

Executing Computer Instructions

Objective

- **DEBUG Program**
 - DEBUG Commands
 - Rules of DEBUG Commands
 - DEBUG Display
 - Viewing Memory Locations
- **Machine and Assembly Language**
 - Keying in program instructions and data
 - Execute program instructions

DEBUG Program

- **DEBUG Commands**
- **Rules of DEBUG Commands**
- **DEBUG Display**
- **Viewing Memory Locations**

DEBUG Program

DEBUG is a DOS program that is used to

- **view memory**
- **enter machine code or assembly code in memory**
- **enter data**
- **trace code execution**
- **single step tracing**

DEBUG Commands

A: Assemble symbolic instructions into machine code

D: Display contents of memory at a specific address

E: Enter data/instructions into memory

G: Run the program

T: Trace the execution of one instruction

P: Proceed or execute a set of instructions

R: display contents of Registers

W: Save a program onto disk

Q: Quit

Rules of DEBUG Commands

- **Not case insensitive**
- **use colon to specify segment and offset**
- **use hexadecimal numbers**
- **use a space to separate parameters in a command**

The DEBUG Display

Display contents of memory at offset 200H in DS using D command

Address	HEX Representation	ASCII
0F6C:0200	B8 00 42 33 C9 8B D1 CD-21 EB 24 3D 05 00 F9 75	..B3....!.\$=...u
0F6C:0210	03 E9 5B FF BE E7 04 33-C9 2E A1 D1 E4 BB 22 00	..[....3.....".
0F6C:0220	BA 12 01 BF 01 00 CD 21-73 03 E9 42 FF 8B D8 B0!s..B....
0F6C:0230	FF 86 47 18 A2 19 00 C3-50 33 C9 FC AC 41 0A C0	..G....P3...A..
0F6C:0240	75 FA 2B F1 58 C3 73 FD-9C 53 51 56 57 55 06 1E	u.+..X.s..SQVWU..
0F6C:0250	50 52 B4 59 CD 21 59 5B-BA A1 80 3D 41 00 74 04	PR.Y.!Y[...=A.t.
0F6C:0260	8B C3 8B D1 1F 07 5D 5F-5E 59 5B 9D C3 E8 D6 FF]_^Y[.....
0F6C:0270	CB 56 57 51 BF CB D7 33-C9 8B C1 57 AC 3C 00 74	.VWQ...3...W.<.t
-		

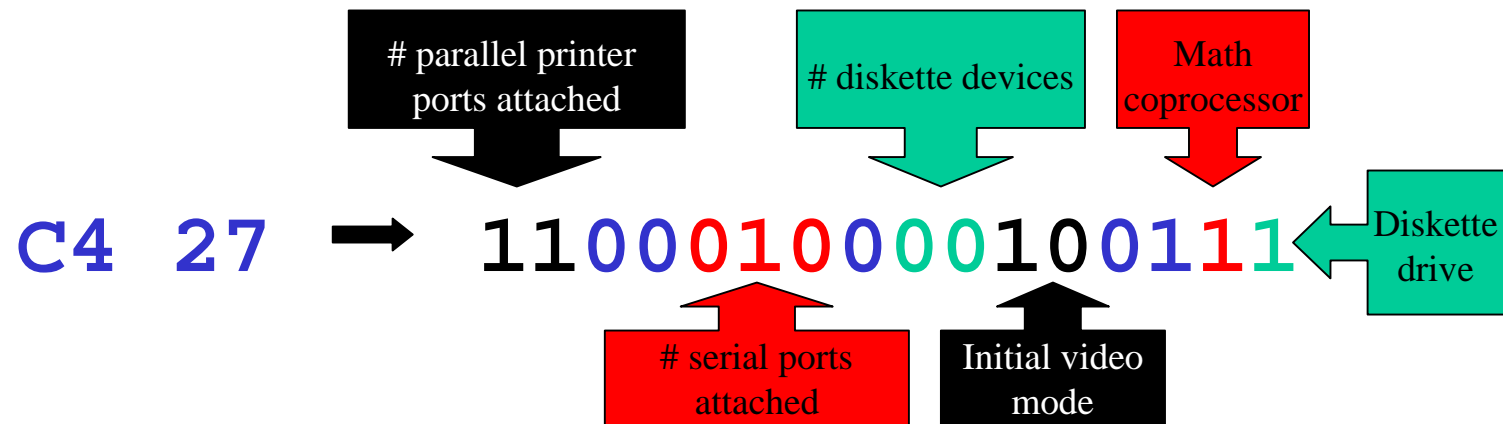
Viewing Memory Locations

- **System equipment**
- **memory size**
- **serial number and copyright notice**
- **ROM BIOS Date**
- **Model ID**

System Equipment

Location of equipment status word in the BIOS data area
is at 410H-411H

```
-d 40:10
0040:0010  27 C4 00 80 02 20 00 00-00 00 32 00 32 00 33 04  '....  ....2.2.3.
0040:0020  0D 1C 64 20 20 39 34 05-30 0B 3A 27 31 02 30 0B  ..d 94.0.: '1.0.
```



Based Memory Size

Size of based memory is at location 413H and 414H

```
-d 40:13
0040:0010      80 02 20 00 00-00 00 22 00 22 00 33 04      .. ....."".3.
0040:0020  0D 1C 08 0E 08 0E 30 0B-30 0B 3A 27 30 0B 0D 1C  .....0.0.: '0...
```

02 80 ↔ 640

Serial Number and Copyright Notice

Computer's serial number is at location FE000H

```
-d fe00:0
FE00:0000  70 E7 AA E7 E5 E7 2E E0-F9 F5 3D F6 00 00 49 42  p.....=...IB
FE00:0010  4D 20 41 54 20 43 6F 6D-70 61 74 69 62 6C 65 20  M AT Compatible
FE00:0020  50 68 6F 65 6E 69 78 20-4E 75 42 49 4F 53 42 E8  Phoenix NuBIOSB.
FE00:0030  04 00 88 46 12 C3 EC 8A-E0 42 EC C3 42 B0 08 EE  ...F.....B..B...
```

ROM BIOS Date

ROM BIOS manufacture date begins at location FFFF5H

```
-d ffff:5
FFFF:0000          31 30 2F-31 36 2F 39 36 00 FC 56          10/16/96..V
FFFF:0010  00 00 00 56 44 49 53 4B-33 2E 33 80 00 01 01 00  ...VDISK3.3.....
```

