

Name : Tazmeen Afroz
Roll no: 22p-9252
Section : BAI-4A
Lab-Task-4

Step 1

first I store the value 0100 in the cs register.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD - x
AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0027 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP 19FE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0

[ R } reg=value
CM >cs = 0100

0100 31C0 XOR AX,AX
0102 BB1F01 MOV BX,011F
0105 0307 ADD AX,[BX]
0107 81C30200 ADD BX,0002
010B 0307 ADD AX,[BX]
010D 81C30200 ADD BX,0002
0111 0307 ADD AX,[BX]
0113 81C30200 ADD BX,0002

DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0008 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
DS:0018 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
DS:0020 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0028 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DS:0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DS:0038 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DS:0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

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 .....ft. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 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
```

Step 2

Then store 0500 in the ds register.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD - x
AX 0000 SI 0000 CS 0100 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0027 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

[CR] register value
CMD >ds = 0500
CD

0100 0000 ADD [BX+SI],AL
0102 0000 ADD [BX+SI],AL
0104 0000 ADD [BX+SI],AL
0106 0000 ADD [BX+SI],AL
0108 0000 ADD [BX+SI],AL
010A 0000 ADD [BX+SI],AL
010C 0000 ADD [BX+SI],AL
010E 0000 ADD [BX+SI],AL

1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00

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 .....ft. ....
DS:0020 FF 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
```

Step 3:

Visualizing the both memory addresses.

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

AX 0000	SI 0000	CS 0100	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 0500		+2 20CD	
CX 0027	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 0000	ADD	[BX+SI],AL
0102 0000	ADD	[BX+SI],AL
0104 0000	ADD	[BX+SI],AL
0106 0000	ADD	[BX+SI],AL
0108 0000	ADD	[BX+SI],AL
010A 0000	ADD	[BX+SI],AL
010C 0000	ADD	[BX+SI],AL
010E 0000	ADD	[BX+SI],AL

1	0	1	2	3	4	5	6	7
CS:0100	00	00	00	00	00	00	00	00
CS:0108	00	00	00	00	00	00	00	00
CS:0110	00	00	00	00	00	00	00	00
CS:0118	00	00	00	00	00	00	00	00
CS:0120	00	00	00	00	00	00	00	00
CS:0128	00	00	00	00	00	00	00	00
CS:0130	00	00	00	00	00	00	00	00
CS:0138	00	00	00	00	00	00	00	00
CS:0140	00	00	00	00	00	00	00	00
CS:0148	00	00	00	00	00	00	00	00

