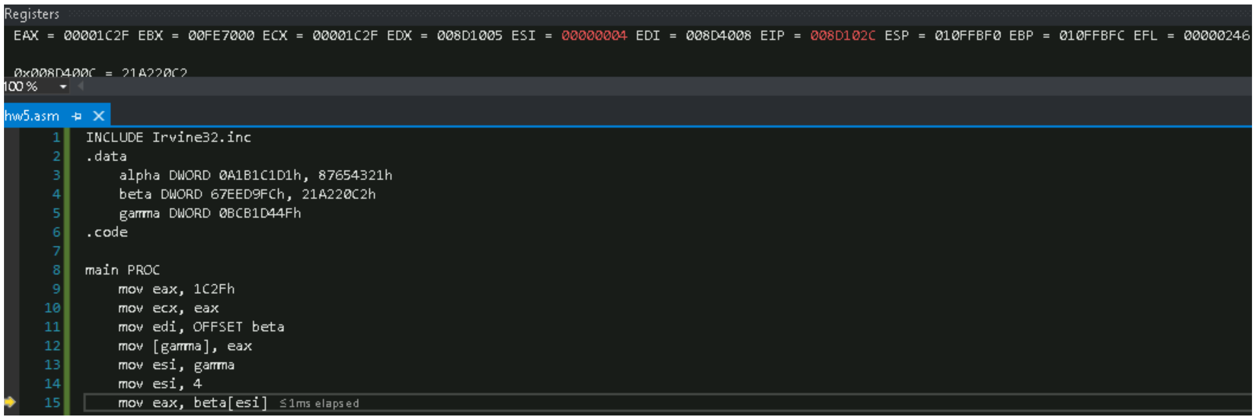
COMP 3350

HW #5



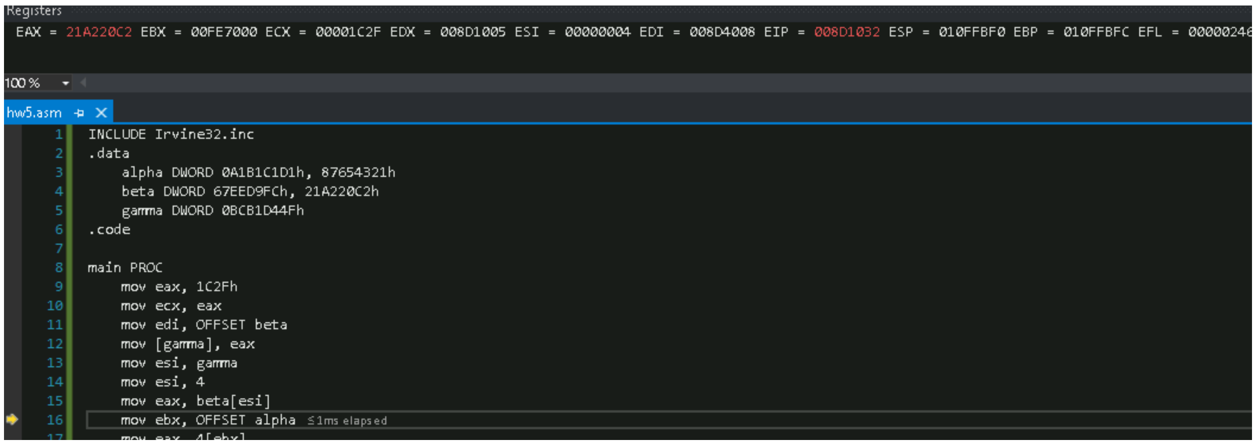
```
Registers
EAX = 00001C2F EBX = 00FE7000 ECX = 00001C2F EDX = 008D1005 ESI = 00001C2F EDI = 008D4008 EIP = 008D1027 ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246
```

```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh
    mov ecx, eax
    mov edi, OFFSET beta
    mov [gamma], eax
    mov esi, gamma
    mov esi, 4  ≤1ms elapsed
```



```
Registers
EAX = 00001C2F EBX = 00FE7000 ECX = 00001C2F EDX = 008D1005 ESI = 00000004 EDI = 008D4008 EIP = 008D102C ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246

0x008D400C = 21A220C2
```

```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh
    mov ecx, eax
    mov edi, OFFSET beta
    mov [gamma], eax
    mov esi, gamma
    mov esi, 4
    mov eax, beta[esi]  ≤1ms elapsed
```



```
Registers
EAX = 21A220C2 EBX = 00FE7000 ECX = 00001C2F EDX = 008D1005 ESI = 00000004 EDI = 008D4008 EIP = 008D1032 ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246
```

