

## ***Example 5.3: -***

### ***Before Calling The Subroutine Of Bubble Sort***

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

AX 0000	SI 0000	CS 19F5	IP 013E	Stack +0 0000	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 004C	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 E93B00	JMP	013E
013E BB0301	MOV	BX,0103
0141 B90400	MOV	CX,0004
0144 E8CDFF	CALL	0114
0147 B8004C	MOV	AX,4C00
014A CD21	INT	21
014C F2	REP NZ	
014D 48	DEC	AX
014E 3B46F6	CMP	AX,[BP-0A]

1	3	4	5	6	7	8	9	A
DS:0103	06	00	07	00	09	00	08	00
DS:010B	00	8B	00	87	40	02	89	00
DS:0113	C3	49	D1	E1	BE	00	00	C6
DS:011B	06	0B	01	00	8B	00	3B	40
DS:0123	02	76	08	E8	E3	FF	C6	06
DS:012B	0B	01	01	81	C6	02	00	39
DS:0133	CE	75	E9	80	3E	0B	01	01
DS:013B	74	DA	C3	BB	03	01	B9	04
DS:0143	00	E8	CD	FF	B8	00	4C	CD
DS:014B	21	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	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

### ***Before Calling The Swap Sub Routine We Are In The bubble Sort Subroutine Notice That The Stack Pointer Has Change***

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

AX 0006	SI 0000	CS 19F5	IP 0124	Stack +0 0147	Flags 7295
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 1 0 1 1 1

CMD > █

0121 3B4002	CMP	AX,[BX+SI+02]
0124 7608	JNA	012E
0126 E8E3FF	CALL	010C
0129 C060B0101	MOV	[010B],01
012E 81C60200	ADD	SI,0002
0132 39CE	CMP	SI,CX
0134 75E9	JNZ	011F
0136 803E0B0101	CMP	[010B],01
013B 74DA	JZ	0117

1	3	4	5	6	7	8	9	A
DS:0103	06	00	07	00	09	00	08	00
DS:010B	00	8B	00	87	40	02	89	00
DS:0113	C3	49	D1	E1	BE	00	00	C6
DS:011B	06	0B	01	00	8B	00	3B	40
DS:0123	02	76	08	E8	E3	FF	C6	06
DS:012B	0B	01	01	81	C6	02	00	39
DS:0133	CE	75	E9	80	3E	0B	01	01
DS:013B	74	DA	C3	BB	03	01	B9	04
DS:0143	00	E8	CD	FF	B8	00	4C	CD
DS:014B	21	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	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

After Calling The Swap Subroutine For The Last Two ValuesNotice That The Stack Has Changed Once Again For The Swap Subroutine

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

AX 0009	SI 0004	CS 19F5	IP 010C	Stack +0 0129	Flags 7200
BX 0103	DI 0000	DS 19F5		+2 0147	
CX 0006	BP 0000	ES 19F5	HS 19F5	+4 0000	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFA	SS 19F5	FS 19F5	+6 20CD	0 0 1 0 0 0 0 0

```

CMD >
0126 E8E3FF CALL 010C
010C 8B00 MOV AX,[BX+SI]
010E 874002 XCHG AX,[BX+SI+02]
0111 8900 MOV [BX+SI],AX
0113 C3 RET
0114 49 DEC CX
0115 D1E1 SHL CX,1
0117 BE0000 MOV SI,0000
011A C6060B0100 MOV [010B],00
  
```

1	3	4	5	6	7	8	9	A
DS:0103	06	00	07	00	09	00	08	00
DS:010B	00	8B	00	87	40	02	89	00
DS:0113	C3	49	D1	E1	BE	00	00	C6
DS:011B	06	0B	01	00	8B	00	3B	40
DS:0123	02	76	08	E8	E3	FF	C6	06
DS:012B	0B	01	01	81	C6	02	00	39
DS:0133	CE	75	E9	80	3E	0B	01	01
DS:013B	74	DA	C3	BB	03	01	B9	04
DS:0143	00	E8	CD	FF	B8	00	4C	CD
DS:014B	21	F2	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

