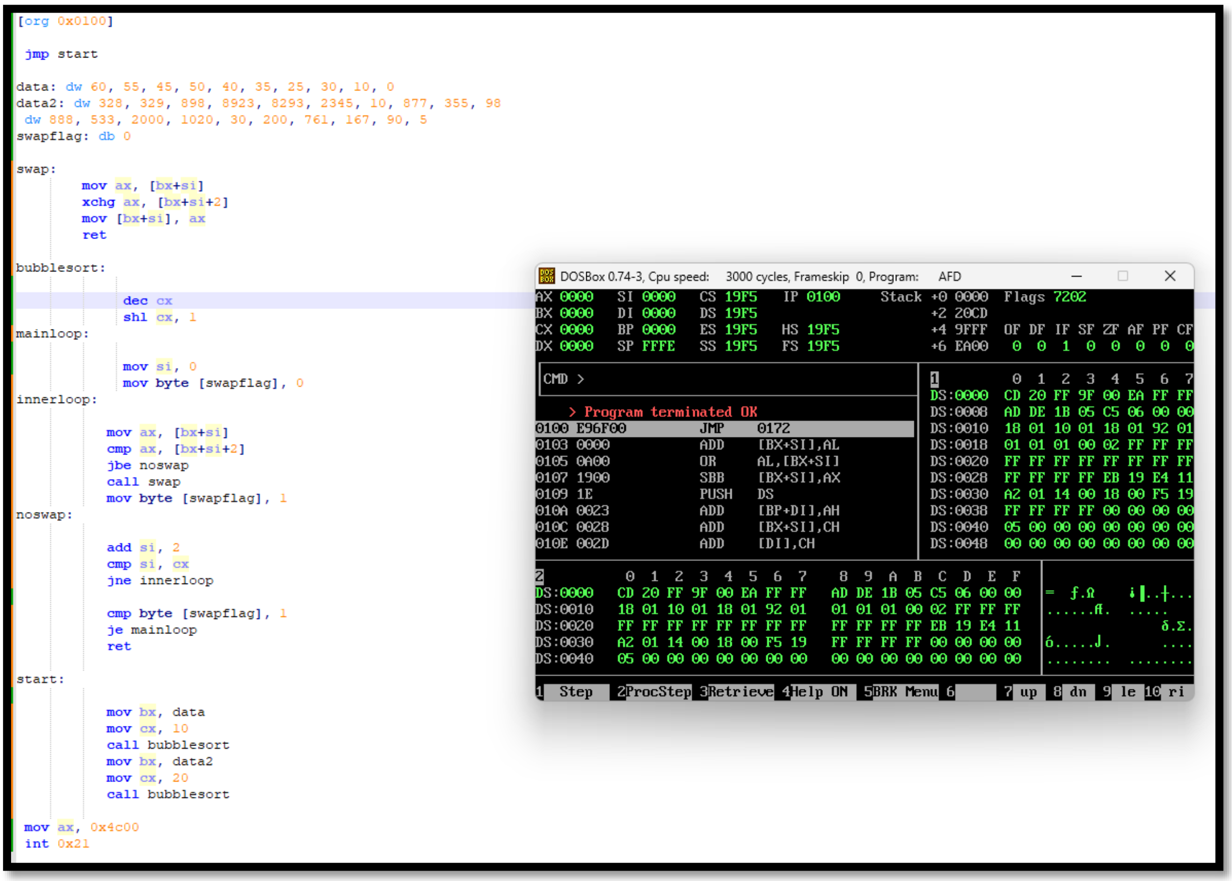
# LAB-10

## Example 5.3

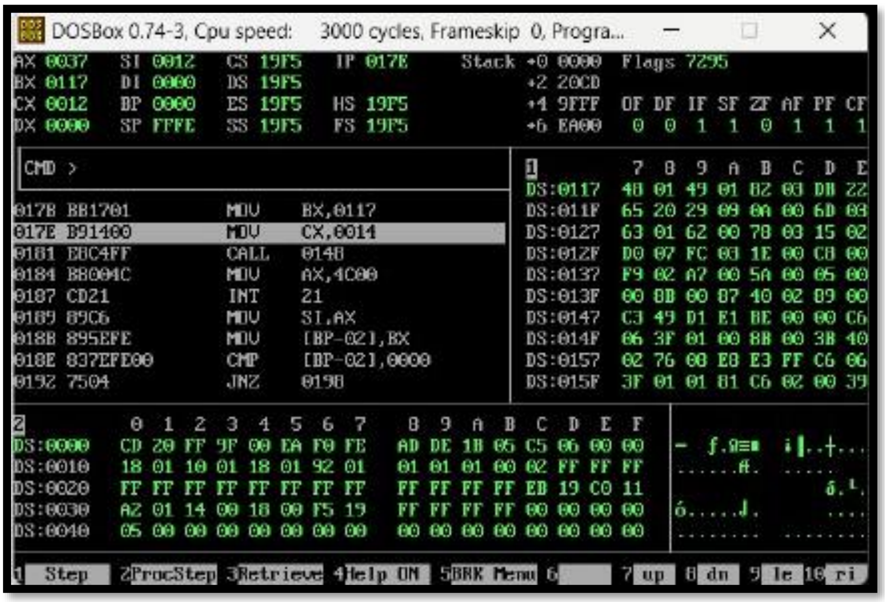


```
[org 0x0100]

 jmp start

data: dw 60, 55, 45, 50, 40, 35, 25, 30, 10, 0
data2: dw 328, 329, 898, 8923, 8293, 2345, 10, 877, 355, 98
 dw 888, 533, 2000, 1020, 30, 200, 761, 167, 90, 5
swapflag: db 0

swap:
        mov ax, [bx+si]
        xchg ax, [bx+si+2]
        mov [bx+si], ax
        ret

bubblesort:

             dec cx
             shl cx, 1
mainloop:

             mov si, 0
             mov byte [swapflag], 0
innerloop:

           mov ax, [bx+si]
           cmp ax, [bx+si+2]
           jbe noswap
           call swap
           mov byte [swapflag], 1
noswap:

           add si, 2
           cmp si, cx
           jne innerloop

           cmp byte [swapflag], 1
           je mainloop
           ret

start:

           mov bx, data
           mov cx, 10
           call bubblesort
           mov bx, data2
           mov cx, 20
           call bubblesort

 mov ax, 0x4c00
 int 0x21
```

