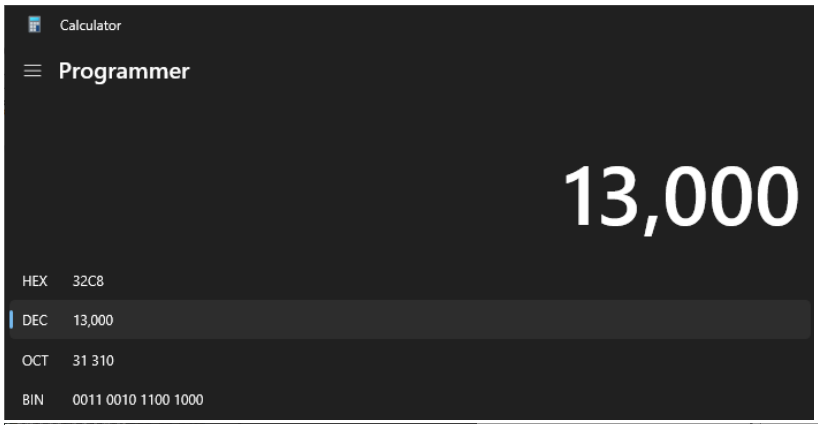
Task 1

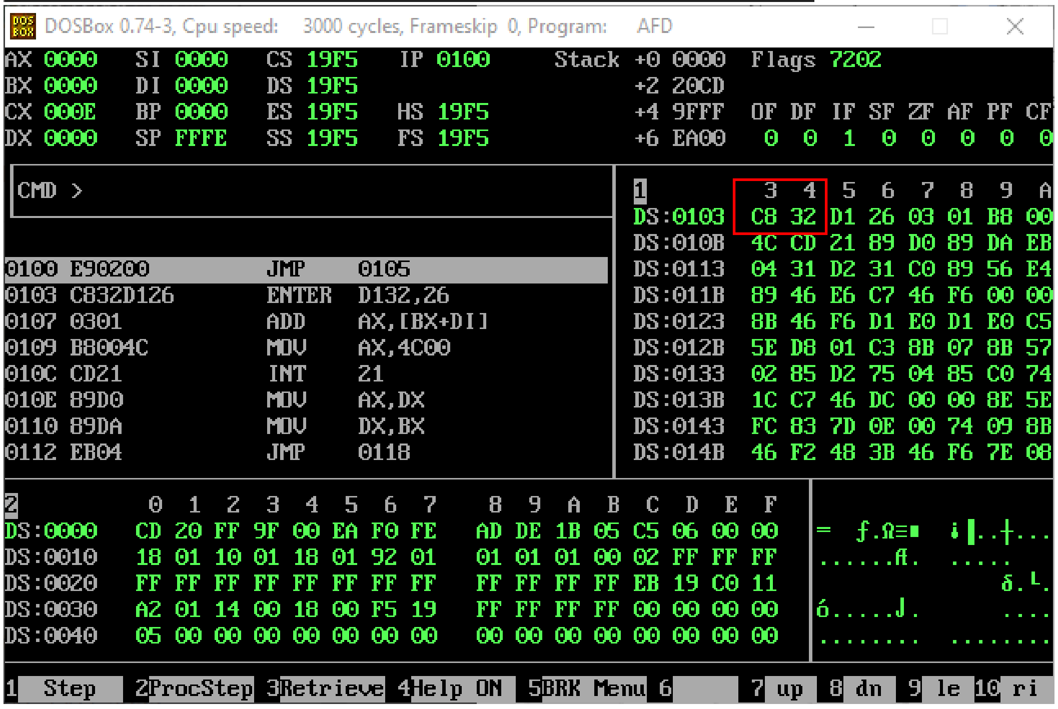
Compose a code in Assembly language to left shift (one time) a 16-bit number of your choice by using only "SHL". Apply the concept of extended shifting as well on the same number and compare the results of both shifting methods.

Show the number you are shifting in 16-bit binary, hexadecimal and decimal format before and after shifting it once. (Hint: You may use the calculator of Windows in its programmer mode and show all these values in a screenshot or notepad++ can also be used for this purpose (notepad++ →plugins → converter → conversion panel).