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

Notice That The xchg command is used to exchange the values between two spaces and after which the value in ax is returned to first space

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

AX 0008	SI 0004	CS 19F5	IP 0113	Stack +0 0129	Flags 7200
BX 0103	DI 0000	DS 19F5		+2 0147	
CX 0006	BP 0000	ES 19F5	HS 19F5	+4 0000	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFA	SS 19F5	FS 19F5	+6 20CD	0 0 1 0 0 0 0 0

```

CMD >
0111 8900 MOV [BX+SI],AX
0113 C3 RET
0114 49 DEC CX
0115 D1E1 SHL CX,1
0117 BE0000 MOV SI,0000
011A C6060B0100 MOV [010B],00
011F 8B00 MOV AX,[BX+SI]
0121 3B4002 CMP AX,[BX+SI+02]
0124 7608 JNA 012E
  
```

1	3	4	5	6	7	8	9	A
DS:0103	06	00	07	00	08	00	09	00
DS:010B	00	8B	00	87	40	02	89	00
DS:0113	C3	49	D1	E1	BE	00	00	C6
DS:011B	06	0B	01	00	8B	00	3B	40
DS:0123	02	76	08	E8	E3	FF	C6	06
DS:012B	0B	01	01	81	C6	02	00	39
DS:0133	CE	75	E9	80	3E	0B	01	01
DS:013B	74	DA	C3	BB	03	01	B9	04
DS:0143	00	E8	CD	FF	B8	00	4C	CD
DS:014B	21	F2	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

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

**After The Swap Subroutine Has Returned To The Bubble Sort Subroutine  
Notice That The Stack Pointer Has popped An Address To Come Back From  
Where It Was Called**

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

AX 0008	SI 0004	CS 19F5	IP 0129	Stack +0 0147	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 >

0113 C3	RET	
0129 C6060B0101	MOV	[010B],01
012E 81C60200	ADD	SI,0002
0132 39CE	CMP	SI,CX
0134 75E9	JNZ	011F
0136 803E0B0101	CMP	[010B],01
013B 74DA	JZ	0117
013D C3	RET	
013E BB0301	MOV	BX,0103

DS:0103	06 00 07 00 08 00 09 00
DS:010B	00 8B 00 87 40 02 89 00
DS:0113	C3 49 D1 E1 BE 00 00 C6
DS:011B	06 0B 01 00 8B 00 3B 40
DS:0123	02 76 08 E8 E3 FF C6 06
DS:012B	0B 01 01 81 C6 02 00 39
DS:0133	CE 75 E9 80 3E 0B 01 01
DS:013B	74 DA C3 BB 03 01 B9 04
DS:0143	00 E8 CD FF B8 00 4C CD
DS:014B	21 F2 48 3B 46 F6 7E 08

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

**As There Were No More Values That Should've Been Swapped So It Didn't Swap Them**

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

AX 0008	SI 0006	CS 19F5	IP 013D	Stack +0 0147	Flags 7295
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 1 0 1 1 1

CMD >

013B 74DA	JZ	0117
013D C3	RET	
013E BB0301	MOV	BX,0103
0141 B90400	MOV	CX,0004
0144 EBCDFF	CALL	0114
0147 B8004C	MOV	AX,4C00
014A CD21	INT	21
014C F2	REP NZ	
014D 4B	DEC	AX

DS:0103	06 00 07 00 08 00 09 00
DS:010B	00 8B 00 87 40 02 89 00
DS:0113	C3 49 D1 E1 BE 00 00 C6
DS:011B	06 0B 01 00 8B 00 3B 40
DS:0123	02 76 08 E8 E3 FF C6 06
DS:012B	0B 01 01 81 C6 02 00 39
DS:0133	CE 75 E9 80 3E 0B 01 01
DS:013B	74 DA C3 BB 03 01 B9 04
DS:0143	00 E8 CD FF B8 00 4C CD
DS:014B	21 F2 48 3B 46 F6 7E 08

