

1. Use the REGISTER command to display the current contents of all the internal registers by typing R.

o List the values of the following registers:

AX	0000	SP	00FD
BX	0000	CS	073F
CX	0000	DS	073F
DX	0000	SS	073F
IP	0100	ES	073F

```
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 C C:\
Mounting c:\ is NOT recommended. Please mount a (sub)directory next time.
Drive C is mounted as local directory C:\

Z:\>C:

C:\>cd MASM

C:\MASM>debug
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100 NU UP EI PL NZ NA PO NC
073F:0100 0000          ADD     [BX+SI],AL          DS:0000=CD
_
```

o What is the address of the next instruction to be executed

Offset=073F

Logical=0100

Physical Address=073F0+0100=74F0

o What is the instruction held at this address?

MOV AX , A

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

Z:\>Mount C C:\
Mounting c:\ is NOT recommended. Please mount a (sub)directory next time.
Drive C is mounted as local directory C:\

Z:\>C:

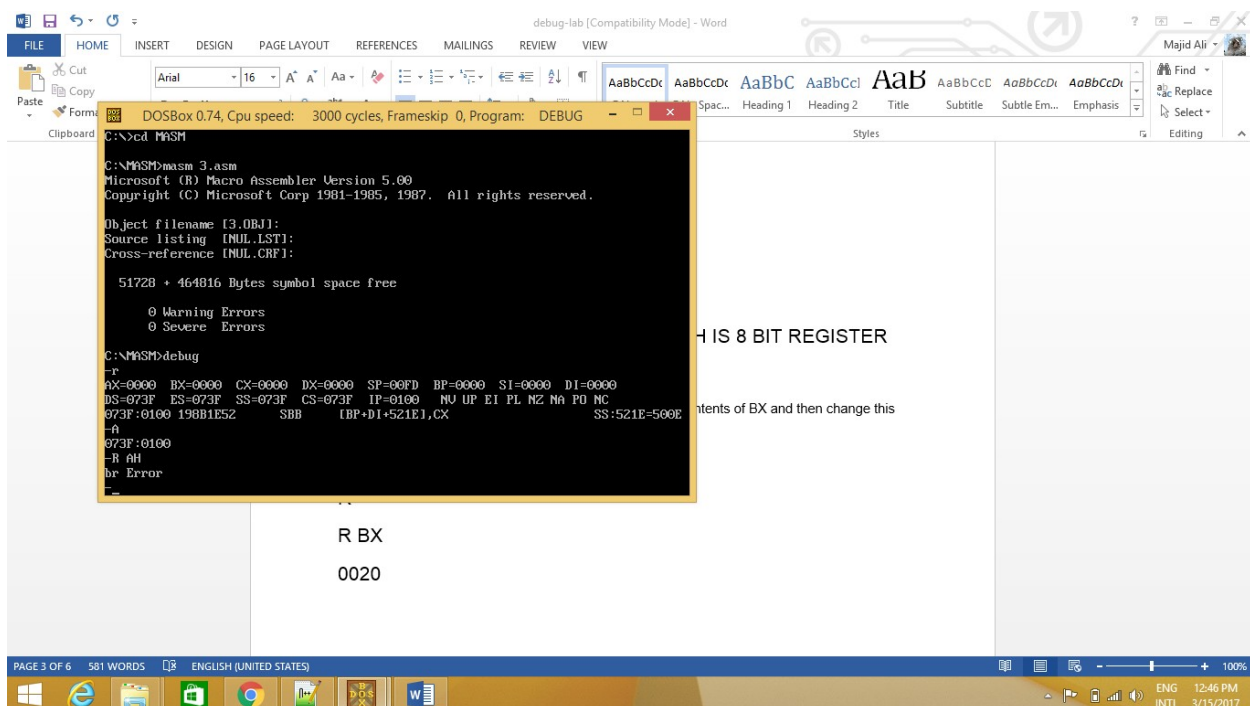
C:\>cd MASM

C:\MASM>debug
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100  NU UP EI PL NZ NA PO NC
073F:0100 0000          ADD     [BX+SI],AL          DS:0000=CD
-a
073F:0100
-a
073F:0100 MOV AX,7
073F:0103
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100  NU UP EI PL NZ NA PO NC
073F:0100 B80700          MOV     AX,0007
-
```

2. Enter the command: R AH (hit ENTER key)

ERROR

- o What happens? Can you find out why?
- o IT SHOWS ERROR BECAUSE AH IS 8 BIT REGISTER



