



National University
Of Computer and Emerging Sciences

LAB 07:

Name – **Muhammad Taha**

Roll NO – **23p-0559**

SECTION – **BCS(3A)**

Subject - **Computer Organization and Assembly Language**
(LAB)

3.6 (a)

DOS Screenshot:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>

Z:\>mount x D://DOS
Drive X is mounted as local directory D://DOS\

Z:\>x:/

X:\>nasm
nasm: error: no input file specified
type 'nasm -h' for help

X:\>nasm 559_6(a).asm -o 559_6(a).com

X:\>559_6(a).com

X:\>afd 559_6(a).com

AFD-Pro is done

X:\>
```

UnSorted:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX	SI	CS	IP	Stack	Flags
00FF	0000	19F5	0100	+0 0000	7202
0000	0000	19F5		+2 20CD	
003C	0000	19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
0000	SP FFFE	19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 0 0

CMD	>
[1] 0 1 2 3 4 5 6 7	
DS:0100 E9 08 00 06 00 04 00 05	
DS:0108 00 02 00 B9 04 00 BB 00	
DS:0110 00 8B 87 03 01 3B 87 05	
DS:0118 01 76 0C 8B 97 05 01 89	
DS:0120 87 05 01 89 97 03 01 81	
DS:0128 C3 02 00 81 FB 06 00 75	
DS:0130 E0 81 E9 01 00 75 D7 B8	
DS:0138 00 4C CD 21 C7 46 DC 00	
DS:0140 00 8E 5E FC 83 7D 0E 00	
DS:0148 74 09 8B 46 F2 48 3B 46	

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0100	E9	08	00	06	00	04	00	05	00	02	00	B9	04	00	BB	00
DS:0110	00	8B	87	03	01	3B	87	05	01	76	0C	8B	97	05	01	89
DS:0120	87	05	01	89	97	03	01	81	C3	02	00	81	FB	06	00	75
DS:0130	E0	81	E9	01	00	75	D7	B8	00	4C	CD	21	C7	46	DC	00
DS:0140	00	8E	5E	FC	83	7D	0E	00	74	09	8B	46	F2	48	3B	46

Step	2ProcStep	3Retrieve	4help ON	5BRK Menu	6	7 up	8 dn	9 le	10 ri
------	-----------	-----------	----------	-----------	---	------	------	------	-------

Sorted:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra... — □ ×																			
AX 0000	SI 0000	CS 19F5	IP 010B	Stack +0	0000	Flags 7200													
BX 0000	DI 0000	DS 19F5			+2	20CD													
CX 0000	BP 0000	ES 19F5	HS 19F5			+4	9FFF	OF	DF	IF	SF	ZF	AF	PF	CF				
DX 0000	SP FFFE	SS 19F5	FS 19F5			+6	EA00	0	0	1	0	0	0	0	0				
CMD >								1	0	1	2	3	4	5	6	7			
00FE 0000	ADD		[BX+SI],AL		DS:0100		E9	08	00	02	00	04	00	05					
0100 E90800	JMP		010B		DS:0108		00	06	00	B9	04	00	BB	00					
0103 0200	ADD		AL,[BX+SI]		DS:0110		00	8B	87	03	01	3B	87	05					
0105 0400	ADD		AL,00		DS:0118		01	76	0C	8B	97	05	01	89					
0107 050006	ADD		AX,0600		DS:0120		87	05	01	89	97	03	01	81					
010A 00B90400	ADD		[0004+BX+DI],BH		DS:0128		C3	02	00	81	FB	06	00	75					
010E BB0000	MOV		BX,0000		DS:0130		E0	81	E9	01	00	75	D7	B8					
0111 8B870301	MOV		AX,[0103+BX]		DS:0138		00	4C	CD	21	C7	46	DC	00					
0115 3B870501	CMP		AX,[0105+BX]		DS:0140		00	8E	5E	FC	83	7D	0E	00					
								DS:0148								74 09 8B 46 F2 48 3B 46			
2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
DS:0100	E9	08	00	02	00	04	00	05	00	06	00	B9	04	00	BB	00			
DS:0110	00	8B	87	03	01	3B	87	05	01	76	0C	8B	97	05	01	89			
DS:0120	87	05	01	89	97	03	01	81	C3	02	00	81	FB	06	00	75			
DS:0130	E0	81	E9	01	00	75	D7	B8	00	4C	CD	21	C7	46	DC	00			
DS:0140	00	8E	5E	FC	83	7D	0E	00	74	09	8B	46	F2	48	3B	46			
																0.....i.s...s. .v.iù..ë s..èù..ü ..üf..u αiø..u .L=t F■. .Ä^"â}.. t.iFzH;F			
1	Step	2	ProcStep	3	Retrieve	4	Help ON	5	BRK Menu	6		7	up	8	dn	9	le	10	ri

3.6 (b):

DOS Screenshot:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
X:\>
X:\>
X:\>afd 559_6(a).com
AFD-Pro is done
X:\>afd 559_6(a).com
AFD-Pro is done
X:\>afd 559_6(a).com
AFD-Pro is done
X:\>nasm 559_6(b).asm -o 559_6(b).com
X:\>559_6(b).com
X:\>afd 559_6(b).com
AFD-Pro is done
X:\>
```