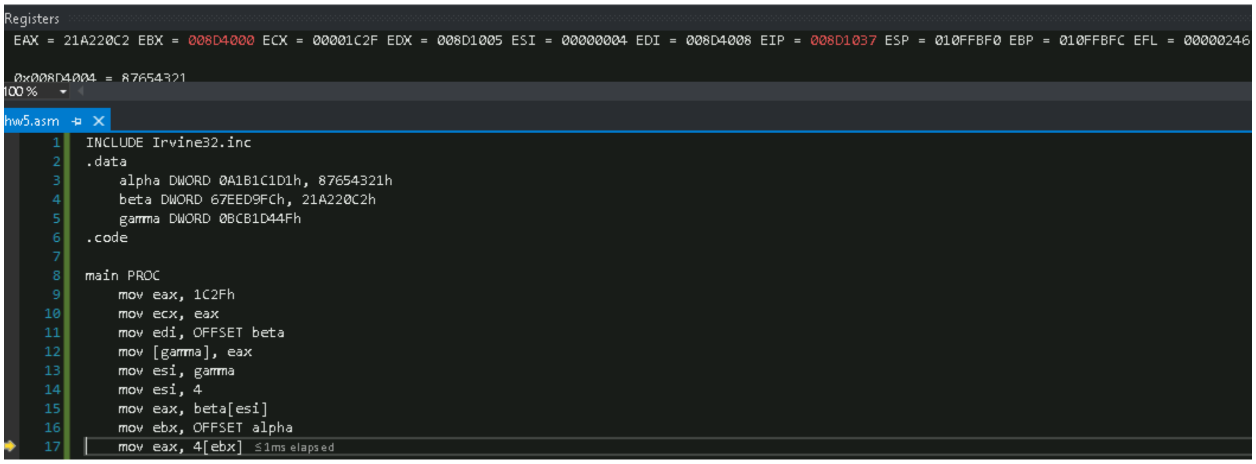
```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh
    mov ecx, eax
    mov edi, OFFSET beta
    mov [gamma], eax
    mov esi, gamma
    mov esi, 4
    mov eax, beta[esi]
    mov ebx, OFFSET alpha  ≤1ms elapsed
    mov eax, 4[ebx]
```

