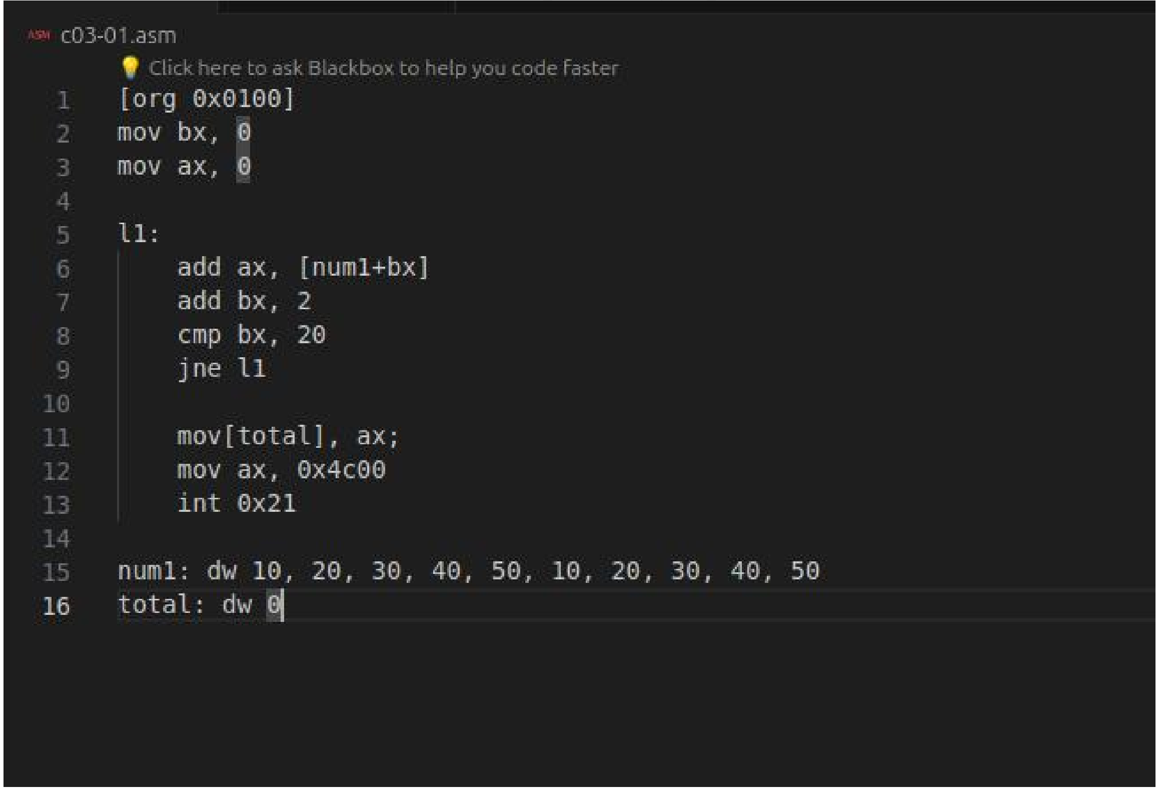
Name : M Sarmad Khan
Roll-Number: 22P-9009

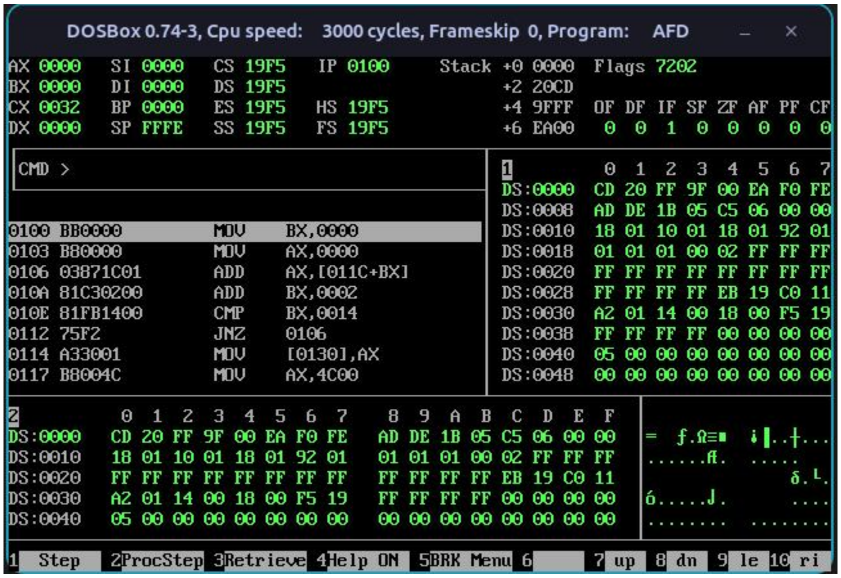
Lab Task # 06
EXAMPLE 3.1:

1. **Start by opening the code file named c03-01.asm.**
2. **Execute the code and test it while providing explanations.**