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

**Finally After Returning From The Bubble Sort Subroutine The Stack Has Again Changed So That It Can Return From Where It Was Called**

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

AX 0008	SI 0006	CS 19F5	IP 0147	Stack +0 0000	Flags 7295
BX 0103	DI 0000	DS 19F5		+2 20CD	
CX 0006	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

CMD >

013D C3	RET	DS:0103	06 00 07 00 08 00 09 00
0147 BB004C	MOV AX,4C00	DS:010B	00 8B 00 87 40 02 89 00
014A CD21	INT 21	DS:0113	C3 49 D1 E1 BE 00 00 C6
014C F2	REPNZ	DS:011B	06 0B 01 00 8B 00 3B 40
014D 4B	DEC AX	DS:0123	02 76 08 E8 E3 FF C6 06
014E 3B46F6	CMP AX,[BP-0A]	DS:012B	0B 01 01 81 C6 02 00 39
0151 7E08	JNG 015B	DS:0133	CE 75 E9 80 3E 0B 01 01
0153 BB0100	MOV AX,0001	DS:013B	74 DA C3 BB 03 01 B9 04
0156 EB05	JMP 015D	DS:0143	00 E8 CD FF B8 00 4C CD
		DS:014B	21 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≡■ ÷ ..†...
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 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 00	.....

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

## Example 5.4

### Before Calling The Subroutine

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

AX 0000	SI 0000	CS 19F5	IP 0149	Stack +0 0000	Flags 7200
BX 0103	DI 0000	DS 19F5		+2 20CD	
CX 0054	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 >

0146 BB0301	MOV	BX,0103	DS:0103	06 00 07 00 09 00 08 00
0149 B90400	MOV	CX,0004	DS:010B	00 50 8B 00 87 40 02 89
014C E8C7FF	CALL	0116	DS:0113	00 58 C3 50 51 56 49 D1
014F B8004C	MOV	AX,4C00	DS:011B	E1 BE 00 00 C6 06 0B 01
0152 CD21	INT	21	DS:0123	00 8B 00 3B 40 02 76 0B
0154 0100	ADD	[BX+SI],AX	DS:012B	E8 DE FF C6 06 0B 01 01
0156 EB05	JMP	015D	DS:0133	81 C6 02 00 39 CE 75 E9
0158 E94201	JMP	029D	DS:013B	80 3E 0B 01 01 74 DA 5E
015B 31C0	XOR	AX,AX	DS:0143	59 58 C3 BB 03 01 B9 04
			DS:014B	00 E8 C7 FF B8 00 4C CD

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 00	.....

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

### After Calling The Sub Routine Of Bubble Sort

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

AX 0000	SI 0000	CS 19F5	IP 0116	Stack +0 014F	Flags 7200
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 FFC	SS 19F5	FS 19F5	+6 9FFF	0 0 1 0 0 0 0 0

CMD >

014C E8C7FF	CALL	0116	DS:0103	06 00 07 00 09 00 08 00
0116 50	PUSH	AX	DS:010B	00 50 8B 00 87 40 02 89
0117 51	PUSH	CX	DS:0113	00 58 C3 50 51 56 49 D1
0118 56	PUSH	SI	DS:011B	E1 BE 00 00 C6 06 0B 01
0119 49	DEC	CX	DS:0123	00 8B 00 3B 40 02 76 0B
011A D1E1	SHL	CX,1	DS:012B	E8 DE FF C6 06 0B 01 01
011C BE0000	MOV	SI,0000	DS:0133	81 C6 02 00 39 CE 75 E9
011F C6060B0100	MOV	[010B],00	DS:013B	80 3E 0B 01 01 74 DA 5E
0124 8B00	MOV	AX,[BX+SI]	DS:0143	59 58 C3 BB 03 01 B9 04
			DS:014B	00 E8 C7 FF B8 00 4C CD

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 00	.....