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

```
AX 0000   SI 0000   CS 19F5   IP 0100     Stack +0 0000  Flags 7202
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0000   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 >                                             1          0  1  2  3  4  5  6  7
                                                  DS:0000   CD 20 FF 9F 00 EA FF FF
   > Program terminated OK                        DS:0008   AD DE 1B 05 C5 06 00 00
0100 E96F00          JMP     0172                 DS:0010   18 01 10 01 18 01 92 01
0103 0000            ADD     [BX+SI],AL           DS:0018   01 01 01 00 02 FF FF FF
0105 0A00            OR      AL,[BX+SI]           DS:0020   FF FF FF FF FF FF FF FF
0107 1900            SBB     [BX+SI],AX           DS:0028   FF FF FF FF EB 19 E4 11
0109 1E              PUSH    DS                   DS:0030   A2 01 14 00 18 00 F5 19
010A 0023            ADD     [BP+DI],AH           DS:0038   FF FF FF FF 00 00 00 00
010C 0028            ADD     [BX+SI],CH           DS:0040   05 00 00 00 00 00 00 00
010E 002D            ADD     [DI],CH              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 FF FF  AD DE 1B 05 C5 06 00 00   =  ƒ.ß     ↓▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 02 FF FF FF   ......Æ.   .....
DS:0020   FF FF FF FF FF FF FF FF  FF FF FF FF EB 19 E4 11               δ.Σ.
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   ........   ........
```

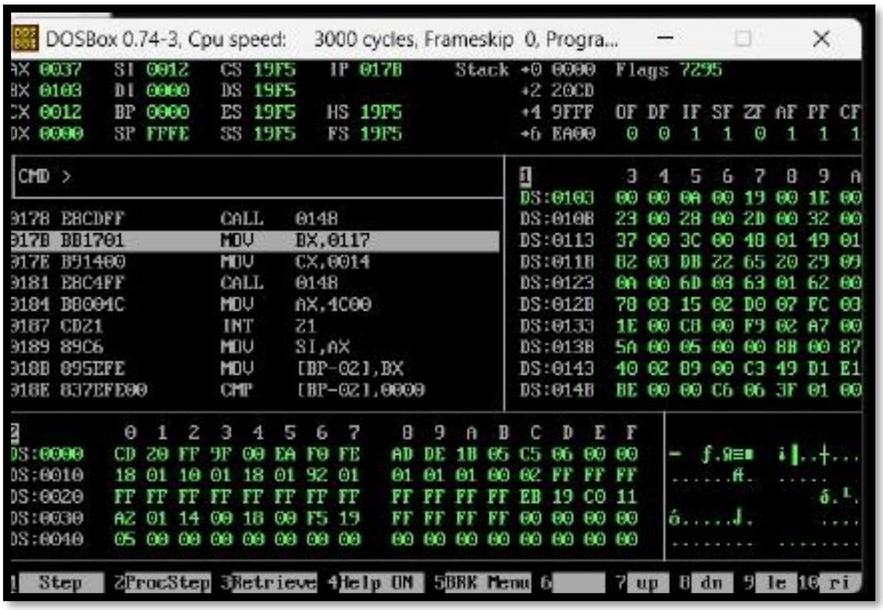
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri



DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Progra...

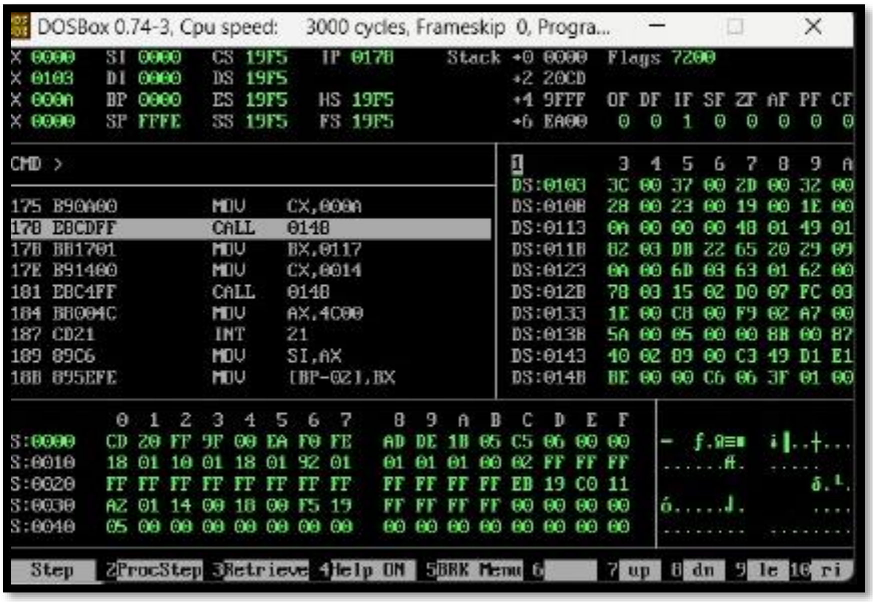
```
AX 0037   SI 0012   CS 19F5   IP 017E     Stack +0 0000  Flags 7295
BX 0117   DI 0000   DS 19F5                     +2 20CD
CX 0012   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  1  0  1  1  1

CMD >                                             1          7  8  9  A  B  C  D  E
                                                  DS:0117   48 01 49 01 B2 03 DB 22
017B BB1701          MOV     BX,0117              DS:011F   65 20 29 09 0A 00 6D 03
017E B91400          MOV     CX,0014              DS:0127   63 01 62 00 78 03 15 02
0181 E8C4FF          CALL    0148                 DS:012F   D0 07 FC 03 1E 00 C8 00
0184 B8004C          MOV     AX,4C00              DS:0137   F9 02 A7 00 5A 00 05 00
0187 CD21            INT     21                   DS:013F   00 8D 00 87 40 02 89 00
0189 89C6            MOV     SI,AX                DS:0147   C3 49 D1 E1 BE 00 00 C6
018B 895EFE          MOV     [BP-02],BX           DS:014F   06 3F 01 00 8B 00 3B 40
018E 837EFE00        CMP     [BP-02],0000         DS:0157   02 76 08 E8 E3 FF C6 06
0192 7504            JNZ     0198                 DS:015F   3F 01 01 81 C6 02 00 39

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   -  ƒ.ß≡■   ↓▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 02 FF FF FF   ......Æ.   .....
DS:0020   FF FF FF FF FF FF FF FF  FF FF FF FF EB 19 C0 11               δ.└.
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



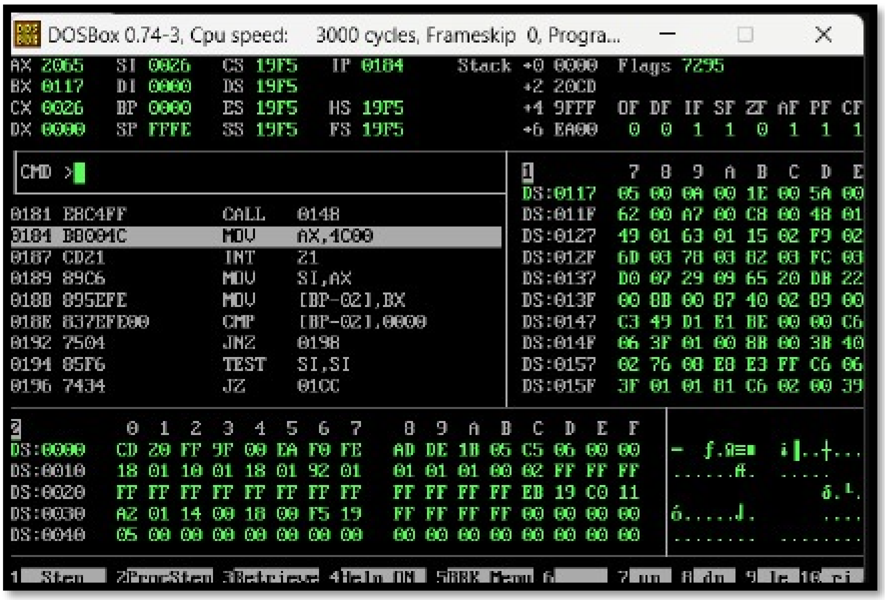
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
AX 0037   SI 0012   CS 19F5   IP 017B      Stack +0 0000  Flags 7295
BX 0103   DI 0000   DS 19F5                      +2 20CD
CX 0012   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
                                             DS:0103    00 00 0A 00 19 00 1E 00
0178 E8CDFF           CALL    0148           DS:010B    23 00 28 00 2D 00 32 00
017B BB1701           MOV     BX,0117        DS:0113    37 00 3C 00 48 01 49 01
017E B91400           MOV     CX,0014        DS:011B    B2 03 DB 22 65 20 29 09
0181 E8C4FF           CALL    0148           DS:0123    0A 00 6D 03 63 01 62 00
0184 B8004C           MOV     AX,4C00        DS:012B    78 03 15 02 D0 07 FC 03
0187 CD21             INT     21             DS:0133    1E 00 C8 00 F9 02 A7 00
0189 89C6             MOV     SI,AX          DS:013B    5A 00 05 00 00 8B 00 87
018B 895EFE           MOV     [BP-02],BX     DS:0143    40 02 89 00 C3 49 D1 E1
018E 837EFE00         CMP     [BP-02],0000   DS:014B    BE 00 00 C6 06 3F 01 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   - ƒ.Ω≡■  ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01   01 01 01 00 02 FF FF FF   ......Æ.  .....
DS:0020   FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 C0 11             δ.└.
DS:0030   A2 01 14 00 18 00 F5 19   FF FF FF FF 00 00 00 00   ó.....⌡.      ....
DS:0040   05 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ........  ........
```



