Name: Amber Khurshid

Section: BAI-4A

Roll No: 22P-9295

COAL Lab Task 07

Code:

```
; a program to simulate 3 loops
 [org 0x0100]
 mov ax,1
 loop1: mov bx,2
 mov dx,[num1]
 add [result],ax
 loop2: mov cx,3

 loop3: NOP
 NOP
 DEC cx
 jne loop3
 DEC bx
 jne loop2
 DEC ax
 jne loop1
 mov ax,0x4c00
 int 0x21
 num1: dw 10
 result: dw 0
```

Explanation:

[org 0x0100]

The statement 'org 0x0100' specifies the starting point of the code at memory address 0x100.

mov ax,1

The instruction 'mov ax, 1' sets the initial value of the outer loop counter AX to 1.

loop1: mov bx,2

LOOP1 Label for the first nested loop

The instruction 'mov bx, 2' sets the initial value of the outer loop counter bx to 2.

mov dx,[num1]

The instruction 'mov dx, [num1]' gets the value stored in memory location num1, which is 10, and stores it in the DX register.

add [result],ax

The command 'add [result], ax' performs an addition operation where the value of AX is added to the content stored at memory address 'result'.

loop2: mov cx,3

LOOP2: Assigns the initial value of 3 to the inner loop counter CX, initiating its iteration.

loop3: NOP NOP

LOOP3: Includes two NOP instructions, serving as placeholder commands to simulate a workload within the loop structure.

DEC cx

The instruction DEC cx reduces the value of CX by 1, while jne LOOP3 directs the program to return to LOOP3 if CX is not zero.

jne loop3

During Iteration 1, the value of CX decreases from 3 to 0, while the values of BX and AX remain constant at 1.

DEC bx

```
jne loop2
```

The instruction DEC bx reduces the value of BX by 1, and jne LOOP2 directs the program to return to LOOP2 if BX is not zero.

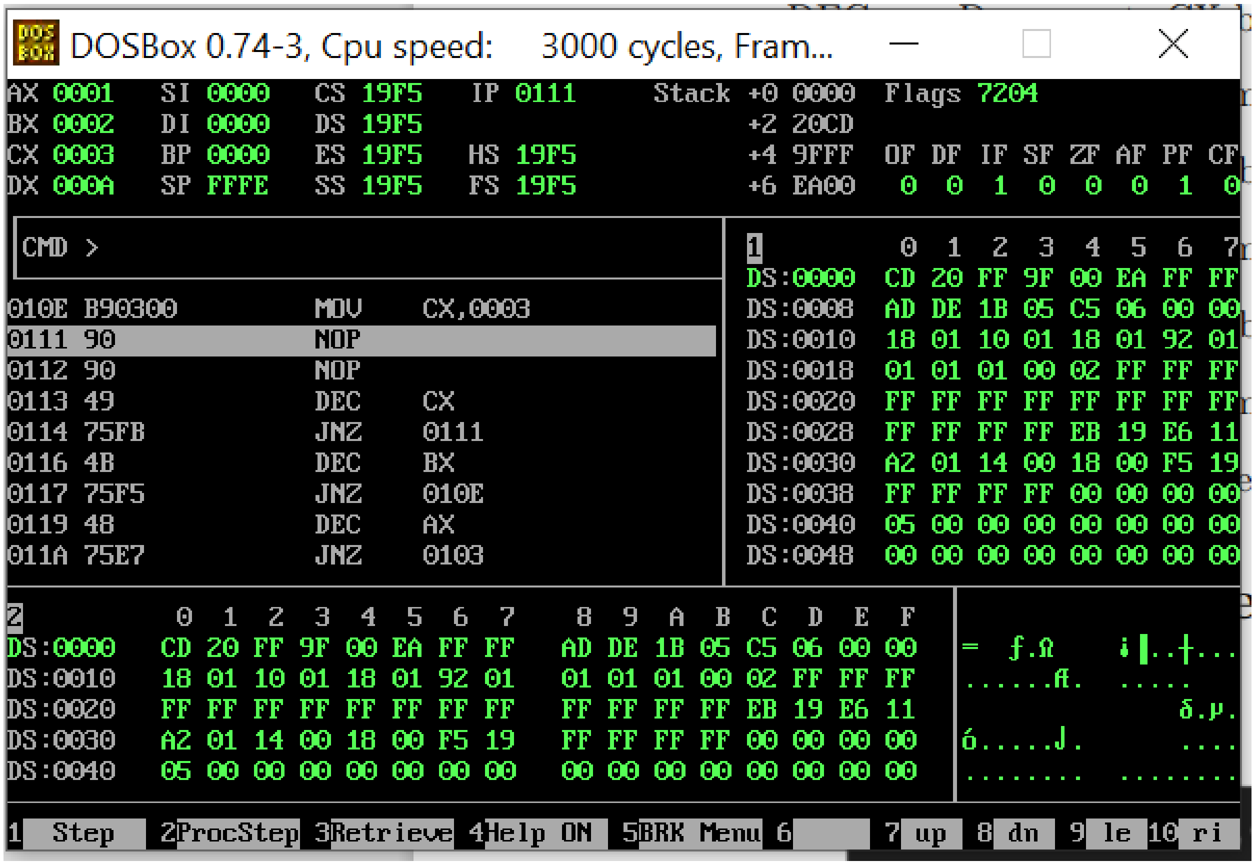
In Iteration 2, CX iterates from 3 to 0 twice, while BX decreases from 2 to 1. AX remains 1.

```
DEC ax
 jne loop1
```

The instruction DEC ax reduces the value of AX by 1, and jne LOOP1 directs the program to return to LOOP1 if AX is not zero.

In Iteration 3, the outer loop counter CX iterates from 3 to 0 three times. Meanwhile, the middle loop counter BX decreases from 2 to 1 twice. Additionally, the innermost loop counter AX decrements from 1 to 0.

The final value stored in [result] after the iterations is 10, which is obtained by adding 0 to the initial value of 10.



DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

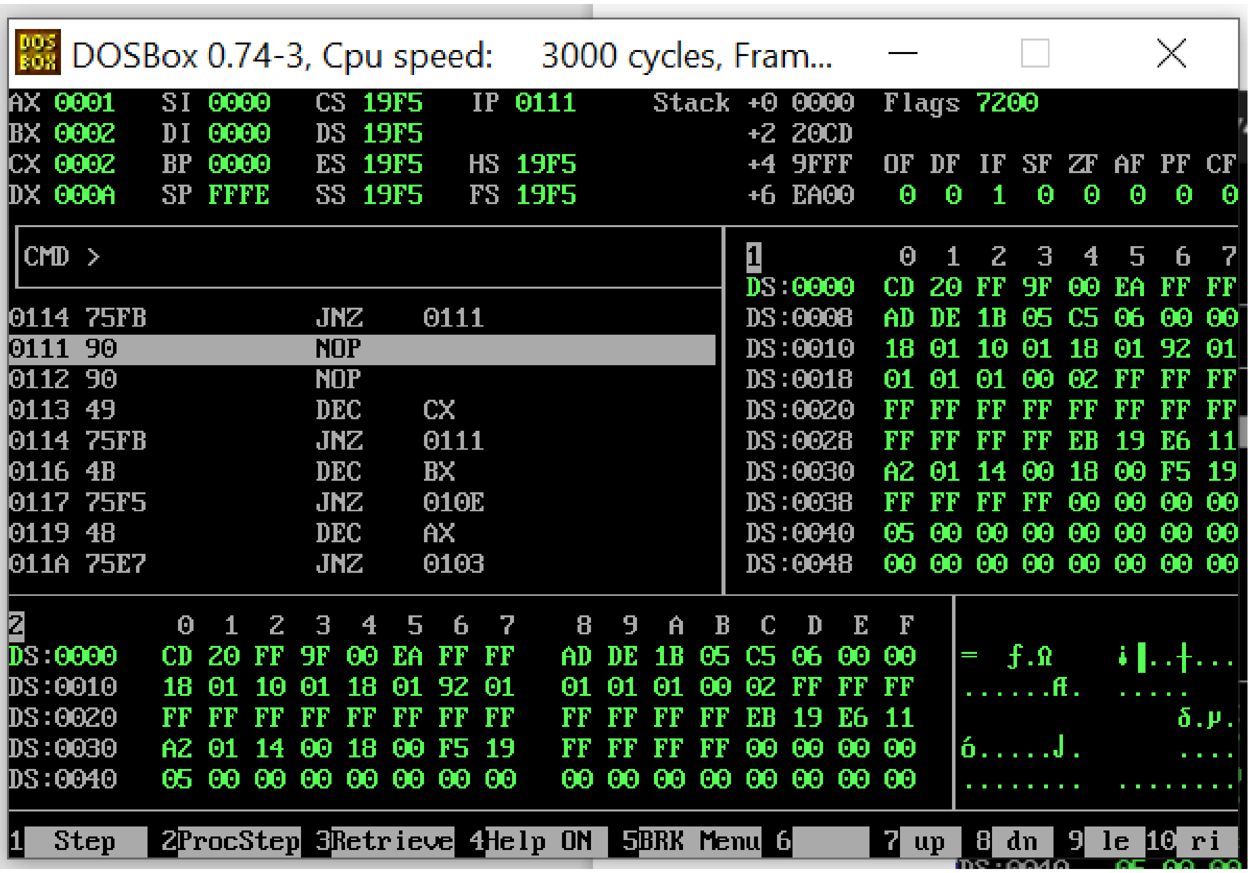
```
AX 0001   SI 0000   CS 19F5   IP 0111      Stack +0 0000  Flags 7204
BX 0002   DI 0000   DS 19F5                      +2 20CD
CX 0003   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 000A   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  1  0

 CMD >                                          1         0  1  2  3  4  5  6  7
                                                DS:0000  CD 20 FF 9F 00 EA FF FF
010E B90300             MOV    CX,0003          DS:0008  AD DE 1B 05 C5 06 00 00
0111 90                 NOP                     DS:0010  18 01 10 01 18 01 92 01
0112 90                 NOP                     DS:0018  01 01 01 00 02 FF FF FF
0113 49                 DEC    CX               DS:0020  FF FF FF FF FF FF FF FF
0114 75FB               JNZ    0111             DS:0028  FF FF FF FF EB 19 E6 11
0116 4B                 DEC    BX               DS:0030  A2 01 14 00 18 00 F5 19
0117 75F5               JNZ    010E             DS:0038  FF FF FF FF 00 00 00 00
0119 48                 DEC    AX               DS:0040  05 00 00 00 00 00 00 00
011A 75E7               JNZ    0103             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 02 FF FF FF  ......Æ.  .....
DS:0020    FF 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
```

loop iteration 1:



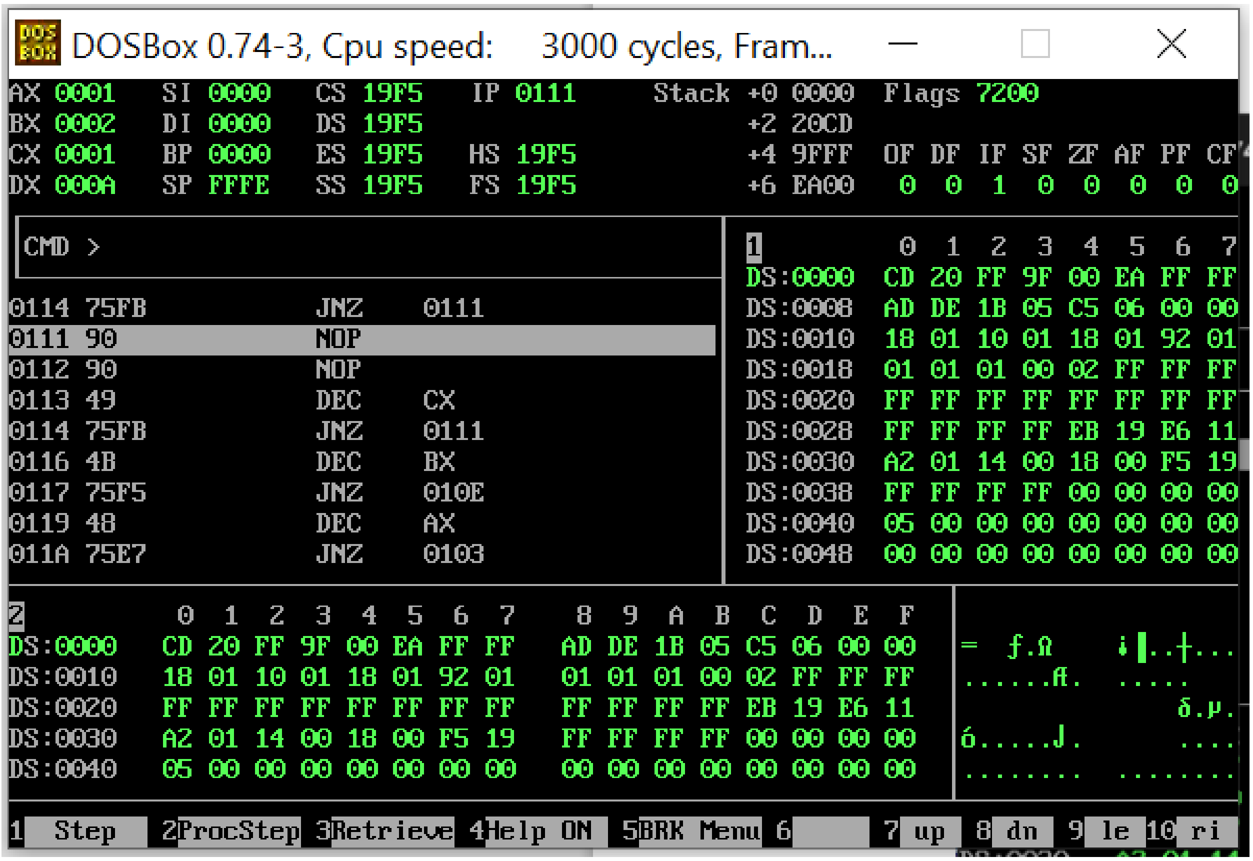
```
AX 0001    SI 0000    CS 19F5    IP 0111       Stack +0 0000  Flags 7200
BX 0002    DI 0000    DS 19F5                        +2 20CD
CX 0002    BP 0000    ES 19F5    HS 19F5             +4 9FFF  OF DF IF SF ZF AF PF CF
DX 000A    SP FFFE    SS 19F5    FS 19F5             +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                          1          0  1  2  3  4  5  6  7
                                                DS:0000   CD 20 FF 9F 00 EA FF FF
0114 75FB          JNZ     0111                 DS:0008   AD DE 1B 05 C5 06 00 00
0111 90            NOP                          DS:0010   18 01 10 01 18 01 92 01
0112 90            NOP                          DS:0018   01 01 01 00 02 FF FF FF
0113 49            DEC     CX                   DS:0020   FF FF FF FF FF FF FF FF
0114 75FB          JNZ     0111                 DS:0028   FF FF FF FF EB 19 E6 11
0116 4B            DEC     BX                   DS:0030   A2 01 14 00 18 00 F5 19
0117 75F5          JNZ     010E                 DS:0038   FF FF FF FF 00 00 00 00
0119 48            DEC     AX                   DS:0040   05 00 00 00 00 00 00 00
011A 75E7          JNZ     0103                 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 02 FF FF FF   ......Æ.   .....
DS:0020    FF 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
```

loop iteration 2:



DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

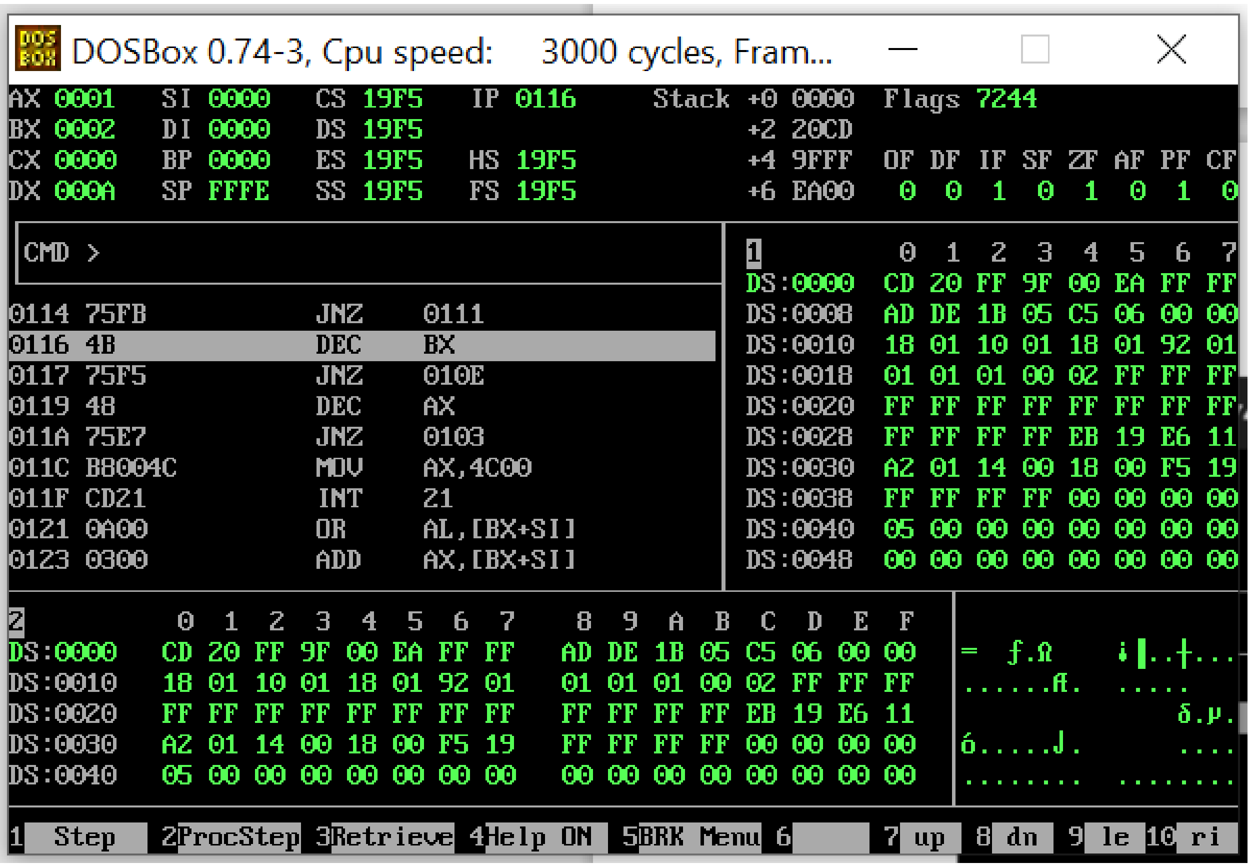
```
AX 0001   SI 0000   CS 19F5   IP 0111      Stack +0 0000  Flags 7200
BX 0002   DI 0000   DS 19F5                      +2 20CD
CX 0001   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 000A   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                       1          0  1  2  3  4  5  6  7
                                             DS:0000  CD 20 FF 9F 00 EA FF FF
0114 75FB           JNZ     0111             DS:0008  AD DE 1B 05 C5 06 00 00
0111 90             NOP                      DS:0010  18 01 10 01 18 01 92 01
0112 90             NOP                      DS:0018  01 01 01 00 02 FF FF FF
0113 49             DEC     CX               DS:0020  FF FF FF FF FF FF FF FF
0114 75FB           JNZ     0111             DS:0028  FF FF FF FF EB 19 E6 11
0116 4B             DEC     BX               DS:0030  A2 01 14 00 18 00 F5 19
0117 75F5           JNZ     010E             DS:0038  FF FF FF FF 00 00 00 00
0119 48             DEC     AX               DS:0040  05 00 00 00 00 00 00 00
011A 75E7           JNZ     0103             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 02 FF FF FF   ......Æ.  .....
DS:0020   FF 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
```