Noah Jones

Dr. Baskiyar

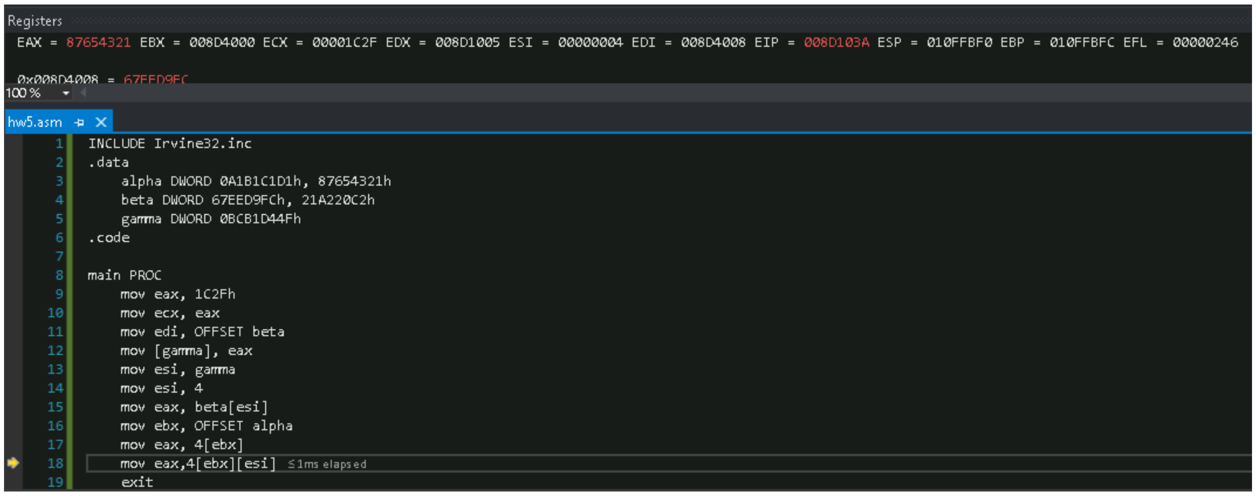
COMP 3350

HW #5



```
EAX = 21A220C2 EBX = 008D4000 ECX = 00001C2F EDX = 008D1005 ESI = 00000004 EDI = 008D4008 EIP = 008D1037 ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246

0x008D4004 = 87654321
```

```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh
    mov ecx, eax
    mov edi, OFFSET beta
    mov [gamma], eax
    mov esi, gamma
    mov esi, 4
    mov eax, beta[esi]
    mov ebx, OFFSET alpha
    mov eax, 4[ebx]  ≤1ms elapsed
```



```
EAX = 87654321 EBX = 008D4000 ECX = 00001C2F EDX = 008D1005 ESI = 00000004 EDI = 008D4008 EIP = 008D103A ESP = 010FFBF0 EBP = 010FFBFC EFL = 00000246

0x008D4008 = 67EED9FC
```

```
INCLUDE Irvine32.inc
.data
    alpha DWORD 0A1B1C1D1h, 87654321h
    beta DWORD 67EED9FCh, 21A220C2h
    gamma DWORD 0BCB1D44Fh
.code

main PROC
    mov eax, 1C2Fh
    mov ecx, eax
    mov edi, OFFSET beta
    mov [gamma], eax
    mov esi, gamma
    mov esi, 4
    mov eax, beta[esi]
    mov ebx, OFFSET alpha
    mov eax, 4[ebx]
    mov eax,4[ebx][esi]  ≤1ms elapsed
    exit
```

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

3. [Indirect addressing] Write a program that first displays all the elements of Array1, Array2 and Array3. Then, the program should subtract all the odd indexed elements of Array2 from the odd indexed elements of Array1 and store the result in Array3; e.g. Array3 [7] = Array1 [7] - Array2 [7]. Next, it must add the even indexed elements of Array1 and Array 2 and store them in the corresponding even indexed elements of Array3, e.g. Array3 [4] = Array1 [4] + Array2 [4]. Next, display the elements of all the arrays after these operations. Submit screenshots of the displays of the elements of all the arrays. You can use WriteInt or WriteHex to display the elements of the arrays. Fill in Array1 and Array2 each by your own ten numbers each using both positive and negative integers.

```
INCLUDE Irvine32.inc

.data

   Array1 SWORD 7, 8, 9, -7, -8, -9, 6, 5, 4, 3

   Array2 SWORD -4, -6, -8, -2, 4, 6, 8, 9, 7, 5

   Array3 SWORD 10 DUP (?)

   Array1Label BYTE "Array1: ",0

   Array2Label BYTE "Array2: ",0

   Array3Label BYTE "Array3: ",0

   Array3aLabel BYTE "Array3 after Addition:     ",0

   Array3sLabel BYTE "Array3 after Subtraction:  ",0

.code


main PROC

   ;Label for a1

   mov edx, OFFSET Array1Label
```

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

```
  call WriteString

  ; Display all elements of Array1
  mov esi, OFFSET Array1
  mov ecx, LENGTHOF Array1

  DisplayArray1:
      movsx eax, WORD PTR [esi]
      call WriteInt
      add esi, 2
      loop DisplayArray1

  call Crlf

      ;Label for a2
  mov edx, OFFSET Array2Label
  call WriteString

  ; Display all elements of Array2
  mov esi, OFFSET Array2
  mov ecx, LENGTHOF Array2
```

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

```
DisplayArray2:
    movsx eax, WORD PTR [esi]
    call WriteInt
    add esi, 2
    loop DisplayArray2

call Crlf
    ;Label for a3
mov edx, OFFSET Array3Label
call WriteString

; Display Array3 Before Operations
mov esi, OFFSET Array3
mov ecx, LENGTHOF Array3        ; loop counter

DisplayArray3:
    movsx eax, WORD PTR [esi]
    call WriteInt
    add esi, 2
    loop DisplayArray3

call Crlf
```

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

```
  ; Subtract odd elements of Array2 from odd elements of Array1 and store the result in Array3
  mov esi, OFFSET Array1 + 2
  mov edi, OFFSET Array2 + 2
  mov ebx, OFFSET Array3 + 2
  mov ecx, LENGTHOF Array1 / 2 ; Divide by 2 to process only odd-indexed elements

  SubtractOdds:
    movsx eax, WORD PTR [esi]
    movsx edx, WORD PTR [edi]
    sub eax, edx
    mov WORD PTR [ebx], ax

    add esi, 4
    add edi, 4
    add ebx, 4
    loop SubtractOdds

  ;Label for a3
  mov edx, OFFSET Array3sLabel
  call WriteString
```

