National University
Of Computer and Emerging Sciences

EE2003 – Computer Organization and Assembly Language (Fall 2023) 10 Oct 2023
Assignment: 03, Marks: 10, Due Date: 20 minutes, CLO: 3 Teacher: Dr Muhammad Usman Abbasi

Name: Ahmed – Mustafa **Roll No.** 22P-9060 **Section: BS-CS-3A**

1. Make separate subroutines for add, subtract, multiply and divide and then perform all these operations between two numbers of your choice using these subroutines and passing them the numbers as parameters on the stack. Also store the results for each of the operation in the variables shown in the starter code.

Answer:

Code:

assign_3.asm

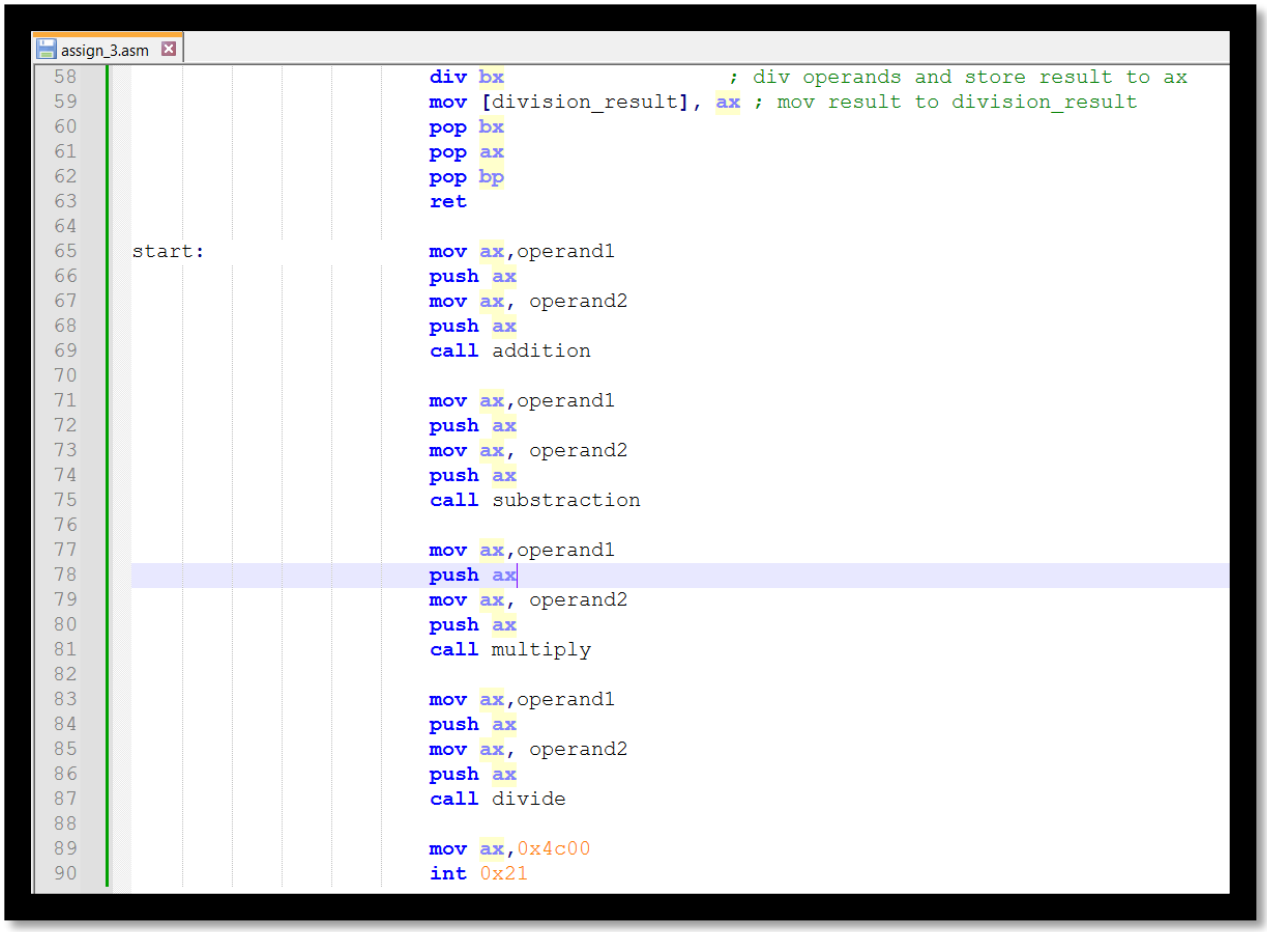
```
37  ; multiply subroutine
38  multiply:               push bp                 ; save old value of base pointer
39                          mov bp,sp               ; mov base pointer to stack pointer
40                          push ax                 ; save old value of ax register
41                          push bx                 ; save old value of bx register
42                          mov ax,[bp+6]           ; mov operand1 to ax from stack
43                          mov bx,[bp+4]           ; mov operand2 to bx from stack
44                          imul ax,bx              ; mul both operands and store result to ax
45                          mov [multiplication_result], ax ;mov result to multiplication_result
46                          pop bx
47                          pop ax
48                          pop bp
49                          ret
50  ; divide subroutine
51  divide:                 push bp                 ; save old value of base pointer
52                          mov bp,sp               ; mov base pointer to stack pointer
53                          push ax                 ; save old value of ax register
54                          push bx                 ; save old value of bx register
55                          mov ax,[bp+6]           ; mov operand1 to ax from stack
56                          mov bx,[bp+4]           ; mov operand2 to bx from stack
57                          xor dx,dx               ; clear dx
58                          div bx                  ; div operands and store result to ax
59                          mov [division_result], ax ; mov result to division_result
60                          pop bx
61                          pop ax
62                          pop bp
63                          ret
64
65  start:                  mov ax,operand1
66                          push ax
67                          mov ax, operand2
68                          push ax
69                          call addition
70
71                          mov ax,operand1
72                          push ax
```

assign_3.asm

```
[org 0x0100]

jmp start ; jump to start label

operand1: dw 20
operand2: dw 12
sum_result: dw 0
subtraction_result: dw 0
multiplication_result: dw 0
division_result: dw 0
; addition subroutine
addition:               push bp                 ; save old value of base pointer
                        mov bp,sp               ; mov base pointer to stack pointer
                        push ax                 ; save old value of ax register
                        push bx                 ; save old value of bx register
                        mov ax,[bp+6]           ; mov operand1 to ax from stack
                        mov bx,[bp+4]           ; mov operand2 to bx from stack
                        add ax,bx               ; add both operands and store result to ax
                        mov [sum_result], ax    ; mov result to sum_result
                        pop bx
                        pop ax
                        pop bp
                        ret                     ; return back using instruction pointer
; substraction subroutine
substraction:           push bp                 ; save old value of base pointer
                        mov bp,sp               ; mov base pointer to stack pointer
                        push ax                 ; save old value of ax register
                        push bx                 ; save old value of bx register
                        mov ax,[bp+6]           ; mov operand1 to ax from stack
                        mov bx,[bp+4]           ; mov operand2 to bx from stack
                        sub ax,bx               ; sub both operands and store result to ax
                        mov [subtraction_result], ax ; mov result to sub_result
                        pop bx
                        pop ax
                        pop bp
                        ret
```

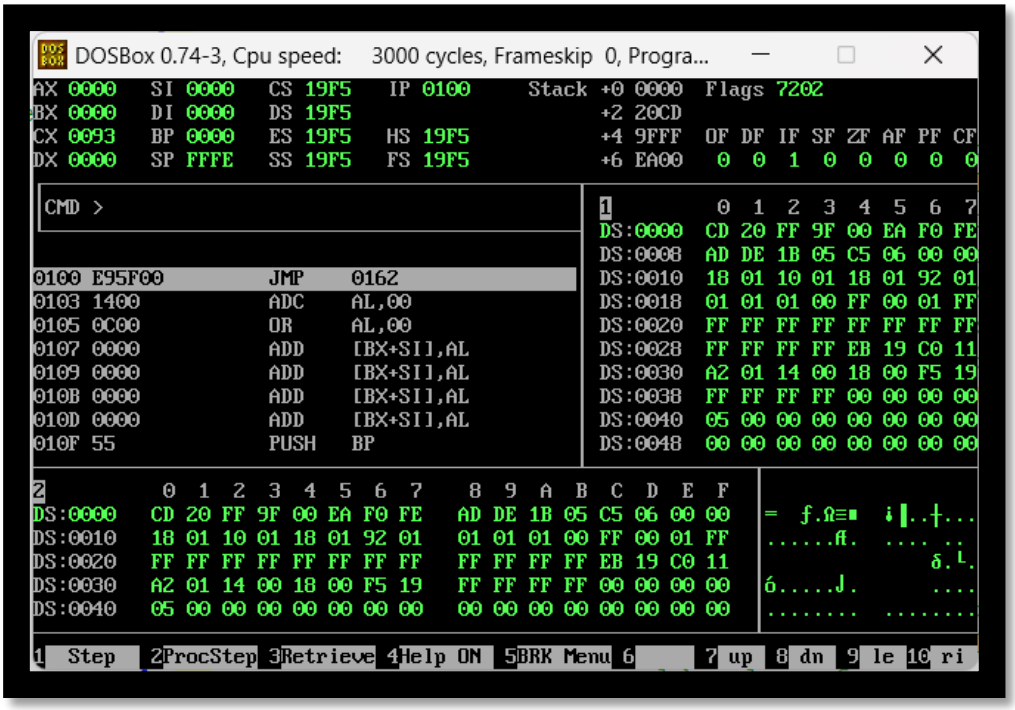


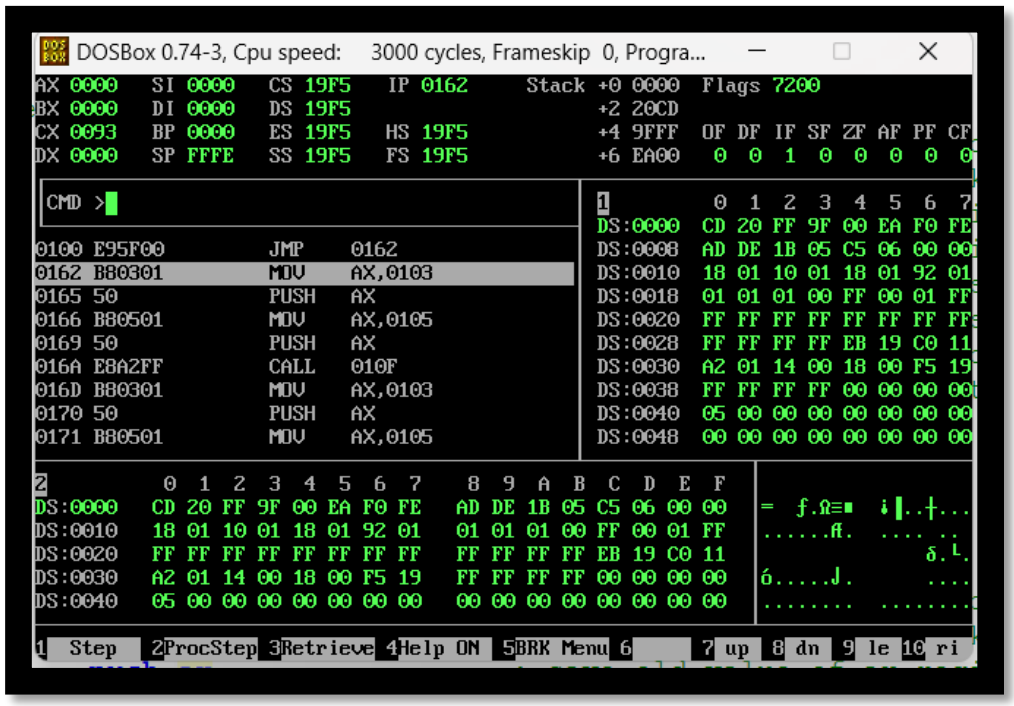
Explanation:

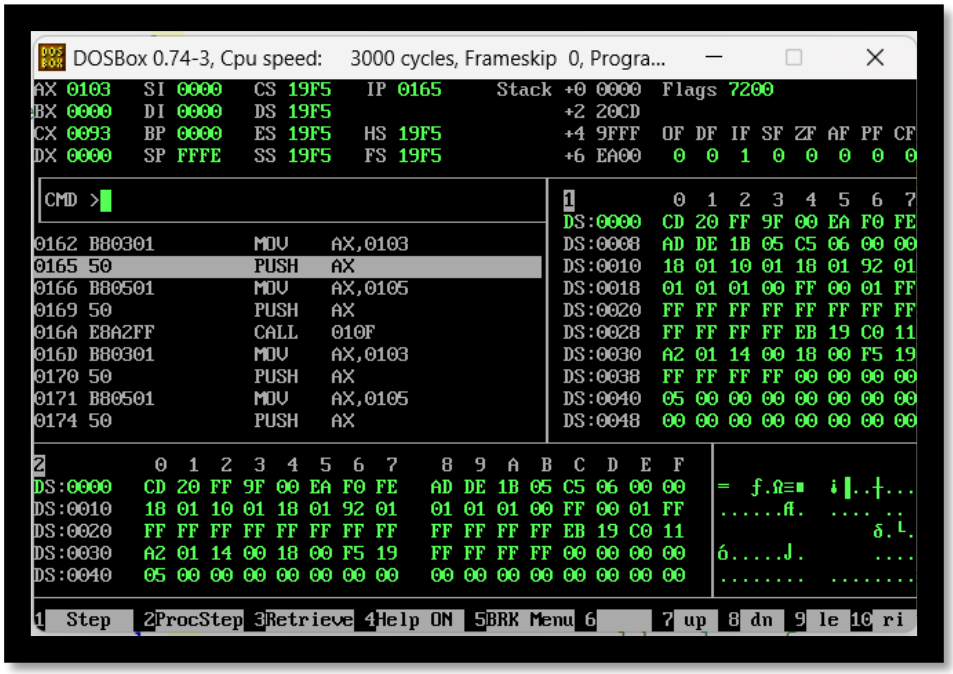
This assembly code defines a program to perform arithmetic operations on operand1 and operand2 using the stack for subroutine calls. It initializes operands and result variables, with four subroutines (addition, subtraction, multiply, and divide) following a pattern of stack preservation, operation execution, and result storage.

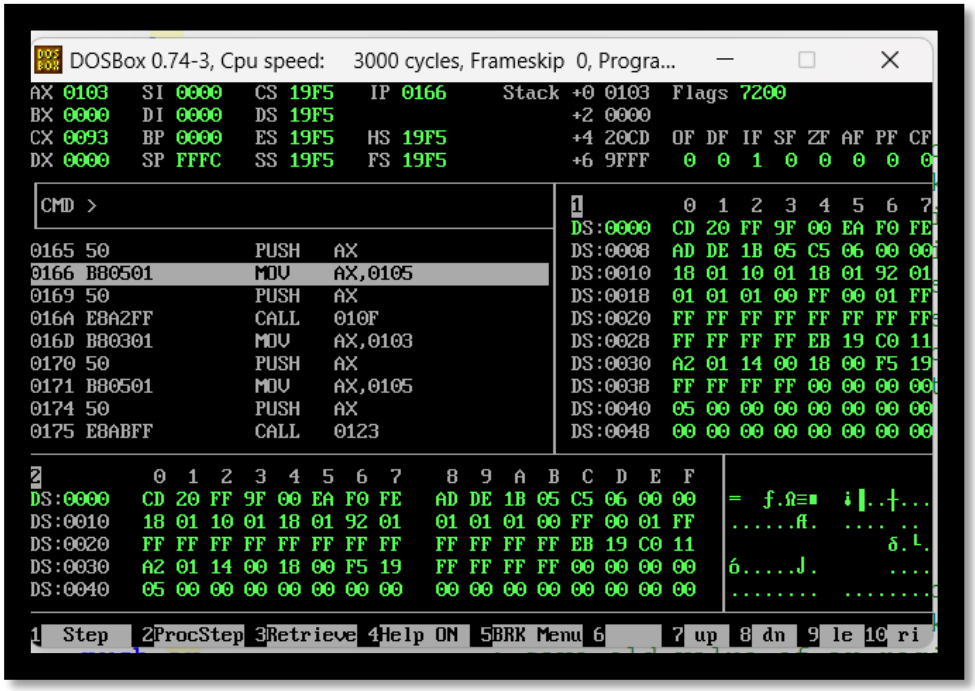
In the start section, the program calls each arithmetic subroutine sequentially, utilizing the stack for parameter passing. The DOS interrupt int 0x21 is employed for program termination. The stack serves as a crucial mechanism for managing data and control flow during subroutine execution.

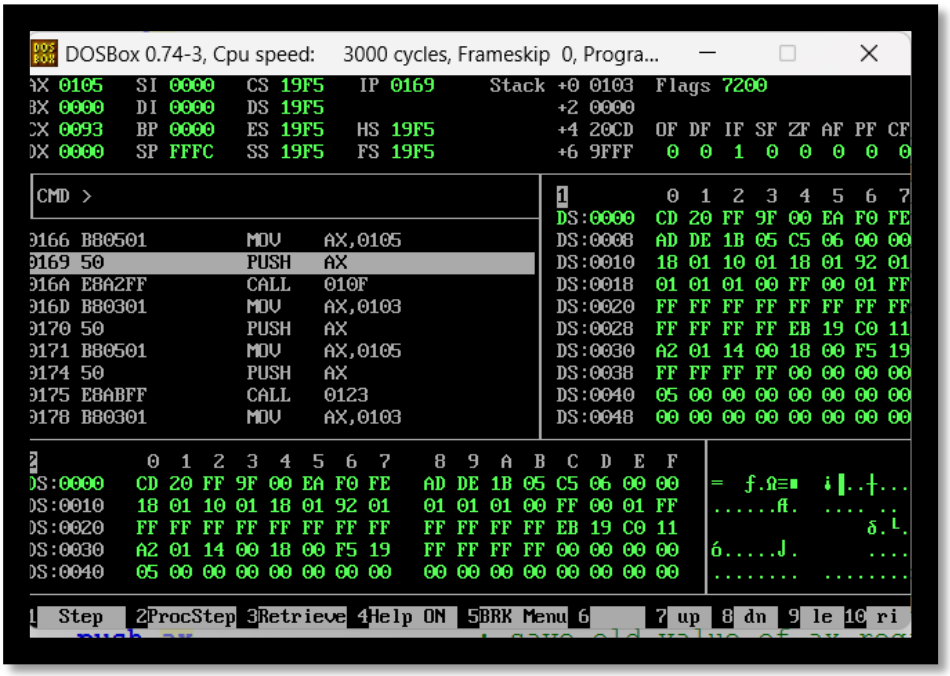
Screen-Shot of Debugger:

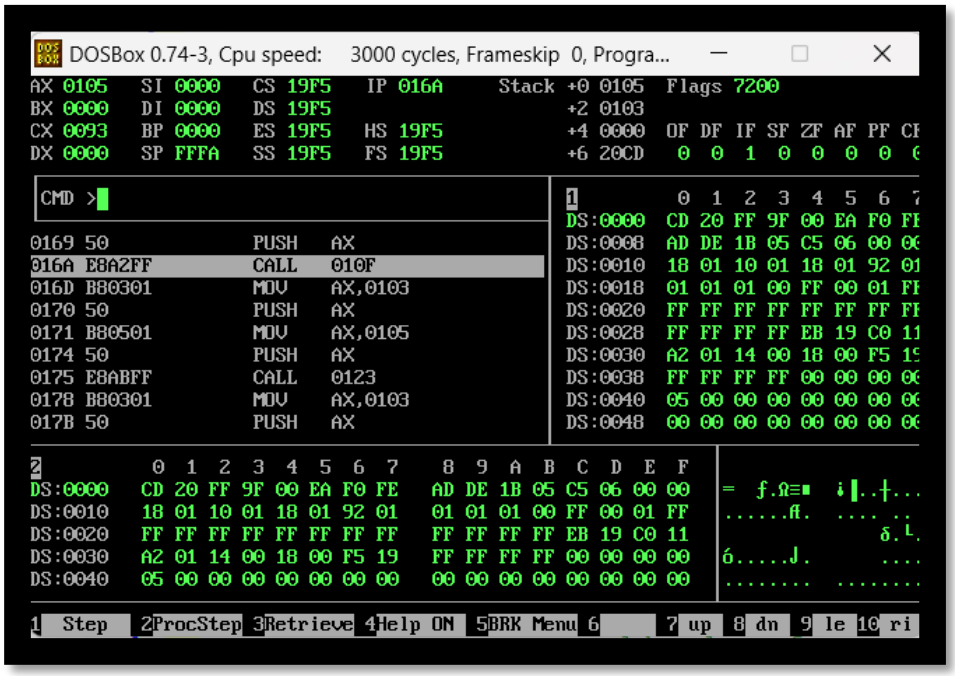

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0000 SI 0000 CS 19F5 IP 0100 Stack +0 0000 Flags 7202
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
0100 E95F00 JMP 0162
0103 1400 ADC AL,00
0105 0C00 OR AL,00
0107 0000 ADD [BX+SI],AL
0109 0000 ADD [BX+SI],AL
010B 0000 ADD [BX+SI],AL
010D 0000 ADD [BX+SI],AL
010F 55 PUSH BP
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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 F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

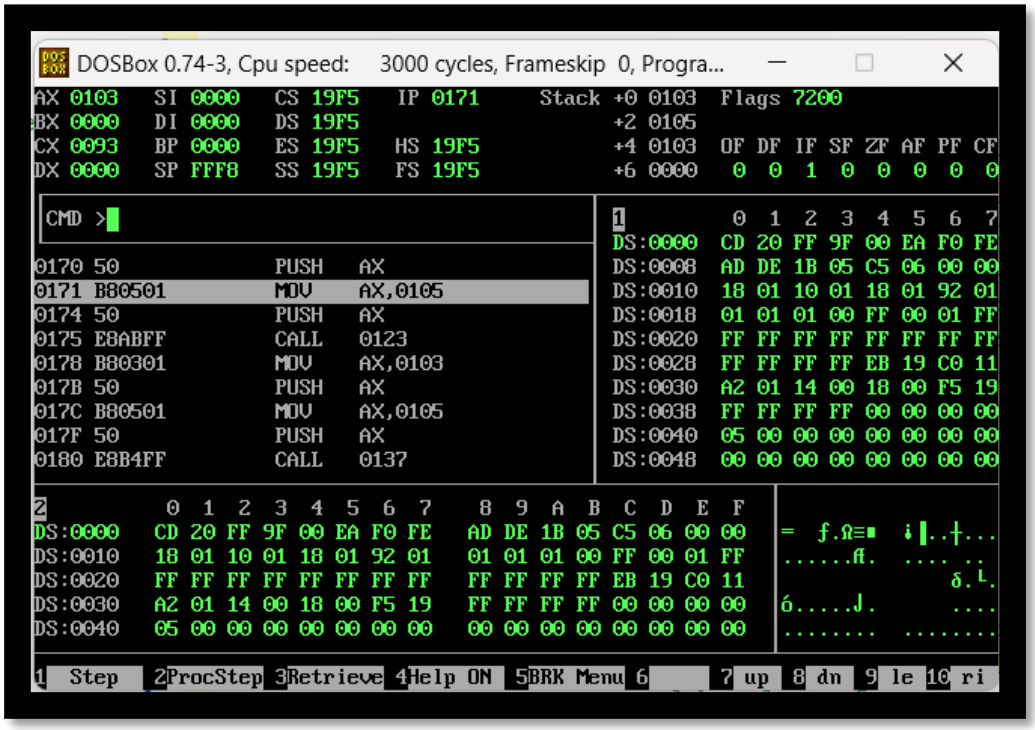

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0000 SI 0000 CS 19F5 IP 0162 Stack +0 0000 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
0100 E95F00 JMP 0162
0162 B80301 MOV AX,0103
0165 50 PUSH AX
0166 B80501 MOV AX,0105
0169 50 PUSH AX
016A E8A2FF CALL 010F
016D B80301 MOV AX,0103
0170 50 PUSH AX
0171 B80501 MOV AX,0105
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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 F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