Computer Model ID

Model ID is at the location FFFF:E

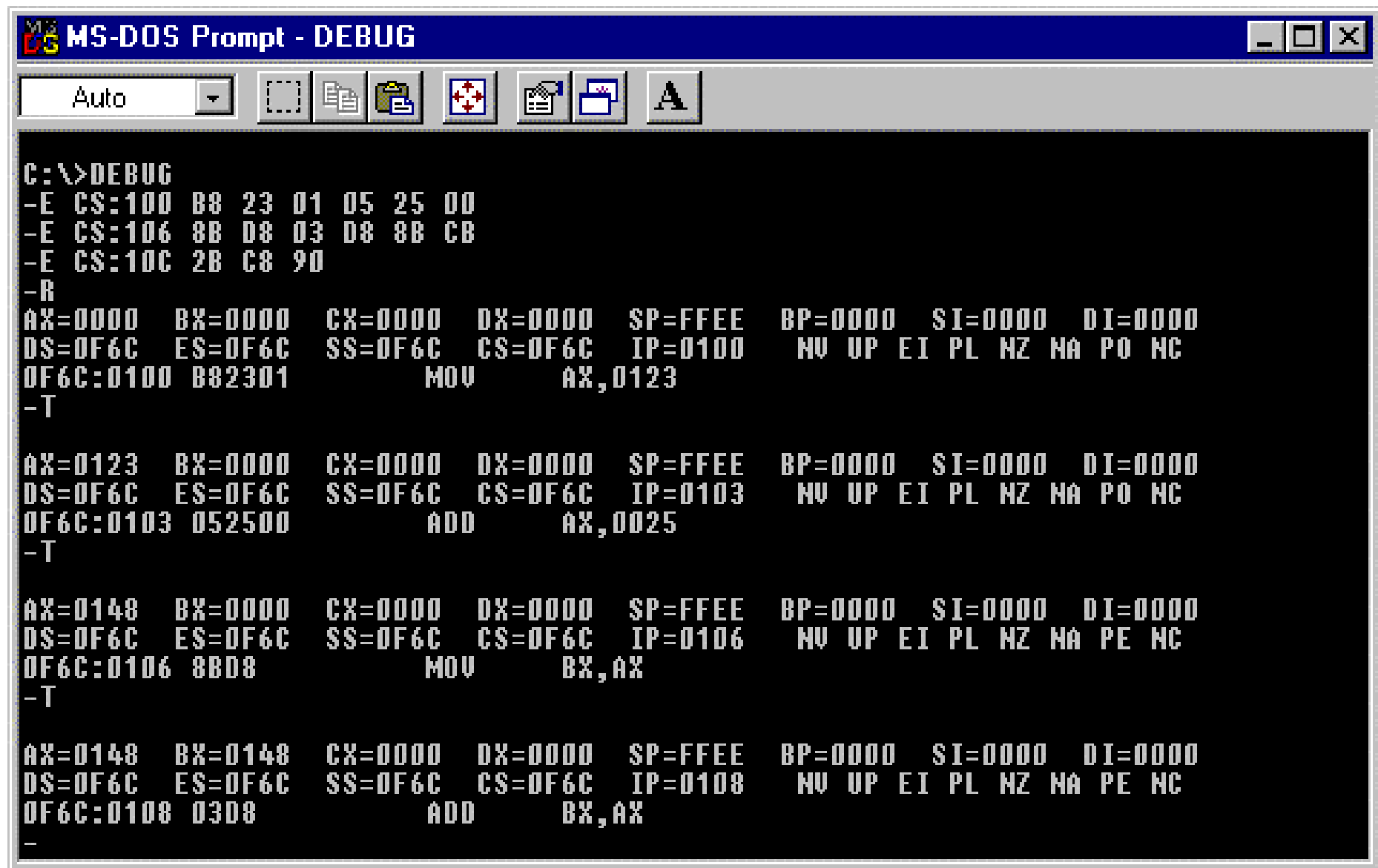
CODE	Model
F8	PS/2 models 70 and 80
FA	PS/2 model 30
FB	PC-XT (1986)
FC	PC-AT (1984), PC-XT model 286, PS/2 models 50 and 60
FE	PC-XT (1982), portable (1982)
FF	Original IBM PC

Machine and Assembly Language

- **Key in program instructions**
- **Execute program instructions**
- **Save a program**

Machine Language Example

Machine Code	Assembly Code
B82301	MOV AX,0123
052500	ADD AX,0025
8BD8	MOVE BX,AX
03D8	ADD BX,AX
8BCB	MOV CX,BX
2BC8	SUB CX,AX
90	NOP



The screenshot shows a Windows-style window titled "MS-DOS Prompt - DEBUG". The window has a menu bar with "Auto" and a dropdown arrow, and a toolbar with icons for file operations and a keyboard icon. The main area is a black console with white text. The text shows the execution of several assembly instructions in the DEBUG program, with register values and status flags displayed before each instruction.

```
C:\>DEBUG
-E CS:100 B8 23 01 05 25 00
-E CS:106 8B D8 03 D8 8B CB
-E CS:10C 2B C8 90
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0100  NV UP EI PL NZ NA PO NC
0F6C:0100 B82301      MOV     AX,0123
-T

AX=0123 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0103  NV UP EI PL NZ NA PO NC
0F6C:0103 052500      ADD     AX,0025
-T

AX=0148 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0106  NV UP EI PL NZ NA PE NC
0F6C:0106 8BD8      MOV     BX,AX
-T

AX=0148 BX=0148 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0108  NV UP EI PL NZ NA PE NC
0F6C:0108 03D8      ADD     BX,AX
-
```


MS-DOS Prompt - DEBUG

Auto

```
AX=0148  BX=0148  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0F6C  ES=0F6C  SS=0F6C  CS=0F6C  IP=0108  NV UP EI PL NZ NA PE NC
0F6C:0108  03D8          ADD     BX,AX
-T

AX=0148  BX=0290  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0F6C  ES=0F6C  SS=0F6C  CS=0F6C  IP=010A  NV UP EI PL NZ AC PE NC
0F6C:010A  8BCB          MOV     CX,BX
-T

AX=0148  BX=0290  CX=0290  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0F6C  ES=0F6C  SS=0F6C  CS=0F6C  IP=010C  NV UP EI PL NZ AC PE NC
0F6C:010C  2BC8          SUB     CX,AX
-T

AX=0148  BX=0290  CX=0148  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0F6C  ES=0F6C  SS=0F6C  CS=0F6C  IP=010E  NV UP EI PL NZ AC PE NC
0F6C:010E  90           NOP
-T

AX=0148  BX=0290  CX=0148  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0F6C  ES=0F6C  SS=0F6C  CS=0F6C  IP=010F  NV UP EI PL NZ AC PE NC
0F6C:010F  C0           DB      C0
-
```