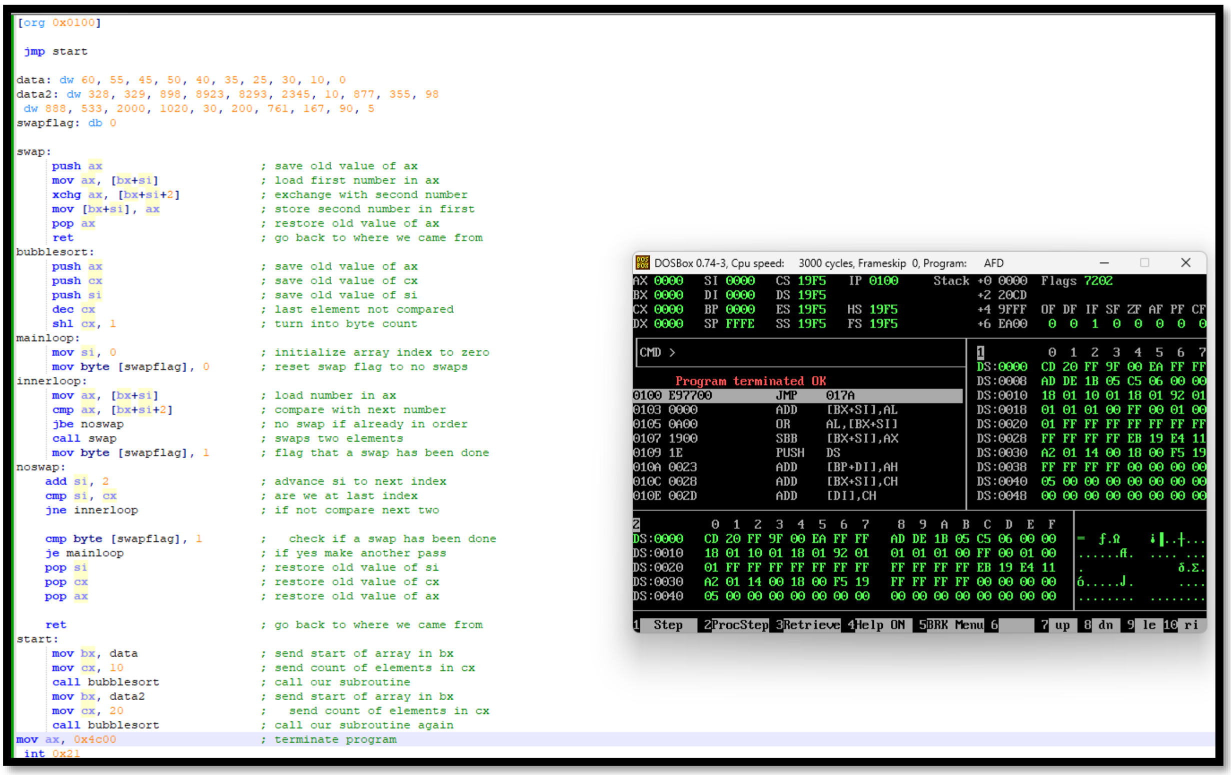
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
X 0000   SI 0000   CS 19F5   IP 0178      Stack +0 0000  Flags 7200
X 0103   DI 0000   DS 19F5                      +2 20CD
X 000A   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
X 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  0  0

CMD >                                       1           3  4  5  6  7  8  9  A
                                            DS:0103    3C 00 37 00 2D 00 32 00
175 B90A00           MOV     CX,000A        DS:010B    28 00 23 00 19 00 1E 00
178 E8CDFF           CALL    0148           DS:0113    0A 00 00 00 48 01 49 01
17B BB1701           MOV     BX,0117        DS:011B    B2 03 DB 22 65 20 29 09
17E B91400           MOV     CX,0014        DS:0123    0A 00 6D 03 63 01 62 00
181 E8C4FF           CALL    0148           DS:012B    78 03 15 02 D0 07 FC 03
184 B8004C           MOV     AX,4C00        DS:0133    1E 00 C8 00 F9 02 A7 00
187 CD21             INT     21             DS:013B    5A 00 05 00 00 8B 00 87
189 89C6             MOV     SI,AX          DS:0143    40 02 89 00 C3 49 D1 E1
18B 895EFE           MOV     [BP-02],BX     DS:014B    BE 00 00 C6 06 3F 01 00

          0  1  2  3  4  5  6  7    8  9  A  B  C  D  E  F
S:0000   CD 20 FF 9F 00 EA F0 FE   AD DE 1B 05 C5 06 00 00   - ƒ.Ω≡■  ¡▌..┼...
S:0010   18 01 10 01 18 01 92 01   01 01 01 00 02 FF FF FF   ......Æ.  .....
S:0020   FF FF FF FF FF FF FF FF   FF FF FF FF EB 19 C0 11             δ.└.
S:0030   A2 01 14 00 18 00 F5 19   FF FF FF FF 00 00 00 00   ó.....⌡.      ....
S:0040   05 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, Progra...
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