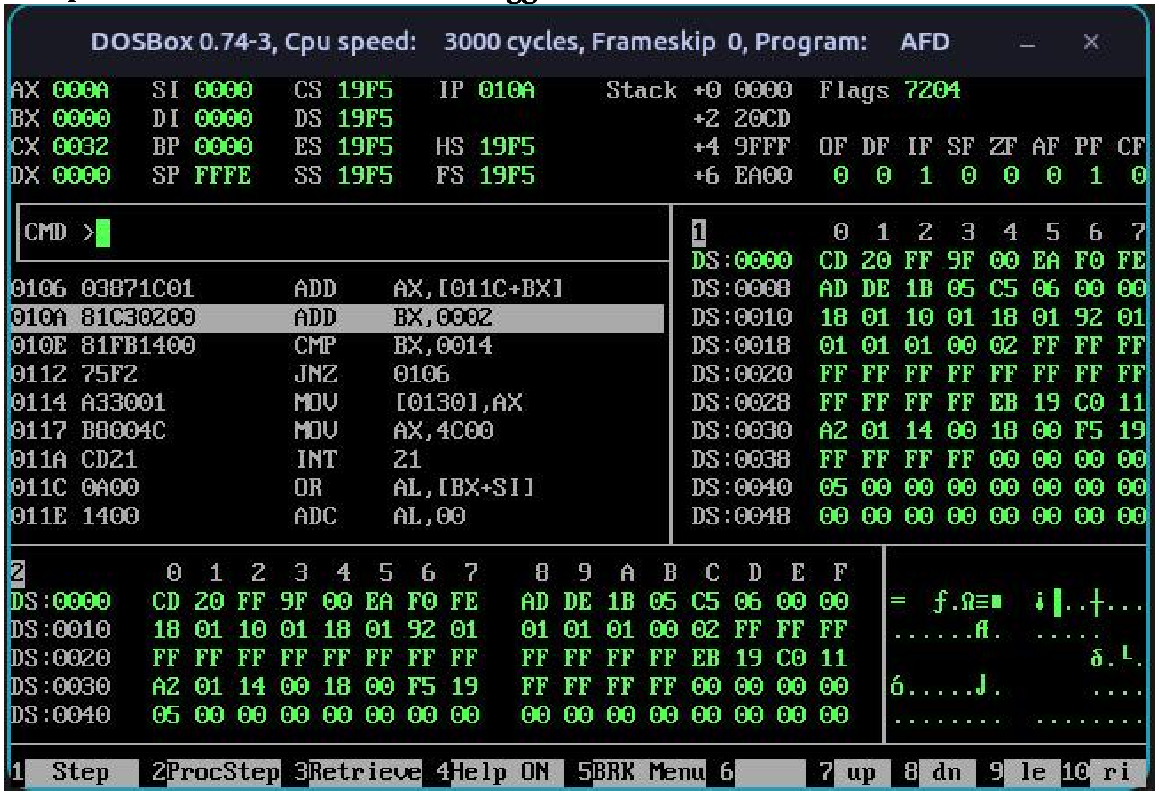
```
[org 0x0100]
mov bx, 0
mov ax, 0

l1:
    add ax, [num1+bx]
    add bx, 2
    cmp bx, 20
    jne l1

    mov[total], ax;
    mov ax, 0x4c00
    int 0x21

num1: dw 10, 20, 30, 40, 50, 10, 20, 30, 40, 50
total: dw 0
```

3. **Examine the code first.**


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

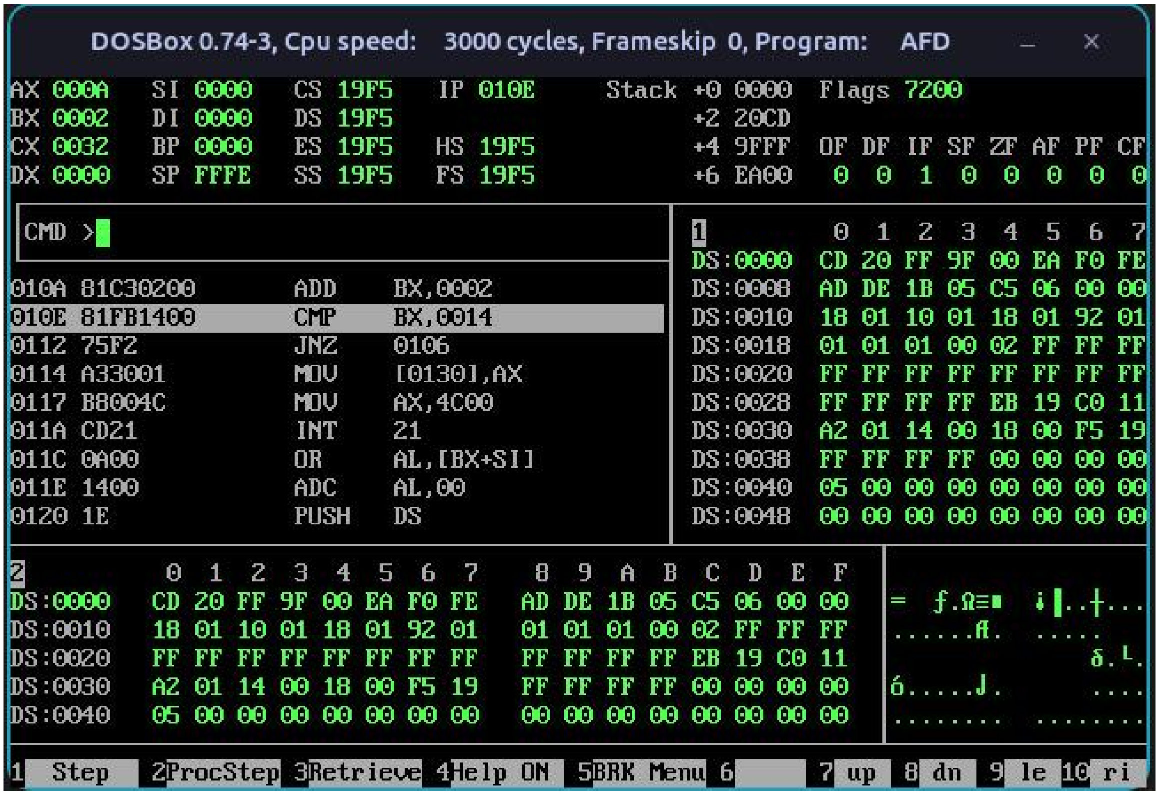
```
AX 0000   SI 0000   CS 19F5   IP 0100     Stack +0 0000  Flags 7202
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0032   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                         1          0  1  2  3  4  5  6  7
                                               DS:0000  CD 20 FF 9F 00 EA F0 FE
                                               DS:0008  AD DE 1B 05 C5 06 00 00
0100 BB0000        MOV     BX,0000             DS:0010  18 01 10 01 18 01 92 01
0103 B80000        MOV     AX,0000             DS:0018  01 01 01 00 02 FF FF FF
0106 03871C01      ADD     AX,[011C+BX]        DS:0020  FF FF FF FF FF FF FF FF
010A 81C30200      ADD     BX,0002             DS:0028  FF FF FF FF EB 19 C0 11
010E 81FB1400      CMP     BX,0014             DS:0030  A2 01 14 00 18 00 F5 19
0112 75F2          JNZ     0106                DS:0038  FF FF FF FF 00 00 00 00
0114 A33001        MOV     [0130],AX           DS:0040  05 00 00 00 00 00 00 00
0117 B8004C        MOV     AX,4C00             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  =  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  ........  ........
```

