Noah Jones

Dr. Baskiyar

COMP-3350

HW #4

1.

A. Write a program fragment that will set the sign flag.

```
.code
  mov eax, 4    ; puts value '4' into eax register
  sub eax, 5    ; EAX = (-1) , SF = 1
```

B. Write a program fragment that will clear the carry flag.

```
.code
  mov al, 0FFh          ; hex value for 255(10)
  add al, 1             ; AL = 00, CF = 1
```

C. Write a program fragment that sets the overflow flag.

```
.code
  mov al, -128  ; puts value '-128' into al register
  sub al, 1     ; AL = 127, OF = 1
```

D. What will be the value of the parity flag after the following lines execute? Show your work.

mov al, 34h

add al, 10h

34h

+ 10h

= 44h

4_{10} -> 0100_2

So binary value of (4)(4) = (0100)(0100) = 01000100

To determine the value of PF, count the number of '1' in the binary string above …

There are two '1' values in 01000100

2 is even, so the PF is set to '1'

So, 1

Noah Jones
Dr. Baskiyar
COMP-3350
HW #4

2. Given the following data declarations: Write instructions that sum the elements of the array into AX and then save the resultant sum in the location Total.

```
.data
 MyArray  BYTE 4Eh, 64h, 9Ah, 7Fh, 3Ch
 Total    WORD ?

.code
                                ; Sum MyArray  into register AX
mov AX, 0                       ; initialize running sum: AX
                                ;get first byte and add to AX
mov BL, byte PTR MyArray        ; BL = 0x4E
MOVZX BX, BL                    ; BX = 0x004E
ADD AX, BX                      ; AX = 0x0000 + 0x004E
                                ;get and add second byte
mov BL, byte PTR MyArray + 1    ;BL = 0x64
MOVZX BX, BL                    ;BX = 0x0064
ADD AX, BX                      ; AX = 0x004E + 0x0064
                                ;get and add third byte
mov BL, byte PTR MyArray + 2    ;BL = 0x9A
MOVZX BX, BL                    ;BX = 0x009A
ADD AX, BX                      ;AX = 0x004E + 0x0064 + 0x009A
                                ;get and add third byte
mov BL, byte PTR MyArray + 3    ;BL = 0x7F
MOVZX BX, BL                    ;BX = 0x007F
ADD AX, BX                      ;AX = 0x004E + 0x0064 + 0x009A + 0x007F
                                ;get and add last byte
mov BL, byte PTR MyArray + 4    ;BL = 0x3C
MOVZX BX, BL                    ;BX = 0x003C
ADD AX, BX                      ;AX = 0x004E + 0x0064 + 0x009A + 0x007F + 0x003C
                                ;Save result in total
```

Noah Jones

Dr. Baskiyar

COMP-3350

HW #4

mov [TOTAL], AX ;move the value from register AX to 'TOTAL'

3. (**Little Endian**) **Fill in the requested register values after the executions of the instructions:**

Show the memory map using an address-data table.

```
.data
;assume data segments starts at memory address 0x0040 2070
myBytes     BYTE   27h, 5Ah, 7Bh, 0CEh
myWords     WORD   45A8h, 49A3h, 0AC32h, 257Bh, 0DF30h
myDoubles   DWORD  71D2h, 0B114h, 67F2h, 0E276h, 1234h
myPointer   DWORD  myDoubles

.code
   mov esi, OFFSET myBytes          ;     ESI =  0x0040 2070 (address of first byte)
   mov ax, WORD PTR [esi+1]         ;     AX =  5Ah (value of myBytes at 0x0040 2071)
   mov eax, DWORD PTR myWords       ;     EAX = 45A8h (value of myWords at 0x0040 2070)
   mov esi, myPointer
   mov ax, WORD PTR [esi+2]         ;     AX =F2h ( low-byte myDoubles(where esi is) + 2)
//does the pointer reset to esi[0] now, or does it stay at esi[2]?
   mov ax, WORD PTR [esi+3]         ;     AX = E276h (value at +3 of esi)
   mov ax, WORD PTR [esi-2]         ;     AX = 7Bh (value of PTR - 2)
```