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

### After Pushing Values Of Ax And Cx And Si

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0000 SI 0000 CS 19F5 IP 0119 Stack +0 0000 Flags 7200
BX 0103 DI 0000 DS 19F5 +2 0004
CX 0004 BP 0000 ES 19F5 HS 19F5 +4 0000 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF6 SS 19F5 FS 19F5 +6 014F 0 0 1 0 0 0 0 0

CMD >

0118 56          PUSH    SI
0119 49          DEC     CX
011A D1E1        SHL     CX,1
011C BE0000      MOV     SI,0000
011F C6060B0100  MOV     [010B],00
0124 8B00        MOV     AX,[BX+SI]
0126 3B4002      CMP     AX,[BX+SI+02]
0129 7608        JNA     0133
012B EBDEFF      CALL    010C

1  DS:0103 06 00 07 00 09 00 08 00
   DS:010B 00 50 8B 00 87 40 02 89
   DS:0113 00 58 C3 50 51 56 49 D1
   DS:011B E1 BE 00 00 C6 06 0B 01
   DS:0123 00 8B 00 3B 40 02 76 08
   DS:012B E8 DE FF C6 06 0B 01 01
   DS:0133 81 C6 02 00 39 CE 75 E9
   DS:013B 80 3E 0B 01 01 74 DA 5E
   DS:0143 59 58 C3 BB 03 01 B9 04
   DS:014B 00 E8 C7 FF B8 00 4C CD

2  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 .....

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

```

### After Calling The Swap Subroutine For The Last Two Numbers

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0009 SI 0004 CS 19F5 IP 010C Stack +0 012E Flags 7200
BX 0103 DI 0000 DS 19F5 +2 0000
CX 0006 BP 0000 ES 19F5 HS 19F5 +4 0004 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF4 SS 19F5 FS 19F5 +6 0000 0 0 1 0 0 0 0 0

CMD >

012B EBDEFF      CALL    010C
010C 50          PUSH    AX
010D 8B00        MOV     AX,[BX+SI]
010F 874002      XCHG    AX,[BX+SI+02]
0112 8900        MOV     [BX+SI],AX
0114 58          POP     AX
0115 C3          RET
0116 50          PUSH    AX
0117 51          PUSH    CX

1  DS:0103 06 00 07 00 09 00 08 00
   DS:010B 00 50 8B 00 87 40 02 89
   DS:0113 00 58 C3 50 51 56 49 D1
   DS:011B E1 BE 00 00 C6 06 0B 01
   DS:0123 00 8B 00 3B 40 02 76 08
   DS:012B E8 DE FF C6 06 0B 01 01
   DS:0133 81 C6 02 00 39 CE 75 E9
   DS:013B 80 3E 0B 01 01 74 DA 5E
   DS:0143 59 58 C3 BB 03 01 B9 04
   DS:014B 00 E8 C7 FF B8 00 4C CD

2  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 .....

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

```



### Again After Pushing The Value Of Ax

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

AX 0009	SI 0004	CS 19F5	IP 010D	Stack +0 0009	Flags 7200
BX 0103	DI 0000	DS 19F5		+2 012E	
CX 0006	BP 0000	ES 19F5	HS 19F5	+4 0000	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF2	SS 19F5	FS 19F5	+6 0004	0 0 1 0 0 0 0 0

CMD > █

010C 50	PUSH	AX	DS:0103 06 00 07 00 09 00 08 00
010D 8B00	MOV	AX, [BX+SI]	DS:010B 00 50 8B 00 87 40 02 89
010F 874002	XCHG	AX, [BX+SI+02]	DS:0113 00 58 C3 50 51 56 49 D1
0112 8900	MOV	[BX+SI], AX	DS:011B E1 BE 00 00 C6 06 0B 01
0114 58	POP	AX	DS:0123 00 8B 00 3B 40 02 76 0B
0115 C3	RET		DS:012B E8 DE FF C6 06 0B 01 01
0116 50	PUSH	AX	DS:0133 81 C6 02 00 39 CE 75 E9
0117 51	PUSH	CX	DS:013B 80 3E 0B 01 01 74 DA 5E
0118 56	PUSH	SI	DS:0143 59 58 C3 BB 03 01 B9 04
			DS:014B 00 E8 C7 FF B8 00 4C CD

