



ASSIGNMENT #1

COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE (EE-2003)

Course Instructors: Mr. Shahbaz Hassan, Miss Farheen Tabassum, Miss Ummarha Qassem

ID: 23i-0061

NAME: Fahid Imrar

SECTION: A1 (B)

Read the Instructions Carefully

- ❖ Assignments are designed for your learning. Copying will lead to plagiarism.
- ❖ Data Declarations highlighted in yellow must be replaced as instructed.

- ✓ Convert your 4-digit ID to two bytes **IDB1** and **IDB2** both should be in HEX. First two digits of your ID will replace **IDB1** and Last two will replace **IDB2** in the assignment.
- ✓ Convert your 4-digit ID to two bytes **IDB3** and **IDB4** both should be in HEX. First and third digits of your ID will replace **IDB3** and second and fourth digit will replace **IDB4** in the assignment.
- ✓ Convert your 4-digit ID to a WORD in HEX and replace **IDW** in assignment with your 4-digit ID in HEX.
- ✓ Concatenate your ID with your ID to create a 8-digit Double WORD in HEX and replace **IDD** in assignment with 8-digit ID.
- ✓ **F 1 F 2 F 3 F 4** in assignment must be replaced with first four character your FIRST Name where FN1 is first character of your first name, FN2 is second character, FN3 is third character and FN4 is fourth character.
- ✓ **S 1 S 2 S 3 S 4** in assignment must be replaced with first four character your Second Name where SN1 is first character of your second name, SN2 is second character, SN3 is third character and SN4 is fourth character.

$$IDB_1 = 00 = 000h$$

$$SN_1 = 'I' = 73 = 049h$$

$$IDB_2 = 61 = 03Dh$$

$$SN_2 = 'M' = 77 = 04Dh$$

$$IDB_3 = 06 = 006h$$

$$SN_3 = 'R' = 82 = 052h$$

$$IDB_4 = 01 = 001h$$

$$SN_4 = 'A' = 65 = 041h$$

$$IDW = 0061 = 03Dh$$

$$IDD = 00610061 = 094F0Dh$$

For Q1

$$FN_1 = 'F' = 70 = 046h$$

$$IDB_1 = 0 \quad IDB_2 = 0$$

$$FN_2 = 'A' = 65 = 041h$$

$$IDB_3 = 6 \quad IDB_4 = 1$$

$$FN_3 = 'H' = 72 = 048h$$

$$FN_4 = 'I' = 73 = 049h$$

Assignment 1
Computer Organization and Assembly Language

Question # 01 Perform each of the following operations on these numbers and update the answer and value of flags after performing the arithmetic. Perform all the steps in the "calculation" box, only filling the answer will not get any credit.

1. $\text{AFFDh} + \langle \text{IDB2}+6, \text{IDB3}+4, \text{IDB4}+6, \text{IDB1} \rangle_d$ $(0+6)16^4 + 1+6, 0>_d = 6070d$

Answer		Calculation
Flags	Sign	1
	Zero	0
	Carry	0
	Overflow	0
	Aux	
	Carry	1
	Parity	0

$6070d = 0110110010101000_2$

$$\begin{array}{r}
 \text{AFFD} \\
 + 17B6 \\
 \hline
 C7B3
 \end{array}$$

$LSB = B3$
 $= 10110011$
 $Parity = 0$

2. $876Fh + \langle \text{IDB2}, \text{IDB1}, \text{IDB4}, \text{IDB3}+2 \rangle_h$ $(0, 0, 1, 6+2)_h = 0018d$

Answer		Calculation
Flags	Sign	1
	Zero	0
	Carry	0
	Overflow	0
	Aux	
	Carry	1
	Parity	1

$0018d = 00012h$

$$\begin{array}{r}
 876F \\
 + 0012 \\
 \hline
 8781
 \end{array}$$

$LSB = 81$
 $= 10000001$
 $Parity = 1$

3. $\langle \text{IDB2}, \text{IDB3}, \text{IDB4}, \text{IDB1}+3 \rangle_d - 7E0Eh$

Answer		Calculation
Flags	Sign	1
	Zero	0
	Carry	1
	Overflow	0
	Aux Carry	1
	Parity	0