Through Normal Shifting

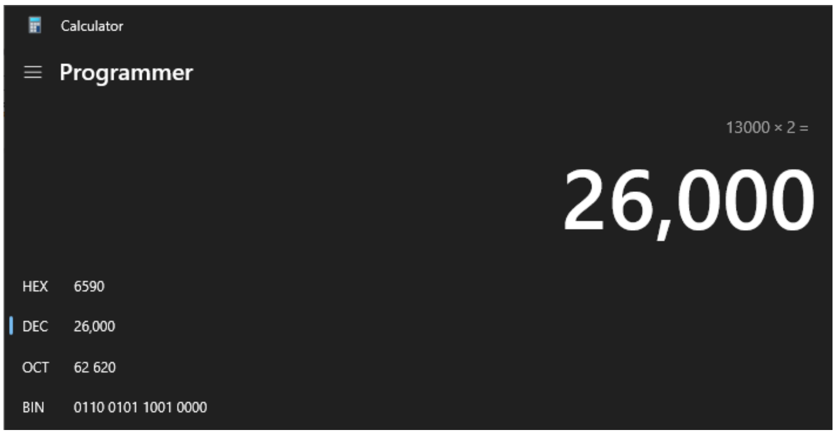


```
13,000
HEX   32C8
DEC   13,000
OCT   31 310
BIN   0011 0010 1100 1000
```


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 000E   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 C8 32 D1 26 03 01 B8 00
                                          DS:010B 4C CD 21 89 D0 89 DA EB
0100 E90200        JMP    0105            DS:0113 04 31 D2 31 C0 89 56 E4
0103 C832D126      ENTER  D132,26         DS:011B 89 46 E6 C7 46 F6 00 00
0107 0301          ADD    AX,[BX+DI]      DS:0123 8B 46 F6 D1 E0 D1 E0 C5
0109 B8004C        MOV    AX,4C00         DS:012B 5E D8 01 C3 8B 07 8B 57
010C CD21          INT    21              DS:0133 02 85 D2 75 04 85 C0 74
010E 89D0          MOV    AX,DX           DS:013B 1C C7 46 DC 00 00 8E 5E
0110 89DA          MOV    DX,BX           DS:0143 FC 83 7D 0E 00 74 09 8B
0112 EB04          JMP    0118            DS:014B 46 F2 48 3B 46 F6 7E 08

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
```

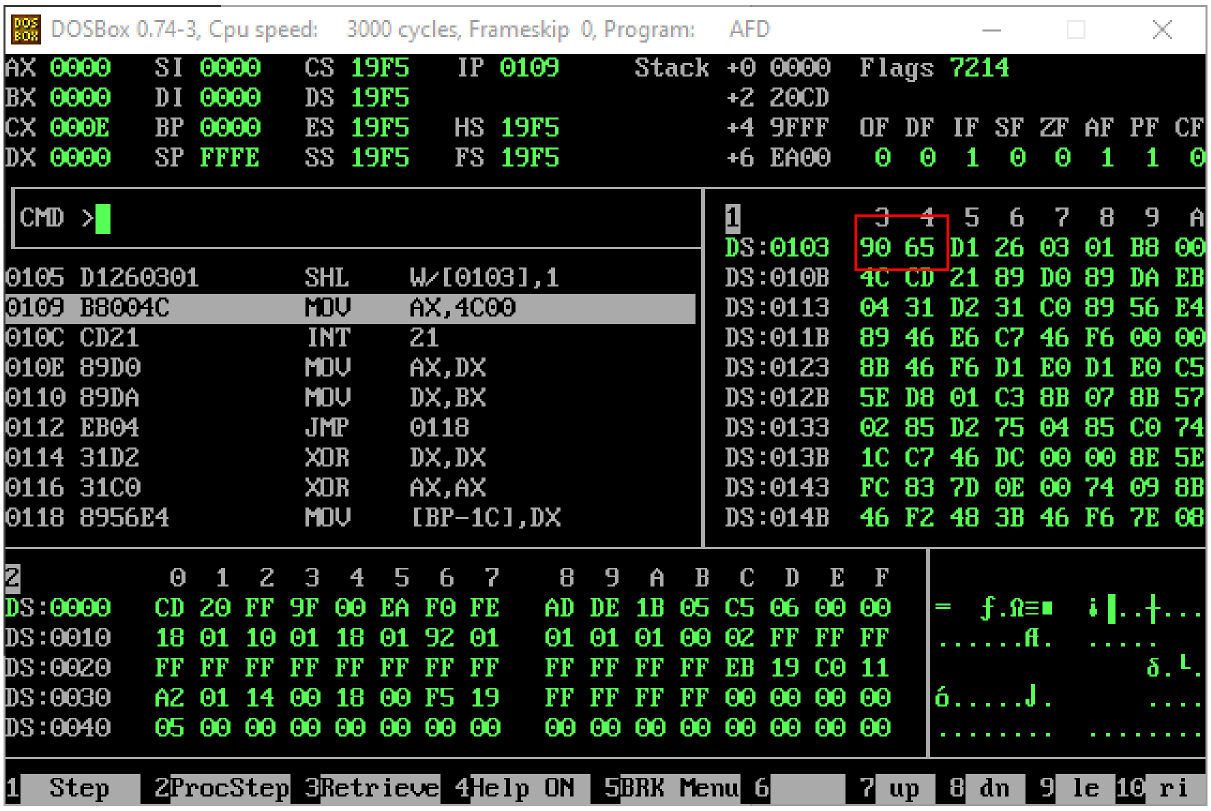
After Shifting



13000 × 2 =

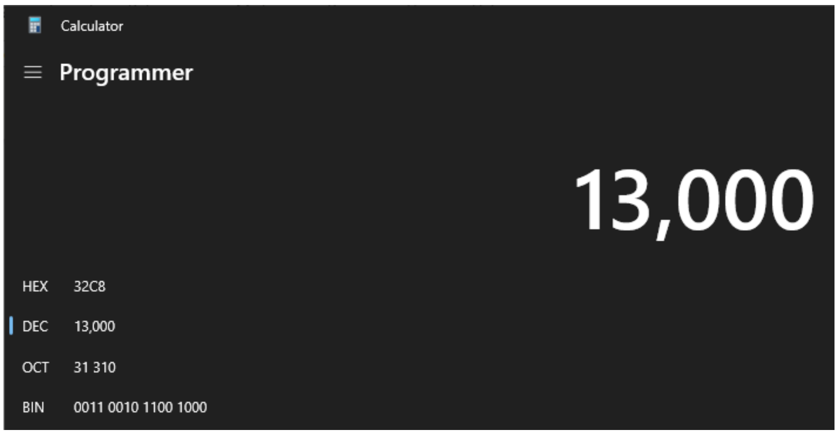
26,000