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≡■ ; ..†...
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

### After Exchanging The Values

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

AX 0008	SI 0004	CS 19F5	IP 0114	Stack +0 0009	Flags 7200
BX 0103	DI 0000	DS 19F5		+2 012E	
CX 0006	BP 0000	ES 19F5	HS 19F5	+4 0000	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF2	SS 19F5	FS 19F5	+6 0004	0 0 1 0 0 0 0 0

CMD > █

0112 8900	MOV	[BX+SI], AX	DS:0103 06 00 07 00 08 00 09 00
0114 58	POP	AX	DS:010B 00 50 8B 00 87 40 02 89
0115 C3	RET		DS:0113 00 58 C3 50 51 56 49 D1
0116 50	PUSH	AX	DS:011B E1 BE 00 00 C6 06 0B 01
0117 51	PUSH	CX	DS:0123 00 8B 00 3B 40 02 76 0B
0118 56	PUSH	SI	DS:012B E8 DE FF C6 06 0B 01 01
0119 49	DEC	CX	DS:0133 81 C6 02 00 39 CE 75 E9
011A D1E1	SHL	CX, 1	DS:013B 80 3E 0B 01 01 74 DA 5E
011C BE0000	MOV	SI, 0000	DS:0143 59 58 C3 BB 03 01 B9 04
			DS:014B 00 E8 C7 FF B8 00 4C CD

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≡■ ; ..†...
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

**Popping The Value Of Ax In The Swap Subroutine As The Value In Ax Before The Subroutine Is Again Put In Ax**

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

Register	Value	Register	Value	Register	Value	Register	Value	Stack	Offset	Value	Flags
AX	0009	SI	0004	CS	19F5	IP	0115		+0	012E	7200
BX	0103	DI	0000	DS	19F5				+2	0000	
CX	0006	BP	0000	ES	19F5	HS	19F5		+4	0004	OF DF IF SF ZF AF PF CF
DX	0000	SP	FFF4	SS	19F5	FS	19F5		+6	0000	0 0 1 0 0 0 0 0

CMD >

Address	Disassembly	Comment
0114 58	POP	AX
0115 C3	RET	
0116 50	PUSH	AX
0117 51	PUSH	CX
0118 56	PUSH	SI
0119 49	DEC	CX
011A D1E1	SHL	CX,1
011C BE0000	MOV	SI,0000
011F C6060B0100	MOV	[010B],00

Address	Disassembly	Comment
DS:0103	06 00 07 00 08 00 09 00	
DS:010B	00 50 8B 00 87 40 02 89	
DS:0113	00 58 C3 50 51 56 49 D1	
DS:011B	E1 BE 00 00 C6 06 0B 01	
DS:0123	00 8B 00 3B 40 02 76 0B	
DS:012B	E8 DE FF C6 06 0B 01 01	
DS:0133	81 C6 02 00 39 CE 75 E9	
DS:013B	80 3E 0B 01 01 74 DA 5E	
DS:0143	59 58 C3 BB 03 01 B9 04	
DS:014B	00 EB C7 FF B8 00 4C CD	

Address	Disassembly	Comment
DS:0000	CD 20 FF 9F 00 EA F0 FE	
DS:0010	18 01 10 01 18 01 92 01	
DS:0020	FF FF FF FF FF FF FF FF	
DS:0030	A2 01 14 00 18 00 F5 19	
DS:0040	05 00 00 00 00 00 00 00	

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



### After Returning From The Swap Subroutine Into The Bubble Sort Subroutine

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

AX 0009	SI 0004	CS 19F5	IP 012E	Stack +0 0000	Flags 7200
BX 0103	DI 0000	DS 19F5		+2 0004	
CX 0006	BP 0000	ES 19F5	HS 19F5	+4 0000	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF6	SS 19F5	FS 19F5	+6 014F	0 0 1 0 0 0 0 0