$0613d = 00265h$

$$\begin{array}{r}
 0265 \\
 - 7E0E \\
 \hline
 8457
 \end{array}$$

$MSB = 57$
 $= 01010111$
 $Parity = 0$

↓ consider it Positive

Assignment 1
Computer Organization and Assembly Language

$$4 \quad -1 \quad 1 \quad 6$$

4. $4785d + <IDB2+4, IDB1-1, IDB2+1, IDB3>h$

Answer			
Flags	Sign	0	
	Zero	0	
	Carry	0	
	Overflow	0	
	Aux Carry	0	
	Parity	1	

3 2 0 5

Calculation

$$\begin{array}{r} 416d = 1014h \\ 4785d = 12B1h \\ \hline \end{array}$$

$$\begin{array}{r} 12B1 \\ + 1014 \\ \hline 22C5 \end{array}$$

$$\begin{array}{l} LSB = C5 \\ = 1100\ 0101 \end{array}$$

Parity = 1

5. $67EBh + <IDB1+3, IDB2+2, IDB2, IDB1+5>h$

Answer			
Flags	Sign	0	
	Zero	0	
	Carry	0	
	Overflow	0	
	Aux Carry	1	
	Parity	0	

5 0 6 2

Calculation

$$\begin{array}{r} 3205d = 0C85h \\ \hline \end{array}$$

$$\begin{array}{r} 67EB \\ + 0C85 \\ \hline 7470 \end{array}$$

$$\begin{array}{l} LSB = 70 \\ = 01110000 \end{array}$$

Parity = 0

6. $EFB2h - <IDB1+5, IDB1, IDB4+5, IDB3+6>h$

Answer			
Flags	Sign	1	
	Zero	0	
	Carry	0	
	Overflow	0	
	Aux Carry	1	
	Parity	0	

Calculation

$$\begin{array}{r} 5062d = 13C6h \\ \hline \end{array}$$

$$\begin{array}{r} EFB2 \\ - 13C6 \\ \hline DBEC \end{array}$$

$$\begin{array}{l} LSB = EC \\ = 11101100 \end{array}$$

Parity = 0

0 0 9 6
 7. $8100d + <IDB2, IDB1, IDB3+3, IDB3>d \quad 8100d = 1FA4h$

Answer			
Flags	Sign	0	
	Zero	0	
	Carry	0	
	Overflow	0	
	Aux Carry	0	
	Parity	1	

Calculation

$$\begin{array}{r} 0096d = 13C6h \\ \hline \end{array}$$

$$\begin{array}{r} 1FA4 \\ + 13C6 \\ \hline 336A \end{array}$$

$$\begin{array}{l} LSB = 6A \\ = 01101010 \end{array}$$

Parity = 1

Assignment 1
Computer Organization and Assembly Language

8. FEEEh + <IDB1 +2, IDB4, IDB1, IDB3>h

Answer	
Flags	Sign 0
	Zero 0
	Carry 1
	Overflow 0
	Aux Carry 1
	Parity 1

Calculation

$$\begin{array}{r} 2106d = 083Ah \\ \hline \end{array}$$
$$\begin{array}{l} LSB = 28 \\ = 00101000 \\ = Parity = 1 \end{array}$$

$$\begin{array}{r} 111 \\ F E E E \\ + 083A \\ \hline 0728 \end{array}$$

Assignment 1
Computer Organization and Assembly Language

Question#2: Calculate Memory Address Space

Given a main memory of 1DB2 MB, compute the following:

- Convert your 4 digit ID into IDB1, IDB2, IDB3, IDB4 and IDW as per the rules above
- Calculate the total number of addressable memory locations in bytes
- Determine the minimum number of address lines required to address the full memory space
- If the starting address of memory is IDW X 100H, calculate the starting and last address of main memory

a): $ID = 0061$

$$\begin{aligned} IDB1 &= 00 = 000h & IDB2 &= 61 = 03Dh \\ IDB3 &= 06 = 006h & IDB4 &= 001h & IDW &= 0061 = 03Dh \end{aligned}$$

b): Total Number of Addresses

$$\begin{aligned} 61 MB &= 61 \times 10^2 \text{ Byte} \\ \text{Total No. of Addresses} &= 63963136 \end{aligned}$$

c): Number of Address Lines

$$\text{Size of Address bus} = \log_2(61 \times 2^{20}) \approx 26 \text{ bit}$$

$$\text{Number of Address lines} = 26$$

d): Starting and Ending Address

$$= IDW \times 100h = 03Dh \times 100h = 03D00h$$

5 | Page Start Address = 0003D00h
End Address = 3 FFFFFFh

This digit have 2 bits ←
which can have hex value 3. all hexa digit have 4 bits
6x4=24

