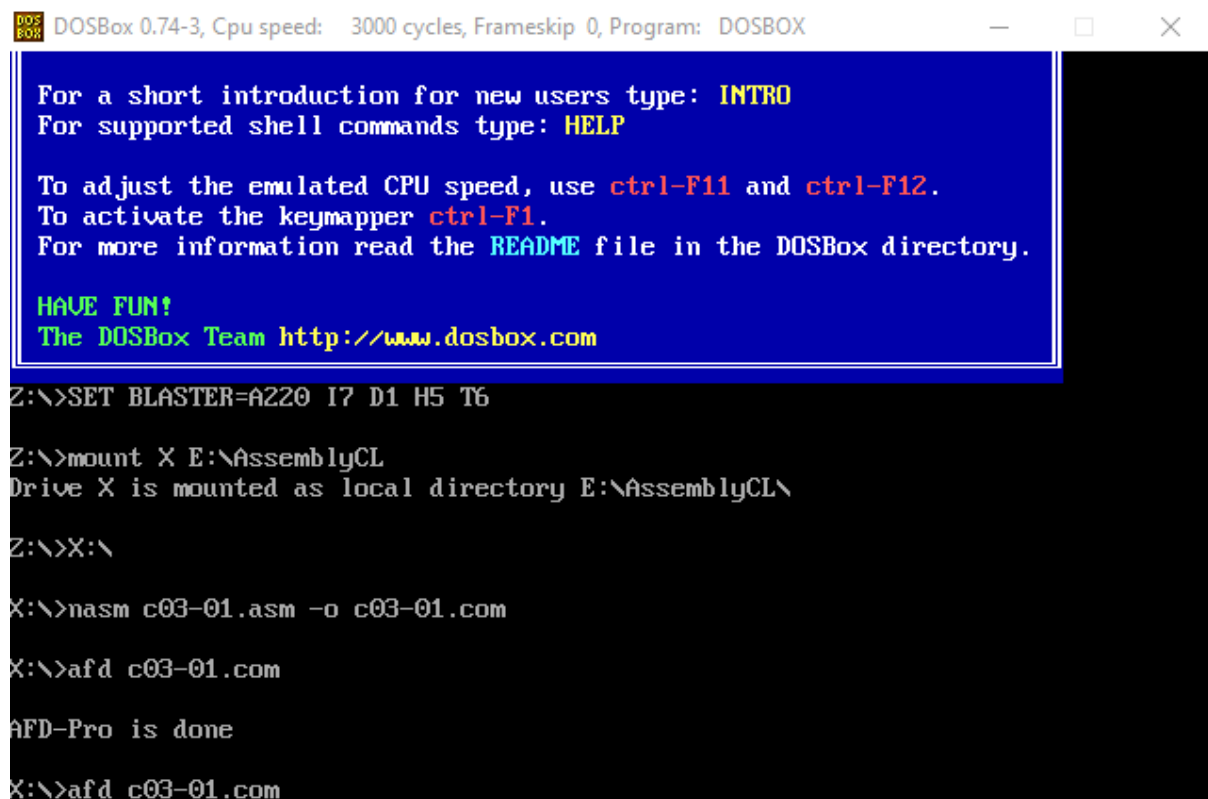


## **Lab Task 1: -**

The difference is only that in 3.3 (a) if the values are already in descending order it will still run till the cx has a value of 0 while 3.3 (b) has a different logic in that first of all we are not using the cx and with that we have made a flag other than the inbuilt flags that helps us to refrain from unnecessary passes we initialize the value in the swap flag as 0 and whenever a swap is done its value is made 1 and it will continue in that pass but when there is no swap in the whole inner loop the value will remain 0 in the swap flag, hence at the end while checking if a swap has happened it will continue to another pass but if its value is 0 it will exit as in that state the numbers will be in ascending order.