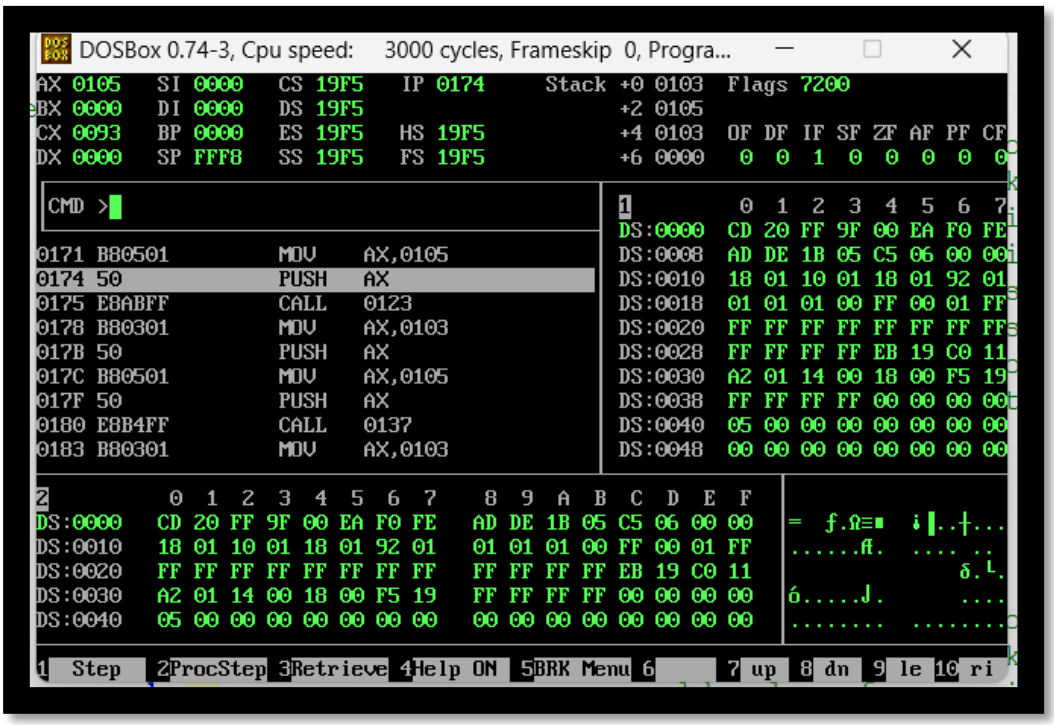

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0103 SI 0000 CS 19F5 IP 0165 Stack +0 0000 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
0162 B80301 MOV AX,0103
0165 50 PUSH AX
0166 B80501 MOV AX,0105
0169 50 PUSH AX
016A E8A2FF CALL 010F
016D B80301 MOV AX,0103
0170 50 PUSH AX
0171 B80501 MOV AX,0105
0174 50 PUSH AX
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0008 AD DE 1B 05 C5 06 00 00 18 01 10 01 18 01 92 01
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0018 01 01 01 00 FF 00 01 FF FF FF FF FF FF FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0028 FF FF FF FF EB 19 C0 11 A2 01 14 00 18 00 F5 19
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0038 FF FF FF FF 00 00 00 00 05 00 00 00 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DS:0048 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

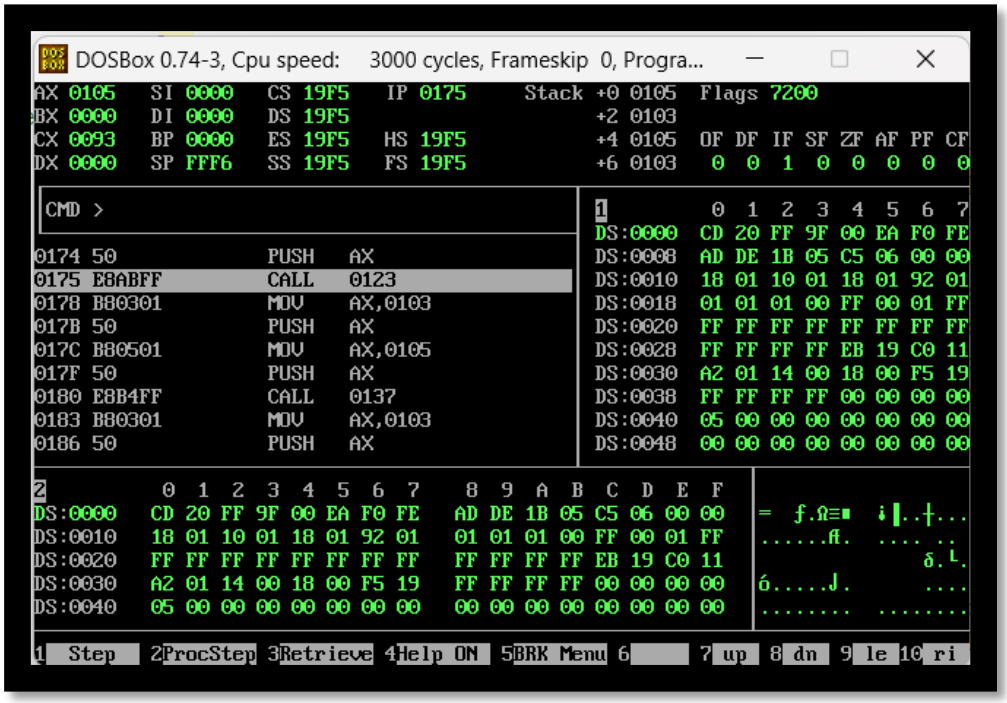

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0103 SI 0000 CS 19F5 IP 0166 Stack +0 0103 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0000
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 20CD OF DF IF SF ZF AF PF CF
DX 0000 SP FFFC SS 19F5 FS 19F5 +6 9FFF 0 0 1 0 0 0 0 0
CMD >
0165 50 PUSH AX
0166 B80501 MOV AX,0105
0169 50 PUSH AX
016A E8A2FF CALL 010F
016D B80301 MOV AX,0103
0170 50 PUSH AX
0171 B80501 MOV AX,0105
0174 50 PUSH AX
0175 E8ABFF CALL 0123
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0008 AD DE 1B 05 C5 06 00 00 18 01 10 01 18 01 92 01
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0018 01 01 01 00 FF 00 01 FF FF FF FF FF FF FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0028 FF FF FF FF EB 19 C0 11 A2 01 14 00 18 00 F5 19
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0038 FF FF FF FF 00 00 00 00 05 00 00 00 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DS:0048 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

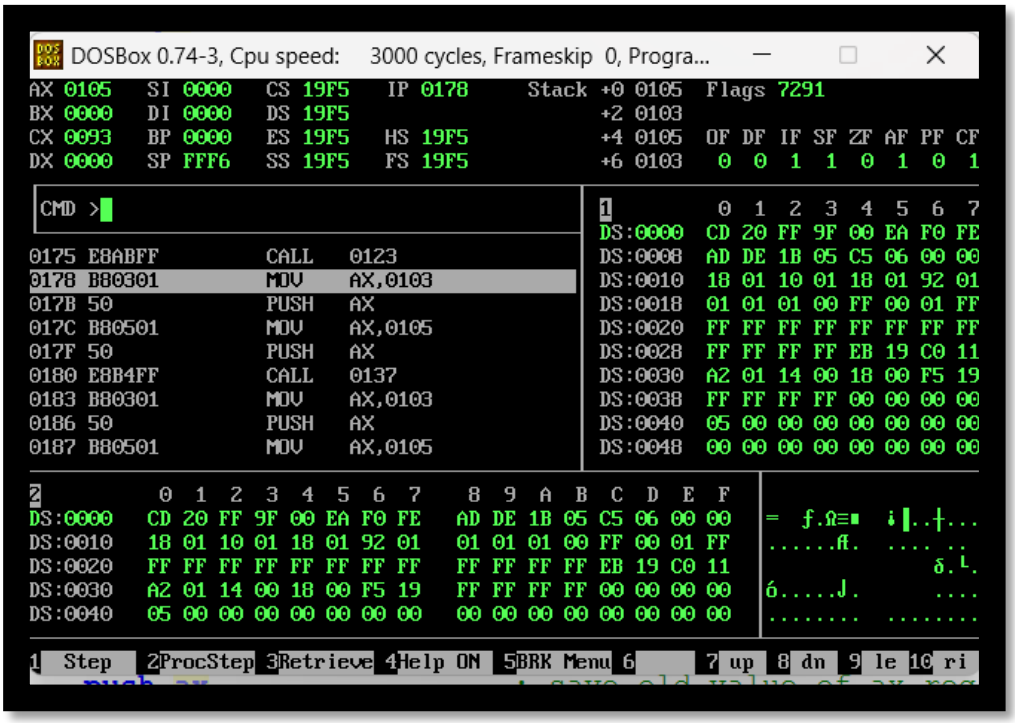

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0105 SI 0000 CS 19F5 IP 0169 Stack +0 0103 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0000
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 20CD OF DF IF SF ZF AF PF CF
DX 0000 SP FFFC SS 19F5 FS 19F5 +6 9FFF 0 0 1 0 0 0 0 0
CMD >
0166 B80501 MOV AX,0105
0169 50 PUSH AX
016A E8A2FF CALL 010F
016D B80301 MOV AX,0103
0170 50 PUSH AX
0171 B80501 MOV AX,0105
0174 50 PUSH AX
0175 E8ABFF CALL 0123
0178 B80301 MOV AX,0103
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