Code Segment Display

```

MS-DOS Prompt - DEBUG
Auto
-D CS:100
0F6C:0100  B8 23 01 05 25 00 8B D8-03 D8 8B CB 2B C8 90 C0  .#..%.....+...
0F6C:0110  90 F9 75 04 3C 3B 75 F6-4E C3 1E 52 34 00 5B 0F  ..u.<;u.N..R4.[.
0F6C:0120  57 2E 8E 1E DE 00 80 3E-43 04 00 75 00 F6 06 21  W.....>C..u...!
0F6C:0130  04 FF 75 06 E8 0B 00 E8-59 00 5F 5E 59 5B 58 5A  ..u.....Y._^Y[XZ
0F6C:0140  1F C3 2E 80 3E 77 E0 00-74 F7 1E 0E 1F BE 77 E0  ....>w..t....w.
0F6C:0150  E8 91 02 2E A1 D1 E4 BB-40 00 BA 01 00 33 FF CD  ....@....3..
0F6C:0160  21 1F 72 0B 8B 08 B0 FF-86 47 18 A2 18 00 C3 DE  !.r.....G.....
0F6C:0170  1F E8 02 00 3D 41 00 74-07 0B FF 74 06 BA 8F 80  ....=A.t...t....
-D CS:100,110
0F6C:0100  B8 23 01 05 25 00 8B D8-03 D8 8B CB 2B C8 90 C0  .#..%.....+...
0F6C:0110  90

```

Debug Operations

- **Keying in program instructions**
- **Executing program instructions**
- **Displaying memory contents**
- **Correcting an entry**

Machine Language Example with Defined Data

DS Offset	Hex Contents
0200H	2301H
0202H	2500H
0204H	0000H
0206H	2A2A2AH

Program Instructions with Defined Data

Machine Code	Assembly Code
A10002	MOV AX,[0200]
03060202	ADD AX,[0202]
A30402	MOVE [0204],AX
90	NOP

```

MS-DOS Prompt - DEBUG
Auto
C:\>DEBUG
-E DS:200 23 10 25 00 00 00
-E DS:206 2A 2A 2A
-E CS:100 A1 00 02 03 06 02 02
-E CS:107 A3 04 02 90
-D DS:200,208
0F6C:0200 23 10 25 00 00 00 2A 2A-2A          #.%.***
-
-D CS:100,10A
0F6C:0100 A1 00 02 03 06 02 02 A3-04 02 90      .....
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0100  NV UP EI PL NZ NA PO NC
0F6C:0100 A10002          MOV     AX,[0200]          DS:0200=1023
-T
AX=1023 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0103  NV UP EI PL NZ NA PO NC
0F6C:0103 03060202      ADD     AX,[0202]          DS:0202=0025
-T
AX=1048 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0107  NV UP EI PL NZ NA PE NC
0F6C:0107 A30402      MOV     [0204],AX          DS:0204=0000
-T
AX=1048 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=010A  NV UP EI PL NZ NA PE NC
0F6C:010A 90          NOP
-

```

Assembly Language Example

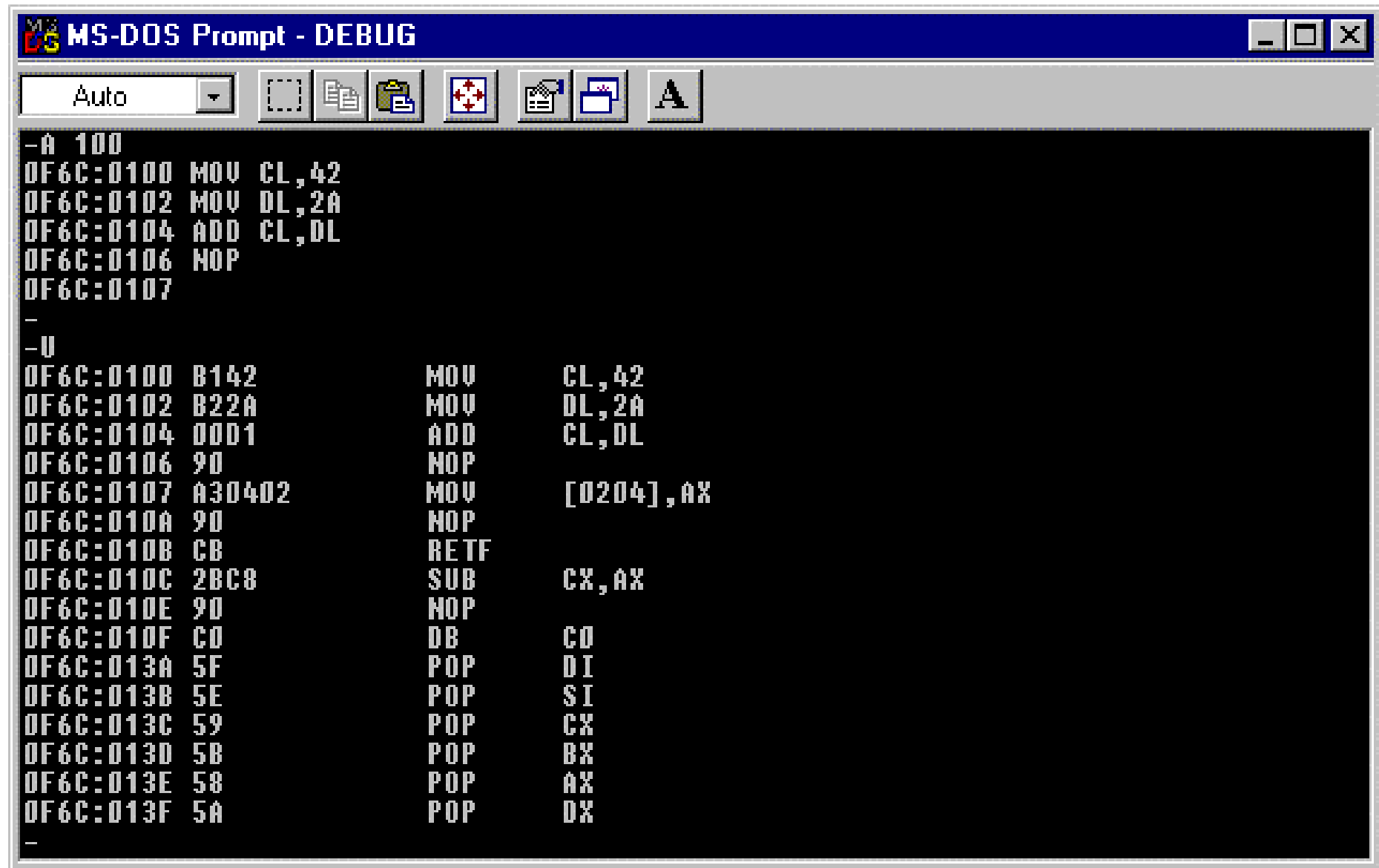
Assemble Command (A)