Assignment 1
Computer Organization and Assembly Language

Question #4

Question #3:

A 16bit 8086 processor uses segment offset addressing to access memory. Each student must use their 4 digit ID to derive key value for segment registers and memory addressing.

Identifier	Rule Applied	Hex Value for ID
IDB1	First two digits	= 000h
IDB2	Last two digits	= 03Dh
IDB3	First and third digits	= 006h
IDB4	Second and fourth digits	= 001h
IDW	Full 4 Digits ID	= 03D0h

You are given the following Segment Registers derived from your Student ID

Segment	Base Address
Code Segment	IDB1 IDB2 H 0003Dh
Data Segment	IDB3 IDB4 H 00601h
Stack Segment	IDW H 0003Dh
Extra Segment	IDB2 IDB1 H 03D00h

1. You are required to compute the physical address of each segment
2. If the stack Segment is set to IDW H , find the offset required to access Physical Address= 49250H
3. If the Extra Segment ES = IDB2 IDB1 H needs to access Physical Address = 503F0H, determine the offset.
4. A program instruction is located at Physical address 24510H. If the code Segment (CS) = IDB1 IDB2 H, Compute the Instruction Pointer (IP)

1): Physical Address of CS = 003Dh x 10h = 0003D0h
 Physical Address of DS = 0601h x 10h = 006010h
 Physical Address of SS = 003Dh x 10h = 0003D0h
 Physical Address of ES = 3D00h x 10h = 03D000h

2): Find Offset:

$$\text{offset} = \text{Physical Address} - \text{Stack Segment Address} \times 10h$$

$$= 49250h - 003Dh \times 10h$$

$$= 49250h - 003D0h$$

$$\text{offset} = 48E80h$$

$$\begin{array}{r}
 49250 \\
 - 003D0 \\
 \hline
 48E80
 \end{array}$$

3): Determine the offset:

$$\begin{aligned}\text{Offset} &= \text{Physical Address} - \text{Extra Segment Address} \times 10h \\ &= 503F0h - 3D00h \times 10h \\ &= 503F0h - 3D000h\end{aligned}$$

$$\begin{array}{r} 503F0 \\ - 3D000 \\ \hline 133F0 \end{array}$$

$$\text{Offset} = 0\ 133F0\ h$$

4): Compute Instructor Pointer

$$\begin{aligned}\text{Offset} &= \text{Physical Address} - \text{Code Segment Address} \times 10h \\ &= 24510h - 003D0h \times 10h \\ &= 24510h - 003D0h\end{aligned}$$

$$\begin{array}{r} 24510h \\ - 003D0h \\ \hline 24140 \end{array}$$

IP (Instructor Pointer contains address)
= 024140h

Assignment 1
Computer Organization and Assembly Language

Question#4: Calculate Memory Address Space

Given a main memory of 1DB3 GB, compute the following:

- Calculate the Start and Last address of the given memory.
- Calculate the Start and Last Address of Code Segment
- Calculate the Start and Last Address of Data Segment
- Calculate the Start and Last Address of Stack Segment
- Calculate the Start and Last Address of Extra Segment

$$06 \text{ GB} = 6 \times 2^{30} \text{ Byte}$$

$$\text{Address Bus Size} = \log_2 (6 \times 2^{30}) = 32.58 \text{ bits} \approx 33 \text{ bits}$$

1 bit + 8x4 bits

1 bit 4 bit all hexa digits have
 ↑ ↑

Start Address of Memory = 0 0 0 0 0 0 0 0 0 0
End Address of Memory = 1 F F F F F F F F F F

Start Address of CS = 0 0 0 0 0 0 0 0 0 0
End Address of CS = 0 7 F F F F F F F F F F

Start Address of DS = 0 8 0 0 0 0 0 0 0 0
End Address of DS = 0 F F F F F F F F F F

Start Address of SS = 1 0 0 0 0 0 0 0 0 0
End Address of SS = 1 7 F F F F F F F F F F

7 | Page

Start Address of ES = 1 8 0 0 0 0 0 0 0 0
End Address of ES = 1 F F F F F F F F F F

Assignment 1
Computer Organization and Assembly Language

DATA

Question#5: Memory Management

- Consider the following Data declaration? Fill given Memory and register value? Look for errors if any.