```
C:\>cd MASM
C:\MASM>masm 3.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [3.0BJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

51728 + 464816 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\MASM>debug
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100 NU UP EI PL NZ NA PO NC
073F:0100 198B1E52 SBB [BP+DI+521E],CX SS:521E=500E
-a
073F:0100
-r AH
br Error
-
```

THIS IS 8 BIT REGISTER

Contents of BX and then change this

R BX

0020

3. Use a REGISTER command to first display the current contents of BX and then change this value to "0020".

Debug

R

R BX

0020

4. Use a REGISTER command to first display the current contents of IP and then change this value to "0200".

R

Debug

R

R IP

0200

```

Z:\>Mount C C:\
Mounting c:\ is NOT recommended. Please mount a (sub)directory next time.
Drive C is mounted as local directory C:\

Z:\>C:

C:\>cd MASM

C:\MASM>debug
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100  NU UP EI PL NZ NA PO NC
073F:0100 0000          ADD     [BX+SI],AL          DS:0000=CD
-r bx
BX 0000
:0020
-r ip
IP 0100
:0200
-r
AX=0000 BX=0020 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0200  NU UP EI PL NZ NA PO NC
073F:0200 0000          ADD     [BX+SI],AL          DS:0020=FF
-

```

5. Redisplay the contents of all the internal registers. Compare the displayed register contents with those observed in step 1 above. What instruction is now pointed by CS: IP?

Examining and modifying the contents of memory

1. Use the DUMP command (D) to display the first 100 bytes of the current data segment.
2. Use the DUMP command (D) to display the first 100 bytes of the code segment starting from the current value of CS: IP.

```
AX=0000 BX=0020 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0200  NU UP EI PL NZ NA PO NC
073F:0200 0000          ADD     [BX+SI],AL          DS:0020=FF
-d
073F:0100 B8 0A 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0110 00 00 00 00 00 00 00 00-00 00 00 00 34 00 ZE 07 .....4...
073F:0120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
073F:0130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
073F:0140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
073F:0150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
073F:0160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
073F:0170 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
-D cs:0200 0263
073F:0200 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0210 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0220 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0230 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0240 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0250 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0260 00 00 00 00 .....
-D Ip=0200 0263
^ Error
-D IP:0200 0263
^ Error
```

3. Use the ENTER command (E) to load locations CS:100, CS:102, and CS:104 with 11, 22, and 33, one-by-one respectively.
4. Use the ENTER command (E) to load five consecutive byte-wide memory locations starting at CS:105 with data "FF".

5. Verify the result of steps 3 and 4 using the DUMP command.

```

073F:0310  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
073F:0320  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0330  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0340  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0350  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0360  00 00 00 00 .....
-D DS:0100
073F:0100  11 00 22 00 33 00 00 00 00-00 00 00 00 00 00 00 ..".3.....
073F:0110  00 00 00 00 00 00 00 00 00-00 00 00 00 34 00 2E 07 .....4...
073F:0120  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0130  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0140  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0150  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0160  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0170  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
-D DS:0000
073F:0000  CD 20 3E A7 00 EA FD FF-AD DE 4F 03 A3 01 8A 03 . >.....0.....
073F:0010  A3 01 17 03 A3 01 92 01-01 01 01 00 02 FF FF FF .....
073F:0020  FF FF FF FF FF FF FF FF-FF FF FF FF 00 00 00 00 .....
073F:0030  00 00 14 00 18 00 3F 07-FF FF FF FF 00 00 00 00 .....?.....
073F:0040  05 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
073F:0050  CD 21 CB 00 00 00 00 00 00-00 00 00 00 20 20 20 .!.....
073F:0060  20 20 20 20 20 20 20 20 20-00 00 00 00 20 20 20 .....
073F:0070  20 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 .....

```

6. Use the FILL command (F) to initialize the 16 storage locations starting at DS:10 with the value "AA", the 16 storage locations starting at address DS:30 with "BB", the 16 storage locations starting at address DS:50 with "CC", and the 16 storage locations starting at address DS:70 with "DD"

7. Verify the result of step 6 using the DUMP command.