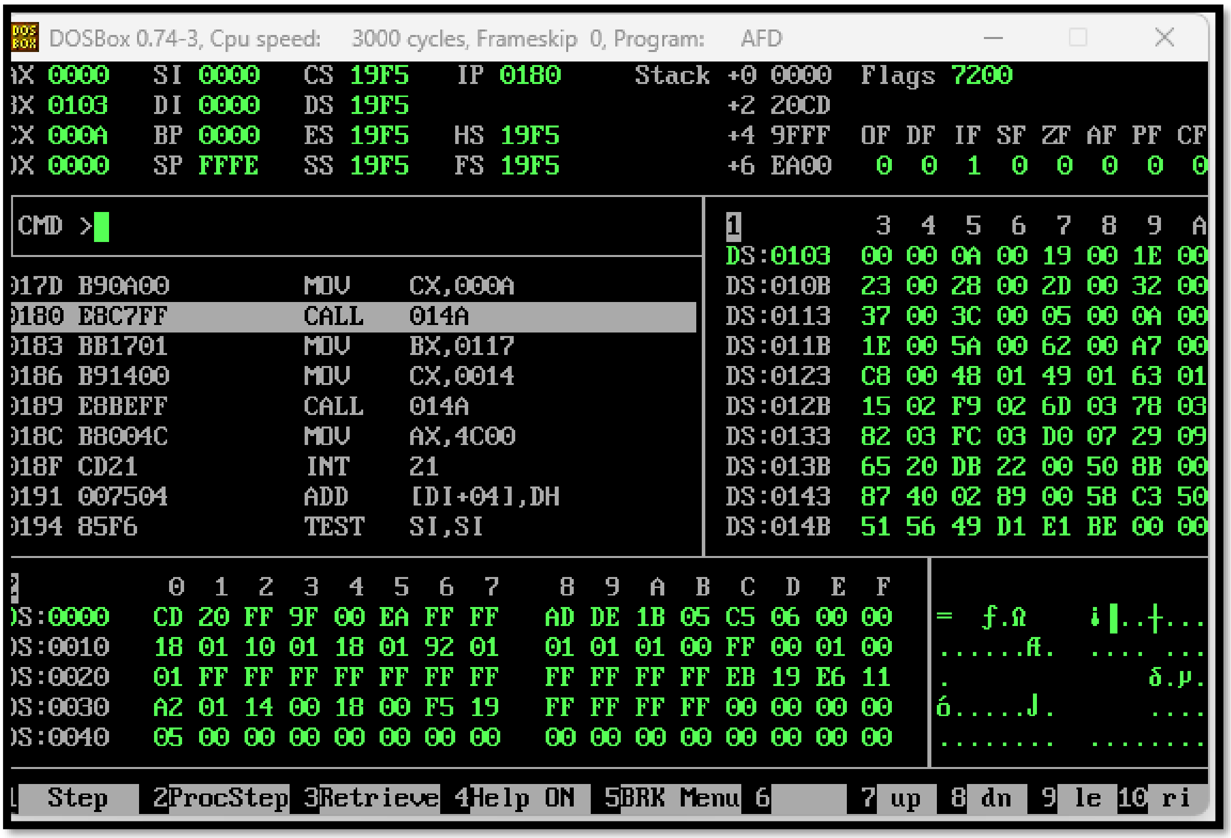
```
AX 2065   SI 0026   CS 19F5   IP 0184      Stack +0 0000  Flags 7295
BX 0117   DI 0000   DS 19F5                      +2 20CD
CX 0026   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  1  0  1  1  1

CMD >                                         1          7  8  9  A  B  C  D  E
                                              DS:0117   05 00 0A 00 1E 00 5A 00
0181 E8C4FF         CALL    0148              DS:011F   62 00 A7 00 C8 00 48 01
0184 B8004C         MOV     AX,4C00           DS:0127   49 01 63 01 15 02 F9 02
0187 CD21           INT     21                DS:012F   6D 03 78 03 82 03 FC 03
0189 89C6           MOV     SI,AX             DS:0137   D0 07 29 09 65 20 DB 22
018B 895EFE         MOV     [BP-02],BX        DS:013F   00 0B 00 07 40 02 89 00
018E 837EFE00       CMP     [BP-02],0000      DS:0147   C3 49 D1 E1 BE 00 00 C6
0192 7504           JNZ     0198              DS:014F   06 3F 01 00 BB 00 3B 40
0194 85F6           TEST    SI,SI             DS:0157   02 76 00 E8 E3 FF C6 06
0196 7434           JZ      01CC              DS:015F   3F 01 01 81 C6 02 00 39

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   = ƒ.Ω≡■  ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 02 FF FF FF   .......Æ.  .....
DS:0020   FF FF FF FF FF FF FF FF  FF FF FF FF EB 19 C0 11             δ.└.
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   ........  ........
```