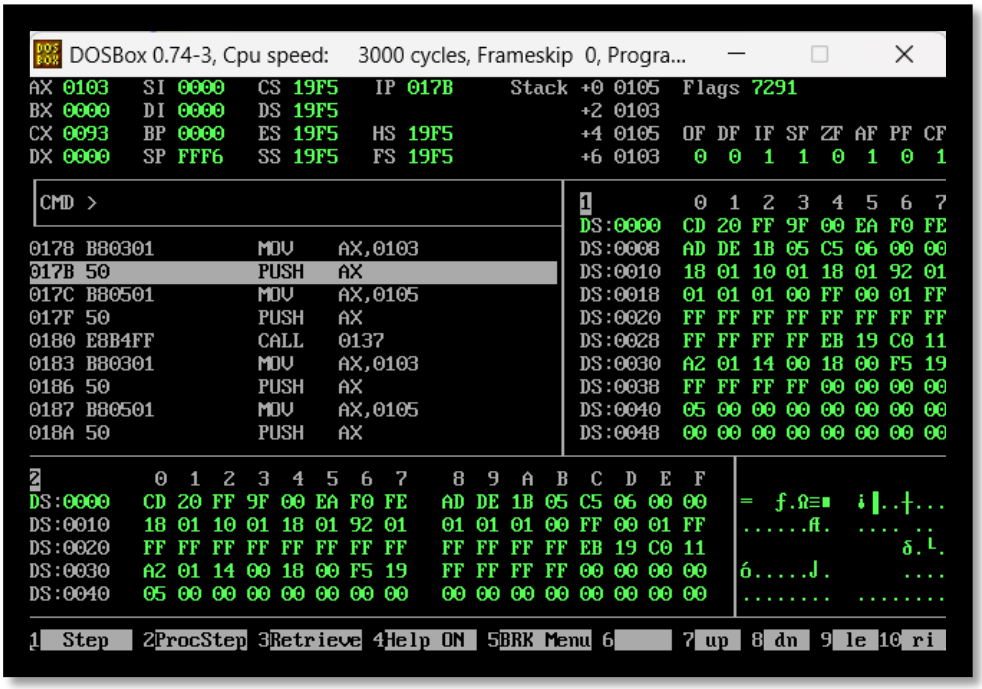

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0105 SI 0000 CS 19F5 IP 016A Stack +0 0105 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0103
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0000 OF DF IF SF ZF AF PF CF
DX 0000 SP FFFA SS 19F5 FS 19F5 +6 20CD 0 0 1 0 0 0 0 0
CMD >
0169 50 PUSH AX
016A E8A2FF CALL 010F
016D B80301 MOV AX,0103
0170 50 PUSH AX
0171 B80501 MOV AX,0105
0174 50 PUSH AX
0175 E8ABFF CALL 0123
0178 B80301 MOV AX,0103
017B 50 PUSH AX
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

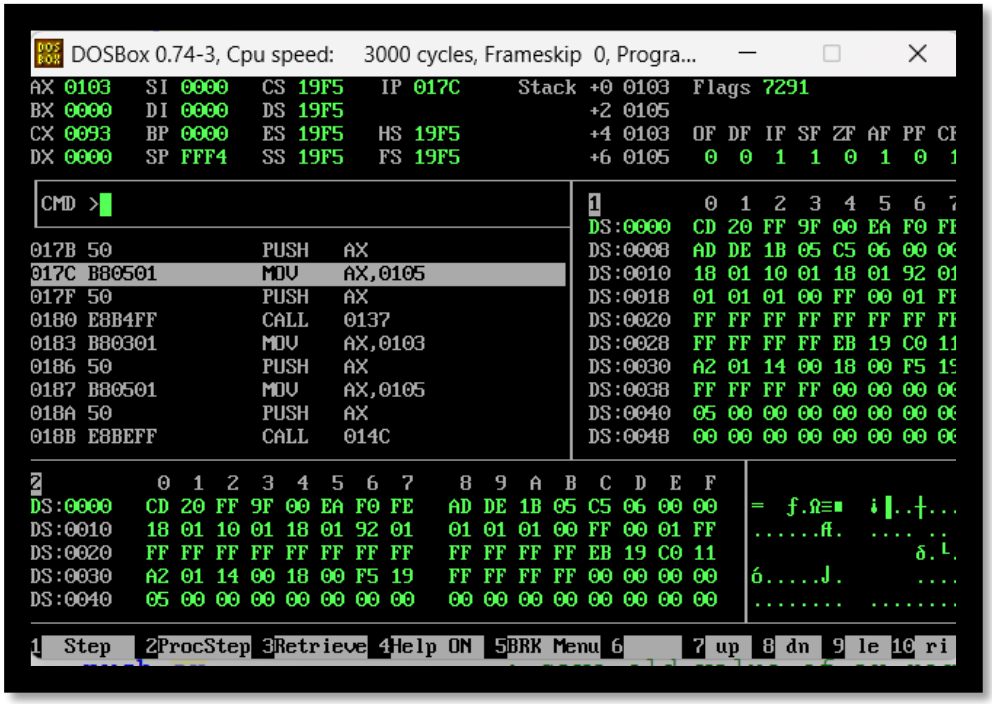

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0103 SI 0000 CS 19F5 IP 0171 Stack +0 0103 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0105
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0103 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF8 SS 19F5 FS 19F5 +6 0000 0 0 1 0 0 0 0 0
CMD >
0170 50 PUSH AX
0171 B80501 MOV AX,0105
0174 50 PUSH AX
0175 E8ABFF CALL 0123
0178 B80301 MOV AX,0103
017B 50 PUSH AX
017C B80501 MOV AX,0105
017F 50 PUSH AX
0180 E8B4FF CALL 0137
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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 F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

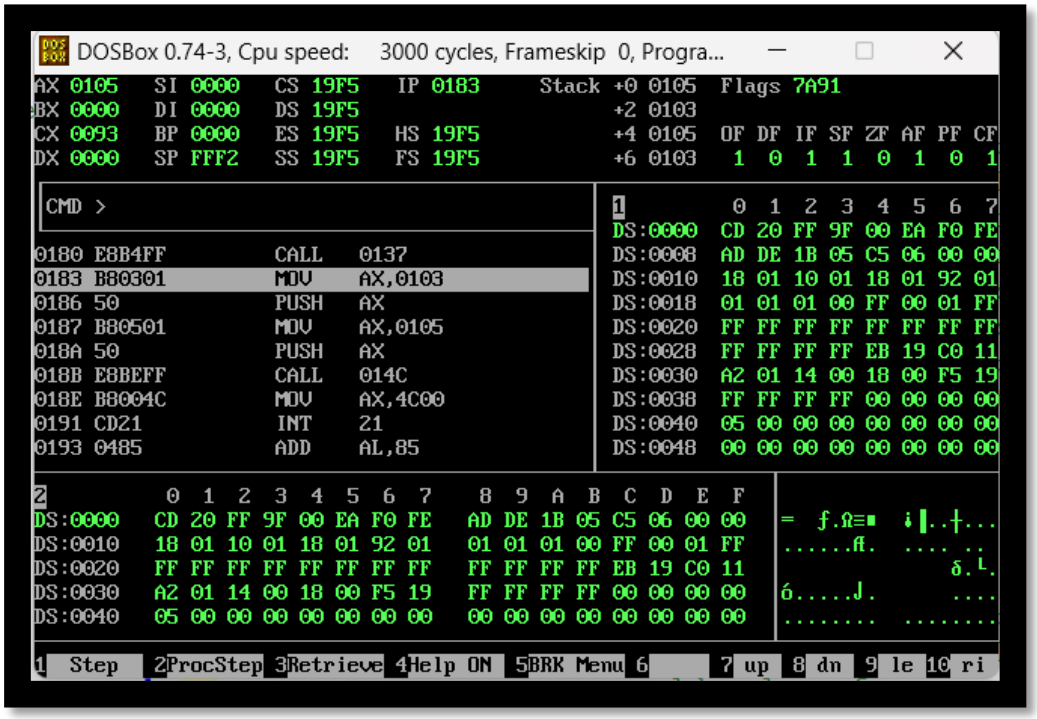

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0105 SI 0000 CS 19F5 IP 0174 Stack +0 0103 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0105
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0103 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF8 SS 19F5 FS 19F5 +6 0000 0 0 1 0 0 0 0 0
CMD >
0171 B80501 MOV AX,0105
0174 50 PUSH AX
0175 E8ABFF CALL 0123
0178 B80301 MOV AX,0103
017B 50 PUSH AX
017C B80501 MOV AX,0105
017F 50 PUSH AX
0180 E8B4FF CALL 0137
0183 B80301 MOV AX,0103
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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 F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

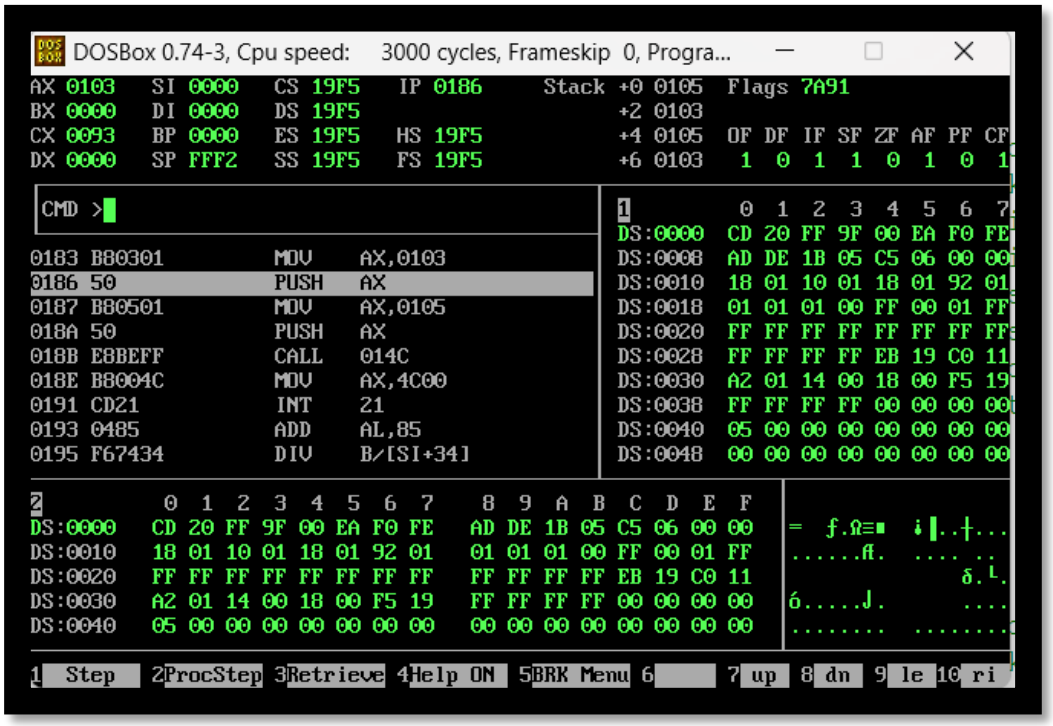

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0105 SI 0000 CS 19F5 IP 0175 Stack +0 0105 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0103
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0105 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF6 SS 19F5 FS 19F5 +6 0103 0 0 1 0 0 0 0 0
CMD >
0174 50 PUSH AX
0175 E8ABFF CALL 0123
0178 B80301 MOV AX,0103
017B 50 PUSH AX
017C B80501 MOV AX,0105
017F 50 PUSH AX
0180 E8B4FF CALL 0137
0183 B80301 MOV AX,0103
0186 50 PUSH AX
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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 F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

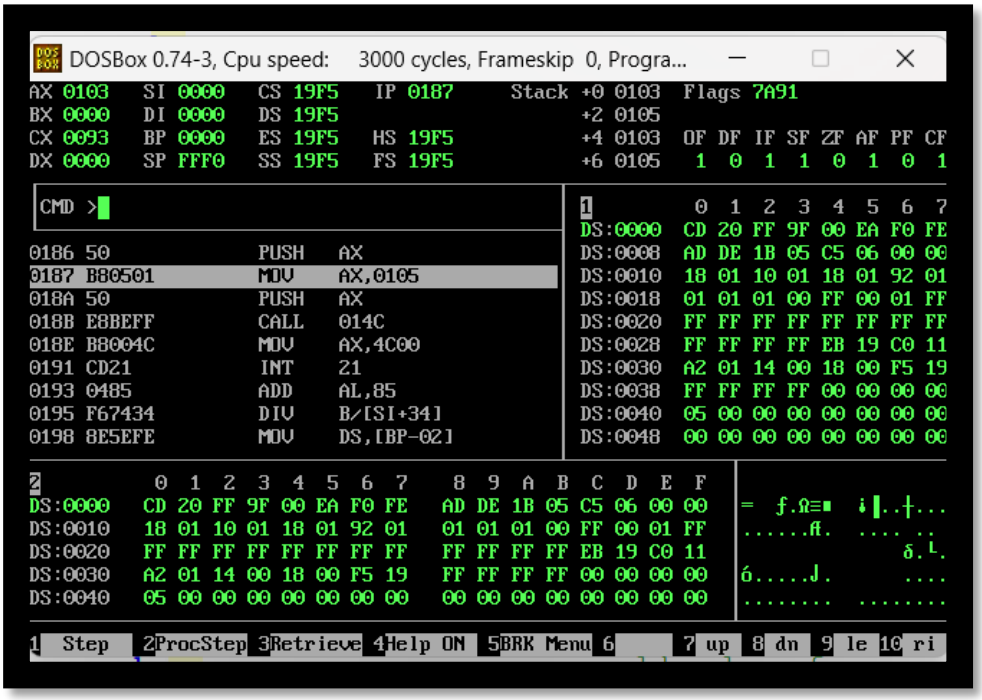

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0105 SI 0000 CS 19F5 IP 0178 Stack +0 0105 Flags 7291
BX 0000 DI 0000 DS 19F5 +2 0103
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0105 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF6 SS 19F5 FS 19F5 +6 0103 0 0 1 1 0 1 0 1
CMD >
0175 E8ABFF CALL 0123
0178 B80301 MOV AX,0103
017B 50 PUSH AX
017C B80501 MOV AX,0105
017F 50 PUSH AX
0180 E8B4FF CALL 0137
0183 B80301 MOV AX,0103
0186 50 PUSH AX
0187 B80501 MOV AX,0105
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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 F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

