

# TASK 1

## 5.1 Example After Changing Values

### Before Sorting

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

AX 0000 SI 0000 CS 19F5 IP 013F Stack +0 0000 Flags 7200  
 BX 0000 DI 0000 DS 19F5 +2 20CD  
 CX 004D 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 >S

Address	Disassembly	Comment
0100	E93C00	JMP 013F
013F	BB0301	MOV BX, 0103
0142	B90600	MOV CX, 0006
0145	E8C8FF	CALL 0110
0148	B8004C	MOV AX, 4C00
014B	CD21	INT 21
014D	48	DEC AX
014E	3B46F6	CMP AX, [BP-0A]
0151	7E08	JNG 015B

Address	Hex	ASCII
DS:0103	36 00 22 00 30 00 46 00	
DS:010B	28 00 37 00 00 49 D1 E1	
DS:0113	BE 00 00 C6 06 0F 01 00	
DS:011B	8B 00 3B 40 02 76 0D 8B	
DS:0123	50 02 89 10 89 40 02 C6	
DS:012B	06 0F 01 01 81 C6 02 00	
DS:0133	39 CE 75 E4 80 3E 0F 01	
DS:013B	01 74 D5 C3 BB 03 01 B9	
DS:0143	06 00 E8 C8 FF B8 00 4C	
DS:014B	CD 21 48 3B 46 F6 7E 08	

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

### When Subroutine Is Called

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

AX 0000 SI 0000 CS 19F5 IP 0110 Stack +0 0148 Flags 7200  
 BX 0103 DI 0000 DS 19F5 +2 0000  
 CX 0006 BP 0000 ES 19F5 HS 19F5 +4 20CD OF DF IF SF ZF AF PF CF  
 DX 0000 SP FFFC SS 19F5 FS 19F5 +6 9FFF 0 0 1 0 0 0 0 0

CMD >S

Address	Disassembly	Comment
0145	E8C8FF	CALL 0110
0110	49	DEC CX
0111	D1E1	SHL CX, 1
0113	BE0000	MOV SI, 0000
0116	C6060F0100	MOV [010F], 00
011B	8B00	MOV AX, [BX+SI]
011D	3B4002	CMP AX, [BX+SI+02]
0120	760D	JNA 012F
0122	8B5002	MOV DX, [BX+SI+02]

Address	Hex	ASCII
DS:0103	36 00 22 00 30 00 46 00	
DS:010B	28 00 37 00 00 49 D1 E1	
DS:0113	BE 00 00 C6 06 0F 01 00	
DS:011B	8B 00 3B 40 02 76 0D 8B	
DS:0123	50 02 89 10 89 40 02 C6	
DS:012B	06 0F 01 01 81 C6 02 00	
DS:0133	39 CE 75 E4 80 3E 0F 01	
DS:013B	01 74 D5 C3 BB 03 01 B9	
DS:0143	06 00 E8 C8 FF B8 00 4C	
DS:014B	CD 21 48 3B 46 F6 7E 08	

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, Program: AFD

AX 0037	SI 000A	CS 19F5	IP 013E	Stack +0 0148	Flags 7295
BX 0103	DI 0000	DS 19F5		+2 0000	
CX 000A	BP 0000	ES 19F5	HS 19F5	+4 20CD	OF DF IF SF ZF AF PF CF
DX 0028	SP FFFC	SS 19F5	FS 19F5	+6 9FFF	0 0 1 1 0 1 1 1

S or SI or SYM

CMD >S

013C 74D5	JZ	0113			
013E C3	RET				
013F BB0301	MOV	BX,0103			
0142 B90600	MOV	CX,0006			
0145 EBC8FF	CALL	0110			
0148 B8004C	MOV	AX,4C00			
014B CD21	INT	21			
014D 4B	DEC	AX			
014E 3B46F6	CMP	AX,[BP-0A]			

1	3	4	5	6	7	8	9	A
DS:0103	22	00	28	00	30	00	36	00
DS:010B	37	00	46	00	00	49	D1	E1
DS:0113	BE	00	00	C6	06	0F	01	00
DS:011B	8B	00	3B	40	02	76	0D	8B
DS:0123	50	02	89	10	89	40	02	C6
DS:012B	06	0F	01	01	81	C6	02	00
DS:0133	39	CE	75	E4	80	3E	0F	01
DS:013B	01	74	D5	C3	BB	03	01	B9
DS:0143	06	00	E8	C8	FF	B8	00	4C
DS:014B	CD	21	48	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
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	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

= f.Ω≡ i|..†...  
.....ff. ....  
δ.L.  
ó.....J. ....  
.....