# Example 5.4

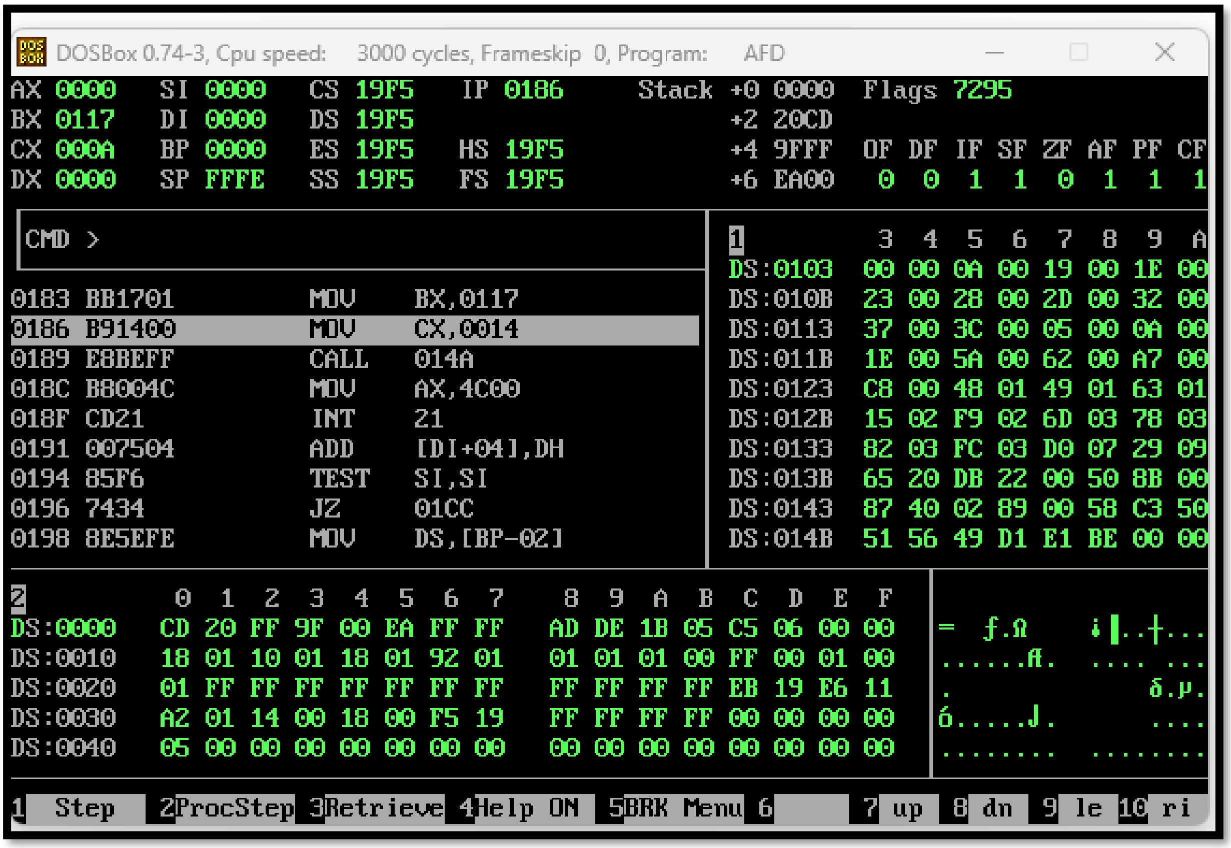

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

```
[org 0x0100]

 jmp start

data: dw 60, 55, 45, 50, 40, 35, 25, 30, 10, 0
data2: dw 328, 329, 898, 8923, 8293, 2345, 10, 877, 355, 98
 dw 888, 533, 2000, 1020, 30, 200, 761, 167, 90, 5
swapflag: db 0

swap:
     push ax                           ; save old value of ax
     mov ax, [bx+si]                   ; load first number in ax
     xchg ax, [bx+si+2]                ; exchange with second number
     mov [bx+si], ax                   ; store second number in first
     pop ax                            ; restore old value of ax
     ret                               ; go back to where we came from
bubblesort:
     push ax                           ; save old value of ax
     push cx                           ; save old value of cx
     push si                           ; save old value of si
     dec cx                            ; last element not compared
     shl cx, 1                         ; turn into byte count
mainloop:
     mov si, 0                         ; initialize array index to zero
     mov byte [swapflag], 0            ; reset swap flag to no swaps
innerloop:
     mov ax, [bx+si]                   ; load number in ax
     cmp ax, [bx+si+2]                 ; compare with next number
     jbe noswap                        ; no swap if already in order
     call swap                         ; swaps two elements
     mov byte [swapflag], 1            ; flag that a swap has been done
noswap:
    add si, 2                          ; advance si to next index
    cmp si, cx                         ; are we at last index
    jne innerloop                      ; if not compare next two

    cmp byte [swapflag], 1             ;   check if a swap has been done
    je mainloop                        ; if yes make another pass
    pop si                             ; restore old value of si
    pop cx                             ; restore old value of cx
    pop ax                             ; restore old value of ax

    ret                                ; go back to where we came from
start:
     mov bx, data                      ; send start of array in bx
     mov cx, 10                        ; send count of elements in cx
     call bubblesort                   ; call our subroutine
     mov bx, data2                     ; send start of array in bx
     mov cx, 20                        ;   send count of elements in cx
     call bubblesort                   ; call our subroutine again
mov ax, 0x4c00                         ; terminate program
 int 0x21
```