loop iteration 3:



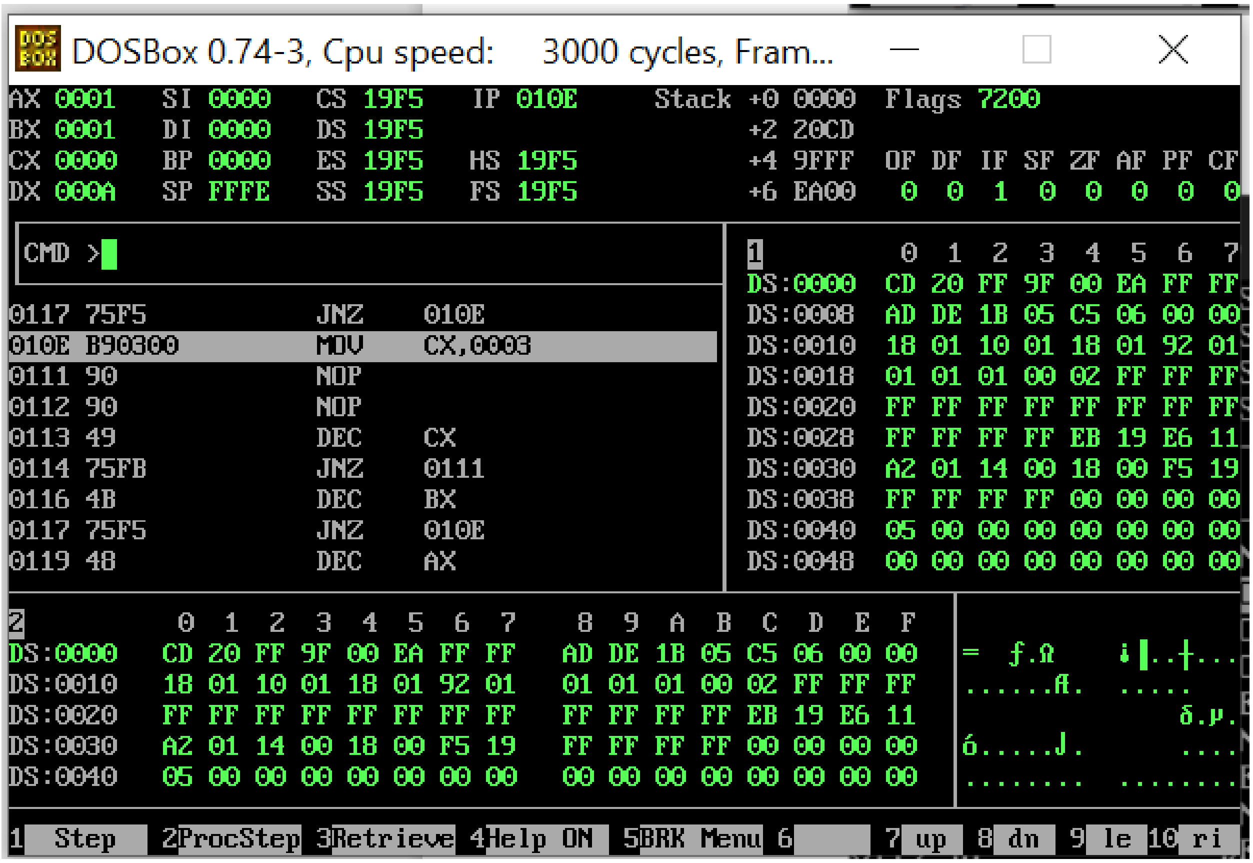
```
AX 0001   SI 0000   CS 19F5   IP 0116      Stack +0 0000  Flags 7244
BX 0002   DI 0000   DS 19F5                      +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 000A   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  1  0  1  0

 CMD >                                         1        0  1  2  3  4  5  6  7
                                               DS:0000  CD 20 FF 9F 00 EA FF FF
0114 75FB           JNZ    0111                DS:0008  AD DE 1B 05 C5 06 00 00
0116 4B             DEC    BX                  DS:0010  18 01 10 01 18 01 92 01
0117 75F5           JNZ    010E                DS:0018  01 01 01 00 02 FF FF FF
0119 48             DEC    AX                  DS:0020  FF FF FF FF FF FF FF FF
011A 75E7           JNZ    0103                DS:0028  FF FF FF FF EB 19 E6 11
011C B8004C         MOV    AX,4C00             DS:0030  A2 01 14 00 18 00 F5 19
011F CD21           INT    21                  DS:0038  FF FF FF FF 00 00 00 00
0121 0A00           OR     AL,[BX+SI]          DS:0040  05 00 00 00 00 00 00 00
0123 0300           ADD    AX,[BX+SI]          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 02 FF FF FF  ......Æ......
DS:0020   FF 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
```

loop iteration 4:



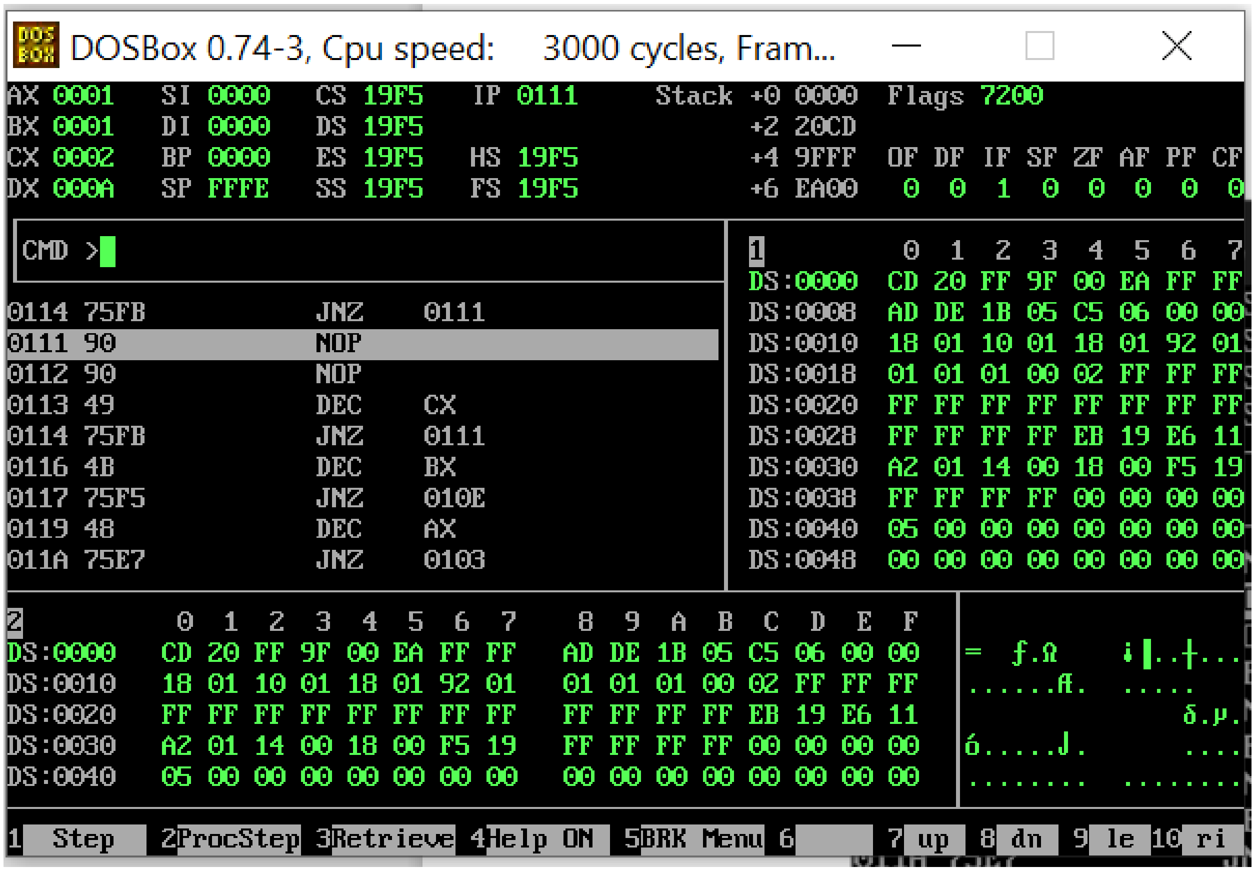
DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

```
AX 0001   SI 0000   CS 19F5   IP 010E      Stack +0 0000  Flags 7200
BX 0001   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 000A   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
0117 75F5           JNZ    010E                 DS:0008   AD DE 1B 05 C5 06 00 00
010E B90300         MOV    CX,0003              DS:0010   18 01 10 01 18 01 92 01
0111 90             NOP                         DS:0018   01 01 01 00 02 FF FF FF
0112 90             NOP                         DS:0020   FF FF FF FF FF FF FF FF
0113 49             DEC    CX                   DS:0028   FF FF FF FF EB 19 E6 11
0114 75FB           JNZ    0111                 DS:0030   A2 01 14 00 18 00 F5 19
0116 4B             DEC    BX                   DS:0038   FF FF FF FF 00 00 00 00
0117 75F5           JNZ    010E                 DS:0040   05 00 00 00 00 00 00 00
0119 48             DEC    AX                   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 02 FF FF FF    ......Æ.  .....
DS:0020   FF 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    ........  ........
```

loop iteration 5:

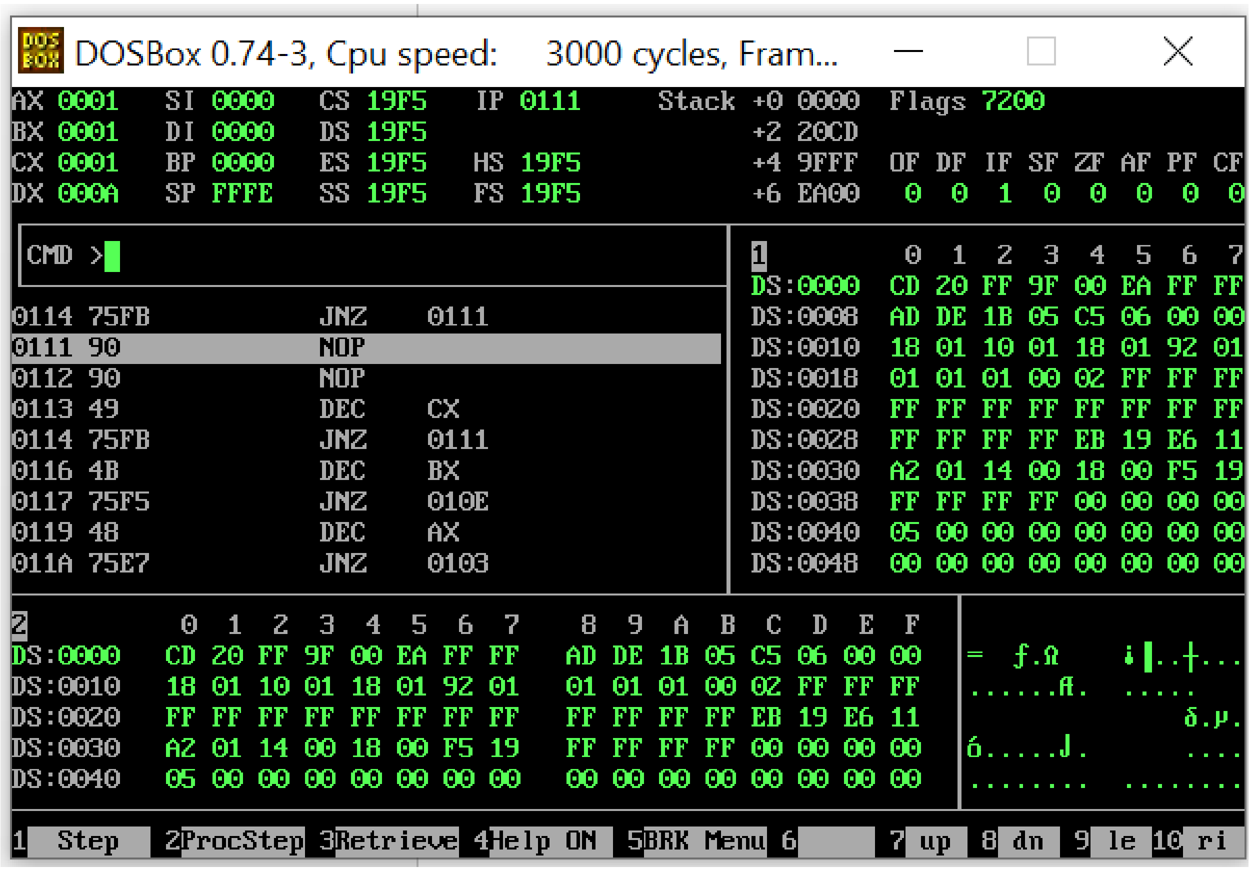


```
AX 0001   SI 0000   CS 19F5   IP 0111      Stack +0 0000  Flags 7200
BX 0001   DI 0000   DS 19F5                      +2 20CD
CX 0002   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 000A   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                          1        0  1  2  3  4  5  6  7
                                                DS:0000  CD 20 FF 9F 00 EA FF FF
0114 75FB          JNZ    0111                  DS:0008  AD DE 1B 05 C5 06 00 00
0111 90            NOP                          DS:0010  18 01 10 01 18 01 92 01
0112 90            NOP                          DS:0018  01 01 01 00 02 FF FF FF
0113 49            DEC    CX                    DS:0020  FF FF FF FF FF FF FF FF
0114 75FB          JNZ    0111                  DS:0028  FF FF FF FF EB 19 E6 11
0116 4B            DEC    BX                    DS:0030  A2 01 14 00 18 00 F5 19
0117 75F5          JNZ    010E                  DS:0038  FF FF FF FF 00 00 00 00
0119 48            DEC    AX                    DS:0040  05 00 00 00 00 00 00 00
011A 75E7          JNZ    0103                  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 02 FF FF FF   ......Æ.  .....
DS:0020   FF 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   ........  ........
```

loop iteration 6:



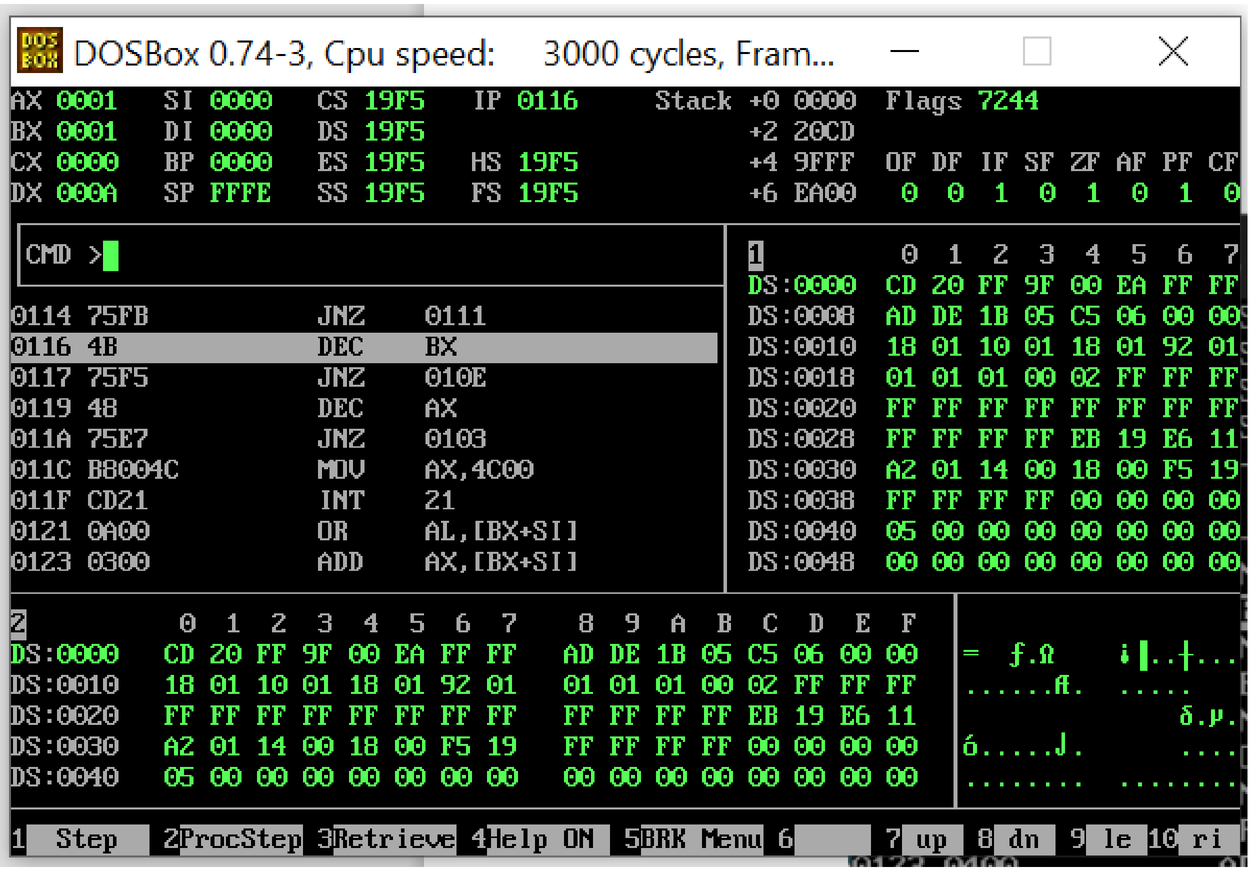
```
AX 0001   SI 0000   CS 19F5   IP 0111     Stack +0 0000  Flags 7200
BX 0001   DI 0000   DS 19F5                     +2 20CD
CX 0001   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 000A   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                      1          0  1  2  3  4  5  6  7
                                            DS:0000  CD 20 FF 9F 00 EA FF FF
0114 75FB             JNZ    0111           DS:0008  AD DE 1B 05 C5 06 00 00
0111 90               NOP                   DS:0010  18 01 10 01 18 01 92 01
0112 90               NOP                   DS:0018  01 01 01 00 02 FF FF FF
0113 49               DEC    CX             DS:0020  FF FF FF FF FF FF FF FF
0114 75FB             JNZ    0111           DS:0028  FF FF FF FF EB 19 E6 11
0116 4B               DEC    BX             DS:0030  A2 01 14 00 18 00 F5 19
0117 75F5             JNZ    010E           DS:0038  FF FF FF FF 00 00 00 00
0119 48               DEC    AX             DS:0040  05 00 00 00 00 00 00 00
011A 75E7             JNZ    0103           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 02 FF FF FF  ......Æ.  .....
DS:0020   FF 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
```