```
HEX   6590
DEC   26,000
OCT   62 620
BIN   0110 0101 1001 0000
```

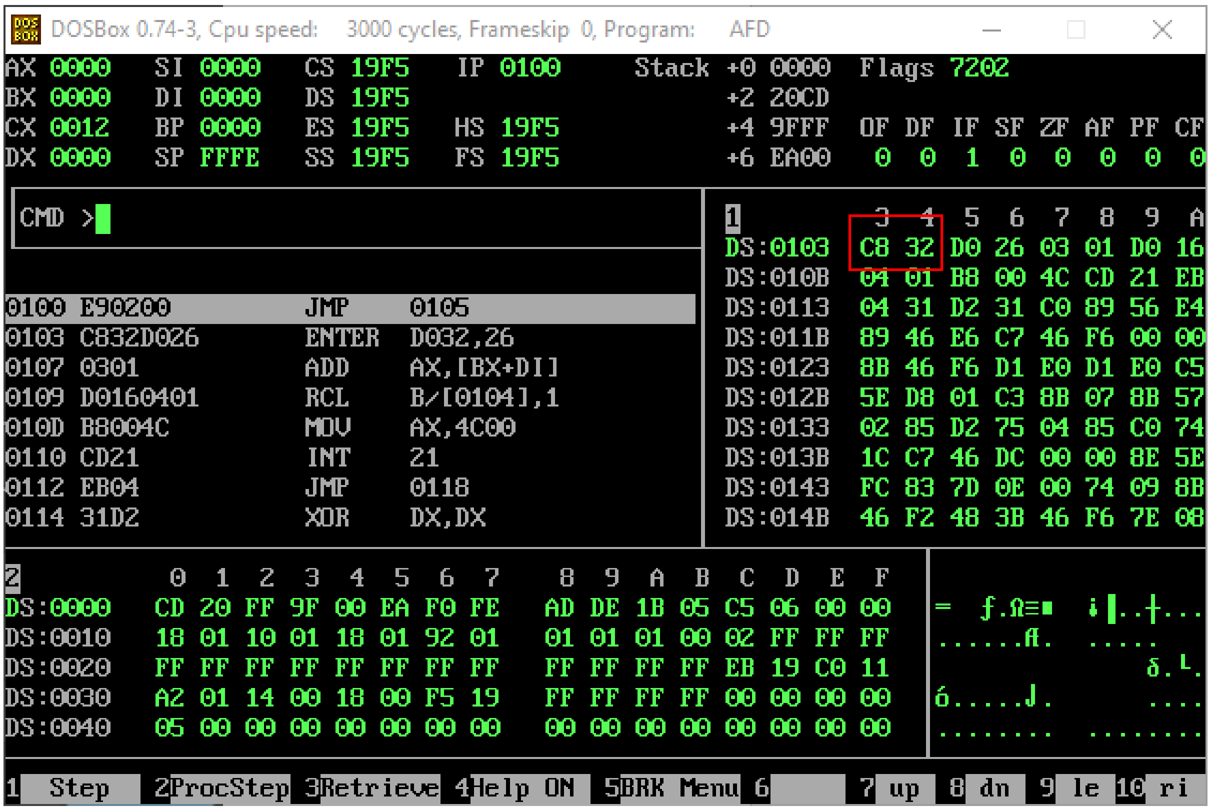

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