4. Proceed to load it into the debugger and execute the first two lines.


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

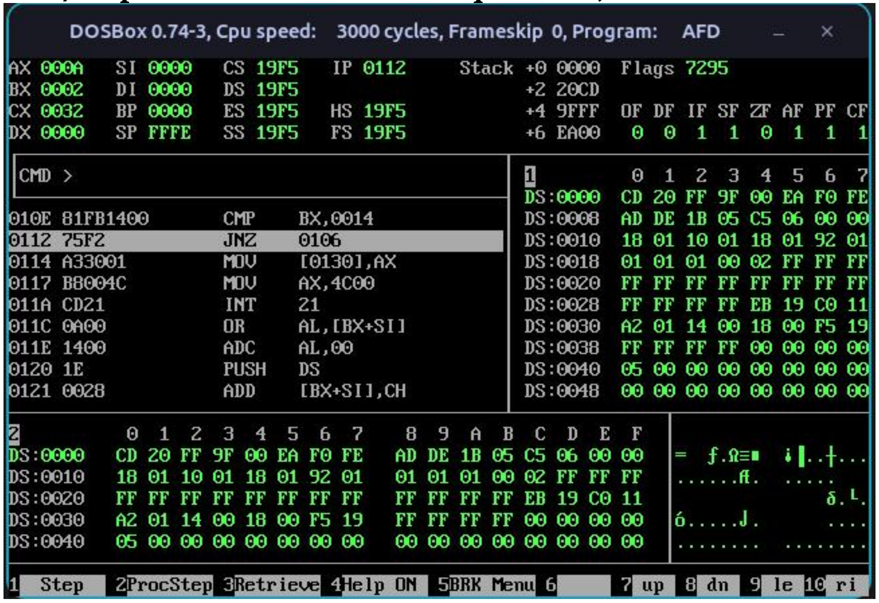
```
AX 000A   SI 0000   CS 19F5   IP 010A     Stack +0 0000  Flags 7204
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0032   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  0  0  1  0

 CMD >                                         1          0  1  2  3  4  5  6  7
                                               DS:0000  CD 20 FF 9F 00 EA F0 FE
0106 03871C01      ADD     AX,[011C+BX]        DS:0008  AD DE 1B 05 C5 06 00 00
010A 81C30200      ADD     BX,0002             DS:0010  18 01 10 01 18 01 92 01
010E 81FB1400      CMP     BX,0014             DS:0018  01 01 01 00 02 FF FF FF
0112 75F2          JNZ     0106                DS:0020  FF FF FF FF FF FF FF FF
0114 A33001        MOV     [0130],AX           DS:0028  FF FF FF FF EB 19 C0 11
0117 B8004C        MOV     AX,4C00             DS:0030  A2 01 14 00 18 00 F5 19
011A CD21          INT     21                  DS:0038  FF FF FF FF 00 00 00 00
011C 0A00          OR      AL,[BX+SI]          DS:0040  05 00 00 00 00 00 00 00
011E 1400          ADC     AL,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  =  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  ........  ........
```

5. Upon execution, notice that AX and BX are initialized with 0 values.
6. Move on to the next instruction: 000A is stored into AX, representing the value 10.