DATA

```
.data
    list1 BYTE 4, idb1 000h
    list2 BYTE 3 DUP(2, IDb2)
    list3 DB 2 DUP(0,1, 'FN1')
    list4 BYTE 2 DUP('FIRST NAME') 'FAHID'
    list5 BYTE 3 DUP("AB")
```

CODE

```
.code
main Proc
    mov edi, Offset list1
    mov al, List2+2
    mov bl, List3+4
    mov cl, List4+3
    mov dl, List5+3
```

```
main ENDP
END main
```

REGISTER

EDI	40000
AL	002h
BL	001h
CL	049h
DL	042h
AH	
BH	
CH	

$$\begin{aligned}
 &= 2d \\
 &= 1d \\
 &= 73d \\
 &= 66d
 \end{aligned}$$

MEMORY

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4000	04	00	02	3D	02	3D	02	3D	00	01	46	00	01	46	46	41
4010	48	49	44	46	41	48	49	44	41	42	41	42	41	42		
4020																
4030																
4040																

- Consider the following Data declaration? Fill given Memory and register value? Look for errors if any.

Assignment 1
Computer Organization and Assembly Language

DATA

```
.data
    word1 DW 1
    word2 DW -1
    word3 SWORD -1
    word4 SWORD +1
    word5 DW 'AB'
    word6 DW 'ABC' error: size mismatch ('ABC' is 3 byte and DW is 2 byte)
;
    List1 WORD 01F2BH,01101011001011b, 45
    list2 DW 0ABh, 01ACBh, 002Dh
    list3 WORD 3 DUP(0)
    list4 WORD 2 DUP(0AB12H)
    list5 DW 2 DUP(0DWB) 03Dh
    list6 DW 3 DUP("AB")
```

$A = 65 = 41h$
 $B = 66 = 42h$

CODE

```
.code
main Proc
    mov edi, Offset list1
    mov al, word1 error: size mismatch error
;
    mov bx, List1+4
    mov cx, List2+1
    mov dx, List4+2
    mov ax, List6+1
    mov bx, word6+2 error: word6 is not declared
;
    mov cx, word5+2
    mov cx, word5+4
```

Main ENDP

END main

I have comment out 3 lines of code due to an error.

REGISTER

EDI	4000A	= 45d
AL		= 512d
BX	00 2Dh	= 43794d
CX	02 00h	= 16961d
DX	AB 12	= 7979d
AX	42 41	= 6859d
CX	1F 2Bh	
CX	1A CBh	
DX		

MEMORY

	word1	word2	word3	word4	word5	list1							
4000	01 00	FF FF	FF FF	01 00	42 41	2B 1F	CB 1A	2D 00					
4010	3D 00	02 00	00 00	00 00	00 00	00 00	12 AB	00					
4020	3D 00	42 41	42 41	42 41	42 41								
4030													
4040													

C
Declara

Assignment 1
Computer Organization and Assembly Language

3. Consider the following Data declaration? Fill given Memory and register value? Look for errors if any.

DATA

.data

```

dword1 DD 1           FFFF FFFF
dword2 DD -1          -1
dword3 DWORD -1       FFFF FFFE
dword4 DWORD +1       +1
dword5 DD 'AB'         FFFFF FFF
dword6 DD 'ABCD'
List1 DWORD 012345678H,0011010011101011001011b,65
list4 DWORD 2 DUP(0AB123456H) 041h

```

CODE

.code

main Proc

```

mov edi,Offset dword1
mov ebx, dword2
mov ecx, dword5
mov edx, dword6
mov eax, List1+4
mov ebx, list4+2
mov ecx, List1+8
; mov ecx, word5+4 ← word5 is not declare
main ENDP
END main