### **(A)**

(Commands)



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

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

Z:\>mount X E:\AssemblyCL
Drive X is mounted as local directory E:\AssemblyCL\

Z:\>X:\

X:\>nasm c03-01.asm -o c03-01.com

X:\>afd c03-01.com

AFD-Pro is done

X:\>afd c03-01.com_
```

(Initial Data)

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 003C 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 >

0100 F90800 JMP 010B
0103 07 POP ES
0104 0005 ADD [DI],AL
0106 00060003 ADD [0300],AL
010A 00B90400 ADD [0004+BX+DI],BH
010E BB0000 MOV BX,0000
0111 8B870301 MOV AX,[0103+BX]
0115 3B870501 CMP AX,[0105+BX]

1 3 4 5 6 7 8 9 A
DS:0103 07 00 05 00 06 00 03 00
DS:010B B9 04 00 BB 00 00 8B 87
DS:0113 03 01 3B 87 05 01 76 0C
DS:011B 8B 97 05 01 89 87 05 01
DS:0123 89 97 03 01 81 C3 02 00
DS:012B 81 FB 06 00 75 E0 81 E9
DS:0133 01 00 75 D7 B8 00 4C CD
DS:013B 21 C7 46 DC 00 00 8E 5E
DS:0143 FC 83 7D 0E 00 74 09 8B
DS:014B 46 F2 48 3B 46 F6 7E 0B

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.Ω≡ i|..†...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ff. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 δ.L.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 ó.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

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

```

(Pass 1)

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0007 SI 0000 CS 19F5 IP 0127 Stack +0 0000 Flags 7200
BX 0004 DI 0000 DS 19F5 +2 20CD
CX 0004 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0003 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

CMD >

0123 89970301 MOV [0103+BX],DX
0127 81C30200 ADD BX,0002
012B 81FB0600 CMP BX,0006
012F 75E0 JNZ 0111
0131 81E90100 SUB CX,0001
0135 75D7 JNZ 010E
0137 BB004C MOV AX,4C00
013A CD21 INT 21
013C C746DC0000 MOV [BP-24],0000

1 3 4 5 6 7 8 9 A
DS:0103 05 00 06 00 03 00 07 00
DS:010B B9 04 00 BB 00 00 8B 87
DS:0113 03 01 3B 87 05 01 76 0C
DS:011B 8B 97 05 01 89 87 05 01
DS:0123 89 97 03 01 81 C3 02 00
DS:012B 81 FB 06 00 75 E0 81 E9
DS:0133 01 00 75 D7 B8 00 4C CD
DS:013B 21 C7 46 DC 00 00 8E 5E
DS:0143 FC 83 7D 0E 00 74 09 8B
DS:014B 46 F2 48 3B 46 F6 7E 0B

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.Ω≡ i|..†...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ff. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 δ.L.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 ó.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

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

```

(Pass 2 Till 4rth Pass As The Data Is Arranged But Due To The CX It Will Run Exactly 4 Times)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

Register	Value	Register	Value	Register	Value	Register	Value	Stack	Flags
AX	0005	SI	0000	CS	19F5	IP	012B	+0	0000
BX	0002	DI	0000	DS	19F5			+2	20CD
CX	0002	BP	0000	ES	19F5	HS	19F5	+4	9FFF
DX	0003	SP	FFFE	SS	19F5	FS	19F5	+6	EA00

CMD >

Address	Instruction	Comment
0127	81C30200	ADD BX,0002
012B	81FB0600	CMP BX,0006
012F	75E0	JNZ 0111
0131	81E90100	SUB CX,0001
0135	75D7	JNZ 010E
0137	B8004C	MOV AX,4C00
013A	CD21	INT 21
013C	C746DC0000	MOV [BP-24],0000
0141	8E5EFC	MOV DS,[BP-04]

Address	Value	Address	Value
DS:0103	03 00 05 00 06 00 07 00	DS:010B	B9 04 00 BB 00 00 8B 87
DS:0113	03 01 3B 87 05 01 76 0C	DS:011B	8B 97 05 01 89 87 05 01
DS:0123	89 97 03 01 81 C3 02 00	DS:012B	81 FB 06 00 75 E0 81 E9
DS:0133	01 00 75 D7 B8 00 4C CD	DS:013B	21 C7 46 DC 00 00 8E 5E
DS:0143	FC 83 7D 0E 00 74 09 8B	DS:014B	46 F2 4B 3B 46 F6 7E 0B

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

(B)

(Commands)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```

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

Z:\>mount X E:\AssemblyCL
Drive X is mounted as local directory E:\AssemblyCL\

Z:\>X:\

X:\>nasm c03-01.asm -o c03-01.com

X:\>afd c03-01.com

AFD-Pro is done

X:\>afd c03-01.com

AFD-Pro is done

X:\>nasm c03-01b.asm -o c03-01b.com

X:\>afd c03-01b.com

AFD-Pro is done

X:\>afd c03-01b.com_
  