```

-F DS:0200 "AA"
-F DS:0210 "AA"
-F DS:0230 "BB"
-F DS:0250 "CC"
-F DS:0270 "DD"
-d
073F:0260          43 43 43 43-43 43 43 43 43 43 43 43 43          CCCCCCCCCCCC
073F:0270  44 44 44 44 44 44 44 44-44 44 44 44 44 44 44  DDDDDDDDDDDDDDDDD
073F:0280  44 44 44 44 44 44 44 44-44 44 44 44 44 44 44  DDDDDDDDDDDDDDDDD
073F:0290  44 44 44 44 44 44 44 44-44 44 44 44 44 44 44  DDDDDDDDDDDDDDDDD
073F:02A0  44 44 44 44 44 44 44 44-44 44 44 44 44 44 44  DDDDDDDDDDDDDDDDD
073F:02B0  44 44 44 44 44 44 44 44-44 44 44 44 44 44 44  DDDDDDDDDDDDDDDDD
073F:02C0  44 44 44 44 44 44 44 44-44 44 44 44 44 44 44  DDDDDDDDDDDDDDDDD
073F:02D0  44 44 44 44 44 44 44 44-44 44 44 44 44 44 44  DDDDDDDDDDDDDDDDD
073F:02E0  44 44 44 44          DDDD
-D DS:0210
073F:0210  41 41 41 41 41 41 41 41-41 41 41 41 41 41 41  AAAAAAAAAAAAAAAAAA
073F:0220  41 41 41 41 41 41 41 41-41 41 41 41 41 41 41  AAAAAAAAAAAAAAAAAA
073F:0230  42 42 42 42 42 42 42 42-42 42 42 42 42 42 42  BBBBBBBBBBBBBBBBBB
073F:0240  42 42 42 42 42 42 42 42-42 42 42 42 42 42 42  BBBBBBBBBBBBBBBBBB
073F:0250  43 43 43 43 43 43 43 43-43 43 43 43 43 43 43  CCCCCCCCCCCCCCCC
073F:0260  43 43 43 43 43 43 43 43-43 43 43 43 43 43 43  CCCCCCCCCCCCCCCC
073F:0270  44 44 44 44 44 44 44 44-44 44 44 44 44 44 44  DDDDDDDDDDDDDDDDD
073F:0280  44 44 44 44 44 44 44 44-44 44 44 44 44 44 44  DDDDDDDDDDDDDDDDD

```

Coding Instructions in Debug

1. Use Assemble command to write following assembly instructions at address CS:0100

MOV AX,1441

MOV BX,45

NEG BX

INC AX

ADD AX,BX

SUB AX,AAAA

DEC AX

INT 3

2. Use Un-assemble command to fill in the following table.

Instruction	Machine Code	Number of Bytes Required	Starting Address
MOV AX,1441	B84114	3	073F:0100
MOV BX,45	BB4500	3	073F:0103
NEG BX	F7DB	2	073F:0106
INC AX	43	1	073F:0108
ADD AX,BX	01DB	2	073F:0109
SUB AX,AAAA	ZDAAAA	3	073F:010B
DEC AX	48	1	073F:010E
INT 3	CC	1	073F:010F

Why starting addresses are not consecutive?

Because for every instruction there is different byte allocation when i.p set the start address

Like MOV has 3 bytes memory so if we start from 0100 the for next instruction the address will be used 0103 as start address of 2nd instruction..

Due to different allocation of bytes depending upon instructions .

3. Execute the above program one instruction at a time using the TRACE command(T).
Observe how the values change for registers: AX, BX, flag register, and IP.