Noah Jones

Dr. Baskiyar

COMP-3350

HW #4

0x0040 2070	27h	*myBytes(1 Byte Each)*
0x0040 2071	5Ah	*myBytes*
0x0040 2072	7Bh	*myBytes*
0x0040 2073	CEh	*myBytes*
0x0040 2074	A8h	*myWords(2 Bytes Each)*
0x0040 2075	45h	*myWords*
0x0040 2076	A3h	*myWords*
0x0040 2077	49h	*myWords*
0x0040 2078	32h	*myWords*
0x0040 2079	ACh	*myWords*
0x0040 207A	7Bh	*myWords*
0x0040 207B	25h	*myWords*
0x0040 207C	30h	*myWords*
0x0040 207D	DFh	*myWords*
0x0040 207E	D2h	*myDoubles(4 Bytes Each)*
0x0040 207F	71h	*myDoubles*
0x0040 2080	00h	*myDoubles*
0x0040 2081	00h	*myDoubles*
0x0040 2082	14h	*myDoubles*
0x0040 2083	B1h	*myDoubles*

Noah Jones

Dr. Baskiyar

COMP-3350

HW #4

0x0040 2084	00h	*myDoubles*
0x0040 2085	00h	*myDoubles*
0x0040 2086	F2h	*myDoubles*
0x0040 2087	67h	*myDoubles*
0x0040 2088	00h	*myDoubles*
0x0040 2089	00h	*myDoubles*
0x0040 208A	76h	*myDoubles*
0x0040 208B	E2h	*myDoubles*
0x0040 208C	00h	*myDoubles*
0x0040 208D	00h	*myDoubles*
0x0040 208E	34h	*myDoubles*
0x0040 208F	12h	*myDoubles*
0x0040 2090	00h	*myDoubles*
0x0040 2091	00h	*myDoubles*

Noah Jones
Dr. Baskiyar
COMP-3350
HW #4

4. **What is the value of ax after each of the following instructions?**

```
.data
myArray WORD 3 DUP (5), 2330, 97, 0CDEh, 7 DUP (30)

.code
   mov ax, TYPE myarray          ; AX = (0x00 + 2 (size of WORD = 16 bits or 2 bytes)) = 0x0002
   mov ax, sizeof myarray        ; AX = 2(3) + 2 + 2 + 2 + 2(7) = 26 bytes ... 26₁₀ = 0x001A
   mov ax, lengthof myarray      ; AX = 3 + 1 + 1 + 1 + 7 = 13 ... 13₁₀ = 0x000D
```

5. (**Sign Extension**) **Fill in the requested register values after executions of the instructions:**

```
.code
   mov bx, 0F26Bh
   movzx eax, bx          ; EAX =  0x0000 F26B
   movzx edx, bh          ; EDX =  0x0000 00F2 (high-order byte)
   movzx cx, bl           ; CX =   0x006B (low-order byte)

   mov bx, 0D312h
   movsx eax, bx          ; EAX = 0x0000 D312
   movsx edx, bl          ; EDX = 0x0000 0012 (low-order byte)
   movsx cx, bh           ; CX  =  0x00D3 (high-order byte)
```

6. (**Indirect, Little Endian**) **What will be the value of the destination operand after each of the following instructions execute?**

```
.data
var1 BYTE   7, 6, 0Fh, 3
```

Noah Jones

Dr. Baskiyar

COMP-3350

HW #4

var2 WORD 2122h, 9396h, 0F10Dh, 9527h

var3 SWORD -55, -25

var4 DWORD 21B3h, 40C2h, 4CAFh, 5D79h

.code

mov ax, [**var1**+**1**] ; **AX** = 0x0006h

mov ax, [**var2**+**2**] ; **AX** = 0xF10Dh //does this remain as 0x**0**F10D??

mov ax, var3 ; **AX** = -55 = 11001001_2 = C9 , so 0xFFC9h

mov ax, [**var3-2**] ; **AX** = -55 again, = 0xFFC9h

7. Write a program that prints your <FirstName Lastname> on your screen. You can use the template provided. Assemble and generate the output using MASM and Visual Studio. Embed your output in your submission.