UnSorted:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 00FF SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
 BX 0000 DI 0000 DS 19F5 +2 20CD
 CX 0045 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
 DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

CMD >

Address	Disassembly	Comment
0100 E90900	JMP	010C
0103 0200	ADD	AL, [BX+SI]
0105 0400	ADD	AL, 00
0107 06	PUSH	ES
0108 0005	ADD	[DI], AL
010A 0000	ADD	[BX+SI], AL
010C BB0000	MOV	BX, 0000
010F C6060B0100	MOV	[010B], 00

DS:0100 E9 09 00 02 00 04 00 06
 DS:0108 00 05 00 00 BB 00 00 C6
 DS:0110 06 0B 01 00 8B 87 03 01
 DS:0118 3B 87 05 01 76 11 8B 97
 DS:0120 05 01 89 87 05 01 89 97
 DS:0128 03 01 C6 06 0B 01 01 81
 DS:0130 C3 02 00 81 FB 06 00 75
 DS:0138 DB 80 3E 0B 01 01 74 CC
 DS:0140 B8 00 4C CD 21 7D 0E 00
 DS:0148 74 09 8B 46 F2 48 3B 46

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
 DS:0100 E9 09 00 02 00 04 00 06 00 05 00 00 BB 00 00 C6
 DS:0110 06 0B 01 00 8B 87 03 01 3B 87 05 01 76 11 8B 97
 DS:0120 05 01 89 87 05 01 89 97 03 01 C6 06 0B 01 01 81
 DS:0130 C3 02 00 81 FB 06 00 75 DB 80 3E 0B 01 01 74 CC
 DS:0140 B8 00 4C CD 21 7D 0E 00 74 09 8B 46 F2 48 3B 46

1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri

When SWAP OCCUR

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0006 SI 0000 CS 19F5 IP 012F Stack +0 0000 Flags 7200
 BX 0004 DI 0000 DS 19F5 +2 20CD
 CX 0045 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
 DX 0005 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

CMD >

Address	Disassembly	Comment
012A C6060B0101	MOV	[010B], 01
012F 81C30200	ADD	BX, 0002
0133 81FB0600	CMP	BX, 0006
0137 75DB	JNZ	0114
0139 803E0B0101	CMP	[010B], 01
013E 74CC	JZ	010C
0140 B8004C	MOV	AX, 4C00
0143 CD21	INT	21
0145 7D0E	JNL	0155