Use assemble command to key in assembly instructions

```
MOV CL, 42  
MOV DL, 2A  
ADD CL, DL  
NOP
```

Unassemble Command (U)

Unassemble command displays the machine code for assembly language instructions



The screenshot shows a window titled "MS-DOS Prompt - DEBUG". The window has a menu bar with "Auto" and a dropdown arrow, and a toolbar with icons for file operations and a keyboard icon. The main area displays assembly code in two sections. The first section starts with "-A 100" and shows instructions at addresses 0F6C:0100 to 0F6C:0107. The second section starts with "-U" and shows a disassembled view of instructions from 0F6C:0100 to 0F6C:013F.

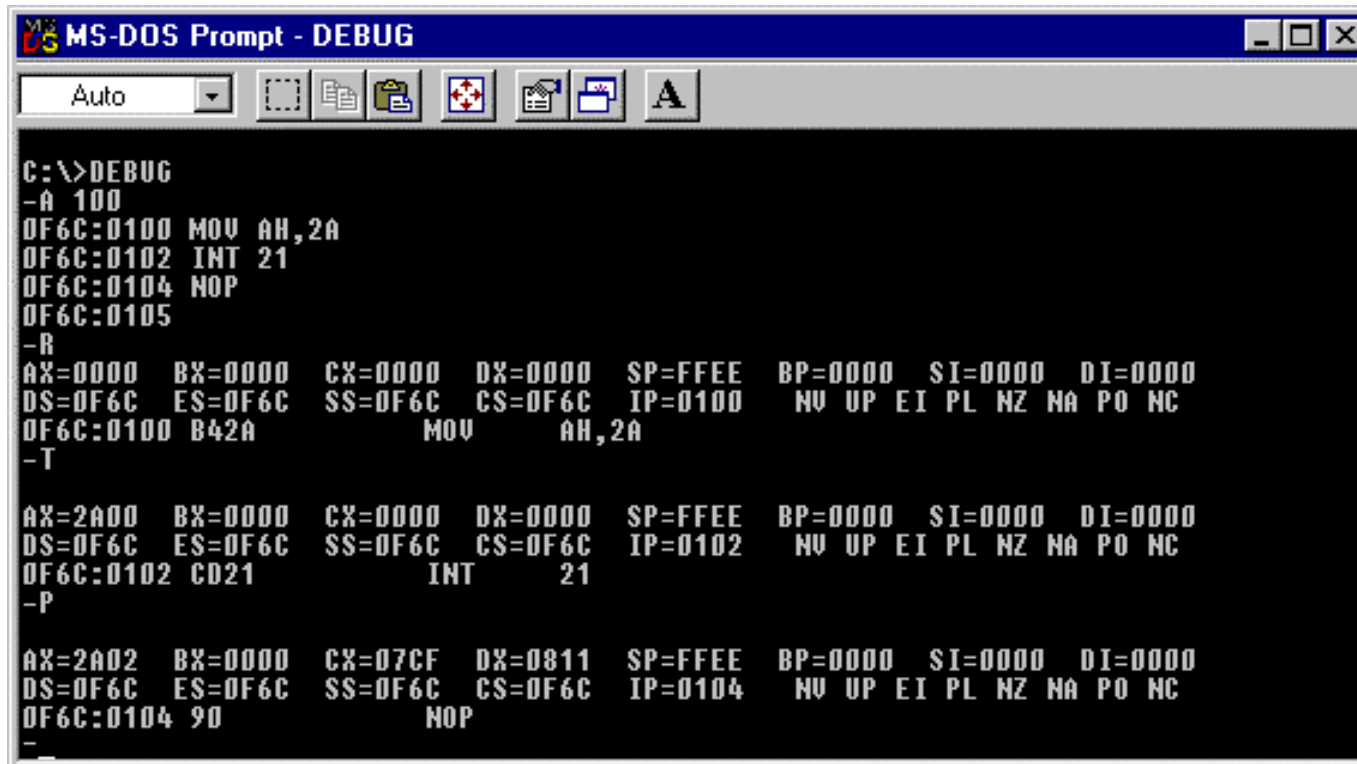
```
-A 100
0F6C:0100 MOV CL,42
0F6C:0102 MOV DL,2A
0F6C:0104 ADD CL,DL
0F6C:0106 NOP
0F6C:0107
-
-U
0F6C:0100 B142          MOV     CL,42
0F6C:0102 B22A          MOV     DL,2A
0F6C:0104 00D1          ADD     CL,DL
0F6C:0106 90            NOP
0F6C:0107 A30402        MOV     [0204],AX
0F6C:010A 90            NOP
0F6C:010B CB            RETF
0F6C:010C 2BC8          SUB     CX,AX
0F6C:010E 90            NOP
0F6C:010F C0            DB     C0
0F6C:013A 5F            POP     DI
0F6C:013B 5E            POP     SI
0F6C:013C 59            POP     CX
0F6C:013D 5B            POP     BX
0F6C:013E 58            POP     AX
0F6C:013F 5A            POP     DX
-
```