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

After Subroutine Returns

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

AX 0037	SI 000A	CS 19F5	IP 0148	Stack +0 0000	Flags 7295
BX 0103	DI 0000	DS 19F5		+2 20CD	
CX 000A	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0028	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 1 0 1 1 1

S or SI or SYM

CMD >S

013E C3	RET				
0148 B8004C	MOV	AX,4C00			
014B CD21	INT	21			
014D 4B	DEC	AX			
014E 3B46F6	CMP	AX,[BP-0A]			
0151 7E08	JNG	015B			
0153 B80100	MOV	AX,0001			
0156 EB05	JMP	015D			
0158 E94201	JMP	029D			

1	3	4	5	6	7	8	9	A
DS:0103	22	00	28	00	30	00	36	00
DS:010B	37	00	46	00	00	49	D1	E1
DS:0113	BE	00	00	C6	06	0F	01	00
DS:011B	8B	00	3B	40	02	76	0D	8B
DS:0123	50	02	89	10	89	40	02	C6
DS:012B	06	0F	01	01	81	C6	02	00
DS:0133	39	CE	75	E4	80	3E	0F	01
DS:013B	01	74	D5	C3	BB	03	01	B9
DS:0143	06	00	E8	C8	FF	B8	00	4C
DS:014B	CD	21	48	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
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	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

= f.Ω≡ i|..†...  
.....ff. ....  
δ.L.  
ó.....J. ....  
.....

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

## TASK 2

### 5.2 Example Running As It Is

#### Before Sorting

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

AX 0000 SI 0000 CS 19F5 IP 0175 Stack +0 0000 Flags 7200  
 BX 0103 DI 0000 DS 19F5 +2 20CD  
 CX 000A BP 0000 ES 19F5 HS 19F5 +4 9FFF 0F 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	Instruction	Comment
0172	B90A00	MOV CX,000A
0175	E8C8FF	CALL 0140
0178	BB1701	MOV BX,0117
017B	B91400	MOV CX,0014
017E	E8BFFF	CALL 0140
0181	B8004C	MOV AX,4C00
0184	CD21	INT 21
0186	8B5C02	MOV BX,[SI+02]
0189	89C6	MOV SI,AX

Address	Value	Value	Value	Value	Value	Value	Value	Value	Value
DS:0103	3C	00	37	00	2D	00	32	00	
DS:010B	28	00	23	00	19	00	1E	00	
DS:0113	0A	00	00	00	48	01	49	01	
DS:011B	82	03	DB	22	65	20	29	09	
DS:0123	0A	00	6D	03	63	01	62	00	
DS:012B	78	03	15	02	D0	07	FC	03	
DS:0133	1E	00	C8	00	F9	02	A7	00	
DS:013B	5A	00	05	00	00	49	D1	E1	
DS:0143	BE	00	00	C6	06	3F	01	00	
DS:014B	8B	00	3B	40	02	76	0D	8B	

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

#### After Sorting

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

AX 2065 SI 0026 CS 19F5 IP 0181 Stack +0 0000 Flags 7295  
 BX 0117 DI 0000 DS 19F5 +2 20CD  
 CX 0026 BP 0000 ES 19F5 HS 19F5 +4 9FFF 0F DF IF SF ZF AF PF CF  
 DX 0005 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 1 0 1 1 1

S or SI or SYM

CMD >S

Address	Instruction	Comment
017E	E8BFFF	CALL 0140
0181	B8004C	MOV AX,4C00
0184	CD21	INT 21
0186	8B5C02	MOV BX,[SI+02]
0189	89C6	MOV SI,AX
018B	895EFE	MOV [BP-02],BX
018E	837EFE00	CMP [BP-02],0000
0192	7504	JNZ 0198
0194	85F6	TEST SI,SI

Address	Value	Value	Value	Value	Value	Value	Value	Value	Value
DS:0103	00	00	0A	00	19	00	1E	00	
DS:010B	23	00	28	00	2D	00	32	00	
DS:0113	37	00	3C	00	05	00	0A	00	
DS:011B	1E	00	5A	00	62	00	A7	00	
DS:0123	C8	00	48	01	49	01	63	01	
DS:012B	15	02	F9	02	6D	03	78	03	
DS:0133	82	03	FC	03	D0	07	29	09	
DS:013B	65	20	DB	22	00	49	D1	E1	
DS:0143	BE	00	00	C6	06	3F	01	00	
DS:014B	8B	00	3B	40	02	76	0D	8B	