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0500	00	00	00	00	00	20	20	20	20	20	20	00	00	00	00	00
DS:0510	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
DS:0520	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
DS:0530	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
DS:0540	00	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
```

Step 4

For pointing to the same memory 0600 I am adding an offset of 100 with ds register and 500 with the cs register.

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

AX 0000	SI 0000	CS 0100	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 0600		+2 20CD	
CX 0027	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

reg=value

CMD >ds = 0500+0100

0100 0000	ADD	[BX+SI],AL	CS:0100	00 00 00 00 00 00 00 00
0102 0000	ADD	[BX+SI],AL	CS:0108	00 00 00 00 00 00 00 00
0104 0000	ADD	[BX+SI],AL	CS:0110	00 00 00 00 00 00 00 00
0106 0000	ADD	[BX+SI],AL	CS:0118	00 00 00 00 00 00 00 00
0108 0000	ADD	[BX+SI],AL	CS:0120	00 00 00 00 00 00 00 00
010A 0000	ADD	[BX+SI],AL	CS:0128	00 00 00 00 00 00 00 00
010C 0000	ADD	[BX+SI],AL	CS:0130	00 00 00 00 00 00 00 00
010E 0000	ADD	[BX+SI],AL	CS:0138	00 00 00 00 00 00 00 00
			CS:0140	00 00 00 00 00 00 00 00
			CS:0148	00 00 00 00 00 00 00 00

DS:0500	61 6E 64 20 6D 75 73 74	20 62 65 20 52 45 47 20	and must be REG
DS:0510	6F 72 20 41 44 44 52 00	00 00 00 00 A8 4D 41 53	or ADDR.MAS
DS:0520	4B 20 6F 72 20 4F 46 46	53 45 54 20 65 78 70 65	K or OFF SET expe
DS:0530	63 74 65 64 00 00 00 00	A9 53 74 61 63 6B 20 6E	cted.... rStack n
DS:0540	6F 74 20 69 6E 20 52 41	4D 20 28 53 53 3A 53 50	ot in RA M (SS:SP

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

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

AX 0000	SI 0000	CS 0100	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 0600		+2 20CD	
CX 0027	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

reg=value

CMD >cs = 0100+0500

0100 0000	ADD	[BX+SI],AL	CS:0100	00 00 00 00 00 00 00 00
0102 0000	ADD	[BX+SI],AL	CS:0108	00 00 00 00 00 00 00 00
0104 0000	ADD	[BX+SI],AL	CS:0110	00 00 00 00 00 00 00 00
0106 0000	ADD	[BX+SI],AL	CS:0118	00 00 00 00 00 00 00 00
0108 0000	ADD	[BX+SI],AL	CS:0120	00 00 00 00 00 00 00 00
010A 0000	ADD	[BX+SI],AL	CS:0128	00 00 00 00 00 00 00 00
010C 0000	ADD	[BX+SI],AL	CS:0130	00 00 00 00 00 00 00 00
010E 0000	ADD	[BX+SI],AL	CS:0138	00 00 00 00 00 00 00 00
			CS:0140	00 00 00 00 00 00 00 00
			CS:0148	00 00 00 00 00 00 00 00

DS:0500	61 6E 64 20 6D 75 73 74	20 62 65 20 52 45 47 20	and must be REG
DS:0510	6F 72 20 41 44 44 52 00	00 00 00 00 A8 4D 41 53	or ADDR.MAS
DS:0520	4B 20 6F 72 20 4F 46 46	53 45 54 20 65 78 70 65	K or OFF SET expe
DS:0530	63 74 65 64 00 00 00 00	A9 53 74 61 63 6B 20 6E	cted.... rStack n
DS:0540	6F 74 20 69 6E 20 52 41	4D 20 28 53 53 3A 53 50	ot in RA M (SS:SP

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

Step 6:

Now they are pointing to the same memory address 0600.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD - x
AX 0000 SI 0000 CS 0600 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 0600 +2 20CD
CX 0027 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 6E OUTSB
0101 7661 JNA 0164
0103 6C INSB
0104 6964206F70 IMUL SP,[SI+20],706F
0109 636F64 ARPL [BX+64],BP
010C 65 DB 65
010D 0000 ADD [BX+SI],AL
010F 0000 ADD [BX+SI],AL

1 0 1 2 3 4 5 6 7
CS:0600 00 00 00 00 00 00 00 B0 55
CS:0608 6E 6B 6E 6F 77 6E 20 6D
CS:0610 6E 65 6D 6F 6E 69 63 00
CS:0618 00 00 00 00 B1 55 6E 73
CS:0620 75 70 70 6F 72 74 65 64
CS:0628 20 6F 70 65 72 61 6E 64
CS:0630 00 00 00 B2 49 6E 76 61
CS:0638 6C 69 64 20 6F 70 65 72
CS:0640 61 6E 64 20 74 79 70 65
CS:0648 20 6F 72 20 73 69 7A 65

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0600 00 00 00 00 00 00 B0 55 6E 6B 6E 6F 77 6E 20 6D .....U nknown m
DS:0610 6E 65 6D 6F 6E 69 63 00 00 00 00 00 B1 55 6E 73 nemonic. ....Uns
DS:0620 75 70 70 6F 72 74 65 64 20 6F 70 65 72 61 6E 64 upported operand
DS:0630 00 00 00 B2 49 6E 76 61 6C 69 64 20 6F 70 65 72 ...Inva lid oper
DS:0640 61 6E 64 20 74 79 70 65 20 6F 72 20 73 69 7A 65 and type or size

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