```
AX 0000   SI 0000   CS 19F5   IP 0109      Stack +0 0000  Flags 7214
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 000E   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  1  1  0

 CMD >                                            1        3  4  5  6  7  8  9  A
                                                  DS:0103  90 65 D1 26 03 01 B8 00
0105 D1260301       SHL     W/[0103],1            DS:010B  4C CD 21 89 D0 89 DA EB
0109 B8004C         MOV     AX,4C00               DS:0113  04 31 D2 31 C0 89 56 E4
010C CD21           INT     21                    DS:011B  89 46 E6 C7 46 F6 00 00
010E 89D0           MOV     AX,DX                 DS:0123  8B 46 F6 D1 E0 D1 E0 C5
0110 89DA           MOV     DX,BX                 DS:012B  5E D8 01 C3 8B 07 8B 57
0112 EB04           JMP     0118                  DS:0133  02 85 D2 75 04 85 C0 74
0114 31D2           XOR     DX,DX                 DS:013B  1C C7 46 DC 00 00 8E 5E
0116 31C0           XOR     AX,AX                 DS:0143  FC 83 7D 0E 00 74 09 8B
0118 8956E4         MOV     [BP-1C],DX            DS:014B  46 F2 48 3B 46 F6 7E 08

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   ........   ........

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

Through Extended Shifting

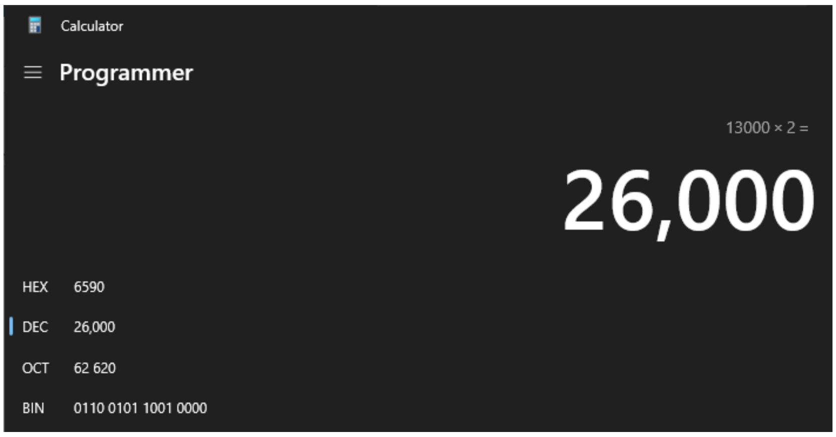


```
Calculator
Programmer

13,000

HEX  32C8
DEC  13,000
OCT  31 310
BIN  0011 0010 1100 1000
```