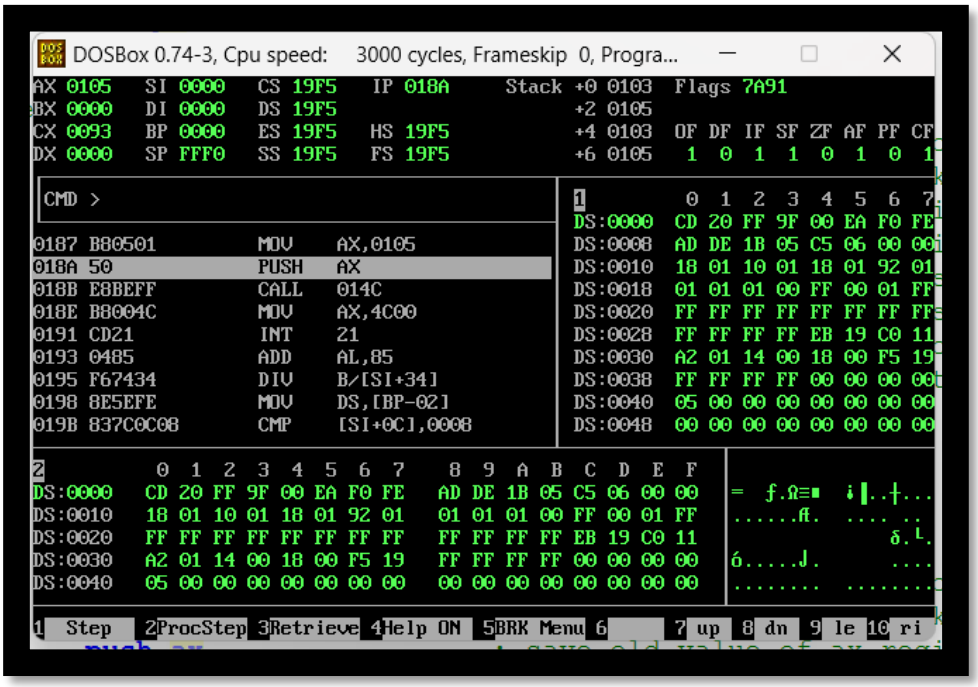

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0103 SI 0000 CS 19F5 IP 017B Stack +0 0105 Flags 7291
BX 0000 DI 0000 DS 19F5 +2 0103
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0105 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF6 SS 19F5 FS 19F5 +6 0103 0 0 1 1 0 1 0 1
CMD >
0178 B80301 MOV AX,0103
017B 50 PUSH AX
017C B80501 MOV AX,0105
017F 50 PUSH AX
0180 E8B4FF CALL 0137
0183 B80301 MOV AX,0103
0186 50 PUSH AX
0187 B80501 MOV AX,0105
018A 50 PUSH AX
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

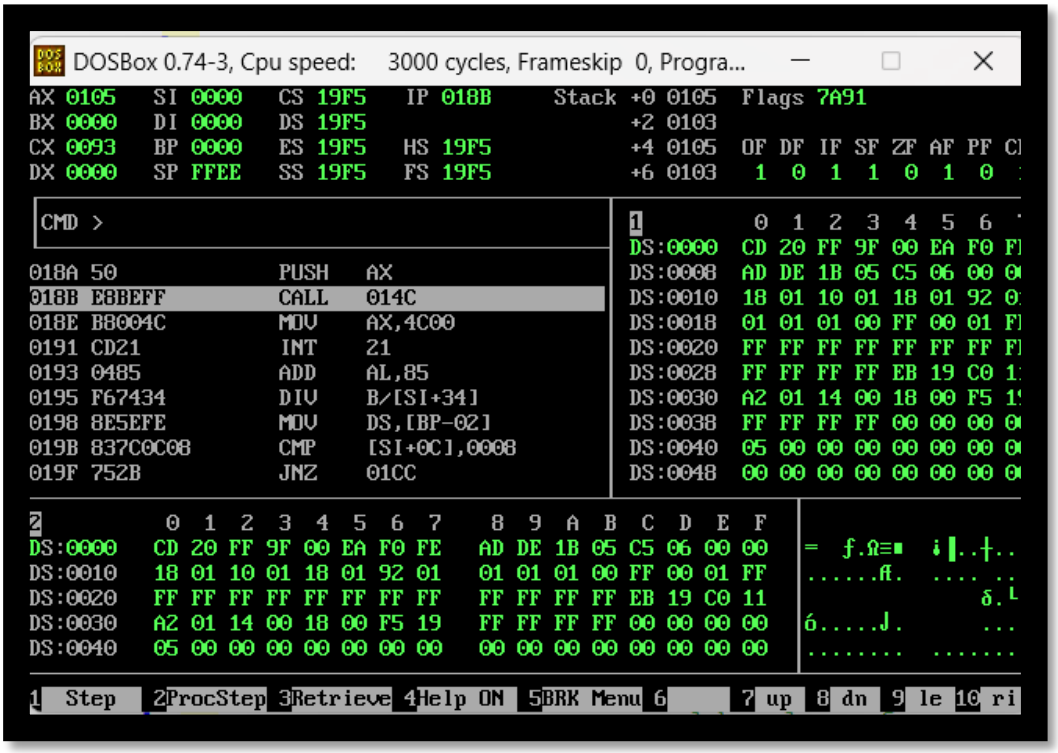

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0103 SI 0000 CS 19F5 IP 017C Stack +0 0103 Flags 7291
BX 0000 DI 0000 DS 19F5 +2 0105
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0103 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF4 SS 19F5 FS 19F5 +6 0105 0 0 1 1 0 1 0 1
CMD >
017B 50 PUSH AX
017C B80501 MOV AX,0105
017F 50 PUSH AX
0180 E8B4FF CALL 0137
0183 B80301 MOV AX,0103
0186 50 PUSH AX
0187 B80501 MOV AX,0105
018A 50 PUSH AX
018B E8BEFF CALL 014C
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

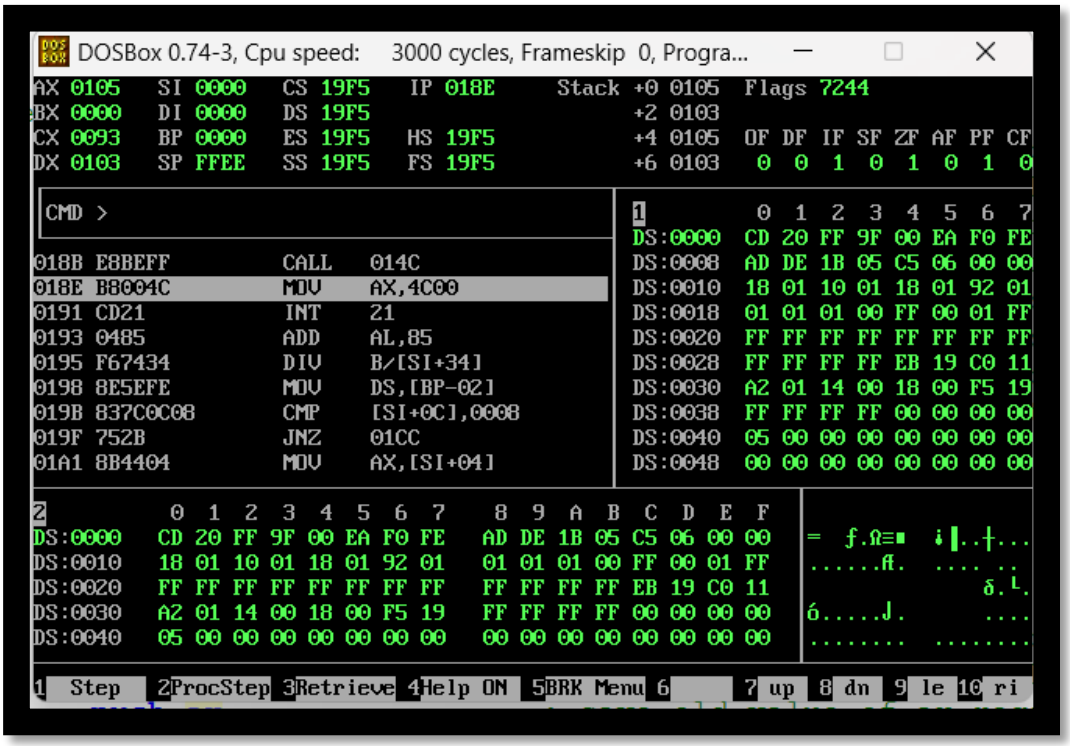

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0105 SI 0000 CS 19F5 IP 0183 Stack +0 0105 Flags 7A91
BX 0000 DI 0000 DS 19F5 +2 0103
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0105 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF2 SS 19F5 FS 19F5 +6 0103 1 0 1 1 0 1 0 1
CMD >
0180 E8B4FF CALL 0137
0183 B80301 MOV AX,0103
0186 50 PUSH AX
0187 B80501 MOV AX,0105
018A 50 PUSH AX
018B E8BEFF CALL 014C
018E B8004C MOV AX,4C00
0191 CD21 INT 21
0193 0485 ADD AL,85
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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 0103 SI 0000 CS 19F5 IP 0186 Stack +0 0105 Flags 7A91
BX 0000 DI 0000 DS 19F5 +2 0103
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0105 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF2 SS 19F5 FS 19F5 +6 0103 1 0 1 1 0 1 0 1
CMD >
0183 B80301 MOV AX,0103
0186 50 PUSH AX
0187 B80501 MOV AX,0105
018A 50 PUSH AX
018B E8BEFF CALL 014C
018E B8004C MOV AX,4C00
0191 CD21 INT 21
0193 0485 ADD AL,85
0195 F67434 DIV B/[SI+34]
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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 0103 SI 0000 CS 19F5 IP 0187 Stack +0 0103 Flags 7A91
BX 0000 DI 0000 DS 19F5 +2 0105
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0103 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF0 SS 19F5 FS 19F5 +6 0105 1 0 1 1 0 1 0 1
CMD >
0186 50 PUSH AX
0187 B80501 MOV AX,0105
018A 50 PUSH AX
018B E8BEFF CALL 014C
018E B8004C MOV AX,4C00
0191 CD21 INT 21
0193 0485 ADD AL,85
0195 F67434 DIV B/[SI+34]
0198 8E5EFE MOV DS,[BP-02]
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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 0105 SI 0000 CS 19F5 IP 018A Stack +0 0103 Flags 7A91
BX 0000 DI 0000 DS 19F5 +2 0105
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0103 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF0 SS 19F5 FS 19F5 +6 0105 1 0 1 1 0 1 0 1
CMD >
0187 B80501 MOV AX,0105
018A 50 PUSH AX
018B E8BEFF CALL 014C
018E B8004C MOV AX,4C00
0191 CD21 INT 21
0193 0485 ADD AL,85
0195 F67434 DIV B/[SI+34]
0198 8E5EFE MOV DS,[BP-02]
019B 837C0C08 CMP [SI+0C],0008
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 FF 00 01 FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