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

```
  mov esi, OFFSET Array3
  mov ecx, LENGTHOF Array3

  DisplayArray3AfterSubtraction:
    movsx eax, WORD PTR [esi]
    call WriteInt
    add esi, 2
    loop DisplayArray3AfterSubtraction

  call Crlf



    ;Label for a3
  mov edx, OFFSET Array3aLabel
  call WriteString
  ; Add even elements of Array1 and Array2 and store them in Array3
  mov esi, OFFSET Array1
  mov edi, OFFSET Array2
  mov ebx, OFFSET Array3
  mov ecx, LENGTHOF Array1 / 2

  AddEvens:
```

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

```
        movsx eax, WORD PTR [esi]
        add esi, 4
        movsx ecx, WORD PTR [edi]
        add edi, 4
        add eax, ecx
        mov WORD PTR [ebx], ax

        add ebx, 4
        loop AddEvens

    mov esi, OFFSET Array3
    mov ecx, LENGTHOF Array3

    DisplayArray3AfterAddition:
        movsx eax, WORD PTR [esi]
        call WriteInt
        add esi, 2
        loop DisplayArray3AfterAddition

    call Crlf
    exit
main ENDP
```

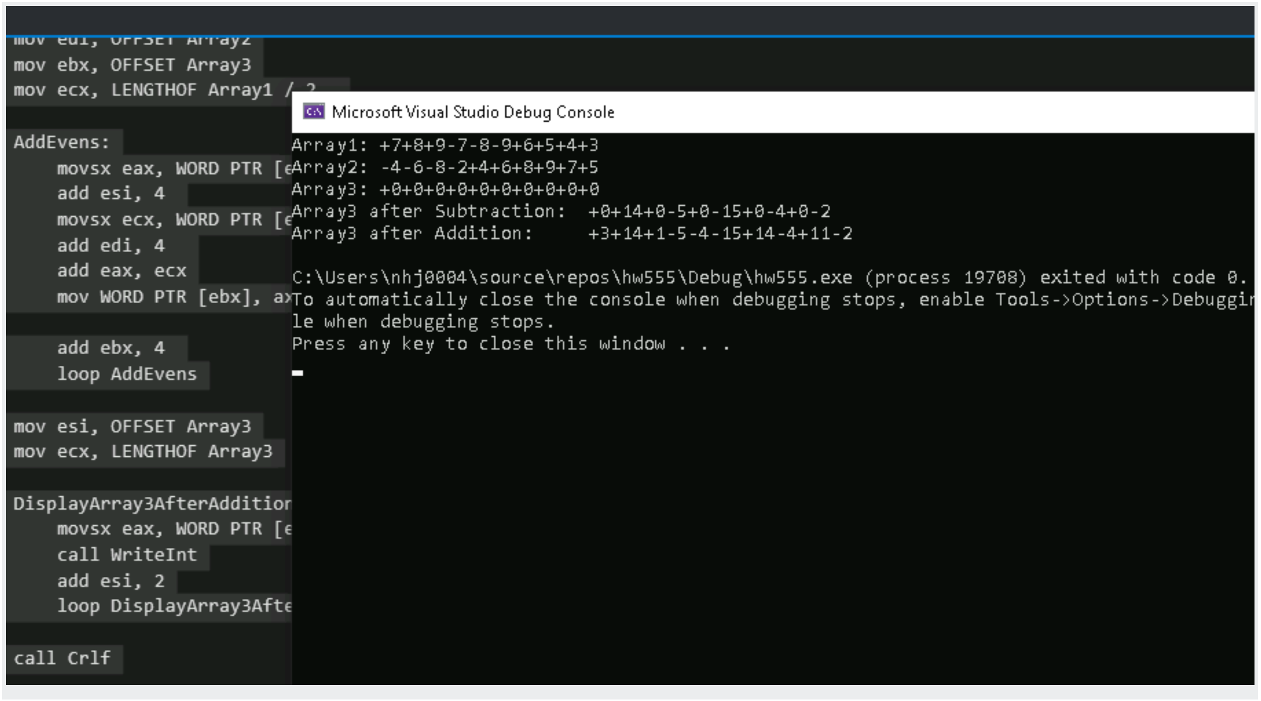
Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