Another Example

```
MOV AX, 5  
ADD AX, 10  
ADD AX, 20  
MOV [0102], AX
```

INT (Interrupt) Instruction

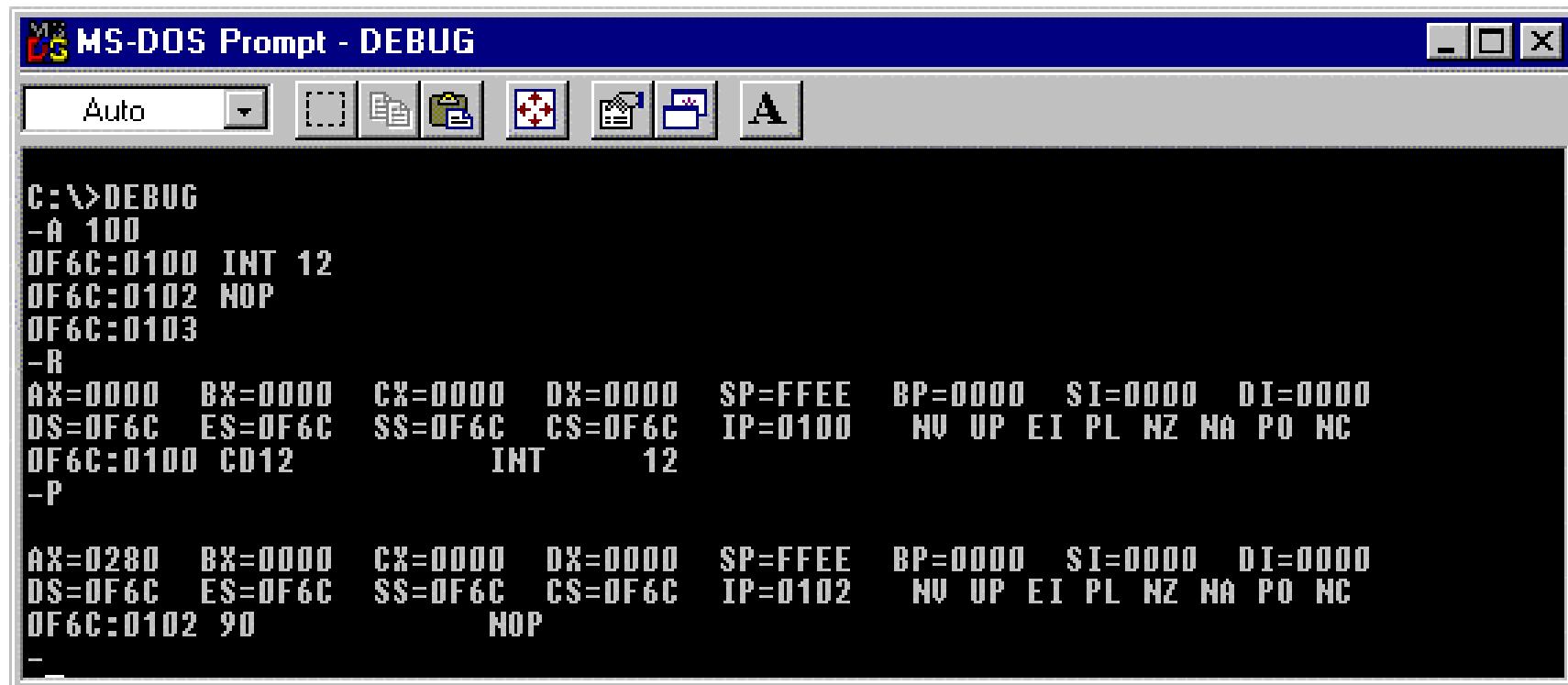
Current date - INT 21H



```
MS-DOS Prompt - DEBUG
Auto
C:\>DEBUG
-A 100
0F6C:0100 MOV AH,2A
0F6C:0102 INT 21
0F6C:0104 NOP
0F6C:0105
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0100  NV UP EI PL NZ NA PO NC
0F6C:0100 B42A      MOV     AH,2A
-T
AX=2A00 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0102  NV UP EI PL NZ NA PO NC
0F6C:0102 CD21      INT     21
-P
AX=2A02 BX=0000 CX=07CF DX=0811 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0104  NV UP EI PL NZ NA PO NC
0F6C:0104 9D      NOP
-
```