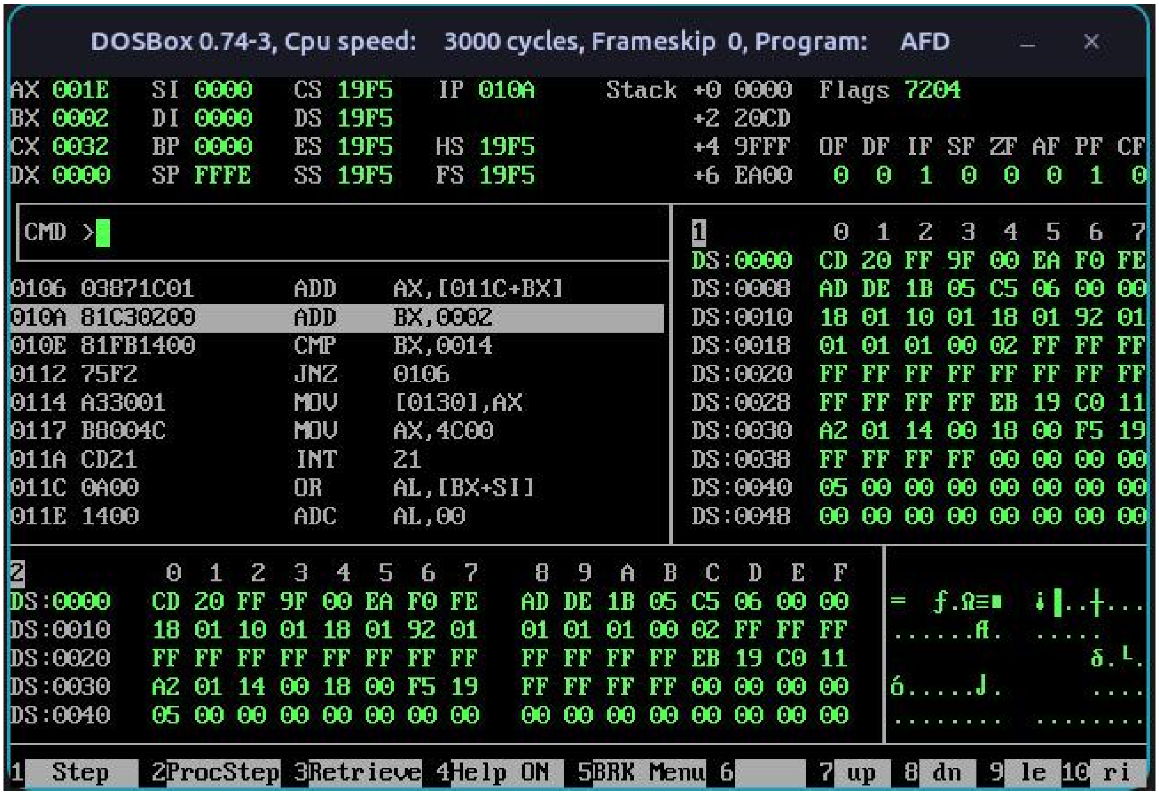
```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:    AFD
AX 000A   SI 0000   CS 19F5   IP 010E    Stack +0 0000  Flags 7200
BX 0002   DI 0000   DS 19F5                    +2 20CD
CX 0032   BP 0000   ES 19F5   HS 19F5          +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   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 F0 FE
010A 81C30200      ADD     BX,0002       DS:0008  AD DE 1B 05 C5 06 00 00
010E 81FB1400      CMP     BX,0014       DS:0010  18 01 10 01 18 01 92 01
0112 75F2          JNZ     0106          DS:0018  01 01 01 00 02 FF FF FF
0114 A33001        MOV     [0130],AX     DS:0020  FF FF FF FF FF FF FF FF
0117 B8004C        MOV     AX,4C00       DS:0028  FF FF FF FF EB 19 C0 11
011A CD21          INT     21            DS:0030  A2 01 14 00 18 00 F5 19
011C 0A00          OR      AL,[BX+SI]    DS:0038  FF FF FF FF 00 00 00 00
011E 1400          ADC     AL,00         DS:0040  05 00 00 00 00 00 00 00
0120 1E            PUSH    DS            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  =  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  ........  ........

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

7. **Upon execution of the subsequent line, the value of BX becomes 0002.**



```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:    AFD
AX 000A   SI 0000   CS 19F5   IP 0112    Stack +0 0000  Flags 7295
BX 0002   DI 0000   DS 19F5                    +2 20CD
CX 0032   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        0  1  2  3  4  5  6  7
                                         DS:0000  CD 20 FF 9F 00 EA F0 FE
010E 81FB1400      CMP     BX,0014       DS:0008  AD DE 1B 05 C5 06 00 00
0112 75F2          JNZ     0106          DS:0010  18 01 10 01 18 01 92 01
0114 A33001        MOV     [0130],AX     DS:0018  01 01 01 00 02 FF FF FF
0117 B8004C        MOV     AX,4C00       DS:0020  FF FF FF FF FF FF FF FF
011A CD21          INT     21            DS:0028  FF FF FF FF EB 19 C0 11
011C 0A00          OR      AL,[BX+SI]    DS:0030  A2 01 14 00 18 00 F5 19
011E 1400          ADC     AL,00         DS:0038  FF FF FF FF 00 00 00 00
0120 1E            PUSH    DS            DS:0040  05 00 00 00 00 00 00 00
0121 0028          ADD     [BX+SI],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 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  ........  ........

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

8. **Now, observe the jump and increment in the loop by modifying the value of BX.**