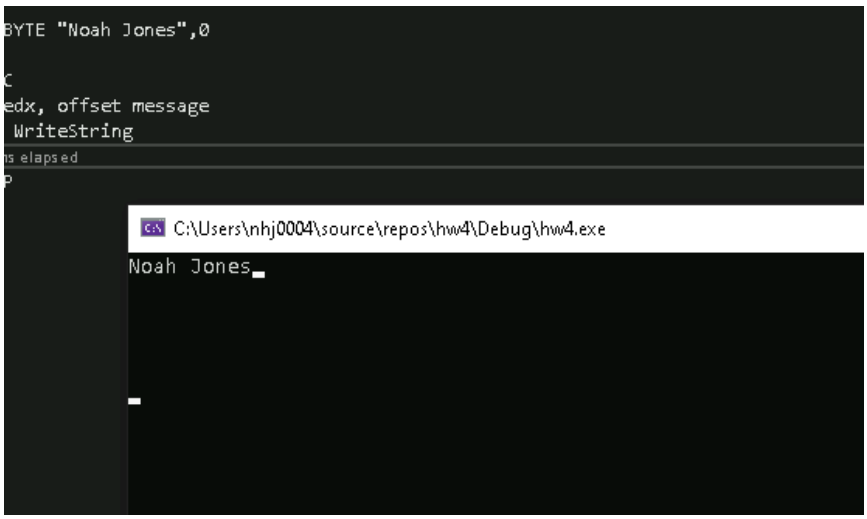
```
TITLE My first assembly program
INCLUDE Irvine32.inc
.DATA
Message BYTE "Noah Jones",0
.CODE
main PROC
    mov edx, offset message
    Call WriteString
exit
main ENDP
END main
```

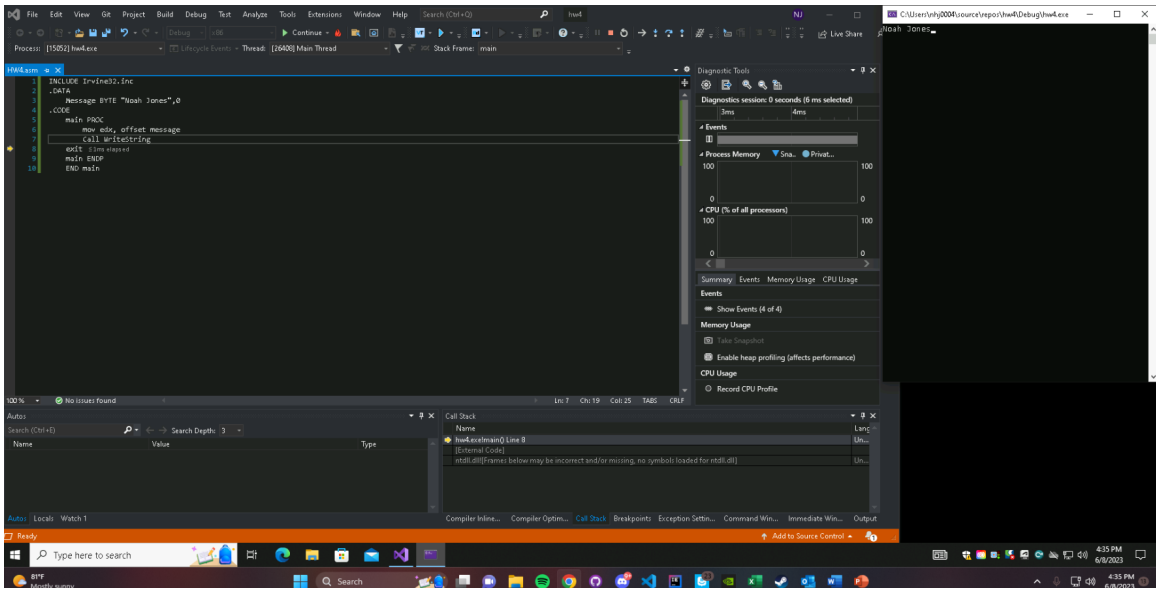
Noah Jones

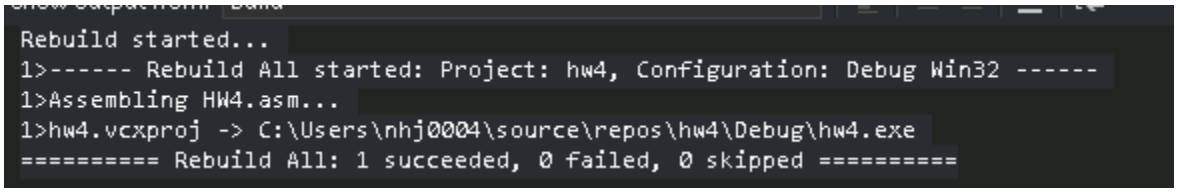
Dr. Baskiyar

COMP-3350

HW #4







//Questions to Ask

Noah Jones

Dr. Baskiyar

COMP-3350

HW #4

1. **Comment on #3, does PTR overflow or reset?**
2. **#6.2 and 6.3**

   ‘Summary of Address modes on Canvas? What is that???

   v