```

(Initial Data)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0000 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

S or SI or SYM

CMD >S

Address	Disassembly	Hex
0100	E90900 JMP	010C
0103	0300 ADD	AX,[BX+SI]
0105	050007 ADD	AX,0700
0108	00060000 ADD	[0000],AL
010C	BB0000 MOV	BX,0000
010F	C6060B0100 MOV	[010B],00
0114	8B870301 MOV	AX,[0103+BX]
0118	3B870501 CMP	AX,[0105+BX]

Address	Hex	ASCII
DS:0103	03 00 05 00 07 00 06 00	
DS:010B	00 BB 00 00 C6 06 0B 01	
DS:0113	00 8B 87 03 01 3B 87 05	
DS:011B	01 76 11 8B 97 05 01 89	
DS:0123	87 05 01 89 97 03 01 C6	
DS:012B	06 0B 01 01 81 C3 02 00	
DS:0133	81 FB 06 00 75 DB 80 3E	
DS:013B	0B 01 01 74 CC B8 00 4C	
DS:0143	CD 21 7D 0E 00 74 09 8B	
DS:014B	46 F2 48 3B 46 F6 7E 0B	

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

(Pass 1 In A Single Pass The Digits Are Arranged And Enters Into Pass 2 Where the Swap Flag Remains Zero Due To Which The Program Ends)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0007 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 0006 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

{R} reg=value

CMD >SS

Address	Disassembly	Hex
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

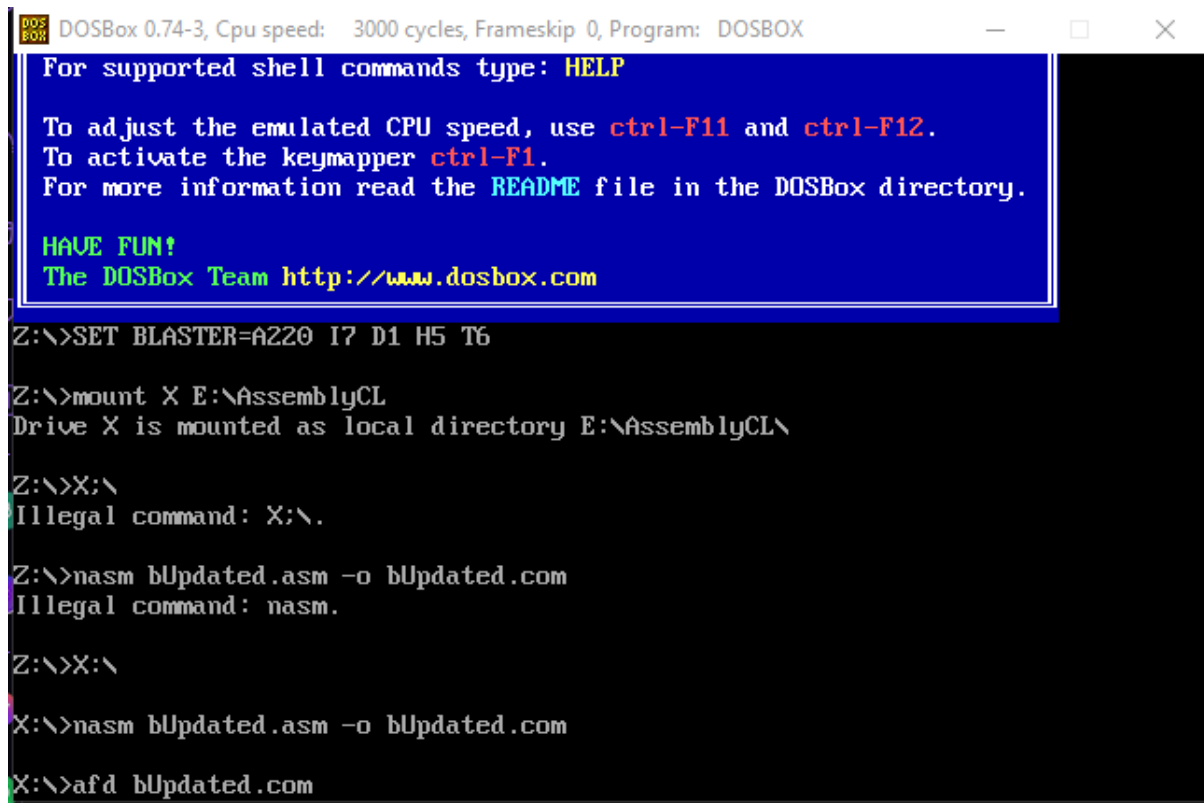
Address	Hex	ASCII
DS:0103	03 00 05 00 06 00 07 00	
DS:010B	01 BB 00 00 C6 06 0B 01	
DS:0113	00 8B 87 03 01 3B 87 05	
DS:011B	01 76 11 8B 97 05 01 89	
DS:0123	87 05 01 89 97 03 01 C6	
DS:012B	06 0B 01 01 81 C3 02 00	
DS:0133	81 FB 06 00 75 DB 80 3E	
DS:013B	0B 01 01 74 CC B8 00 4C	
DS:0143	CD 21 7D 0E 00 74 09 8B	
DS:014B	46 F2 48 3B 46 F6 7E 0B	