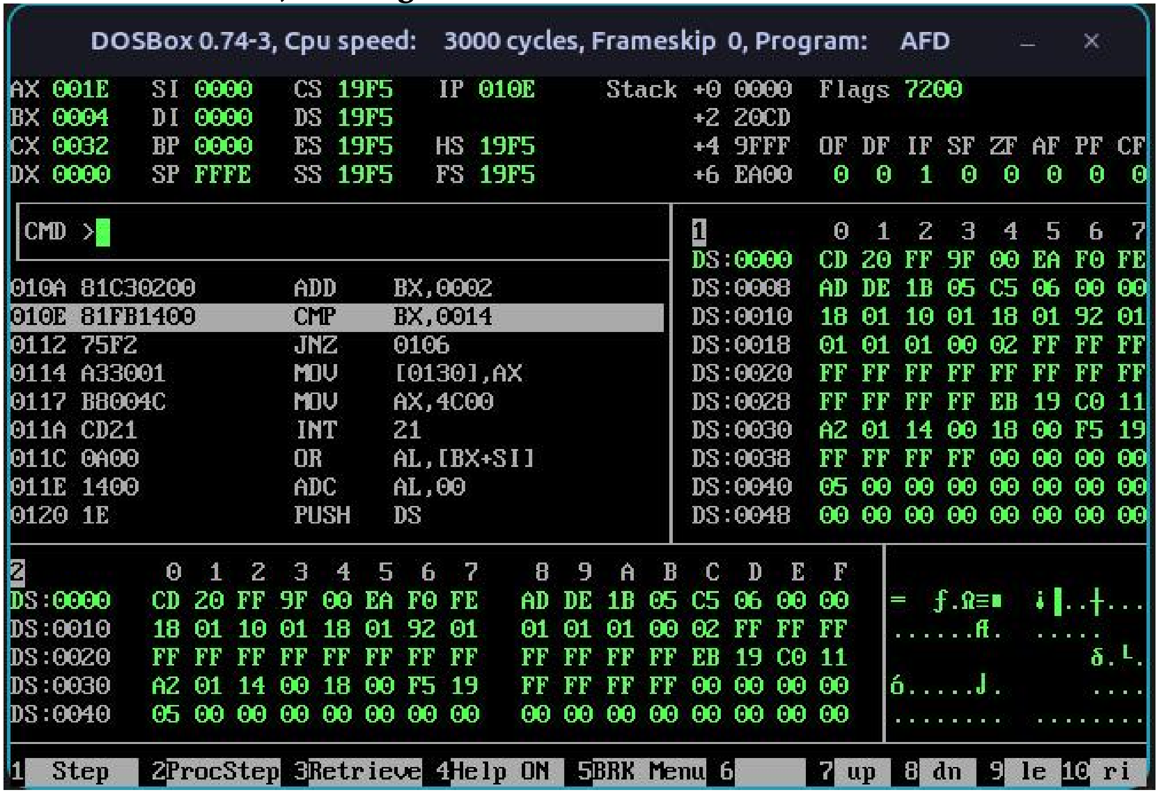


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

```
AX 001E   SI 0000   CS 19F5   IP 010A     Stack +0 0000  Flags 7204
BX 0002   DI 0000   DS 19F5                     +2 20CD
CX 0032   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  1  0

 CMD >                                        1       0  1  2  3  4  5  6  7
                                              DS:0000 CD 20 FF 9F 00 EA F0 FE
0106 03871C01     ADD   AX,[011C+BX]          DS:0008 AD DE 1B 05 C5 06 00 00
010A 81C30200     ADD   BX,0002               DS:0010 18 01 10 01 18 01 92 01
010E 81FB1400     CMP   BX,0014               DS:0018 01 01 01 00 02 FF FF FF
0112 75F2         JNZ   0106                  DS:0020 FF FF FF FF FF FF FF FF
0114 A33001       MOV   [0130],AX             DS:0028 FF FF FF FF EB 19 C0 11
0117 B8004C       MOV   AX,4C00               DS:0030 A2 01 14 00 18 00 F5 19
011A CD21         INT   21                    DS:0038 FF FF FF FF 00 00 00 00
011C 0A00         OR    AL,[BX+SI]            DS:0040 05 00 00 00 00 00 00 00
011E 1400         ADC   AL,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   =  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   ........ ........
```

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

9. Check the loop condition as depicted in the figure above.
10. Add the change in the value of BX to the address of AX, and then store the result back into AX, resulting in the value 001E.

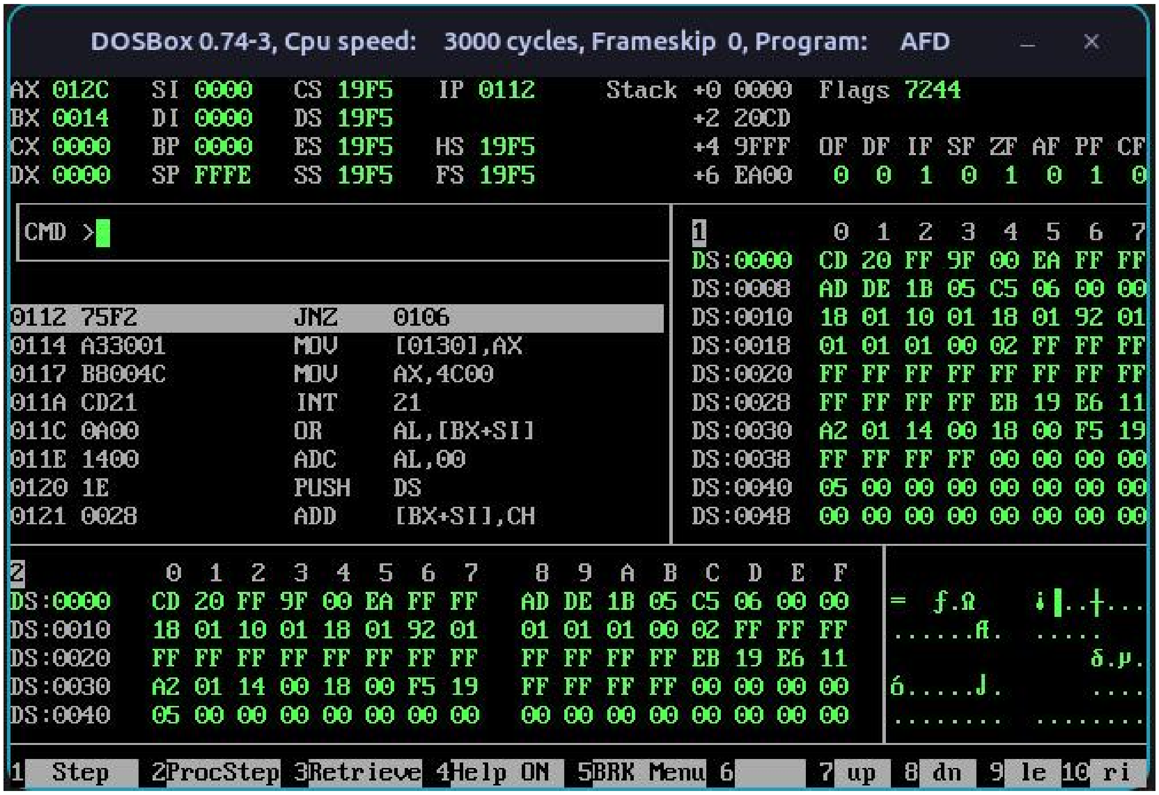


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

```
AX 001E   SI 0000   CS 19F5   IP 010E     Stack +0 0000  Flags 7200
BX 0004   DI 0000   DS 19F5                     +2 20CD
CX 0032   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 F0 FE
010A 81C30200     ADD   BX,0002               DS:0008 AD DE 1B 05 C5 06 00 00
010E 81FB1400     CMP   BX,0014               DS:0010 18 01 10 01 18 01 92 01
0112 75F2         JNZ   0106                  DS:0018 01 01 01 00 02 FF FF FF
0114 A33001       MOV   [0130],AX             DS:0020 FF FF FF FF FF FF FF FF
0117 B8004C       MOV   AX,4C00               DS:0028 FF FF FF FF EB 19 C0 11
011A CD21         INT   21                    DS:0030 A2 01 14 00 18 00 F5 19
011C 0A00         OR    AL,[BX+SI]            DS:0038 FF FF FF FF 00 00 00 00
011E 1400         ADC   AL,00                 DS:0040 05 00 00 00 00 00 00 00
0120 1E           PUSH  DS                    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   =  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   ........ ........
```