DS:0100 E9 09 00 02 00 04 00 05
 DS:0108 00 06 00 01 BB 00 00 C6
 DS:0110 06 0B 01 00 8B 87 03 01
 DS:0118 3B 87 05 01 76 11 8B 97
 DS:0120 05 01 89 87 05 01 89 97
 DS:0128 03 01 C6 06 0B 01 01 81
 DS:0130 C3 02 00 81 FB 06 00 75
 DS:0138 DB 80 3E 0B 01 01 74 CC
 DS:0140 B8 00 4C CD 21 7D 0E 00
 DS:0148 74 09 8B 46 F2 48 3B 46

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
 DS:0100 E9 09 00 02 00 04 00 05 00 06 00 01 BB 00 00 C6
 DS:0110 06 0B 01 00 8B 87 03 01 3B 87 05 01 76 11 8B 97
 DS:0120 05 01 89 87 05 01 89 97 03 01 C6 06 0B 01 01 81
 DS:0130 C3 02 00 81 FB 06 00 75 DB 80 3E 0B 01 01 74 CC
 DS:0140 B8 00 4C CD 21 7D 0E 00 74 09 8B 46 F2 48 3B 46

1 Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri

Sorted:

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 4C00 SI 0000 CS F000 IP 14A0 Stack +0 42BD Flags 7095
BX 0006 DI 0000 DS 19F5 +2 06C5
CX 0045 BP 0000 ES 19F5 HS 19F5 +4 7095 OF DF IF SF ZF AF PF CF
DX 0005 SP FFF2 SS 19F5 FS 19F5 +6 0145 0 0 0 1 0 1 1 1

CMD >
0143 CD21 INT 21
14A0 FB STI
14A1 FE DB FE
14A2 3B25 CMP [DI], AH
14A4 00CF ADD BH, CL
14A6 CB RET Far
14A7 51 PUSH CX
14A8 B94001 MOV CX, 0140
14AB E2FE LOOP 14AB

1 0 1 2 3 4 5 6 7
DS:0100 E9 09 00 02 00 04 00 05
DS:0108 00 06 00 00 BB 00 00 C6
DS:0110 06 0B 01 00 8B 87 03 01
DS:0118 3B 87 05 01 76 11 8B 97
DS:0120 05 01 89 87 05 01 89 97
DS:0128 03 01 C6 06 0B 01 01 81
DS:0130 C3 02 00 81 FB 06 00 75
DS:0138 DB 80 3E 0B 01 01 74 CC
DS:0140 BB 00 4C CD 21 7D 0E 00
DS:0148 74 09 8B 46 F2 48 3B 46

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0100 E9 09 00 02 00 04 00 05 00 06 00 00 BB 00 00 C6 0...
DS:0110 06 0B 01 00 8B 87 03 01 3B 87 05 01 76 11 8B 97 ...
DS:0120 05 01 89 87 05 01 89 97 03 01 C6 06 0B 01 01 81 ..
DS:0130 C3 02 00 81 FB 06 00 75 DB 80 3E 0B 01 01 74 CC |.
DS:0140 BB 00 4C CD 21 7D 0E 00 74 09 8B 46 F2 48 3B 46 |.

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri
```

DIFFERENCE:

The first code sorts the array data: dw 6, 4, 5, 2, while the second one sorts data: dw 2, 4, 6, 5. In the first code, the swap flag is not used to track if any swaps happen, whereas in the second code, the swap flag is set to 1 when a swap occurs, helping to determine if another pass is needed. The first code runs a fixed number of passes (4), while the second code checks the swap flag to decide if it should run the outer loop again; if no swaps were made, it stops early. Both codes have comments, but the second code includes more explanations about why certain instructions are used, like the importance of specifying byte and using registers for comparisons. Lastly, the first code includes a commented-out line `mov cx, 4`, which is not present in the second code, as it doesn't need a fixed number of passes due to the swap flag. In summary, the second code is more efficient because it uses the swap flag to avoid unnecessary passes, making it more adaptable to different data arrangements.