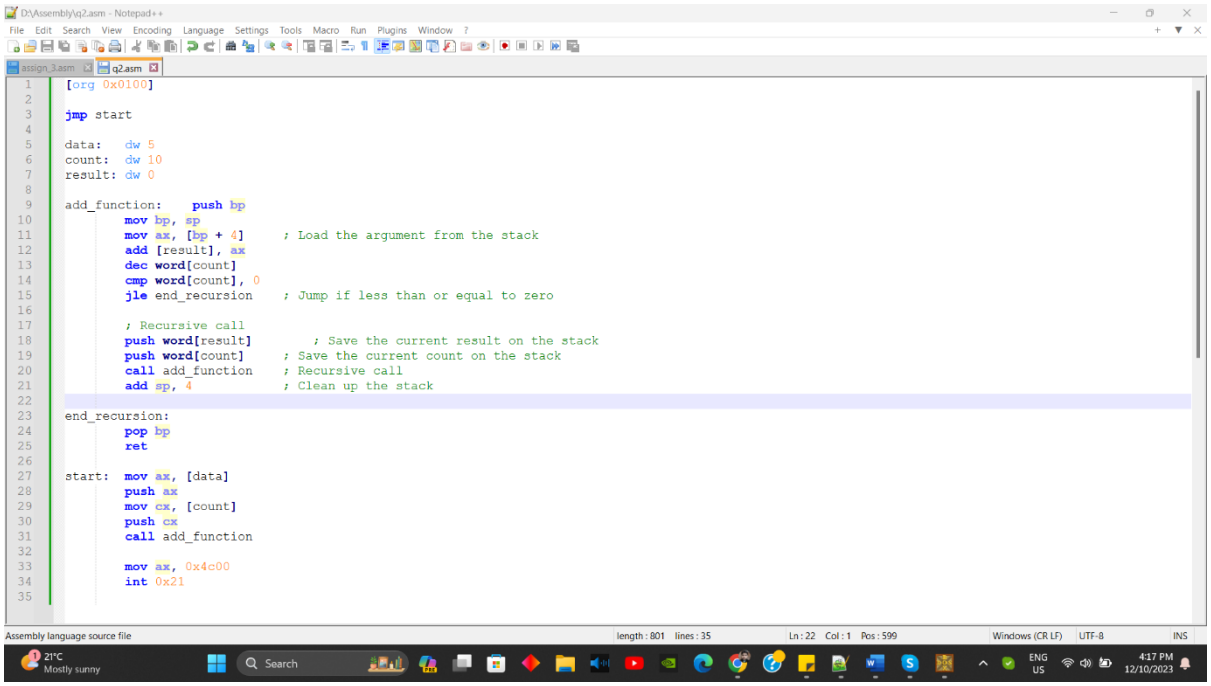




2. Perform recursion in assembly language using subroutines of your choice.

Answer:

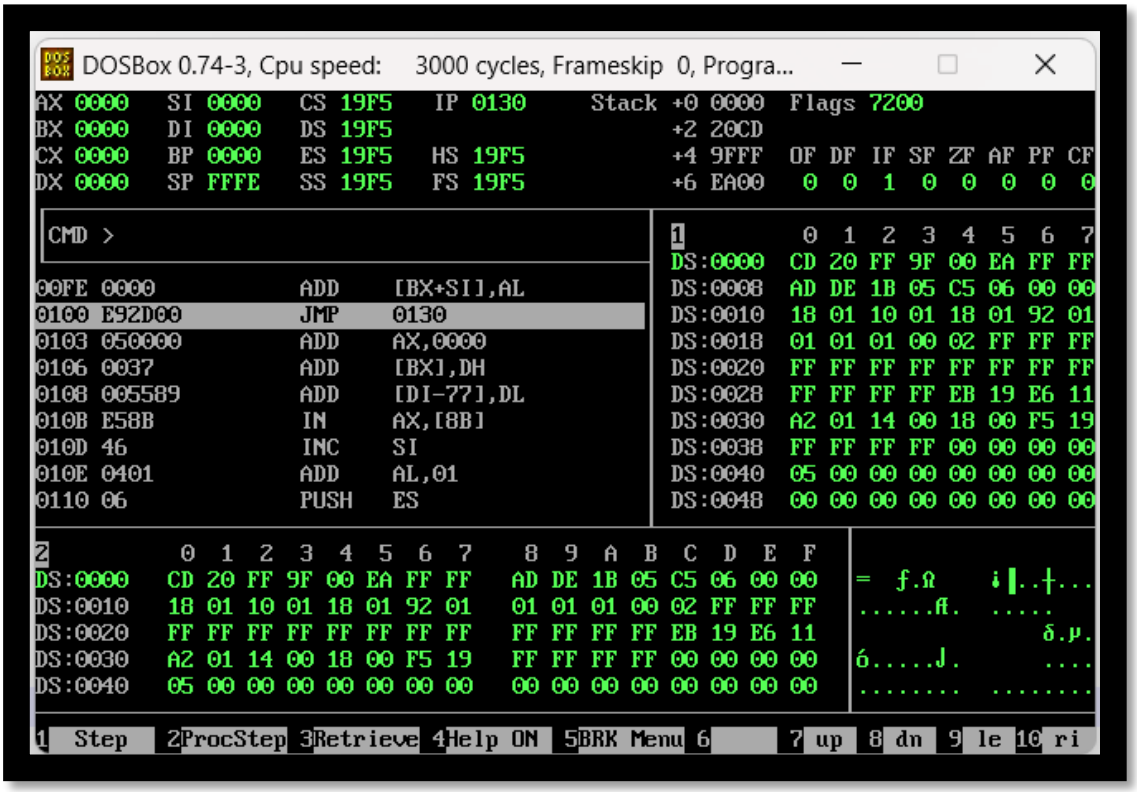
Code:

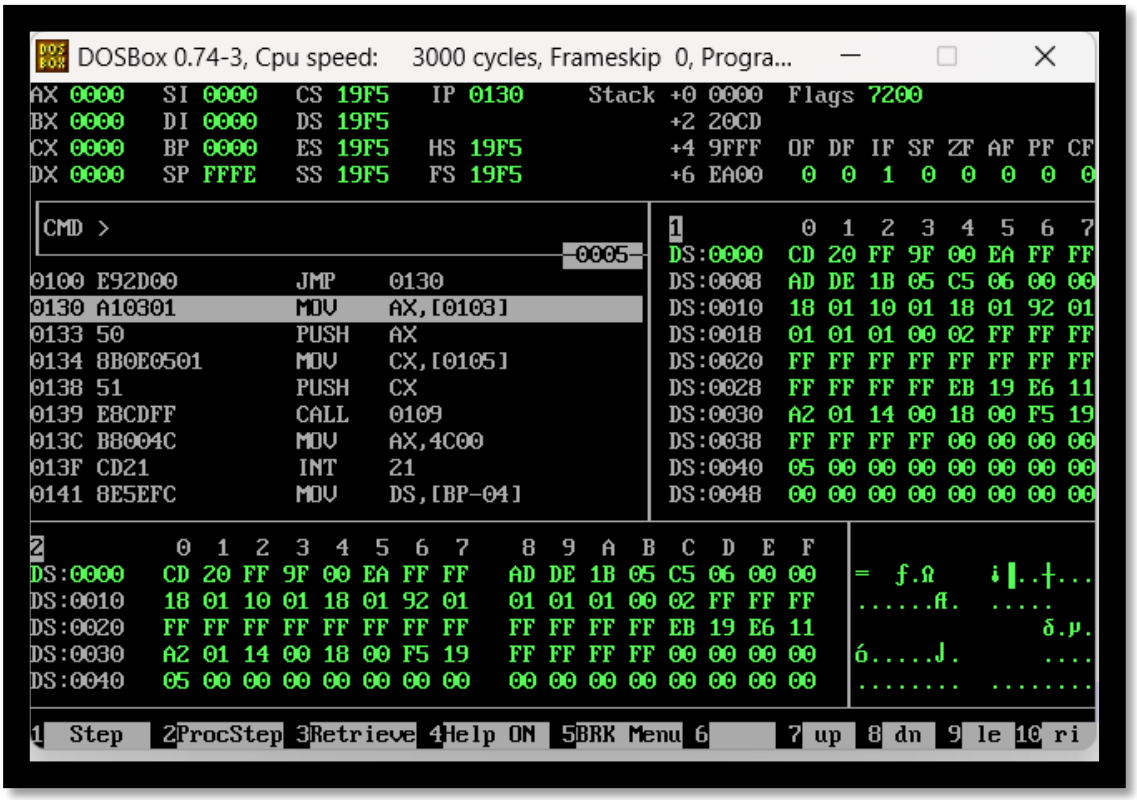


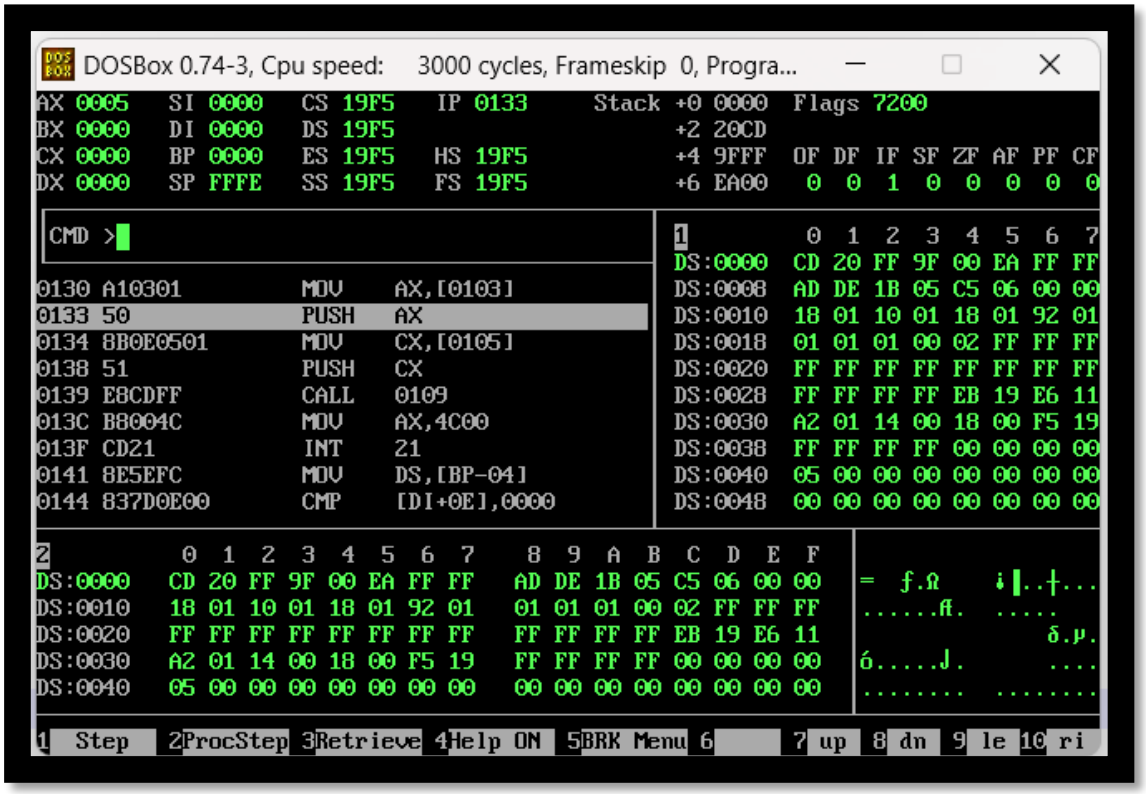
Explanation:

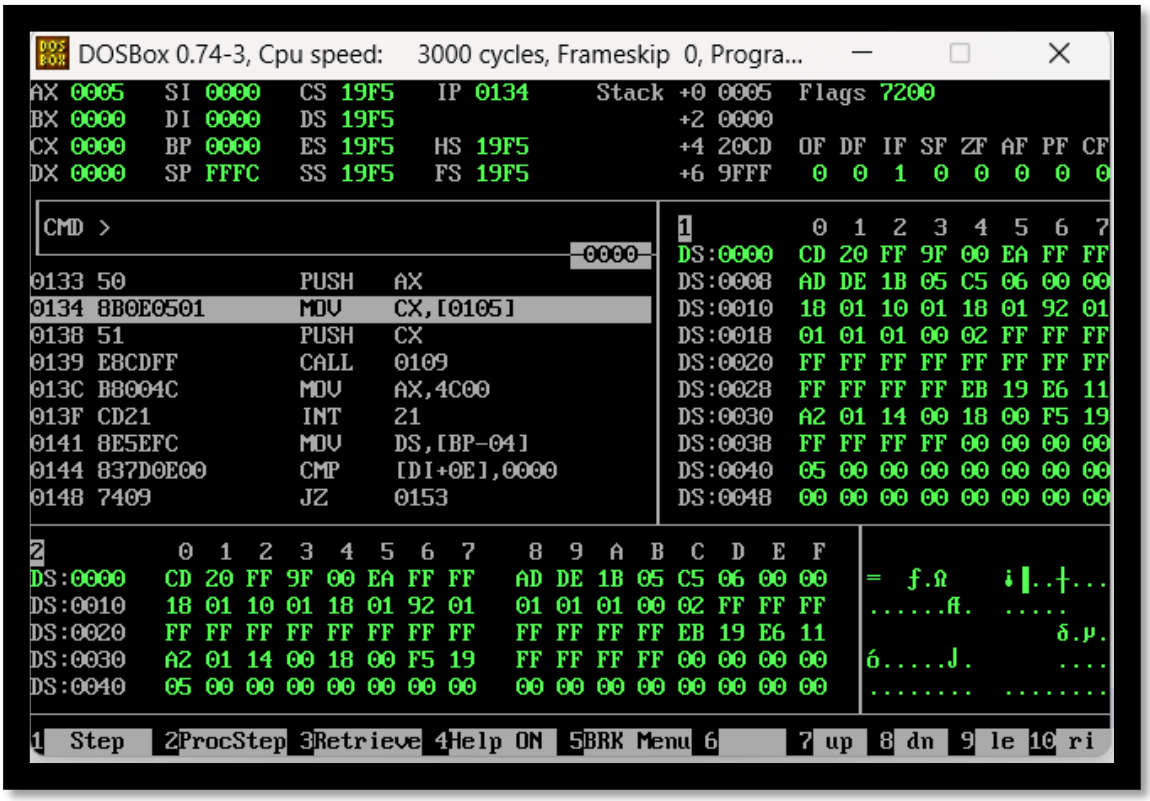
The provided assembly code is a concise implementation of recursive addition. It defines a function called add function that takes an argument from the stack, adds it to the result variable, decrements the count variable, and recursively calls itself until count becomes less than or equal to zero. The recursion is managed by saving and restoring the state (result and count) on the stack. The start section initializes data and count, pushes them onto the stack, and calls the add function. Finally, the program prints the result and exits gracefully. This code exemplifies a modular and clear approach to recursive addition in assembly language**.**

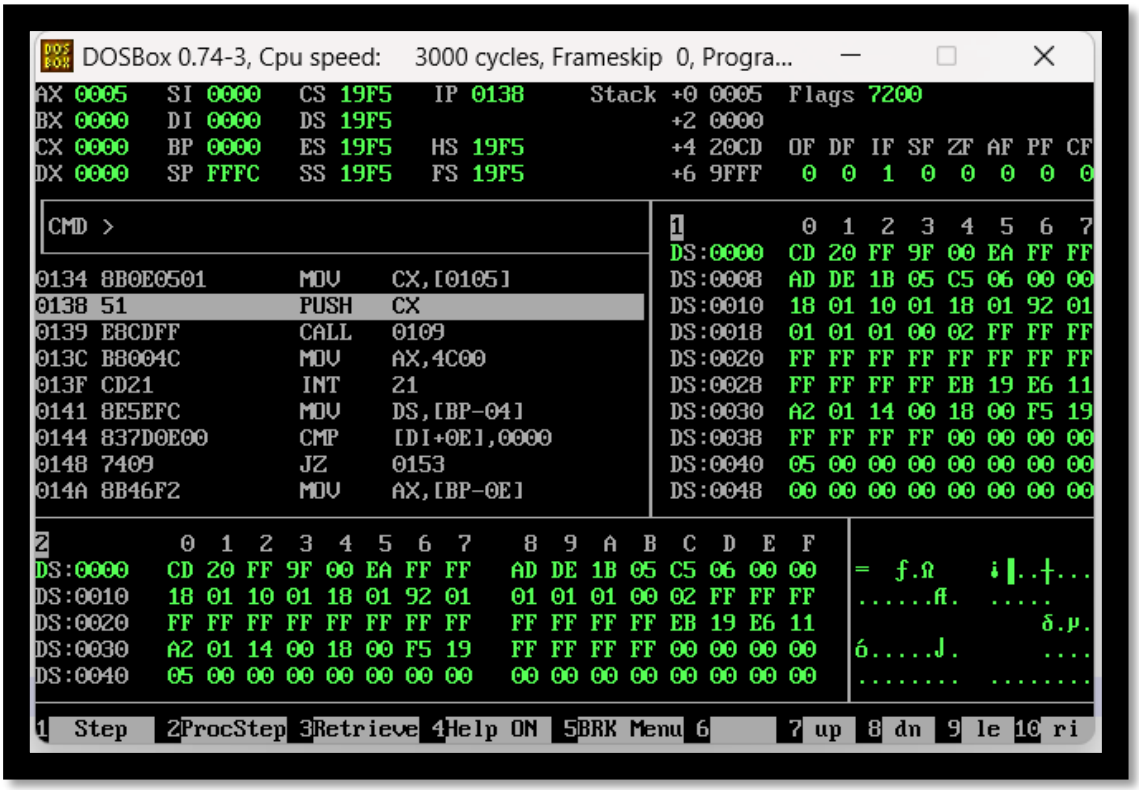
Screen-Shot of Debugger:

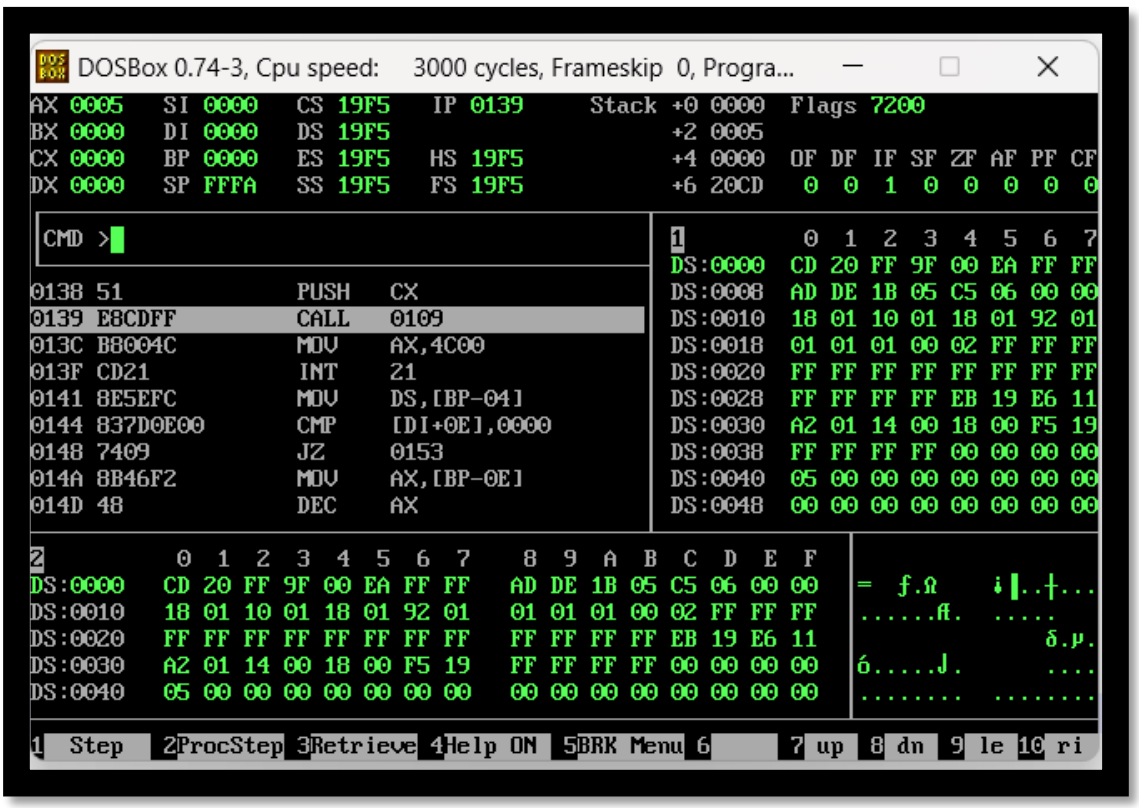

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0000 SI 0000 CS 19F5 IP 0130 Stack +0 0000 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
00FE 0000 ADD [BX+SI],AL
0100 E92D00 JMP 0130
0103 050000 ADD AX,0000
0106 0037 ADD [BX],DH
0108 005589 ADD [DI-77],DL
010B E58B IN AX,[8B]
010D 46 INC SI
010E 0401 ADD AL,01
0110 06 PUSH ES
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0000 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

