1.) Exchanging with direct addressing


```
-a 100
0859:0100 mov al,[0200]
0859:0103 mov bl,[0210]
0859:0107 mov [0200],bl
0859:010B mov [0201],al
0859:010E
-g 100
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=0100 NV UP EI NG NZ NA PE NC
0859:0100 A00002          MOV     AL,[0200]              DS:0200=00
-_
```

Exchanging with register relative addressing


```
-a 100
0859:0100 mov bp, 200
0859:0103 mov al, 0[bp]
0859:0106 mov bl, 10[bp]
0859:0109 mov 0[bp], bl
0859:010C mov 10[bp], al
0859:010F
-g 100
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=0100 NV UP EI NG NZ NA PE NC
0859:0100 BD0002          MOV     BP,0200
-_
```

Exchanging with Base-Indexed addressing


```
-a 100
0859:0100 mov bp, 200
0859:0103 mov si, 0
0859:0106 mov al, [bp+si]
0859:0108 mov bl, [bp+si+10]
0859:010B mov [bp+si], bl
0859:010D mov [bp+si+10], al
0859:0110
-g 100
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=0100 NV UP EI NG NZ NA PE NC
0859:0100 BD0002          MOV     BP,0200
-_
```

Using E command to add data

Tracing the program

Before and after the swap:

Exchanging 16 bit values:


```
-a 100
0859:0100 mov ax,[200]
0859:0103 mov bx,[210]
0859:0107 mov [200],bx
0859:010B mov [210],ax
0859:010E
```

Executing for 16 bits


```
-r ip 100
-t
AX=3412 BX=0042 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=0103 OV UP EI NG NZ NA PO NC
0859:0103 8B1E1002          MOV     BX,[0210]                  DS:0210=7856
-t
AX=3412 BX=7856 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=0107 OV UP EI NG NZ NA PO NC
0859:0107 891E0002          MOV     [0200],BX                  DS:0200=3412
-t
AX=3412 BX=7856 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=010B OV UP EI NG NZ NA PO NC
0859:010B A31002            MOV     [0210],AX                  DS:0210=7856
-t
AX=3412 BX=7856 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=010E OV UP EI NG NZ NA PO NC
0859:010E 0000              ADD     [BX+SI],AL                 DS:7856=00
-
```

Values before and after


```
-d 200
0859:0200  12 34 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .4..............
0859:0210  56 78 00 00 00 00 00 00-00 00 00 00 00 00 00 00  Vx..............
0859:0220  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0230  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0240  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0250  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0260  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0270  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
-
```

```
-d 200
0859:0200  56 78 00 00 00 00 00 00-00 00 00 00 00 00 00 00  Vx..............
0859:0210  12 34 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .4..............
0859:0220  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0230  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0240  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0250  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0260  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0270  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
-
```

Exchanging 32 bit values


```
-a 100
0859:0100 mov eax,[200]
0859:0104 mov ebx,[210]
0859:0109 mov [200],ebx
0859:010E mov [210],eax
0859:0112
-
```

Executing for 32 bits


```
-r ip 100
-t
AX=3412 BX=7856 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=0104 OV UP EI NG NZ NA PO NC
0859:0104 668B1E2002      MOV     EBX,[0220]              DS:0220=00000000
-t
AX=3412 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=0109 OV UP EI NG NZ NA PO NC
0859:0109 66891E0002      MOV     [0200],EBX              DS:0200=78563412
-t
AX=3412 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=010E OV UP EI NG NZ NA PO NC
0859:010E 66A32002        MOV     [0220],EAX              DS:0220=00000000
-t
AX=3412 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0859 ES=0859 SS=0859 CS=0859 IP=0112 OV UP EI NG NZ NA PO NC
0859:0112 0000            ADD     [BX+SI],AL              DS:0000=CD
-
```

Values before and after:

Checking and verifying

Values before and after

2


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

-
-
-
-
-
-
-
-
-
-
-
-N c:\new.txt
-L 100
-s 100 L 50 "virus"
0859:0119
-d 100
0859:0100  54 68 69 73 20 69 73 20-64 65 66 69 6E 69 74 65  This is definite
0859:0110  6C 79 20 4E 4F 54 20 61-20 76 69 72 75 73 21 21  ly NOT a virus!!
0859:0120  21 0A 00 00 00 00 00 00-00 00 00 00 00 00 00 00  !...............
0859:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0859:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
-_
```

3


```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip 0, Program: DEBUGX  ×
-
-
-
-
-e DS:200
0859:0200  12.12   34.34   00.
-A 100
0859:0100 mov ax,[200]
0859:0103 mov bx,[200]
0859:0107 mov cx,[200]
0859:010B mov dx,[200]
0859:010F moc si,[200]
         ^ Error
0859:010F mov si,[200]
0859:0113 mov di,[200]
0859:0117
-N c:\newfile.com
-r bx 0
-r cx 100
-w 100
Writing 0100 bytes
-_
```

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip 0, Program: DEBUGX  ×
-w 100
Writing 0100 bytes
-q

C:\MASM611\BIN>debugx c:\NEWFILE.COM
-l 100
-d 100
0863:0100  A1 00 02 8B 1E 00 02 8B-0E 00 02 8B 16 00 02 8B  ................
0863:0110  36 00 02 8B 3E 00 02 00-00 00 00 00 00 00 00 00  6...>...........
0863:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................

0863:0180  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:0190  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:01A0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:01B0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
0863:01F0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
-_
```

```
                DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip 0, Program: DEBUGX  ×
-
-
-
-
-
-
-
-
-
-e 500
0859:0500  00.12    00.34    00.56    00.78    00.
-M 500 L 4 600
-r ds:600
     ^ Error
-d ds:600
0859:0600   12 34 56 78 00 00 00 00-00 00 00 00 00 00 00 00   .4Vx............
0859:0610   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0859:0620   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0859:0630   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0859:0640   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0859:0650   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0859:0660   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0859:0670   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-_
```

```
                DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip 0, Program: DEBUGX  ×
-
-
-
-
-                 The H command gives the sum and
-                 difference of the 2 input numbers.
-
-                 So here, 0171 is summation of 14f and 22
-                 and 012D is the difference between them.
-
-
-
-
-
-
-
-H 14f 22
0171  012D
-_
```

It also keeps repeating the message till the byte limit is reached.