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

## Lab Task 2: -

Replacing the jbe with jna will make no difference so we can write jna in place of jbe. With that as the logic is the same as jbe means jump if below or equal and jna means jump if not above which both means the same so both of these will give us the same results.

(Commands)



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

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

Z:\>mount X E:\AssemblyCL
Drive X is mounted as local directory E:\AssemblyCL\

Z:\>X:\
Illegal command: X:\.

Z:\>nasm bUpdated.asm -o bUpdated.com
Illegal command: nasm.

Z:\>X:\

X:\>nasm bUpdated.asm -o bUpdated.com

X:\>afd bUpdated.com
```

(Initial Data)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0000	SI 0000	CS 19F5	IP 010C	Stack +0 0000	Flags 7200
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 >	1	3	4	5	6	7	8	9	A
0100 E90900 JMP 010C	DS:0103	03	00	05	00	07	00	04	00
010C BB0000 MOV BX,0000	DS:010B	00	BB	00	00	C6	06	0B	01
010F C6060B0100 MOV [010B],00	DS:0113	00	8B	87	03	01	3B	87	05
0114 8B870301 MOV AX,[0103+BX]	DS:011B	01	76	11	8B	97	05	01	89
0118 3B870501 CMP AX,[0105+BX]	DS:0123	87	05	01	89	97	03	01	C6
011C 7611 JNA 012F	DS:012B	06	0B	01	01	81	C3	02	00
011E 8B970501 MOV DX,[0105+BX]	DS:0133	81	FB	06	00	75	DB	80	3E
0122 89870501 MOV [0105+BX],AX	DS:013B	0B	01	01	74	CC	B8	00	4C
0126 89970301 MOV [0103+BX],DX	DS:0143	CD	21	7D	0E	00	74	09	8B
	DS:014B	46	F2	48	3B	46	F6	7E	0B

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

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

(Pass 1 7 takes the place of 4 and 4 takes the place of 7 by swapping)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0007	SI 0000	CS 19F5	IP 0133	Stack +0 0000	Flags 7204
BX 0006	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 0004	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 1 0

CMD >	1	3	4	5	6	7	8	9	A
012F 81C30200 ADD BX,0002	DS:0103	03	00	05	00	04	00	07	00
0133 81FB0600 CMP BX,0006	DS:010B	01	BB	00	00	C6	06	0B	01
0137 75DB JNZ 0114	DS:0113	00	8B	87	03	01	3B	87	05
0139 803E0B0101 CMP [010B],01	DS:011B	01	76	11	8B	97	05	01	89
013E 74CC JZ 010C	DS:0123	87	05	01	89	97	03	01	C6
0140 B8004C MOV AX,4C00	DS:012B	06	0B	01	01	81	C3	02	00
0143 CD21 INT 21	DS:0133	81	FB	06	00	75	DB	80	3E
0145 7D0E JNL 0155	DS:013B	0B	01	01	74	CC	B8	00	4C
0147 007409 ADD [SI+09],DH	DS:0143	CD	21	7D	0E	00	74	09	8B
	DS:014B	46	F2	48	3B	46	F6	7E	0B

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

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

(Pass 3 4 takes the place of 5 and 5 takes the place of 4 by swapping in the next pass the program ends)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

Register	Value	Register	Value	Stack	Flags
AX	0005	SI	0000	CS	19F5
BX	0002	DI	0000	DS	19F5
CX	0045	BP	0000	ES	19F5
DX	0004	SP	FFFE	FS	19F5

CMD >

Address	Code	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

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	00	02	FF	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11	
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

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

## Lab Task 3: -

### Descending Order Using JNBE

(Commands)

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

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

Z:\>mount x E:\AssemblyCL
Drive X is mounted as local directory E:\AssemblyCL\

Z:\>X:\

X:\>nasm DOrder.asm -o DOrder.com

X:\>-l DOrder.lst
Illegal command: -l.

X:\>nasm DOrder.asm -o DOrder.com -l DOrder.lst

X:\>afd DOrder.com_
  