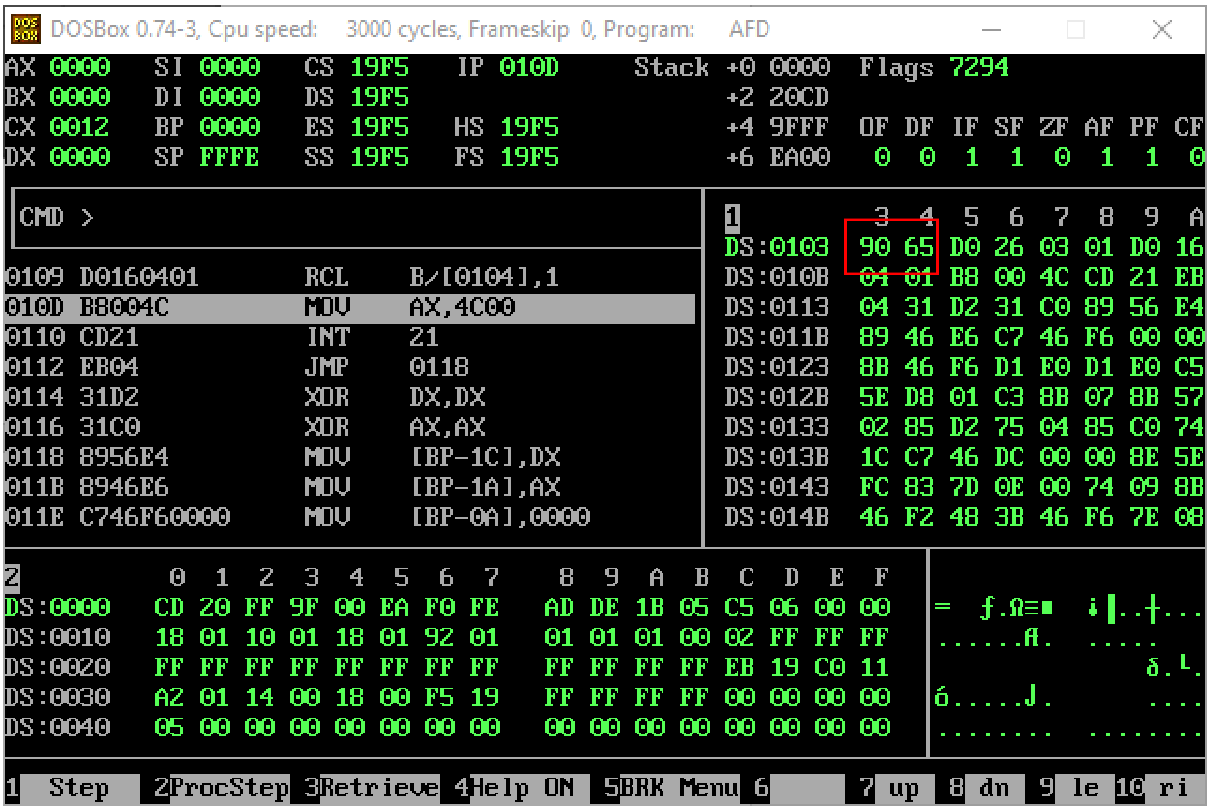


```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0000   SI 0000   CS 19F5   IP 0100      Stack +0 0000  Flags 7202
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 0012   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  0  0

 CMD >                                          1        3  4  5  6  7  8  9  A
                                                DS:0103 C8 32 D0 26 03 01 D0 16
                                                DS:010B 04 01 B8 00 4C CD 21 EB
0100 E90200         JMP    0105                 DS:0113 04 31 D2 31 C0 89 56 E4
0103 C832D026       ENTER  D032,26              DS:011B 89 46 E6 C7 46 F6 00 00
0107 0301           ADD    AX,[BX+DI]           DS:0123 8B 46 F6 D1 E0 D1 E0 C5
0109 D0160401       RCL    B/[0104],1           DS:012B 5E D8 01 C3 8B 07 8B 57
010D B8004C         MOV    AX,4C00              DS:0133 02 85 D2 75 04 85 C0 74
0110 CD21           INT    21                   DS:013B 1C C7 46 DC 00 00 8E 5E
0112 EB04           JMP    0118                 DS:0143 FC 83 7D 0E 00 74 09 8B
0114 31D2           XOR    DX,DX                DS:014B 46 F2 48 3B 46 F6 7E 08

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 02 FF FF 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
```

After Shifting



```
13000 × 2 =
26,000

HEX  6590
DEC  26,000
OCT  62 620
BIN  0110 0101 1001 0000
```


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

```
AX 0000   SI 0000   CS 19F5   IP 010D     Stack +0 0000  Flags 7294
BX 0000   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  0

 CMD >                                          1         3  4  5  6  7  8  9  A
                                                DS:0103  90 65 D0 26 03 01 D0 16
0109 D0160401       RCL     B/[0104],1          DS:010B  04 01 B8 00 4C CD 21 EB
010D B8004C         MOV     AX,4C00             DS:0113  04 31 D2 31 C0 89 56 E4
0110 CD21           INT     21                  DS:011B  89 46 E6 C7 46 F6 00 00
0112 EB04           JMP     0118                DS:0123  8B 46 F6 D1 E0 D1 E0 C5
0114 31D2           XOR     DX,DX               DS:012B  5E D8 01 C3 8B 07 8B 57
0116 31C0           XOR     AX,AX               DS:0133  02 85 D2 75 04 85 C0 74
0118 8956E4         MOV     [BP-1C],DX          DS:013B  1C C7 46 DC 00 00 8E 5E
011B 8946E6         MOV     [BP-1A],AX          DS:0143  FC 83 7D 0E 00 74 09 8B
011E C746F60000     MOV     [BP-0A],0000        DS:014B  46 F2 48 3B 46 F6 7E 08

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
```