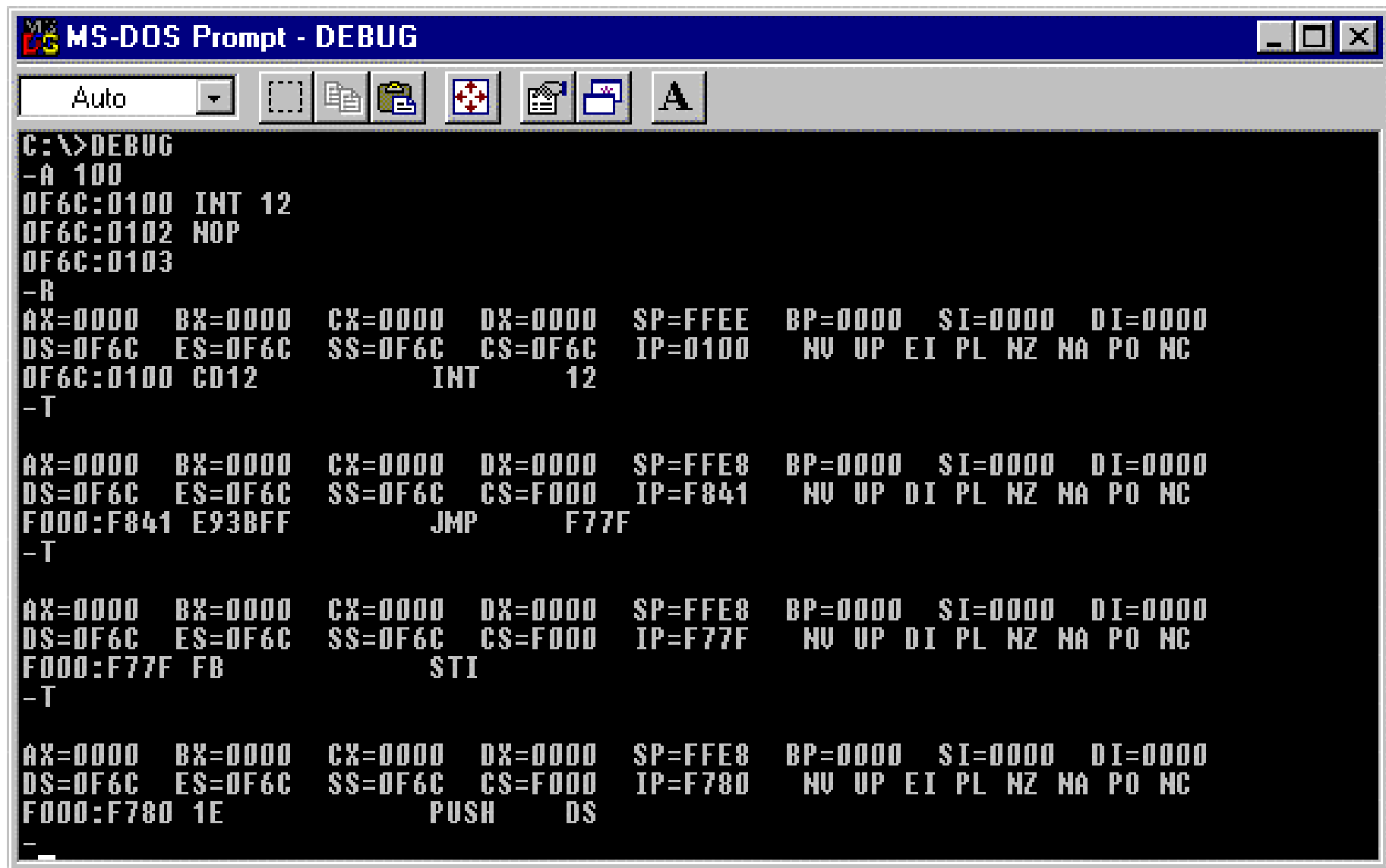
Size of Memory

INT 12H



The screenshot shows a Windows-style window titled "MS-DOS Prompt - DEBUG". The window has a menu bar with "Auto" and a toolbar with icons for file operations and debugging. The command prompt shows the following sequence of commands and their outputs:

```
C:\>DEBUG
-A 100
DF6C:0100 INT 12
DF6C:0102 NOP
DF6C:0103
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=DF6C ES=DF6C SS=DF6C CS=DF6C IP=0100  NV UP EI PL NZ NA PO NC
DF6C:0100 CD12          INT     12
-P
AX=0280 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=DF6C ES=DF6C SS=DF6C CS=DF6C IP=0102  NV UP EI PL NZ NA PO NC
DF6C:0102 90          NOP
-
```



The screenshot shows a Windows-style window titled "MS-DOS Prompt - DEBUG". The window has a menu bar with "Auto" and a toolbar with icons for file operations and a keyboard icon. The main area is a black console with white text showing the execution of assembly instructions in the DEBUG utility.

```
C:\>DEBUG
-A 100
0F6C:0100 INT 12
0F6C:0102 NOP
0F6C:0103
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0100  NV UP EI PL NZ NA PO NC
0F6C:0100 CD12          INT     12
-T

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFE8 BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=F000 IP=F841  NV UP DI PL NZ NA PO NC
F000:F841 E93BFF      JMP     F77F
-T

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFE8 BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=F000 IP=F77F  NV UP DI PL NZ NA PO NC
F000:F77F FB          STI
-T

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFE8 BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=F000 IP=F780  NV UP EI PL NZ NA PO NC
F000:F780 1E          PUSH    DS
-
```

MS-DOS Prompt - DEBUG

Auto

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFE6 BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=F000 IP=F781 NV UP EI PL NZ NA PO NC
F000:F781 2E CS:
F000:F782 8E1E0DC4 MOV DS,[C400] CS:C400=0040
-T

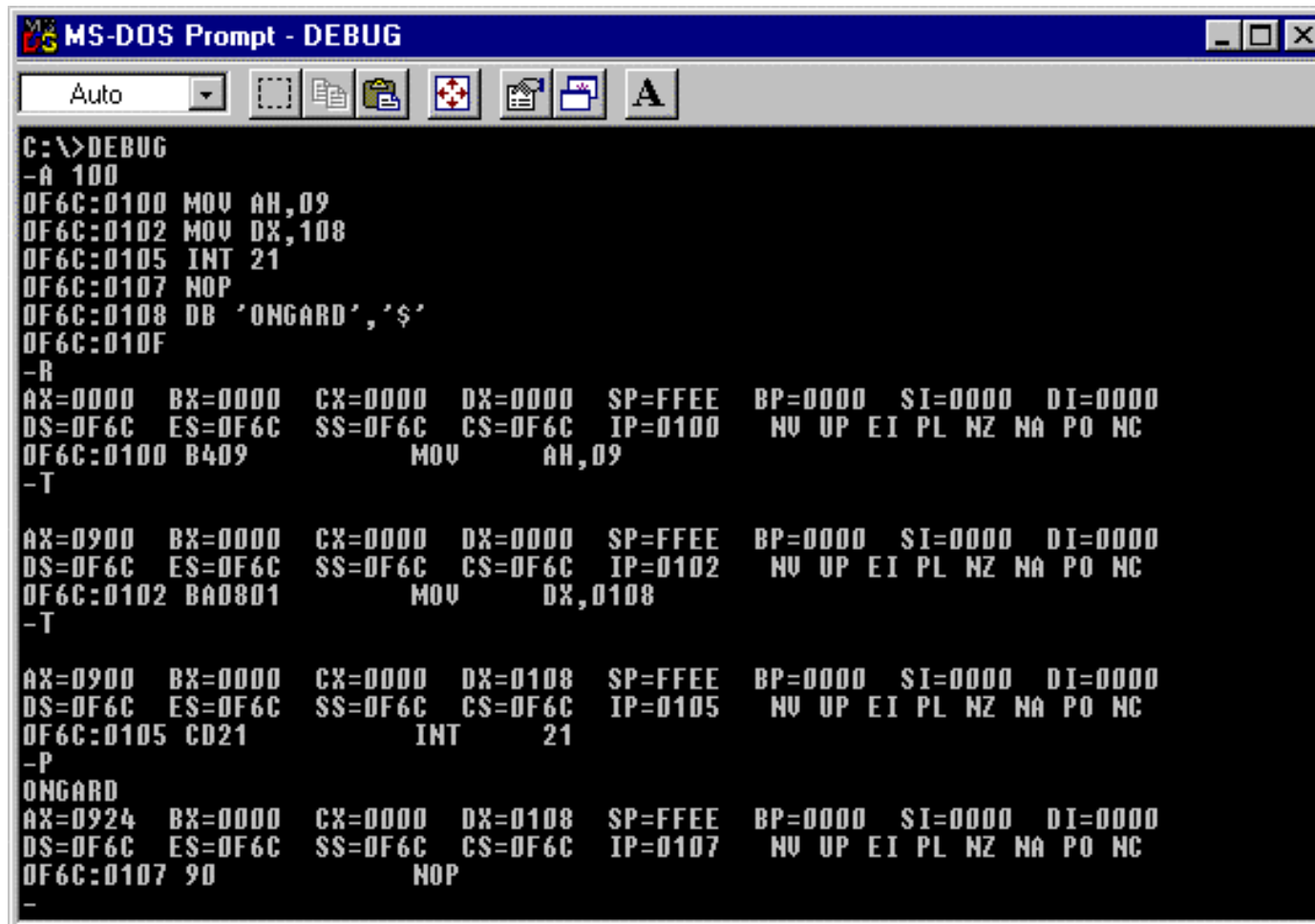
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFE6 BP=0000 SI=0000 DI=0000
DS=0040 ES=0F6C SS=0F6C CS=F000 IP=F786 NV UP EI PL NZ NA PO NC
F000:F786 A11300 MOV AX,[0013] DS:0013=0280
-T