```

REGISTER

EDI	40000
EAL	
EBX	FFFFFFFFFF
ECX	00004142h
EDX	41424344h
EAX	000D3ACBh
EBX	3456AB12h
ECX	00000041h
EDX	

$\frac{d}{d} = 4294967296$
 $= 16706d$
 $= 1094861636$
 $= 867019d$
 $= 878095122$
 $= 65d$

MEMORY

	0	1	2	3	dword1	dword2	dword3	dword4	C	D	E	F
4000	01	00	00	00	FF	FF	FF	FF	FF	FF	FF	01
4010	42	41	00	00	44	43	42	41	78	36	34	12
4020	41	00	00	00	36	34	12	AB	36	34	12	AB
4030												

Look for

Assignment 1 Computer Organization and Assembly Language

4. Declare Data in following order and store it in memory?

Your Name in Capital Letters: As String
Your Name in Small Letters: As String
Your ID in HEX: using Declaration type as WORD
Your ID in HEX: using Declaration type as DWORD
Your ID in BIN: using Declaration type as WORD
Your ID in BIN: using Declaration type as DWORD
Your ID in DEC: using Declaration type as WORD
Your ID in DEC: using Declaration type as DWORD
Your Age in BIN: using Declaration type as BYTE
Your Age in DEC: using Declaration type as BYTE
Your Age in BIN: using Declaration type as BYTE
Your CNIC in HEX: using Declaration as QWORD (Ignore – in your ID CARD Number)

CNIC = 3420118785403
Hexa = 031C4EF71D7Bh

Age = 19
Hexa = 013h

.data

CName Byte 'FAHID'
SName Byte 'fahid'
var1 Word 03dh
var2 DWORD 03dh
var3 Word 111101b
var4 DWord 111101b
var5 Word 0061
var6 DWORD 0061
var7 BYTE 10011b
var8 BYTE 19
var9 BYTE 10011b
var10 QWORD 031C4EF71D7Bh

	CName	SName	var1	var2
0	46 41 48 49 44	66 61 68 69 64	3D 00	3D 00 00 00
1	3D 00 13D 00 00	3D 00 13D 00 00	13 13 13 13	13 13 13 13
2	1D F7 4E 1C 03	00 00		
3				
4				

Assignment 1

Computer Organization and Assembly Language

5. Consider the following Data declaration? Fill given Memory and register value? Look for errors if any.

DATA	CODE	REGISTER																																		
.data bval1 BYTE 16 = 10h bv2 BYTE -16 = F0h wval WORD 0		$\begin{array}{r} FF \\ -10 \\ \hline EF \\ +1 \\ \hline FO \end{array}$																																		
.code main Proc	<pre>movsx bx,al → Set suffix 0 value in al movsx cx,bv2 movsx eax,bv2 movsx ebx,bval1 movzx ecx,bv2 movzx edx,bval1</pre> <p>main ENDP END main</p>	<table border="1"> <tr> <td>BX</td><td>00 00</td></tr> <tr> <td>CX</td><td>FF F0</td></tr> <tr> <td>EAX</td><td>FFFFFFFO</td></tr> <tr> <td>EBX</td><td>00000010</td></tr> <tr> <td>ECX</td><td>000000F0</td></tr> <tr> <td>EDX</td><td>00000010</td></tr> </table>	BX	00 00	CX	FF F0	EAX	FFFFFFFO	EBX	00000010	ECX	000000F0	EDX	00000010																						
BX	00 00																																			
CX	FF F0																																			
EAX	FFFFFFFO																																			
EBX	00000010																																			
ECX	000000F0																																			
EDX	00000010																																			
MEMORY	<p>bval1 bv2 wval</p> <table border="1"> <tr> <td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr> <tr> <td>4000</td><td>10</td><td>F0</td><td>00 00</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	4000	10	F0	00 00														
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																				
4000	10	F0	00 00																																	

6. Consider the following Data declaration? Fill given Memory and register value after the code execution? Look for errors if any.