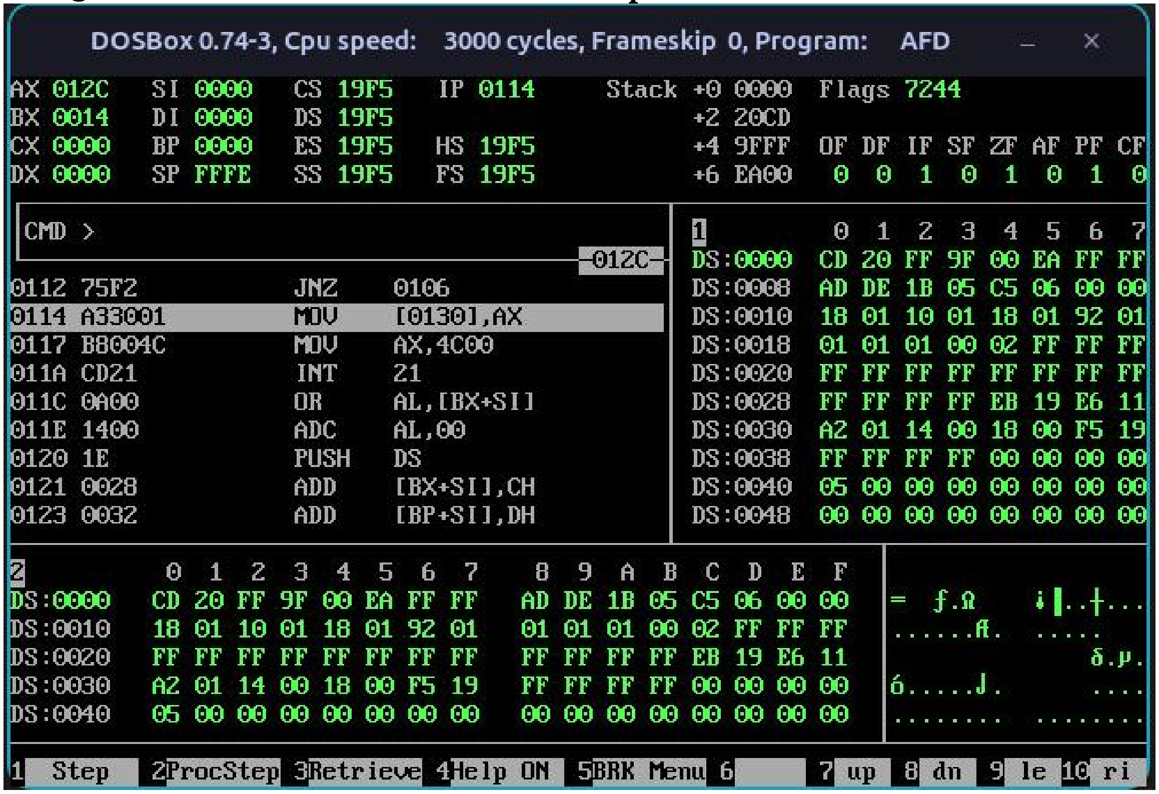
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

11. Increment BX again.



```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 012C   SI 0000   CS 19F5   IP 0112     Stack +0 0000  Flags 7244
BX 0014   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  1  0  1  0

 CMD >                                        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
0112 75F2           JNZ    0106               DS:0010  18 01 10 01 18 01 92 01
0114 A33001         MOV    [0130],AX          DS:0018  01 01 01 00 02 FF FF FF
0117 B8004C         MOV    AX,4C00            DS:0020  FF FF FF FF FF FF FF FF
011A CD21           INT    21                 DS:0028  FF FF FF FF EB 19 E6 11
011C 0A00           OR     AL,[BX+SI]         DS:0030  A2 01 14 00 18 00 F5 19
011E 1400           ADC    AL,00              DS:0038  FF FF FF FF 00 00 00 00
0120 1E             PUSH   DS                 DS:0040  05 00 00 00 00 00 00 00
0121 0028           ADD    [BX+SI],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 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.....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
```

12. Recheck the loop condition to determine if it meets the criteria for BX. If yes, break the loop; otherwise, return to AX and execute statements in line until the loop terminates.
13. The final value stored in AX after the loop breaks is noted.