AX=0280 BX=0000 CX=0000 DX=0000 SP=FFE6 BP=0000 SI=0000 DI=0000
DS=0040 ES=0F6C SS=0F6C CS=F000 IP=F789 NV UP EI PL NZ NA PO NC
F000:F789 1F POP DS
-T

AX=0280 BX=0000 CX=0000 DX=0000 SP=FFE8 BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=F000 IP=F78A NV UP EI PL NZ NA PO NC
F000:F78A CF IRET
-T

AX=0280 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0102 NV UP EI PL NZ NA PO NC
0F6C:0102 90 NOP
-

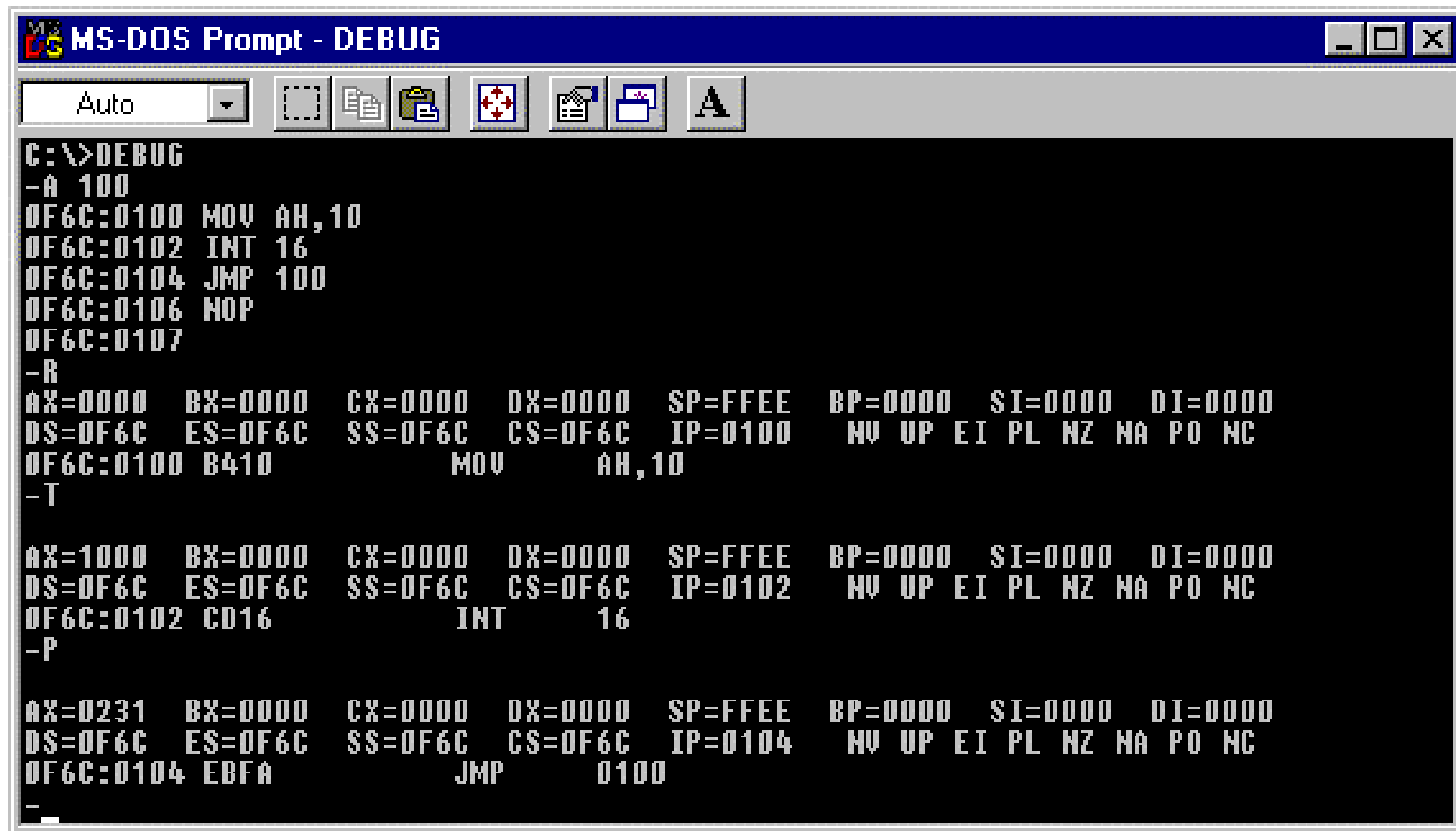
Display with INT



The screenshot shows a Windows-style window titled "MS-DOS Prompt - DEBUG". The window has a menu bar with "Auto" and a toolbar with icons for file operations and a command prompt. The main area is a black console with white text. The text shows the execution of assembly code in the DEBUG utility. The code starts at address 0F6C:0100 and includes instructions like MOV AH,09, MOV DX,108, INT 21, and DB 'ONGARD', '\$'. The output shows the state of registers (AX, BX, CX, DX, SP, BP, SI, DI) and flags (NV, UP, EI, PL, NZ, NA, PO, NC) at various points in the execution. The final output is "ONGARD".