END main



```
mov edi, OFFSET Array2
mov ebx, OFFSET Array3
mov ecx, LENGTHOF Array1 / 2

AddEvens:
    movsx eax, WORD PTR [e
    add esi, 4
    movsx ecx, WORD PTR [e
    add edi, 4
    add eax, ecx
    mov WORD PTR [ebx], ax

    add ebx, 4
    loop AddEvens

mov esi, OFFSET Array3
mov ecx, LENGTHOF Array3

DisplayArray3AfterAdditior
    movsx eax, WORD PTR [e
    call WriteInt
    add esi, 2
    loop DisplayArray3Afte

call Crlf
```

Microsoft Visual Studio Debug Console

```
Array1: +7+8+9-7-8-9+6+5+4+3
Array2: -4-6-8-2+4+6+8+9+7+5
Array3: +0+0+0+0+0+0+0+0+0+0
Array3 after Subtraction:  +0+14+0-5+0-15+0-4+0-2
Array3 after Addition:     +3+14+1-5-4-15+14-4+11-2

C:\Users\nhj0004\source\repos\hw555\Debug\hw555.exe (process 19708) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debuggir
le when debugging stops.
Press any key to close this window . . .
```

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

4. [Loops] Declare a signed word array. Write a program to print on screen the first *n* positive elements of the array, using the Loop instruction. One sample array is given below. You should test with other sample created arrays and with multiple sized arrays. Make sure you have a good mix of positive and negative integers.

.data

MySigned Array SWORD -1, 78, 0AC, 4567, -7, -273, 92

1. **Prompt user for integer *n*,**
2. **Read the value of *n* from user input**

Please use the "WriteInt" procedure, not "DumpRegs". Other relevant procedures: "ReadInt" and "WriteString." In your homework submission, please embed both the code and one screen shot for *n* = *6*.

```
INCLUDE Irvine32.inc

.data

   MySignedArray SWORD -1, 78, 6798, 4567, -7, -273, 92, 65346, 5465, 645, 35, 4

   promptMsg BYTE "Enter the value of n: ", 0


.code

main PROC

   mov edx, OFFSET promptMsg

   call WriteString


   ; Read the value of n from user input

   call ReadInt

   mov ecx, eax ; Move n to the loop counter
```

Noah Jones

Dr. Baskiyar

COMP 3350

HW #5

```
mov esi, OFFSET MySignedArray

mov ebx, 0 ; Counter for positive elements encountered


PrintPositiveElements:

    cmp ebx, ecx ; Compare the counter with n

    jge EndLoop ; If counter  >=  n, exit loop


    movsx eax, WORD PTR [esi] ; Load the signed word into eax

    cmp eax, 0 ; Compare the value with zero

    jle SkipNonPositive ; If the value is not positive or zero, skip printing


    call WriteInt ; Write the positive element

    call Crlf ; Add a line break


    inc ebx ; Increment the counter for positive elements


    SkipNonPositive:

    add esi, 2 ; Move to the next element in the array

    jmp PrintPositiveElements ; Continue the loop


EndLoop:

exit
```

Noah Jones

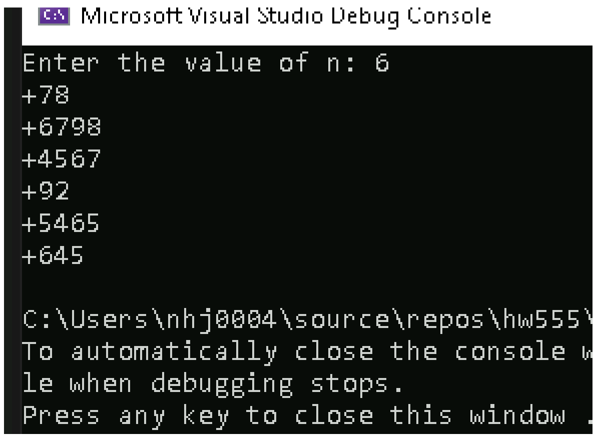
Dr. Baskiyar

COMP 3350

HW #5

main ENDP

END main



```
Enter the value of n: 6
+78
+6798
+4567
+92
+5465
+645

C:\Users\nhj0004\source\repos\hw555\
To automatically close the console w
le when debugging stops.
Press any key to close this window .
```