```

073F:0106 MOV BX,45
073F:0109 NEG BXQ
          ^ Error
073F:0109 Q
          ^ Error
073F:0109
-Q

C:\MASM>DEBUG
-A
073F:0100 MOV AX,1441
073F:0103 MOV BX,45
073F:0106 NEG BX
073F:0108 INC AX
073F:0109 ADD AX,BX
073F:010B SUB AX,AAAA
073F:010E DEC AX
073F:010F INT 3
073F:0110
-T

AX=1441 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103  NU UP EI PL NZ NA PO NC
073F:0103 BB4500          MOV     BX,0045
-
```

AX=1441,BX=0045,IP=0103,

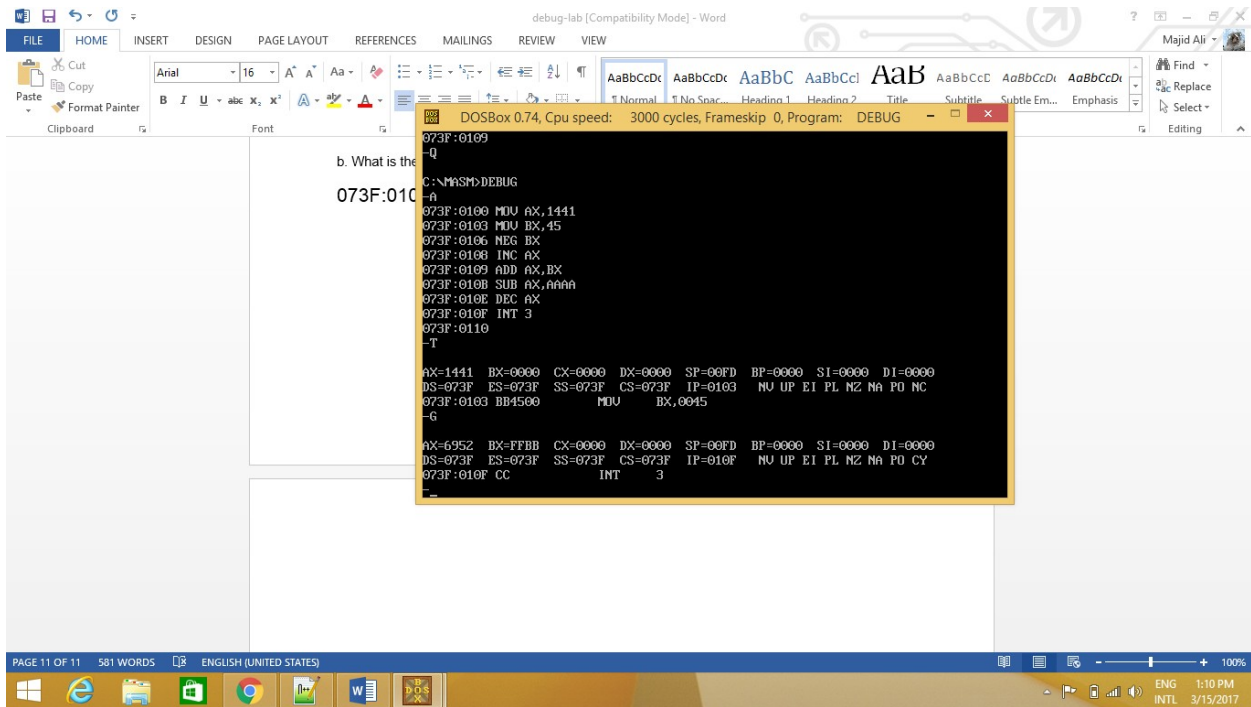
6. Run the complete program by issuing a single GO command (G).

a. What is the starting address for this command?

073F:0100

b. What is the ending address for this command?

073F:010F



7. Check the contents of 16 memory locations starting from address DS:300.

8. Compare the values of these locations with the values in locations starting from DS:200.