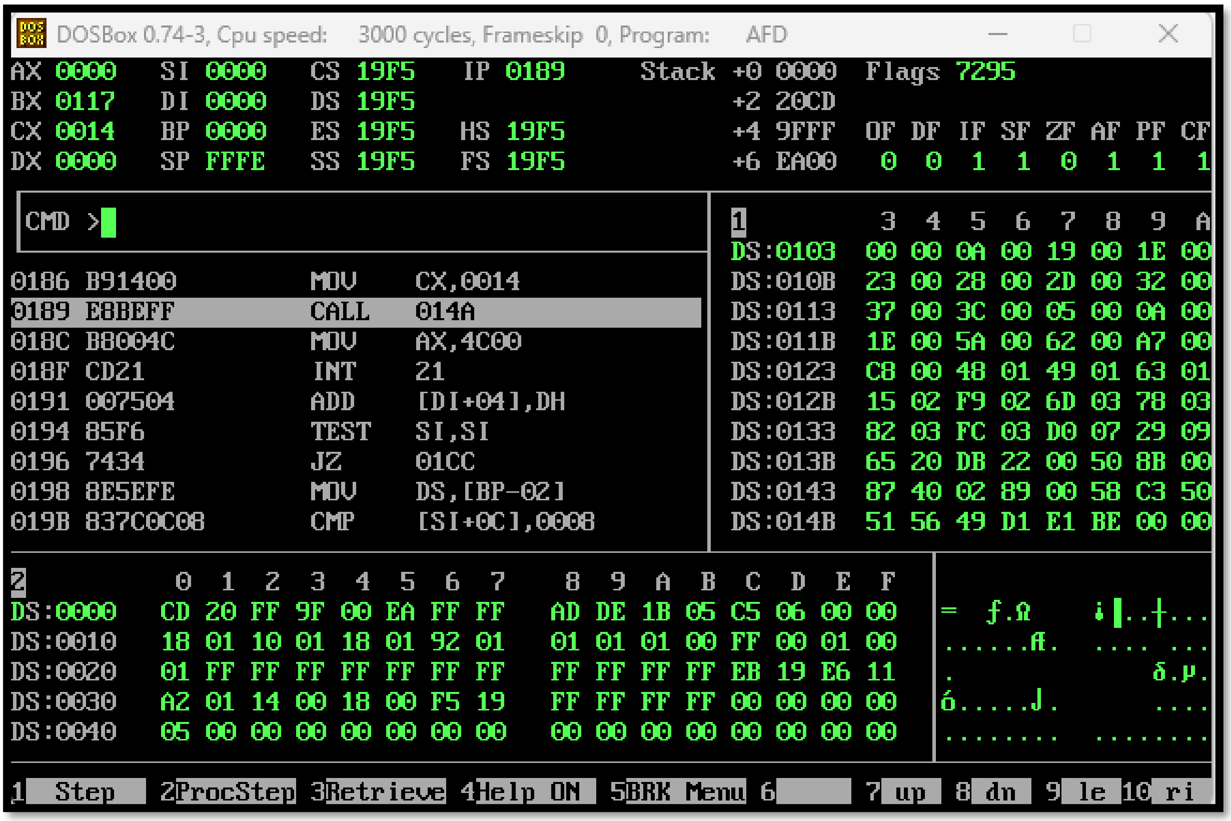
```
AX 0000   SI 0000   CS 19F5   IP 0100      Stack +0 0000  Flags 7202
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  0  0

CMD >                                    1          0  1  2  3  4  5  6  7
                                         DS:0000   CD 20 FF 9F 00 EA FF FF
      Program terminated OK              DS:0008   AD DE 1B 05 C5 06 00 00
0100 E97700         JMP     017A         DS:0010   18 01 10 01 18 01 92 01
0103 0000           ADD     [BX+SI],AL   DS:0018   01 01 01 00 FF 00 01 00
0105 0A00           OR      AL,[BX+SI]   DS:0020   01 FF FF FF FF FF FF FF
0107 1900           SBB     [BX+SI],AX   DS:0028   FF FF FF FF EB 19 E4 11
0109 1E             PUSH    DS           DS:0030   A2 01 14 00 18 00 F5 19
010A 0023           ADD     [BP+DI],AH   DS:0038   FF FF FF FF 00 00 00 00
010C 0028           ADD     [BX+SI],CH   DS:0040   05 00 00 00 00 00 00 00
010E 002D           ADD     [DI],CH      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 FF FF  AD DE 1B 05 C5 06 00 00   =  ƒ.Ω    ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 FF 00 01 00   .......Æ.  .... ...
DS:0020   01 FF FF FF FF FF FF FF  FF FF FF FF EB 19 E4 11   .             δ.Σ.
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   ........  ........
```



```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Program:    AFD
X 0000   SI 0000   CS 19F5   IP 0180      Stack +0 0000  Flags 7200
X 0103   DI 0000   DS 19F5                      +2 20CD
X 000A   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
X 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  0  0

CMD >                                     1         3  4  5  6  7  8  9  A
                                          DS:0103  00 00 0A 00 19 00 1E 00
17D B90A00        MOV    CX,000A          DS:010B  23 00 28 00 2D 00 32 00
180 E8C7FF        CALL   014A             DS:0113  37 00 3C 00 05 00 0A 00
183 BB1701        MOV    BX,0117          DS:011B  1E 00 5A 00 62 00 A7 00
186 B91400        MOV    CX,0014          DS:0123  C8 00 48 01 49 01 63 01
189 E8BEFF        CALL   014A             DS:012B  15 02 F9 02 6D 03 78 03
18C B8004C        MOV    AX,4C00          DS:0133  82 03 FC 03 D0 07 29 09
18F CD21          INT    21               DS:013B  65 20 DB 22 00 50 8B 00
191 007504        ADD    [DI+04],DH       DS:0143  87 40 02 89 00 58 C3 50
194 85F6          TEST   SI,SI            DS:014B  51 56 49 D1 E1 BE 00 00

          0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
S:0000   CD 20 FF 9F 00 EA FF FF  AD DE 1B 05 C5 06 00 00   = ƒ.♫    ↓▌.┼...
S:0010   18 01 10 01 18 01 92 01  01 01 01 00 FF 00 01 00   ......Æ.  .... ...
S:0020   01 FF FF FF FF FF FF FF  FF FF FF FF EB 19 E6 11   .            δ.µ.
S:0030   A2 01 14 00 18 00 F5 19  FF FF FF FF 00 00 00 00   ó.....J.      ....
S:0040   05 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........  ........

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

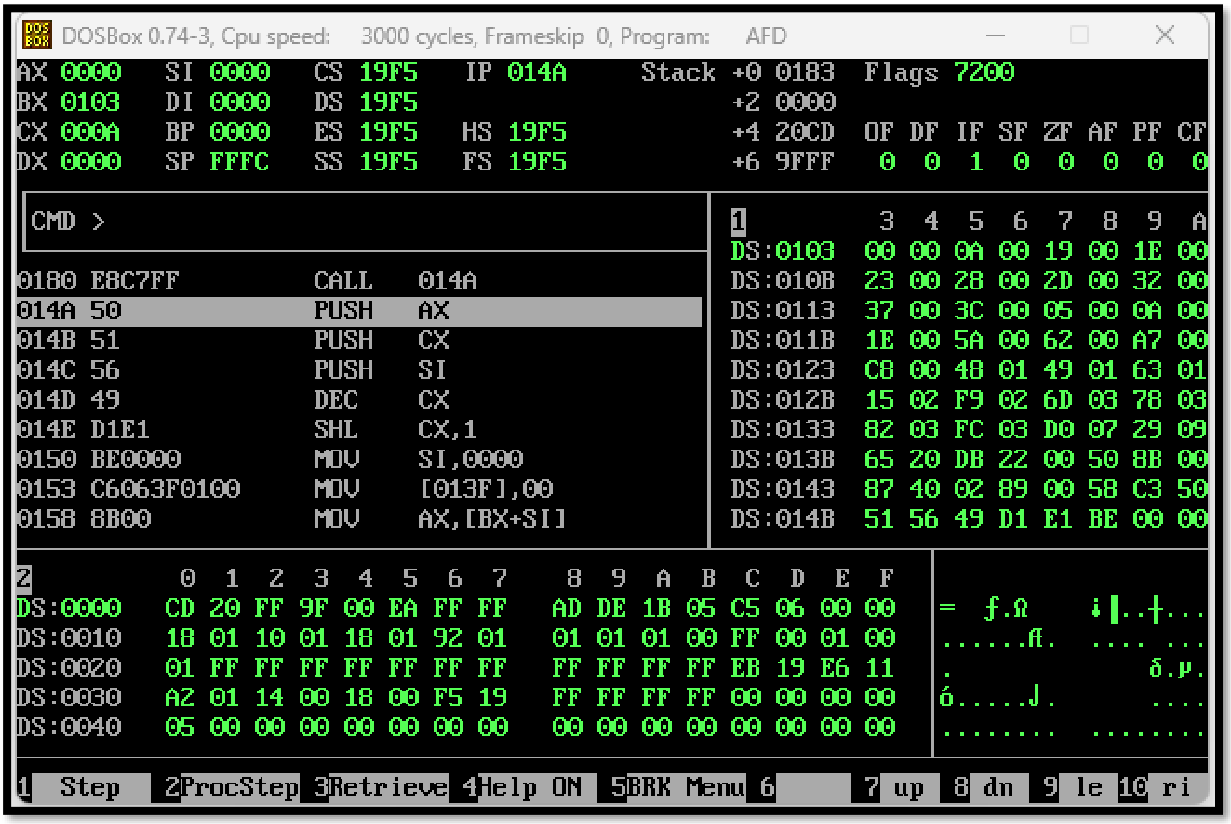


```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Program:    AFD
AX 0000   SI 0000   CS 19F5   IP 0186     Stack +0 0000  Flags 7295
BX 0117   DI 0000   DS 19F5                     +2 20CD
CX 000A   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  1  0  1  1  1