```

(Initial Data)

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5      +2 20CD
CX 0048 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 >

0100 E90900 JMP 010C
0103 06      PUSH ES
0104 0007    ADD [BX],AL
0106 0008    ADD [BX+SI],CL
0108 0009    ADD [BX+DI],CL
010A 0000    ADD [BX+SI],AL
010C BB0000    MOV AX,0000
010F BB0000    MOV BX,0000

DS:0103 06 00 07 00 08 00 09 00
DS:010B 00 B8 00 00 BB 00 00 C6
DS:0113 06 0B 01 00 8B 87 03 01
DS:011B 3B 87 05 01 77 11 8B 97
DS:0123 05 01 89 97 03 01 89 87
DS:012B 05 01 C6 06 0B 01 01 81
DS:0133 C3 02 00 81 FB 06 00 75
DS:013B DB 80 3E 0B 01 01 74 C9
DS:0143 B8 00 4C CD 21 74 09 8E
DS:014B 46 F2 48 3B 46 F6 7E 0E

Z 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.Ω= i|.+.
DS:0010 18 01 10 01 18 01 92 01 01 01 01 FF FF FF 01 FF .....f. ...
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 δ.L.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 6.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

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



(Pass 1 In The Inner Loop Multiple Swaps Happen That Lead To The Swap Flag Is Raised)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0007	SI 0000	CS 19F5	IP 012D	Stack +0 0000	Flags 7291
BX 0002	DI 0000	DS 19F5		+2 20CD	
CX 0048	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0009	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 1 0 1 0 1

CMD >

```

0129 89870501      MOV     [0105+BX],AX
012D C6060B0101    MOV     [010B],01
0132 81C30200      ADD     BX,0002
0136 81FB0600      CMP     BX,0006
013A 75DB          JNZ     0117
013C 803E0B0101    CMP     [010B],01
0141 74C9          JZ      010C
0143 B8004C          MOV     AX,4C00
0146 CD21          INT     21
  
```

DS:0000	CD 20 FF 9F 00 EA F0 FE	AD DE 1B 05 C5 06 00 00	= f.Ω≡ i  .+....
DS:0010	18 01 10 01 18 01 92 01	01 01 01 FF FF FF 01 FF	.....ff. ... .
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF EB 19 C0 11	δ.L.
DS:0030	A2 01 14 00 18 00 F5 19	FF FF FF FF 00 00 00 00	6.....J. ....
DS:0040	05 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....

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

(Pass 2 Same Swapping Occurs And The Swap Flag Is Raised After Which In The Next Pass No Swaps Happen And The Program Exits As The Swap Flag Remains Un Raised Or 0)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0008	SI 0000	CS 19F5	IP 0132	Stack +0 0000	Flags 7295
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0048	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0009	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 1 0 1 1 1

CMD >

```

012D C6060B0101    MOV     [010B],01
0132 81C30200      ADD     BX,0002
0136 81FB0600      CMP     BX,0006
013A 75DB          JNZ     0117
013C 803E0B0101    CMP     [010B],01
0141 74C9          JZ      010C
0143 B8004C          MOV     AX,4C00
0146 CD21          INT     21
0148 7409          JZ      0153
  