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

## Task 2

### Example 5.2 After Changing Values

#### Before Sorting And Calling Subroutine

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

AX 0000 SI 0000 CS 19F5 IP 013F Stack +0 0000 Flags 7200  
 BX 0000 DI 0000 DS 19F5 +2 20CD  
 CX 0056 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	Comment
0100	E93C00	JMP 013F
013F	BB0301	MOV BX,0103
0142	B90300	MOV CX,0003
0145	E8C8FF	CALL 0110
0148	BB0901	MOV BX,0109
014B	B90300	MOV CX,0003
014E	E8BFFF	CALL 0110
0151	B8004C	MOV AX,4C00
0154	CD21	INT 21

Address	Hex	ASCII
DS:0103	0B 00 0C 00 1B 00 0A 00	.....I T B ^ . .   . .
DS:010B	0F 00 1E 00 00 49 D1 E1	..i.:e.v .iP.e.ee
DS:0113	BE 00 00 C6 06 0F 01 00	. . . . . u   . . 9 u s c >
DS:011B	8B 00 3B 40 02 76 0D 8B	...t f   . .   . . x u
DS:0123	50 02 89 10 89 40 02 C6	..  ..x_   .L=?d.e
DS:012B	06 0F 01 01 81 C6 02 00	
DS:0133	39 CE 75 E4 80 3E 0F 01	
DS:013B	01 74 D5 C3 BB 03 01 B9	
DS:0143	03 00 E8 C8 FF BB 09 01	
DS:014B	B9 03 00 E8 BF FF B8 00	

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

#### After Sorting First Data Set And Before Returning From The Subroutine For The First Time

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

AX 0000 SI 0004 CS 19F5 IP 013E Stack +0 0148 Flags 7295  
 BX 0103 DI 0000 DS 19F5 +2 0000  
 CX 0004 BP 0000 ES 19F5 HS 19F5 +4 20CD OF DF IF SF ZF AF PF CF  
 DX 0000 SP FFFC SS 19F5 FS 19F5 +6 9FFF 0 0 1 1 0 1 1 1

{R} reg=value  
 CMD >SS

Address	Disassembly	Comment
013C	74D5	JZ 0113
013E	C3	RET
013F	BB0301	MOV BX,0103
0142	B90300	MOV CX,0003
0145	E8C8FF	CALL 0110
0148	BB0901	MOV BX,0109
014B	B90300	MOV CX,0003
014E	E8BFFF	CALL 0110
0151	B8004C	MOV AX,4C00

Address	Hex	ASCII
DS:0103	0B 00 0C 00 1B 00 0A 00	.....I T B ^ . .   . .
DS:010B	0F 00 1E 00 00 49 D1 E1	..i.:e.v .iP.e.ee
DS:0113	BE 00 00 C6 06 0F 01 00	. . . . . u   . . 9 u s c >
DS:011B	8B 00 3B 40 02 76 0D 8B	...t f   . .   . . x u
DS:0123	50 02 89 10 89 40 02 C6	..  ..x_   .L=?d.e
DS:012B	06 0F 01 01 81 C6 02 00	
DS:0133	39 CE 75 E4 80 3E 0F 01	
DS:013B	01 74 D5 C3 BB 03 01 B9	
DS:0143	03 00 E8 C8 FF BB 09 01	
DS:014B	B9 03 00 E8 BF FF B8 00	

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

**After Sorting First Data Set And Returning After The First Subroutine Call And Second Data Set Is Still Not Sorted**

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

AX 000C SI 0004 CS 19F5 IP 0148 Stack +0 0000 Flags 7295  
 BX 0103 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 0000 SP FFFE SS 19F5 FS 19F5 +6 EAO0 0 0 1 1 0 1 1 1

{R} reg=value  
 CMD >SSS

013E C3	RET	DS:0103	0B 00 0C 00 1B 00 0A 00
014B BB0901	MOV BX,0109	DS:010B	0F 00 1E 00 00 49 D1 E1
014B B90300	MOV CX,0003	DS:0113	BE 00 00 C6 06 0F 01 00
014E E8BFFF	CALL 0110	DS:011B	8B 00 3B 40 02 76 0D 8B
0151 B8004C	MOV AX,4C00	DS:0123	50 02 89 10 89 40 02 C6
0154 CD21	INT 21	DS:012B	06 0F 01 01 81 C6 02 00
0156 EB05	JMP 015D	DS:0133	39 CE 75 E4 80 3E 0F 01
0158 E94201	JMP 029D	DS:013B	01 74 D5 C3 BB 03 01 B9
015B 31C0	XOR AX,AX	DS:0143	03 00 E8 C8 FF BB 09 01
		DS:014B	B9 03 00 E8 BF FF B8 00