CMD >

0115 C3	RET	
012E C6060B0101	MOV	[010B],01
0133 81C60200	ADD	SI,0002
0137 39CE	CMP	SI,CX
0139 75E9	JNZ	0124
013B 803E0B0101	CMP	[010B],01
0140 74DA	JZ	011C
0142 5E	POP	SI
0143 59	POP	CX

DS:0103	06 00 07 00 08 00 09 00
DS:010B	00 50 8B 00 87 40 02 89
DS:0113	00 58 C3 50 51 56 49 D1
DS:011B	E1 BE 00 00 C6 06 0B 01
DS:0123	00 8B 00 3B 40 02 76 0B
DS:012B	E8 DE FF C6 06 0B 01 01
DS:0133	81 C6 02 00 39 CE 75 E9
DS:013B	80 3E 0B 01 01 74 DA 5E
DS:0143	59 58 C3 BB 03 01 B9 04
DS:014B	00 E8 C7 FF B8 00 4C CD

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	.....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	ó.....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

In The Bubble Sort Subroutine As The Data Is All In Ascending Order No Swaps Other Than The One Mentioned Above is Performed And The Values Of SI CX and AX Are Popped As They Were Previously Pushed Into The Stack To Maintain The Before Values Of These Registers

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

AX 0000	SI 0000	CS 19F5	IP 0145	Stack +0 014F	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

CMD >

0144 58	POP	AX
0145 C3	RET	
0146 BB0301	MOV	BX,0103
0149 B90400	MOV	CX,0004
014C E8C7FF	CALL	0116
014F B8004C	MOV	AX,4C00
0152 CD21	INT	21
0154 0100	ADD	[BX+SI],AX
0156 EB05	JMP	015D

DS:0103	06 00 07 00 08 00 09 00
DS:010B	00 50 8B 00 87 40 02 89
DS:0113	00 58 C3 50 51 56 49 D1
DS:011B	E1 BE 00 00 C6 06 0B 01
DS:0123	00 8B 00 3B 40 02 76 0B
DS:012B	E8 DE FF C6 06 0B 01 01
DS:0133	81 C6 02 00 39 CE 75 E9
DS:013B	80 3E 0B 01 01 74 DA 5E
DS:0143	59 58 C3 BB 03 01 B9 04
DS:014B	00 E8 C7 FF B8 00 4C CD

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	.....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	ó.....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

**After Returning From The Bubble Sort Subroutine The Stack Has Changed Again As to Return To The Place From Where It Was Called**

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

AX 0000	SI 0000	CS 19F5	IP 014F	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 EA00	0 0 1 1 0 1 1 1

  

CMD >		1	3	4	5	6	7	8	9	A
0145 C3	RET	DS:0103	06 00	07 00	08 00	09 00				
014F B8004C	MOV AX,4C00	DS:010B	00 50	8B 00	87 40	02 89				
0152 CD21	INT 21	DS:0113	00 58	C3 50	51 56	49 D1				
0154 0100	ADD [BX+SI],AX	DS:011B	E1 BE	00 00	C6 06	0B 01				
0156 EB05	JMP 015D	DS:0123	00 8B	00 3B	40 02	76 0B				
0158 E94201	JMP 029D	DS:012B	E8 DE	FF C6	06 0B	01 01				
015B 31C0	XOR AX,AX	DS:0133	81 C6	02 00	39 CE	75 E9				
015D 8946E2	MOV [BP-1E],AX	DS:013B	80 3E	0B 01	01 74	DA 5E				
0160 8B46F6	MOV AX,[BP-0A]	DS:0143	59 58	C3 BB	03 01	B9 04				
		DS:014B	00 E8	C7 FF	B8 00	4C CD				

  

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

### **Note: -**

The main difference in both the examples is that in the first example we are not using stack to store the previous data in the registers while in the second example we are using the stack to store the values of the register in the stack and after using these registers for our required functions we pop back the values of the registers from the stack back to the registers to maintain the values in the registers so they are not overwrote and it stays as they were before we used them.