```

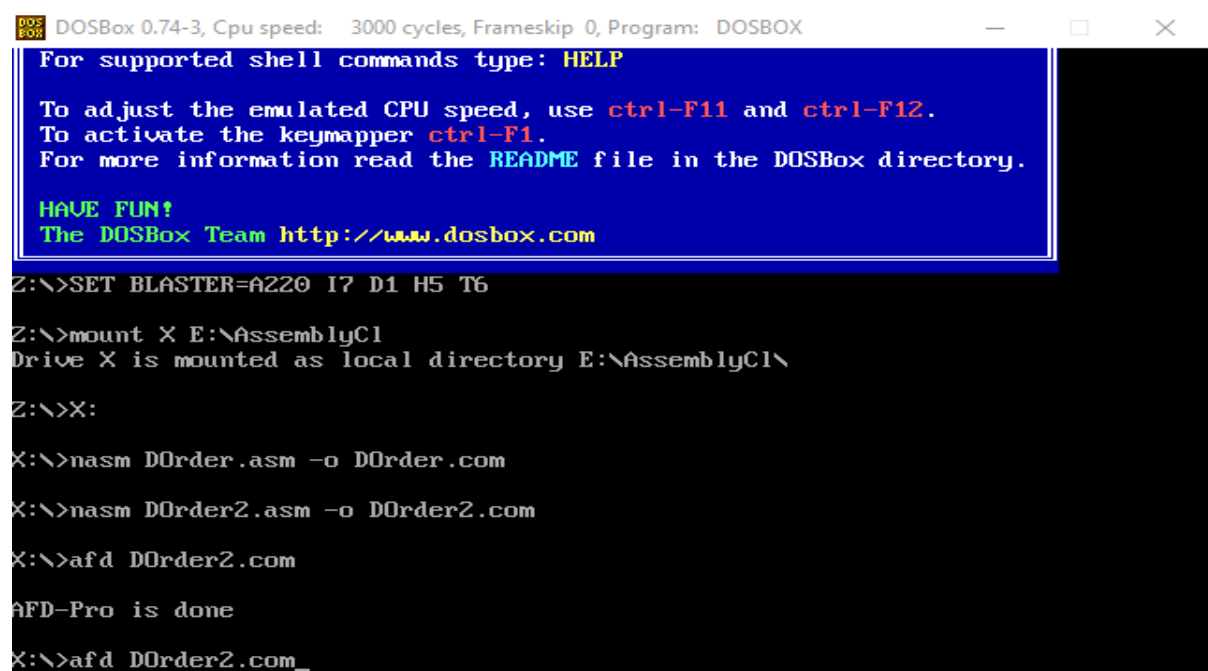
DS:0000	CD 20 FF 9F 00 EA F0 FE	AD DE 1B 05 C5 06 00 00	= f.Ω≡ i  .+....
DS:0010	18 01 10 01 18 01 92 01	01 01 01 FF FF FF 01 FF	.....ff. ... .
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF EB 19 C0 11	δ.L.
DS:0030	A2 01 14 00 18 00 F5 19	FF FF FF FF 00 00 00 00	6.....J. ....
DS:0040	05 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....

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

### **Lab Task 3: -**

Using 2 jump statements one it will compare and jump if the two numbers are equal to noswap code (je) and the other one is if its greater it will use (ja) to jump if it is greater.

(Commands)



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
For supported shell commands type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com

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

Z:\>mount X E:\AssemblyC1
Drive X is mounted as local directory E:\AssemblyC1\

Z:\>X:

X:\>nasm DOrder.asm -o DOrder.com

X:\>nasm DOrder2.asm -o DOrder2.com

X:\>afd DOrder2.com

AFD-Pro is done

X:\>afd DOrder2.com_
```

(Initial Data)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0006	SI 0000	CS 19F5	IP 0133	Stack +0 0000	Flags 7291
BX 0004	DI 0000	DS 19F5		+2 20CD	
CX 004E	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0009	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 1 0 1 0 1

CMD >

012F 89870501	MOV	[0105+BX],AX	DS:0103	07 00 08 00 09 00 06 00
0133 C6060B0101	MOV	[010B],01	DS:010B	01 B8 00 00 BB 00 00 C6
0138 81C30200	ADD	BX,0002	DS:0113	06 0B 01 00 8B 87 03 01
013C 81FB0600	CMP	BX,0006	DS:011B	3B 87 05 01 74 17 3B 87
0140 75D5	JNZ	0117	DS:0123	05 01 77 11 8B 97 05 01
0142 803E0B0101	CMP	[010B],01	DS:012B	89 97 03 01 89 87 05 01
0147 74C3	JZ	010C	DS:0133	C6 06 0B 01 01 81 C3 02
0149 B8004C	MOV	AX,4C00	DS:013B	00 81 FB 06 00 75 D5 80
014C CD21	INT	21	DS:0143	3E 0B 01 01 74 C3 B8 00
			DS:014B	4C CD 21 3B 46 F6 7E 0B

DS:0000	CD 20 FF 9F 00 EA F0 FE	AD DE 1B 05 C5 06 00 00	= f.Ω≡■ i ..†...
DS:0010	18 01 10 01 18 01 92 01	01 01 01 00 02 FF FF FF	.....ff. ....
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF EB 19 C0 11	δ.L.
DS:0030	A2 01 14 00 18 00 F5 19	FF FF FF FF 00 00 00 00	6.....J. ....
DS:0040	05 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....

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

(Pass 1 Swap has Happened Swap Flag Was Raised And Pass 2 Has Started Showing The Swap Flag Rest To Zero)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0008	SI 0000	CS 19F5	IP 011B	Stack +0 0000	Flags 7244
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 004E	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0009	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 1 0 1 0

CMD >

0117 8B870301	MOV	AX,[0103+BX]	DS:0103	08 00 09 00 07 00 06 00
011B 3B870501	CMP	AX,[0105+BX]	DS:010B	00 B8 00 00 BB 00 00 C6
011F 7417	JZ	0138	DS:0113	06 0B 01 00 8B 87 03 01
0121 3B870501	CMP	AX,[0105+BX]	DS:011B	3B 87 05 01 74 17 3B 87
0125 7711	JA	0138	DS:0123	05 01 77 11 8B 97 05 01
0127 8B970501	MOV	DX,[0105+BX]	DS:012B	89 97 03 01 89 87 05 01
012B 89970301	MOV	[0103+BX],DX	DS:0133	C6 06 0B 01 01 81 C3 02
012F 89870501	MOV	[0105+BX],AX	DS:013B	00 81 FB 06 00 75 D5 80
0133 C6060B0101	MOV	[010B],01	DS:0143	3E 0B 01 01 74 C3 B8 00
			DS:014B	4C CD 21 3B 46 F6 7E 0B

DS:0000	CD 20 FF 9F 00 EA F0 FE	AD DE 1B 05 C5 06 00 00	= f.Ω≡■ i ..†...
DS:0010	18 01 10 01 18 01 92 01	01 01 01 00 02 FF FF FF	.....ff. ....
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF EB 19 C0 11	δ.L.
DS:0030	A2 01 14 00 18 00 F5 19	FF FF FF FF 00 00 00 00	6.....J. ....
DS:0040	05 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....

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

(Pass 2 Caused The Data To Be Completely Arranged In Descending Order And Pass 3 Started With No Swaps That Caused The Swap Flag To Be Remained Unraised After Which The Program Ends)

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX 0007	SI 0000	CS 19F5	IP 0149	Stack +0 0000	Flags 7295
BX 0006	DI 0000	DS 19F5		+2 20CD	
CX 004E	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0009	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 1 0 1 1 1

S or SI or SYM

CMD >S

0147 74C3	JZ	010C	DS:0103 09 00 08 00 07 00 06 00
0149 BB04C	MOV	AX,4C00	DS:010B 00 B8 00 00 BB 00 00 C6
014C CD21	INT	21	DS:0113 06 0B 01 00 8B 87 03 01
014E 3B46F6	CMP	AX,[BP-0A]	DS:011B 3B 87 05 01 74 17 3B 87
0151 7E08	JNG	015B	DS:0123 05 01 77 11 8B 97 05 01
0153 B80100	MOV	AX,0001	DS:012B 89 97 03 01 89 87 05 01
0156 EB05	JMP	015D	DS:0133 C6 06 0B 01 01 81 C3 02
0158 E94201	JMP	029D	DS:013B 00 81 FB 06 00 75 D5 80
015B 31C0	XOR	AX,AX	DS:0143 3E 0B 01 01 74 C3 B8 00
			DS:014B 4C CD 21 3B 46 F6 7E 08

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00	= f.Ω≡■ i  ..†...
DS:0010	18	01	10	01	18	01	92	01	01	01	00	02	FF	FF	FF	FF	.....ff. ....
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11		δ. L.
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00	6.....J. ....
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

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

-----THE END-----