loop iteration 7:



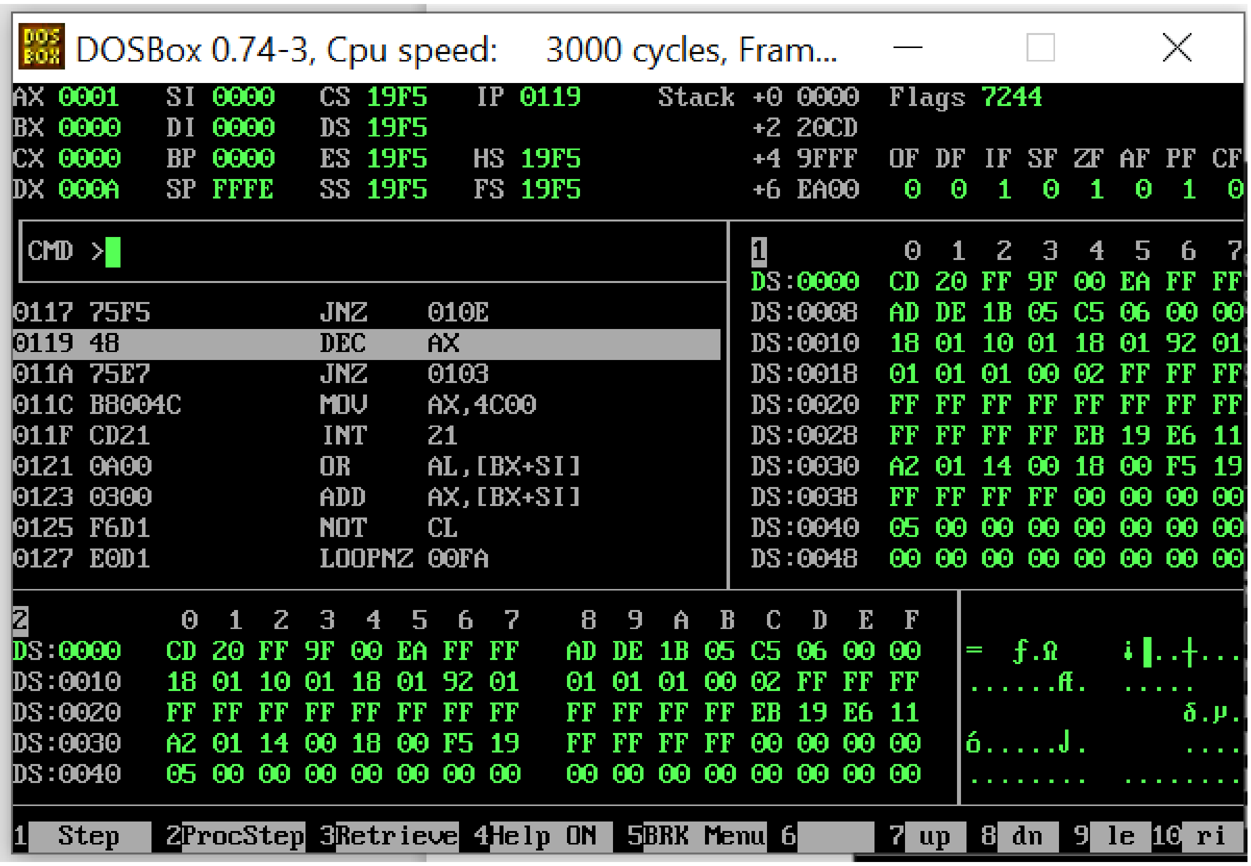
```
AX 0001    SI 0000    CS 19F5    IP 0116      Stack +0 0000  Flags 7244
BX 0001    DI 0000    DS 19F5                       +2 20CD
CX 0000    BP 0000    ES 19F5    HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 000A    SP FFFE    SS 19F5    FS 19F5            +6 EA00   0  0  1  0  1  0  1  0

 CMD >                                        1          0  1  2  3  4  5  6  7
                                              DS:0000  CD 20 FF 9F 00 EA FF FF
0114 75FB            JNZ    0111              DS:0008  AD DE 1B 05 C5 06 00 00
0116 4B              DEC    BX                DS:0010  18 01 10 01 18 01 92 01
0117 75F5            JNZ    010E              DS:0018  01 01 01 00 02 FF FF FF
0119 48              DEC    AX                DS:0020  FF FF FF FF FF FF FF FF
011A 75E7            JNZ    0103              DS:0028  FF FF FF FF EB 19 E6 11
011C B8004C          MOV    AX,4C00           DS:0030  A2 01 14 00 18 00 F5 19
011F CD21            INT    21                DS:0038  FF FF FF FF 00 00 00 00
0121 0A00            OR     AL,[BX+SI]        DS:0040  05 00 00 00 00 00 00 00
0123 0300            ADD    AX,[BX+SI]        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 02 FF FF FF   ......Æ.  .....
DS:0020     FF 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
```

loop iteration 8:



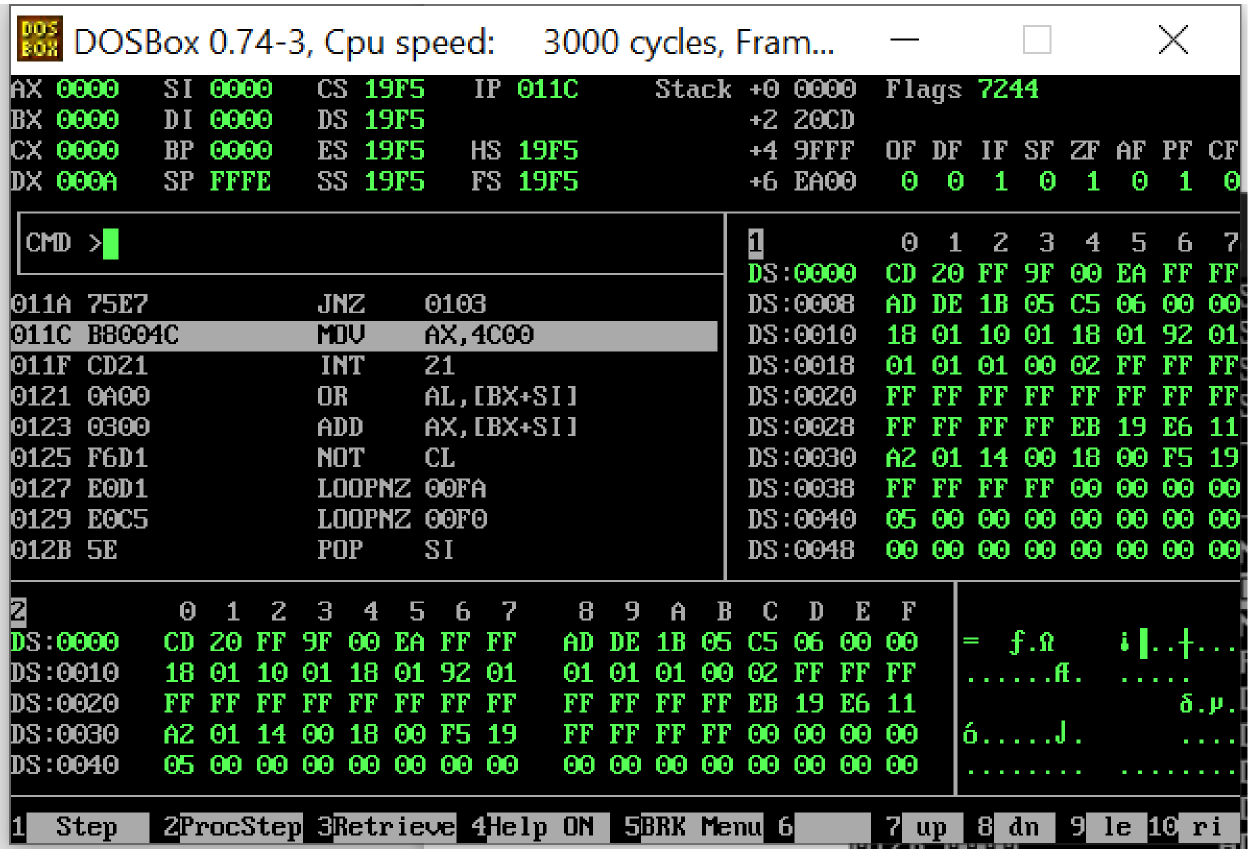
```
AX 0001   SI 0000   CS 19F5   IP 0119      Stack +0 0000  Flags 7244
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 000A   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  1  0  1  0

 CMD >                                      1          0  1  2  3  4  5  6  7
                                            DS:0000  CD 20 FF 9F 00 EA FF FF
0117 75F5             JNZ    010E           DS:0008  AD DE 1B 05 C5 06 00 00
0119 48               DEC    AX             DS:0010  18 01 10 01 18 01 92 01
011A 75E7             JNZ    0103           DS:0018  01 01 01 00 02 FF FF FF
011C B8004C           MOV    AX,4C00        DS:0020  FF FF FF FF FF FF FF FF
011F CD21             INT    21             DS:0028  FF FF FF FF EB 19 E6 11
0121 0A00             OR     AL,[BX+SI]     DS:0030  A2 01 14 00 18 00 F5 19
0123 0300             ADD    AX,[BX+SI]     DS:0038  FF FF FF FF 00 00 00 00
0125 F6D1             NOT    CL             DS:0040  05 00 00 00 00 00 00 00
0127 E0D1             LOOPNZ 00FA           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 02 FF FF FF   ......Æ.  .....
DS:0020    FF 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
```

loop iteration 9:



```
AX 0000   SI 0000   CS 19F5   IP 011C      Stack +0 0000  Flags 7244
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 000A   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  1  0  1  0

 CMD >                                       1         0  1  2  3  4  5  6  7
                                             DS:0000  CD 20 FF 9F 00 EA FF FF
011A 75E7          JNZ    0103               DS:0008  AD DE 1B 05 C5 06 00 00
011C B8004C        MOV    AX,4C00            DS:0010  18 01 10 01 18 01 92 01
011F CD21          INT    21                 DS:0018  01 01 01 00 02 FF FF FF
0121 0A00          OR     AL,[BX+SI]         DS:0020  FF FF FF FF FF FF FF FF
0123 0300          ADD    AX,[BX+SI]         DS:0028  FF FF FF FF EB 19 E6 11
0125 F6D1          NOT    CL                 DS:0030  A2 01 14 00 18 00 F5 19
0127 E0D1          LOOPNZ 00FA               DS:0038  FF FF FF FF 00 00 00 00
0129 E0C5          LOOPNZ 00F0               DS:0040  05 00 00 00 00 00 00 00
012B 5E            POP    SI                 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 02 FF FF FF   ......Æ. .....
DS:0020   FF 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
```