CMD >                                     1         3  4  5  6  7  8  9  A
                                          DS:0103  00 00 0A 00 19 00 1E 00
0183 BB1701        MOV    BX,0117         DS:010B  23 00 28 00 2D 00 32 00
0186 B91400        MOV    CX,0014         DS:0113  37 00 3C 00 05 00 0A 00
0189 E8BEFF        CALL   014A            DS:011B  1E 00 5A 00 62 00 A7 00
018C B8004C        MOV    AX,4C00         DS:0123  C8 00 48 01 49 01 63 01
018F CD21          INT    21              DS:012B  15 02 F9 02 6D 03 78 03
0191 007504        ADD    [DI+04],DH      DS:0133  82 03 FC 03 D0 07 29 09
0194 85F6          TEST   SI,SI           DS:013B  65 20 DB 22 00 50 8B 00
0196 7434          JZ     01CC            DS:0143  87 40 02 89 00 58 C3 50
0198 8E5EFE        MOV    DS,[BP-02]      DS:014B  51 56 49 D1 E1 BE 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 FF FF  AD DE 1B 05 C5 06 00 00   = ƒ.♫    ↓▌.┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 FF 00 01 00   ......Æ.  .... ...
DS:0020   01 FF FF FF FF FF FF FF  FF FF FF FF EB 19 E6 11   .            δ.µ.
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
```


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

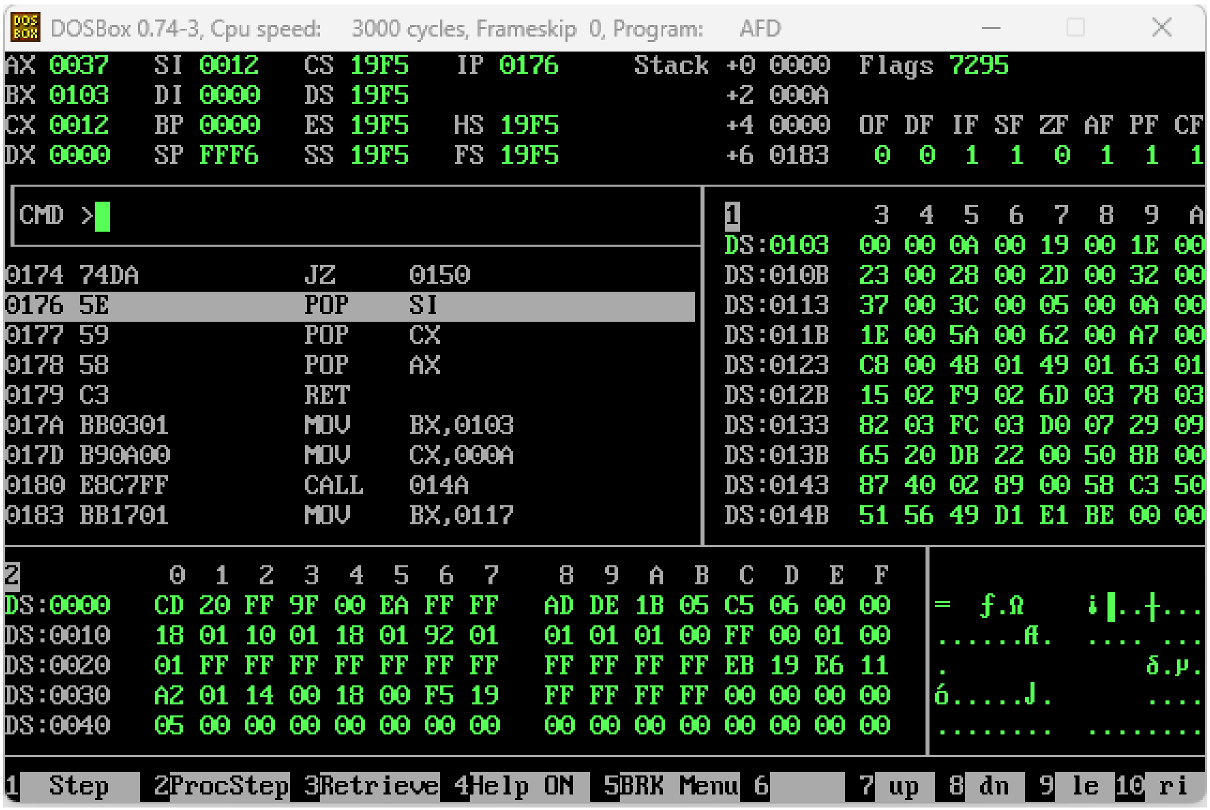
```
AX 0000   SI 0000   CS 19F5   IP 0189     Stack +0 0000  Flags 7295
BX 0117   DI 0000   DS 19F5                     +2 20CD
CX 0014   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  1  0  1  1  1

 CMD >                                        1        3  4  5  6  7  8  9  A
                                              DS:0103  00 00 0A 00 19 00 1E 00
0186 B91400        MOV    CX,0014             DS:010B  23 00 28 00 2D 00 32 00
0189 E8BEFF        CALL   014A                DS:0113  37 00 3C 00 05 00 0A 00
018C B8004C        MOV    AX,4C00             DS:011B  1E 00 5A 00 62 00 A7 00
018F CD21          INT    21                  DS:0123  C8 00 48 01 49 01 63 01
0191 007504        ADD    [DI+04],DH          DS:012B  15 02 F9 02 6D 03 78 03
0194 85F6          TEST   SI,SI               DS:0133  82 03 FC 03 D0 07 29 09
0196 7434          JZ     01CC                DS:013B  65 20 DB 22 00 50 8B 00
0198 8E5EFE        MOV    DS,[BP-02]          DS:0143  87 40 02 89 00 58 C3 50
019B 837C0C08      CMP    [SI+0C],0008        DS:014B  51 56 49 D1 E1 BE 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 FF FF  AD DE 1B 05 C5 06 00 00   = ƒ.Ω   ╡▌..┼...
DS:0010    18 01 10 01 18 01 92 01  01 01 01 00 FF 00 01 00   ......Æ. .... ...
DS:0020    01 FF FF FF FF FF FF FF  FF FF FF FF EB 19 E6 11   .           δ.µ.
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