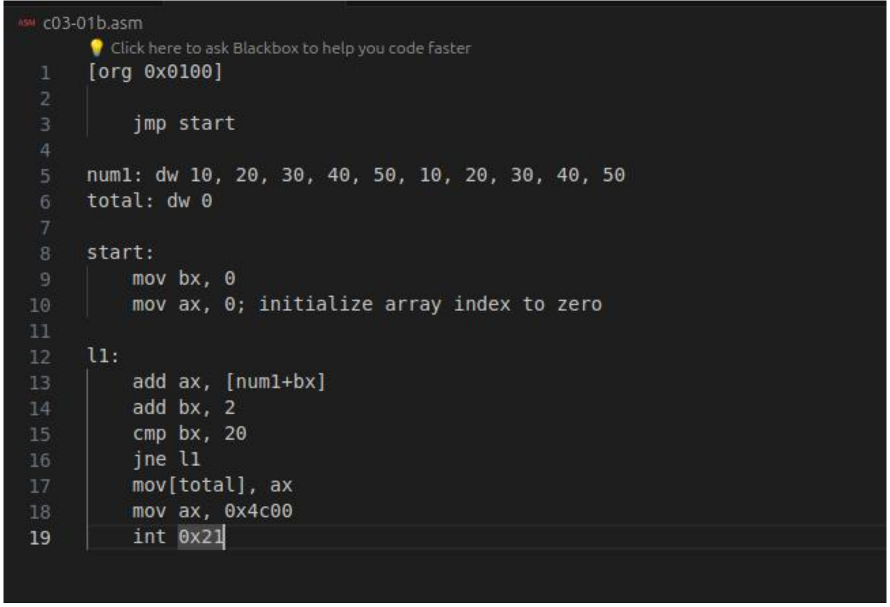


```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 012C   SI 0000   CS 19F5   IP 0114     Stack +0 0000  Flags 7244
BX 0014   DI 0000   DS 19F5                     +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  0  1  0  1  0

 CMD >                                        1         0  1  2  3  4  5  6  7
                                      -012C-  DS:0000  CD 20 FF 9F 00 EA FF FF
0112 75F2           JNZ    0106               DS:0008  AD DE 1B 05 C5 06 00 00
0114 A33001         MOV    [0130],AX          DS:0010  18 01 10 01 18 01 92 01
0117 B8004C         MOV    AX,4C00            DS:0018  01 01 01 00 02 FF FF FF
011A CD21           INT    21                 DS:0020  FF FF FF FF FF FF FF FF
011C 0A00           OR     AL,[BX+SI]         DS:0028  FF FF FF FF EB 19 E6 11
011E 1400           ADC    AL,00              DS:0030  A2 01 14 00 18 00 F5 19
0120 1E             PUSH   DS                 DS:0038  FF FF FF FF 00 00 00 00
0121 0028           ADD    [BX+SI],CH         DS:0040  05 00 00 00 00 00 00 00
0123 0032           ADD    [BP+SI],DH         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.....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
```

14. Move the value of AX into another memory address.
15. Upon completing these statements, intercept the program and exit.

EXAMPLE 3.2:

- Example 3.2 utilizes similar code but with a minor alteration involving an unconditional jump.

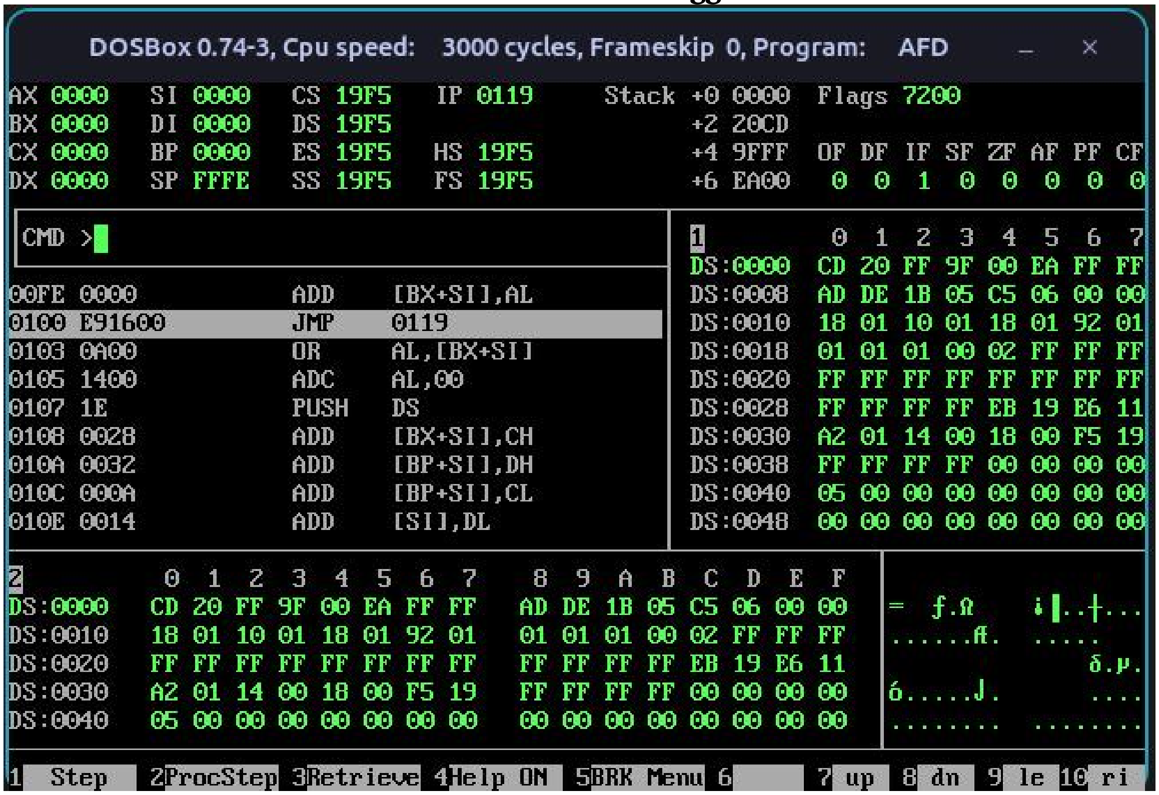


c03-01b.asm

💡 Click here to ask Blackbox to help you code faster

```
[org 0x0100]

    jmp start

num1: dw 10, 20, 30, 40, 50, 10, 20, 30, 40, 50
total: dw 0

start:
    mov bx, 0
    mov ax, 0; initialize array index to zero

l1:
    add ax, [num1+bx]
    add bx, 2
    cmp bx, 20
    jne l1
    mov[total], ax
    mov ax, 0x4c00
    int 0x21
```