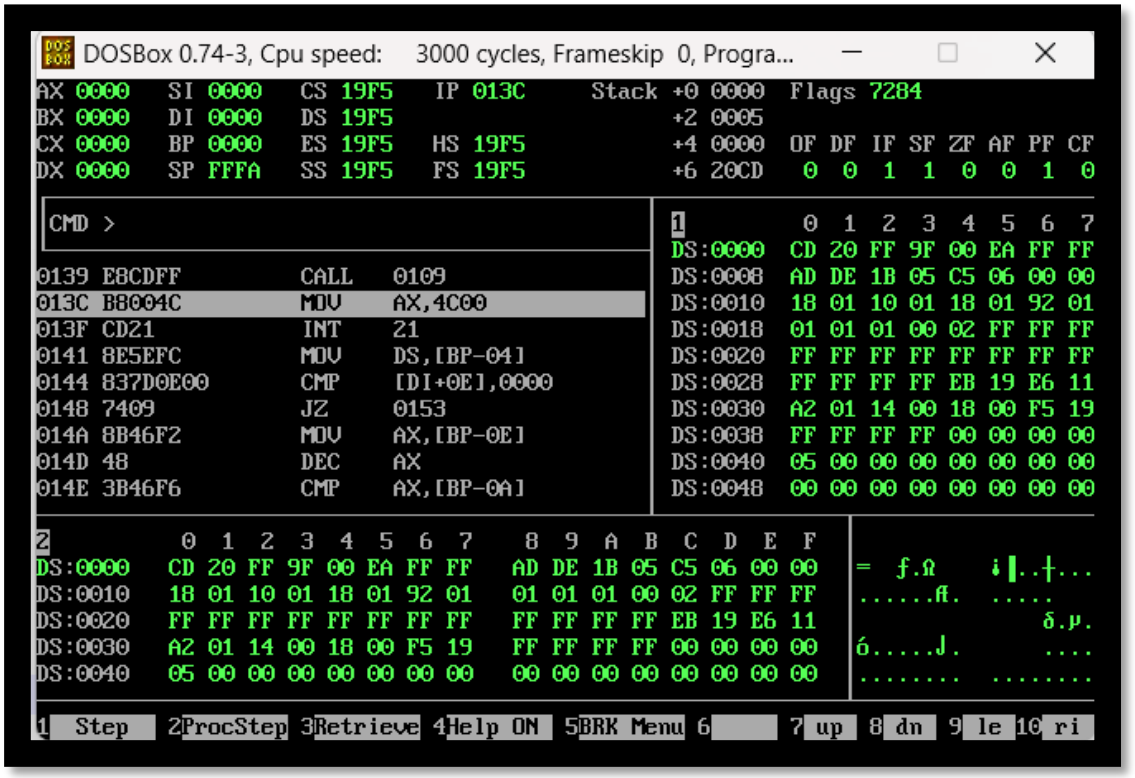

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0000 SI 0000 CS 19F5 IP 0130 Stack +0 0000 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
0005
0100 E92D00 JMP 0130
0130 A10301 MOV AX,[0103]
0133 50 PUSH AX
0134 8B0E0501 MOV CX,[0105]
0138 51 PUSH CX
0139 E8CDFF CALL 0109
013C B8004C MOV AX,4C00
013F CD21 INT 21
0141 8E5EFC MOV DS,[BP-04]
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00
DS:0000 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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

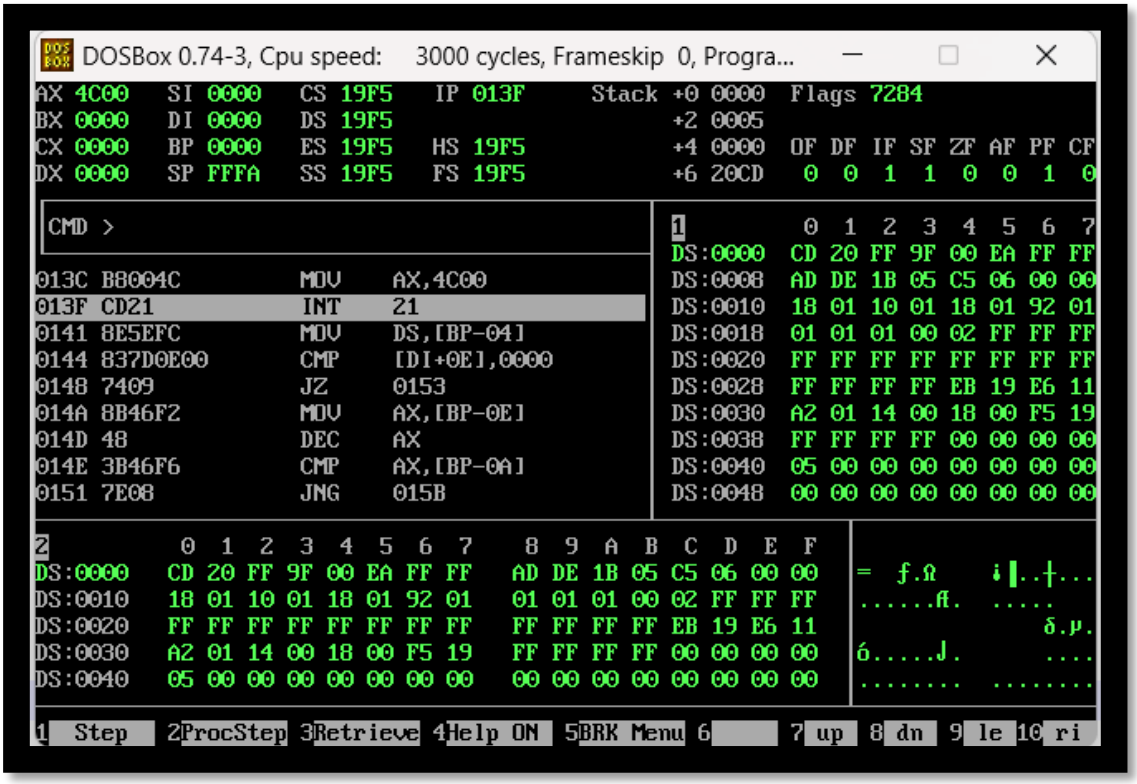

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0005 SI 0000 CS 19F5 IP 0133 Stack +0 0000 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
CMD >
0130 A10301 MOV AX,[0103]
0133 50 PUSH AX
0134 8B0E0501 MOV CX,[0105]
0138 51 PUSH CX
0139 E8CDFF CALL 0109
013C B8004C MOV AX,4C00
013F CD21 INT 21
0141 8E5EFC MOV DS,[BP-04]
0144 837D0E00 CMP [DI+0E],0000
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 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

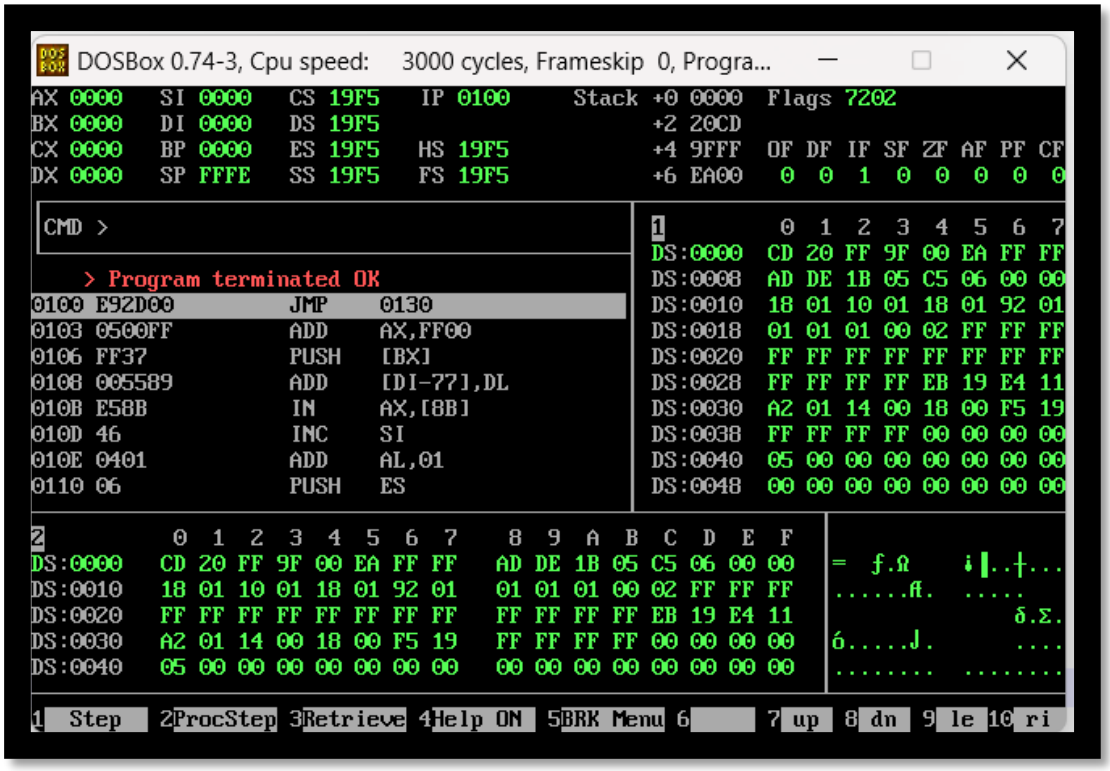

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0005 SI 0000 CS 19F5 IP 0134 Stack +0 0005 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0000
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 20CD OF DF IF SF ZF AF PF CF
DX 0000 SP FFFC SS 19F5 FS 19F5 +6 9FFF 0 0 1 0 0 0 0 0
CMD >
0133 50 PUSH AX
0134 8B0E0501 MOV CX,[0105]
0138 51 PUSH CX
0139 E8CDFF CALL 0109
013C B8004C MOV AX,4C00
013F CD21 INT 21
0141 8E5EFC MOV DS,[BP-04]
0144 837D0E00 CMP [DI+0E],0000
0148 7409 JZ 0153
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 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 0005 SI 0000 CS 19F5 IP 0138 Stack +0 0005 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0000
CX 0000 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 >
0134 8B0E0501 MOV CX,[0105]
0138 51 PUSH CX
0139 E8CDFF CALL 0109
013C B8004C MOV AX,4C00
013F CD21 INT 21
0141 8E5EFC MOV DS,[BP-04]
0144 837D0E00 CMP [DI+0E],0000
0148 7409 JZ 0153
014A 8B46F2 MOV AX,[BP-0E]
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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 0005 SI 0000 CS 19F5 IP 0139 Stack +0 0000 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0005
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 0000 OF DF IF SF ZF AF PF CF
DX 0000 SP FFFA SS 19F5 FS 19F5 +6 20CD 0 0 1 0 0 0 0 0
CMD >
0138 51 PUSH CX
0139 E8CDFF CALL 0109
013C B8004C MOV AX,4C00
013F CD21 INT 21
0141 8E5EFC MOV DS,[BP-04]
0144 837D0E00 CMP [DI+0E],0000
0148 7409 JZ 0153
014A 8B46F2 MOV AX,[BP-0E]
014D 48 DEC AX
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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 0000 SI 0000 CS 19F5 IP 013C Stack +0 0000 Flags 7284
BX 0000 DI 0000 DS 19F5 +2 0005
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 0000 OF DF IF SF ZF AF PF CF
DX 0000 SP FFFA SS 19F5 FS 19F5 +6 20CD 0 0 1 1 0 0 1 0
CMD >
0139 E8CDFF CALL 0109
013C B8004C MOV AX,4C00
013F CD21 INT 21
0141 8E5EFC MOV DS,[BP-04]
0144 837D0E00 CMP [DI+0E],0000
0148 7409 JZ 0153
014A 8B46F2 MOV AX,[BP-0E]
014D 48 DEC AX
014E 3B46F6 CMP AX,[BP-0A]
DS:0000 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 00
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DS:0048 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 4C00 SI 0000 CS 19F5 IP 013F Stack +0 0000 Flags 7284
BX 0000 DI 0000 DS 19F5 +2 0005
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 0000 OF DF IF SF ZF AF PF CF
DX 0000 SP FFFA SS 19F5 FS 19F5 +6 20CD 0 0 1 1 0 0 1 0
CMD >
013C B8004C MOV AX,4C00
013F CD21 INT 21
0141 8E5EFC MOV DS,[BP-04]
0144 837D0E00 CMP [DI+0E],0000
0148 7409 JZ 0153
014A 8B46F2 MOV AX,[BP-0E]
014D 48 DEC AX
014E 3B46F6 CMP AX,[BP-0A]
0151 7E08 JNG 015B
DS:0000 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 00
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E6 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DS:0048 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 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 >
> Program terminated OK
0100 E92D00 JMP 0130
0103 0500FF ADD AX,FF00
0106 FF37 PUSH [BX]
0108 005589 ADD [DI-77],DL
010B E58B IN AX,[8B]
010D 46 INC SI
010E 0401 ADD AL,01
0110 06 PUSH ES
1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA FF FF
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 E4 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
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
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E4 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
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