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

**After Calling The Subroutine For The Second Data Set Which Is Not Sorted Till Now As Can Be Seen The Stack Has Pushed The Address Where It Will Return After Sorting Out The Data**

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

AX 000C SI 0004 CS 19F5 IP 0110 Stack +0 0151 Flags 7295  
 BX 0109 DI 0000 DS 19F5 +2 0000  
 CX 0003 BP 0000 ES 19F5 HS 19F5 +4 20CD OF DF IF SF ZF AF PF CF  
 DX 0000 SP FFFC SS 19F5 FS 19F5 +6 9FFF 0 0 1 1 0 1 1 1

S or SI or SYM  
 CMD >S

014E E8BFFF	CALL 0110	DS:0103	0B 00 0C 00 1B 00 0A 00
0110 49	DEC CX	DS:010B	0F 00 1E 00 00 49 D1 E1
0111 D1E1	SHL CX,1	DS:0113	BE 00 00 C6 06 0F 01 00
0113 BE0000	MOV SI,0000	DS:011B	8B 00 3B 40 02 76 0D 8B
0116 C6060F0100	MOV [010F],00	DS:0123	50 02 89 10 89 40 02 C6
011B 8B00	MOV AX,[BX+SI]	DS:012B	06 0F 01 01 81 C6 02 00
011D 3B4002	CMP AX,[BX+SI+02]	DS:0133	39 CE 75 E4 80 3E 0F 01
0120 760D	JNA 012F	DS:013B	01 74 D5 C3 BB 03 01 B9
0122 8B5002	MOV DX,[BX+SI+02]	DS:0143	03 00 E8 C8 FF BB 09 01
		DS:014B	B9 03 00 E8 BF FF B8 00

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

Second Data Set Is Also Sorted And The Subroutine Has Returned

So Both The Data Sets Are Now Sorted

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

AX 000F SI 0004 CS 19F5 IP 0151 Stack +0 0000 Flags 7295  
 BX 0109 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 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 1 0 1 1 1

S or SI or SYM

CMD >S

013E C3 RET  
 0151 B8004C MOV AX,4C00  
 0154 CD21 INT 21  
 0156 EB05 JMP 015D  
 0158 E94201 JMP 029D  
 015B 31C0 XOR AX,AX  
 015D 8946E2 MOV [BP-1E],AX  
 0160 8B46F6 MOV AX,[BP-0A]  
 0163 D1E0 SHL AX,1

1 3 4 5 6 7 8 9 A  
 DS:0103 0B 00 0C 00 1B 00 0A 00  
 DS:010B 0F 00 1E 00 00 49 D1 E1  
 DS:0113 BE 00 00 C6 06 0F 01 00  
 DS:011B 8B 00 3B 40 02 76 0D 8B  
 DS:0123 50 02 89 10 89 40 02 C6  
 DS:012B 06 0F 01 01 81 C6 02 00  
 DS:0133 39 CE 75 E4 80 3E 0F 01  
 DS:013B 01 74 D5 C3 BB 03 01 B9  
 DS:0143 03 00 E8 C8 FF BB 09 01  
 DS:014B B9 03 00 E8 BF FF B8 00

2 9 A B C D E F 0 1 2 3 4 5 6 7 8  
 DS:0109 0A 00 0F 00 1E 00 00 49 D1 E1 BE 00 00 C6 06 0F .....I 7B¹..f..  
 DS:0119 01 00 8B 00 3B 40 02 76 0D 8B 50 02 89 10 89 40 ..i.:e.v iP.ë.ëe  
 DS:0129 02 C6 06 0F 01 01 81 C6 02 00 39 CE 75 E4 80 3E .f....üf ..9HusQ>  
 DS:0139 0F 01 01 74 D5 C3 BB 03 01 B9 03 00 E8 C8 FF BB ..t f|7. .f..x"7  
 DS:0149 09 01 B9 03 00 E8 BF FF B8 00 4C CD 21 EB 05 E9 ..f|..x7 7.L=†δ.θ

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

HENCE THE MAIN DIFFERENCE IN BOTH THE EXAMPLES IS THAT , THAT IN THE 5.1 EXAMPLE THERE IS ONLY ONE DATA SET AND THE SUBROUTINE IS CALLED ONLY ONCE FOR THAT DATA SET ONLY HOWEVER IN 5.2 THERE ARE 2 DATA SETS AND THE SUBROUTINE IS CALLED TWICE FOR BOTH THE DATA SETS WHICH SORTS OUT BOTH THE DATA SETS.