DATA	CODE	REGISTER
.data bval1 BYTE 5 bval2 BYTE 7 wval1 WORD 0ABCDH wval2 word 01234h		

Assignment 1
Computer Organization and Assembly Language

```
.code
main Proc
    mov ax,@data
    mov ds,ax
    mov al,bval1
    mov bl,bval2
    xchg al,bl
    mov bval1,bl
    mov bval2,al
    mov cx,wval1
    xchg cx,wval2
    mov wval2,cx
main ENDP
END main
```

*@data is typically used in TASM (Turbo Assembler)
in MASM version it is not valid.*

@data is used to load segment address of data segment into a register.

AL	05h
BL	07h
CL	07h
DL	05h
DL	ABCDh
DL	1234h
ESI	

EDI	
AH	
BH	
CX	
DH	
DL	
ESI	

MEMORY

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
4000	05	07	CD	AB	34	12											
	05	07	CD	AB	CD	AB											

bval1 bval2 wval1 wval2

7. Consider the following Data declaration? Fill given Memory and register value? Look for errors if any.

DATA

```
.data
bary BYTE 0ABh,0450h,0Db2h,25h,0Db1h
wary WORD 0DW,34560h,0DB2h,0DB1h,cdh
Dary dd 0DDh,'ABcd','SN1SN2SN3SN4',0AB12CD34h
Qary dq 12345678abcdehf,'AbCd'
```

CODE

094F0Dh

.code

main Proc

```
    mov al,bary
    mov bl,bary+1
    mov cl,bary+2
    mov dl,bary+3
```

AL	AB h
BL	25 h
CL	3D h
DL	19 h
AX	003Dh
BX	8700h

=171
=37
=61
=25
=61
=34560
=Dary

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0	AB	25	3D	19	00	3D	00	00	87	3D	00	00	00	64	63	0D
1	4F	09	00	64	63	42	41	41	S2	4D	49	34	CD	12	AB	ef
2	cd	ab	78	56	34	12	00									

bary worry

Assignment 1

Computer Organization and Assembly Language

```
mov bx,wary+2  
mov cx,wary+6  
mov dx,wary+12  
  
mov eax,dary+4  
;mov ebx,wary+8 } size miss match error  
;mov ecx,wary+00CF }  
;mov edx,wary } non digit in number
```

CX	00 00h
DX	00 09h
EAX	41426364
EBX	
ECX	
EDX	

$$= 94$$

- 8 Consider the following Data declaration? Fill given Memory and register value? Look for errors if any.

DATA	
.data	
bary BYTE 0ABh,0450, IDb2, 25T, IDb1 wary WORD 34560, 34561, IDb2, IDb1, "cd" Dary dd IDD, 'AbCd', 'SN1\$N2\$N3\$N4', 0AB12CD34h Qary dq 12345678abcdefh, 'AbCd'	023h 030h 01fh 00 34560 = 8700h C = 99 = 63h d = 100 = 64h
CODE	094F00h
.code	
main Proc	
mov esi, OFFSET bary mov edi, OFFSET pary ← Pary is not define mov esi, OFFSET wary mov edi, OFFSET wary+2 mov esi, OFFSET qary	
main ENDP	
END main	
	10F
	005
	007
	000

All are memory
address my

ESI	000
EDI	
ESI	005
EDI	007
ESI	10F

Assignment 1
Computer Organization and Assembly Language

9. Consider the following Data declaration? Fill the registers value? Look for errors if any.

DATA																															
<i>bval1</i>							wval2																								
<i>wval2</i>							dval3																								
<i>dval3</i>																															
<pre>.data bval1 DB 034h, IDb1 wval2 DW IDW03Dh dval3 DD IDD094F00h</pre>																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr> <td>34</td><td>00</td><td>3D</td><td>00</td><td>0D</td><td>4F</td><td>09</td><td>00</td></tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>								0	1	2	3	4	5	6	7	34	00	3D	00	0D	4F	09	00								
0	1	2	3	4	5	6	7																								
34	00	3D	00	0D	4F	09	00																								
CODE																															
<pre>.code main Proc mov esi,OFFSET bval1 mov ax, WORD PTR bval1 mov al, BYTE PTR wval2 mov bx, WORD PTR dval3 mov cx, WORD PTR [dval3+2] main ENDP END main</pre>																															
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>ESI</td><td>00</td></tr> <tr> <td>AX</td><td>60 34h</td></tr> <tr> <td>AL</td><td>3Dh</td></tr> <tr> <td>BX</td><td>4F 0Dh</td></tr> <tr> <td>CX</td><td>00 09h</td></tr> </table>								ESI	00	AX	60 34h	AL	3Dh	BX	4F 0Dh	CX	00 09h														
ESI	00																														
AX	60 34h																														
AL	3Dh																														
BX	4F 0Dh																														
CX	00 09h																														