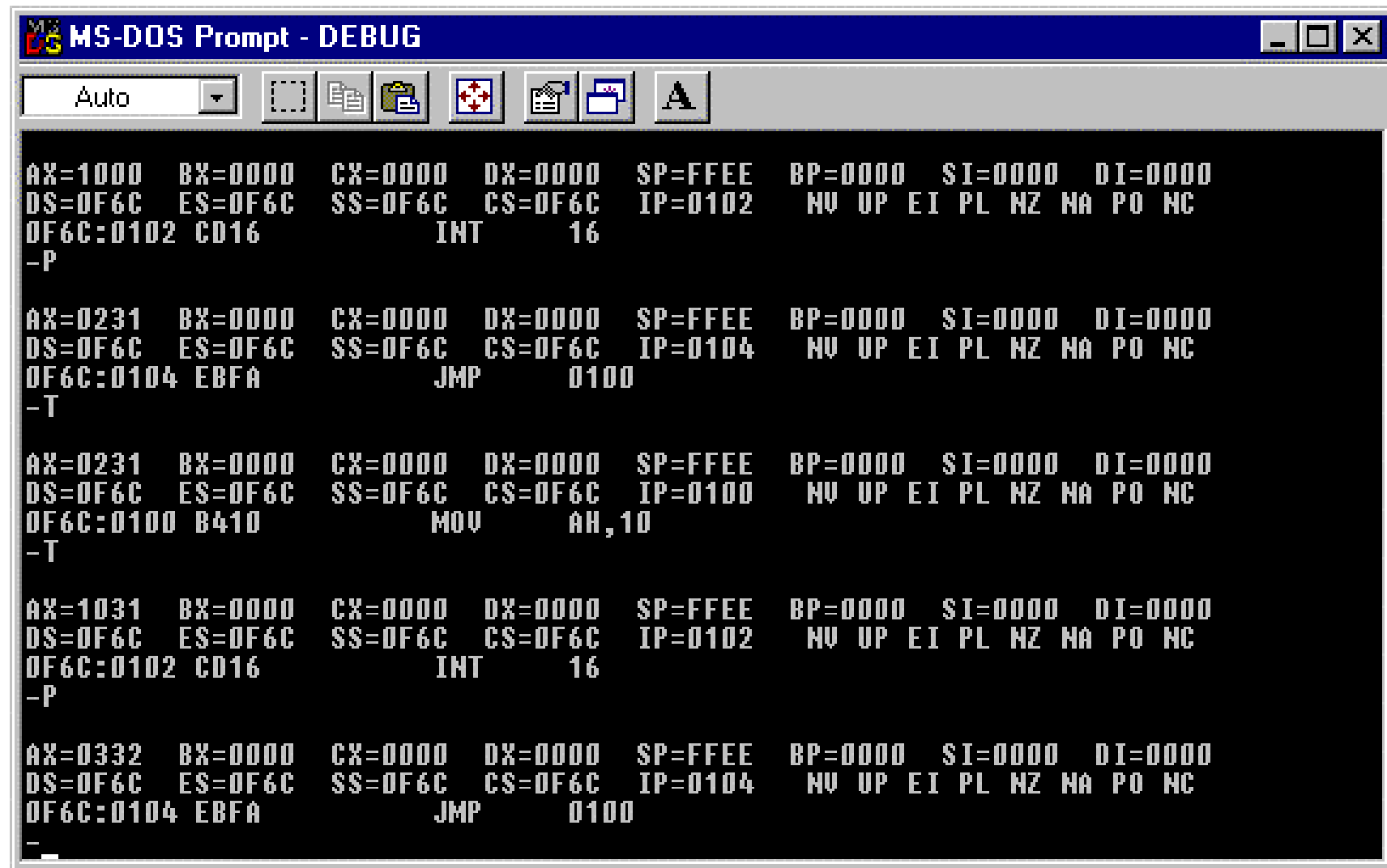
```
C:\>DEBUG
-A 100
0F6C:0100 MOV AH,09
0F6C:0102 MOV DX,108
0F6C:0105 INT 21
0F6C:0107 NOP
0F6C:0108 DB 'ONGARD','$'
0F6C:010F
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0100 NV UP EI PL NZ NA PO NC
0F6C:0100 B409 MOV AH,09
-T
AX=0900 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0102 NV UP EI PL NZ NA PO NC
0F6C:0102 BA0801 MOV DX,0108
-T
AX=0900 BX=0000 CX=0000 DX=0108 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0105 NV UP EI PL NZ NA PO NC
0F6C:0105 CD21 INT 21
-P
ONGARD
AX=0924 BX=0000 CX=0000 DX=0108 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0107 NV UP EI PL NZ NA PO NC
0F6C:0107 90 NOP
-
```

INT for Keyboard Input



The screenshot shows a Windows-style window titled "MS-DOS Prompt - DEBUG". The window has a menu bar with "Auto" and a toolbar with icons for file operations and execution. The main area displays the following text:

```
C:\>DEBUG
-A 100
0F6C:0100 MOV AH,10
0F6C:0102 INT 16
0F6C:0104 JMP 100
0F6C:0106 NOP
0F6C:0107
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0100  NV UP EI PL NZ NA PO NC
0F6C:0100 B410          MOV     AH,10
-T
AX=1000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0102  NV UP EI PL NZ NA PO NC
0F6C:0102 CD16          INT     16
-P
AX=0231 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0104  NV UP EI PL NZ NA PO NC
0F6C:0104 EBFA          JMP     0100
-
```



The screenshot shows the MS-DOS Prompt - DEBUG window. The title bar is blue with the text "MS-DOS Prompt - DEBUG" and standard window controls. Below the title bar is a menu bar with "Auto" and a dropdown arrow, followed by several icons: a grid, a document, a folder, a crosshair, a document with a magnifying glass, a document with a printer, and a large letter 'A'. The main area is a black window with white text displaying assembly instructions and register values. The instructions are:
1. `INT 16` at address `0F6C:0102`, followed by `-P`.
2. `JMP 0100` at address `0F6C:0104`, followed by `-T`.
3. `MOV AH,10` at address `0F6C:0100`, followed by `-T`.
4. `INT 16` at address `0F6C:0102`, followed by `-P`.
5. `JMP 0100` at address `0F6C:0104`, followed by `-`.
The register values are displayed in two columns:
AX=1000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0102
Flags: `NU UP EI PL NZ NA PO NC`

Save Program in DEBUG

- **Create and save a program**
 - use A or E to key in the source code
 - use N filename.COM to name the file
 - clear BX by using the command R BX
 - replace CX with the size of the program in bytes
 - Write or save to a disk with the command W
- **Modify an existing program**
 - type DEBUG filename.com at the DOS prompt
 - use E to make changes
 - use W to save on a disk

PTR Operator

```
100    MOV    AX,[11A]
103    ADD    AX,[11C]
107    ADD    AX,25
10A    MOV    [11E],AX
10D    MOV    WORD PTR [120],25
113    MOV    BYTE PTR [122],30
118    NOP
119    NOP
11A    DB      14    23
11C    DB      05    00
11E    DB      00    00
120    DB      00    00    00
```