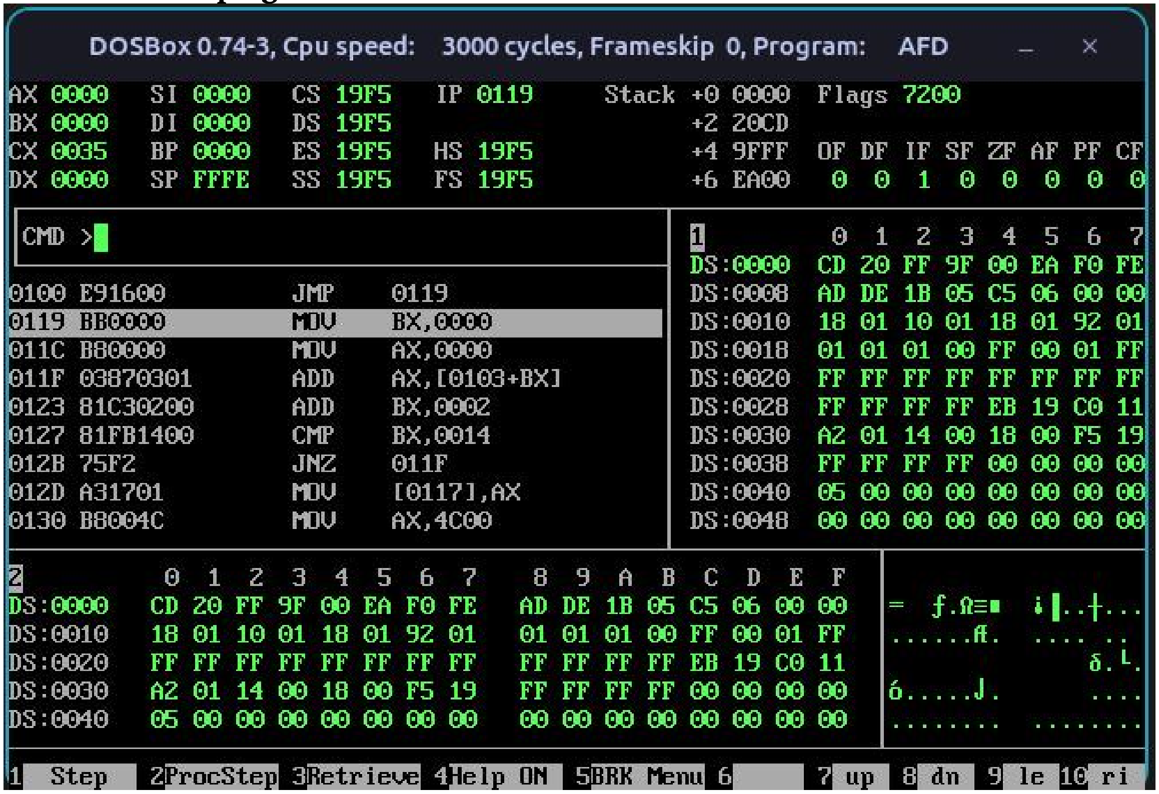
- Execute the code and examine it within the debugger.



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

```
AX 0000   SI 0000   CS 19F5   IP 0119      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 >                                          1          0  1  2  3  4  5  6  7
                                                DS:0000   CD 20 FF 9F 00 EA FF FF
00FE 0000            ADD    [BX+SI],AL          DS:0008   AD DE 1B 05 C5 06 00 00
0100 E91600          JMP    0119                DS:0010   18 01 10 01 18 01 92 01
0103 0A00            OR     AL,[BX+SI]          DS:0018   01 01 01 00 02 FF FF FF
0105 1400            ADC    AL,00               DS:0020   FF FF FF FF FF FF FF FF
0107 1E              PUSH   DS                  DS:0028   FF FF FF FF EB 19 E6 11
0108 0028            ADD    [BX+SI],CH          DS:0030   A2 01 14 00 18 00 F5 19
010A 0032            ADD    [BP+SI],DH          DS:0038   FF FF FF FF 00 00 00 00
010C 000A            ADD    [BP+SI],CL          DS:0040   05 00 00 00 00 00 00 00
010E 0014            ADD    [SI],DL             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   ........  ........
```

- **Utilize the "jmp" keyword, enabling an unconditional jump to location 0119.**
- **Observe in the subsequent figure how it abruptly transitions to line 0119 and commences program execution from there.**



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

```
AX 0000    SI 0000    CS 19F5    IP 0119      Stack +0 0000  Flags 7200
BX 0000    DI 0000    DS 19F5                       +2 20CD
CX 0035    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 F0 FE
0100 E91600          JMP     0119                  DS:0008  AD DE 1B 05 C5 06 00 00
0119 BB0000          MOV     BX,0000               DS:0010  18 01 10 01 18 01 92 01
011C B80000          MOV     AX,0000               DS:0018  01 01 01 00 FF 00 01 FF
011F 03870301        ADD     AX,[0103+BX]          DS:0020  FF FF FF FF FF FF FF FF
0123 81C30200        ADD     BX,0002               DS:0028  FF FF FF FF EB 19 C0 11
0127 81FB1400        CMP     BX,0014               DS:0030  A2 01 14 00 18 00 F5 19
012B 75F2            JNZ     011F                  DS:0038  FF FF FF FF 00 00 00 00
012D A31701          MOV     [0117],AX             DS:0040  05 00 00 00 00 00 00 00
0130 B8004C          MOV     AX,4C00               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   = ƒ.Ω≡■  ¡▌..┼...
DS:0010   18 01 10 01 18 01 92 01  01 01 01 00 FF 00 01 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   ........  ........
```