10. Consider the following Data declaration? Fill given registers value. Look for errors if any.

DATA											
.data	'F'										
bval1 DB 'F 1'	'F'										
wval2 DW 0ABCDH,'F 1'											
dval3 DD IDD094F0Dh											
qval4 QWORD 01243567812435678H											
tval5 tbyte -1											
CODE											
.code											
main Proc											
mov al, TYPE bval1											
mov bl, TYPE wval2											
mov cl, TYPE dval3											
mov dl, TYPE qval4											
mov ah, TYPE tval5											
main ENDP											
END main											
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>AL</td><td>1</td></tr> <tr> <td>BL</td><td>2</td></tr> <tr> <td>CL</td><td>4</td></tr> <tr> <td>DL</td><td>8</td></tr> <tr> <td>AH</td><td>10</td></tr> </table>		AL	1	BL	2	CL	4	DL	8	AH	10
AL	1										
BL	2										
CL	4										
DL	8										
AH	10										
} all values are in decimal											

11. Consider the following Data declaration? Fill given registers value. Look for errors if any.

Assignment 1
Computer Organization and Assembly Language

Computer
main Env-

DATA

```
.data
    bAry BYTE 010H,020h,030H
    wAry WORD 5 DUP(0),0 ,0
    string1 BYTE OFFH,OFFH
    dAry DWORD 2 DUP(3 DUP(0)),0
    string2 BYTE "ABCDEFGHIJKLMNO!QRSTUVWXYZ",0
    bAry2 BYTE 1,2,3,
                4,5,6
                BYte 4,5,6
    r dq 2 dup(5),2 dup('abcd')
```

CODE

```
.code
main Proc
    mov bl, LENGTHOF bAry
    mov cl, LENGTHOF wAry
    mov dl, LENGTHOF dAry
    mov ah, LENGTHOF string2
    mov al, LENGTHOF bAry2
    mov bl, LENGTHOF r
main ENDP
END main
```

BL	3
CL	7
DL	7
AH	28
AL	6
BL	4

{ All values are in decimal }

12. Consider the following Data declaration? Fill given registers value. Look for errors if any.

DATA

```
.data
    bAry BYTE 010H,020h,030H
    wAry WORD 5 DUP(0),0 ,0
    string1 BYTE OFFH,OFFH
    dAry DWORD 2 DUP(3 DUf(0)),0
    string2 BYTE "ABCDEFGHIJKLMNO!QRSTUVWXYZ",0
    r dq 2 dup(5),2 dup('abcd')
```

Typing Error:
There is no DUFL instruction in Assembly.
Replace it with DUP.

→ I'm considering it (fl)

CODE

```
.code
main Proc
    mov bl, SIZEOF bAry   3x1 = 3
    mov cl, SIZEOF wAry   7x2 = 14
    mov dl, SIZEOF dAry   7x4 = 28
    mov ah, SIZEOF string1 2x1 = 2
    mov al, SIZEOF string2 28x1 = 28
    mov bl, SIZEOF r       4x8 = 32
```

BL	3
CL	14
DL	28
AH	2
BH	

{ All values in decimal }

Assignment 1
Computer Organization and Assembly Language

Question no 17: Convert the following C++ code into Assembly language

```
#include <iostream>

using namespace std;

int main() {
    int a = 9, b = 15;
    int max;

    if (a > b)
        max = a;
    else
        max = b;

    cout << "Maximum: " << max << endl;
    return 0;
}
```

Best of Luck

```
include Irvine32.inc
· 386
· model flat, stdcall
· stack 4096
· data

msg BYTE 'Maximum: '
num1 DW 9
num2 DW 15
MAX DW ?

·code

main PROC

    mov ax, num1
    mov bx, num2
```

cmp ax, bx
jle Part-A

mov max, ax
jmp Display

Part-A:

mov max, bx

Display:

mov edx, offset msg
call writestring
mov dx, offset max
call writeint

main endp

end main