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

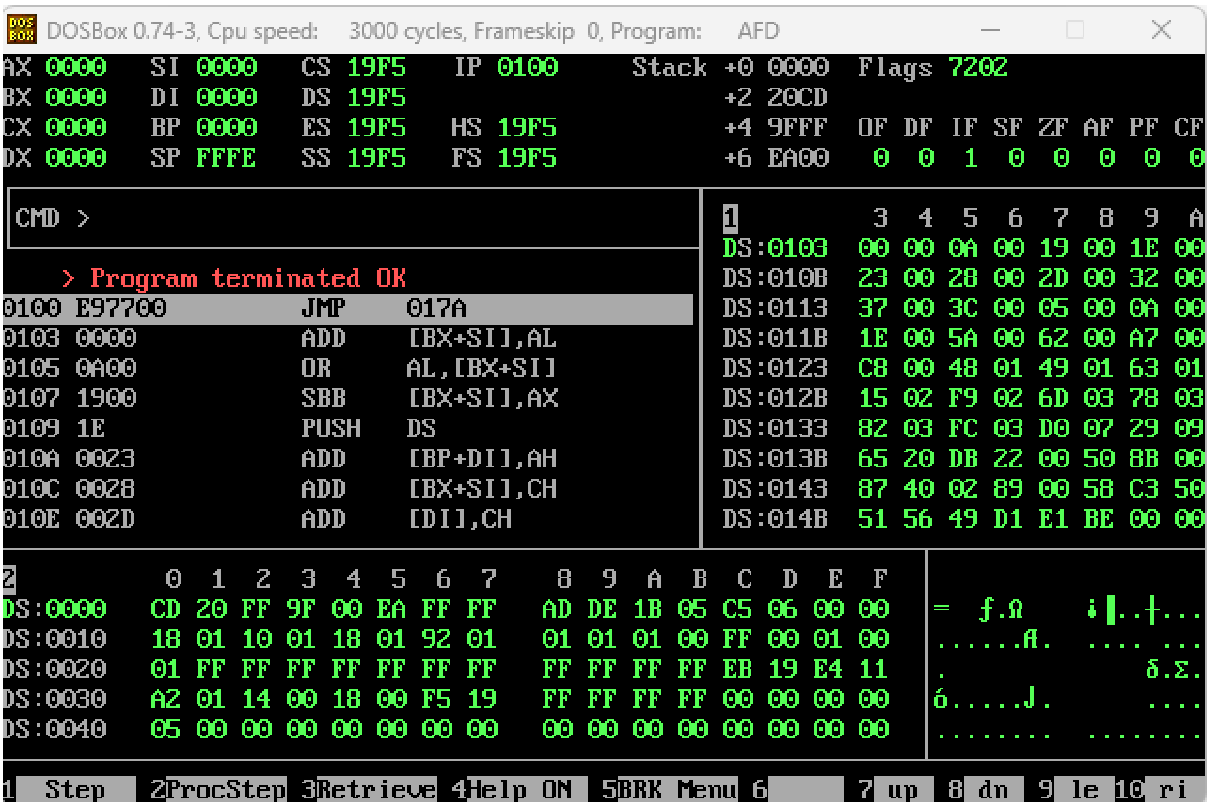
```
AX 0000   SI 0000   CS 19F5   IP 014A     Stack +0 0183  Flags 7200
BX 0103   DI 0000   DS 19F5                     +2 0000
CX 000A   BP 0000   ES 19F5   HS 19F5           +4 20CD  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFC   SS 19F5   FS 19F5           +6 9FFF   0  0  1  0  0  0  0  0

 CMD >                                        1        3  4  5  6  7  8  9  A
                                              DS:0103  00 00 0A 00 19 00 1E 00
0180 E8C7FF        CALL   014A                DS:010B  23 00 28 00 2D 00 32 00
014A 50            PUSH   AX                  DS:0113  37 00 3C 00 05 00 0A 00
014B 51            PUSH   CX                  DS:011B  1E 00 5A 00 62 00 A7 00
014C 56            PUSH   SI                  DS:0123  C8 00 48 01 49 01 63 01
014D 49            DEC    CX                  DS:012B  15 02 F9 02 6D 03 78 03
014E D1E1          SHL    CX,1                DS:0133  82 03 FC 03 D0 07 29 09
0150 BE0000        MOV    SI,0000             DS:013B  65 20 DB 22 00 50 8B 00
0153 C6063F0100    MOV    [013F],00           DS:0143  87 40 02 89 00 58 C3 50
0158 8B00          MOV    AX,[BX+SI]          DS:014B  51 56 49 D1 E1 BE 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 FF FF  AD DE 1B 05 C5 06 00 00   = ƒ.Ω   ╡▌..┼...
DS:0010    18 01 10 01 18 01 92 01  01 01 01 00 FF 00 01 00   ......Æ. .... ...
DS:0020    01 FF FF FF FF FF FF FF  FF FF FF FF EB 19 E6 11   .           δ.µ.
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



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

```
AX 0037   SI 0012   CS 19F5   IP 0176       Stack +0 0000  Flags 7295
BX 0103   DI 0000   DS 19F5                       +2 000A
CX 0012   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 0183   0  0  1  1  0  1  1  1

 CMD >                                        1         3  4  5  6  7  8  9  A
                                              DS:0103  00 00 0A 00 19 00 1E 00
0174 74DA            JZ     0150              DS:010B  23 00 28 00 2D 00 32 00
0176 5E              POP    SI                DS:0113  37 00 3C 00 05 00 0A 00
0177 59              POP    CX                DS:011B  1E 00 5A 00 62 00 A7 00
0178 58              POP    AX                DS:0123  C8 00 48 01 49 01 63 01
0179 C3              RET                      DS:012B  15 02 F9 02 6D 03 78 03
017A BB0301          MOV    BX,0103           DS:0133  82 03 FC 03 D0 07 29 09
017D B90A00          MOV    CX,000A           DS:013B  65 20 DB 22 00 50 8B 00
0180 E8C7FF          CALL   014A              DS:0143  87 40 02 89 00 58 C3 50
0183 BB1701          MOV    BX,0117           DS:014B  51 56 49 D1 E1 BE 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 FF FF  AD DE 1B 05 C5 06 00 00   =  ƒ.Ω    ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 FF 00 01 00   ......Æ.  .... ...
DS:0020   01 FF FF FF FF FF FF FF  FF FF FF FF EB 19 E6 11   .             δ.µ.
DS:0030   A2 01 14 00 18 00 F5 19  FF FF FF FF 00 00 00 00   ó.....⌡.      ....
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
```



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

```
AX 0000   SI 0000   CS 19F5   IP 0100       Stack +0 0000  Flags 7202
BX 0000   DI 0000   DS 19F5                       +2 20CD
CX 0000   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 >                                        1         3  4  5  6  7  8  9  A
                                              DS:0103  00 00 0A 00 19 00 1E 00
    > Program terminated OK                   DS:010B  23 00 28 00 2D 00 32 00
0100 E97700          JMP    017A              DS:0113  37 00 3C 00 05 00 0A 00
0103 0000            ADD    [BX+SI],AL        DS:011B  1E 00 5A 00 62 00 A7 00
0105 0A00            OR     AL,[BX+SI]        DS:0123  C8 00 48 01 49 01 63 01
0107 1900            SBB    [BX+SI],AX        DS:012B  15 02 F9 02 6D 03 78 03
0109 1E              PUSH   DS                DS:0133  82 03 FC 03 D0 07 29 09
010A 0023            ADD    [BP+DI],AH        DS:013B  65 20 DB 22 00 50 8B 00
010C 0028            ADD    [BX+SI],CH        DS:0143  87 40 02 89 00 58 C3 50
010E 002D            ADD    [DI],CH           DS:014B  51 56 49 D1 E1 BE 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 FF FF  AD DE 1B 05 C5 06 00 00   =  ƒ.Ω    ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 FF 00 01 00   ......Æ.  .... ...
DS:0020   01 FF FF FF FF FF FF FF  FF FF FF FF EB 19 E4 11   .             δ.Σ.
DS:0030   A2 01 14 00 18 00 F5 19  FF FF FF FF 00 00 00 00   ó.....⌡.      ....
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
```