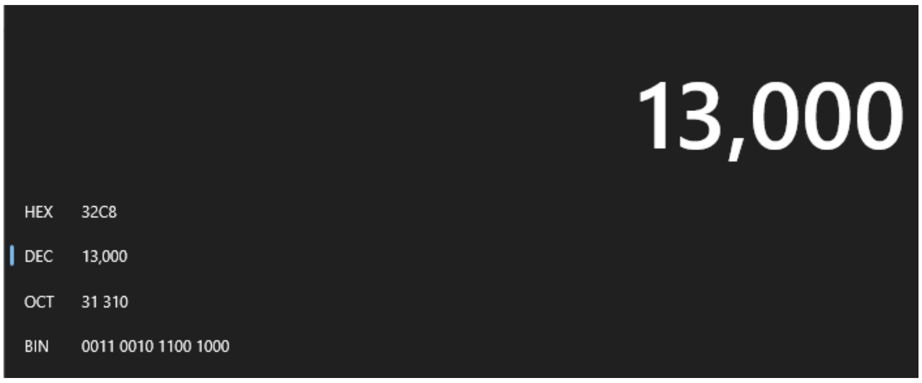
By Comparing the Results Both Are Same.

Task 2

Write a code in Assembly language to add two 16-bit numbers of your own choice by applying the concept of extended addition. Show the numbers you are adding in 16-bit binary, hexadecimal and decimal format before addition. Do the same once you have added the numbers

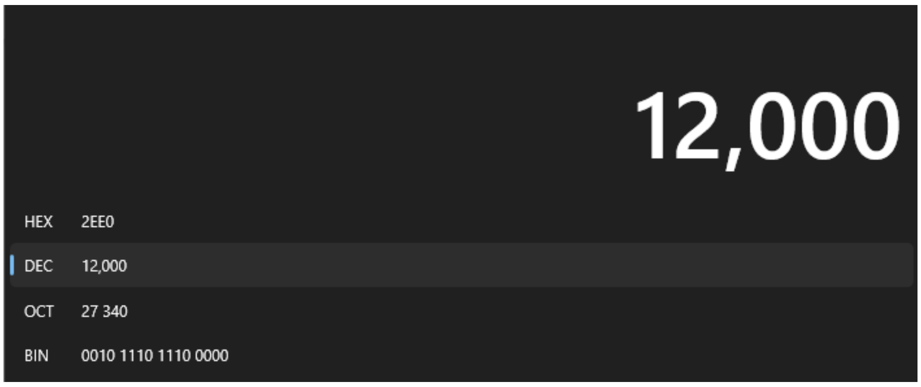
and show the result as said above.

First Num



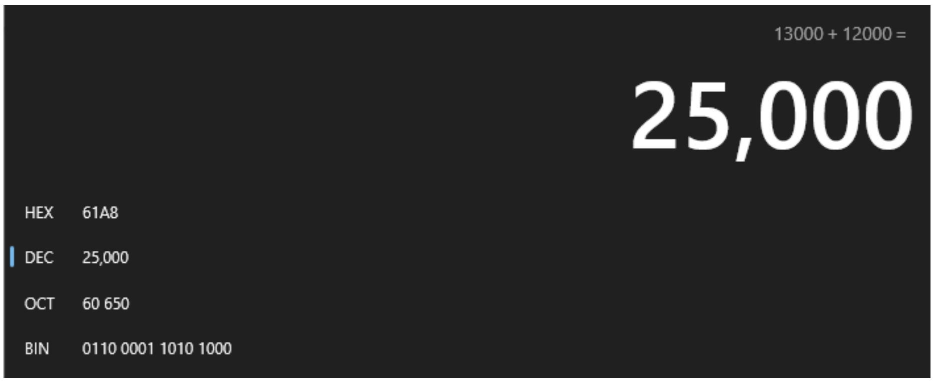
```
13,000

HEX   32C8
DEC   13,000
OCT   31 310
BIN   0011 0010 1100 1000
```

Second Number



```
12,000

HEX   2EE0
DEC   12,000
OCT   27 340
BIN   0010 1110 1110 0000
```

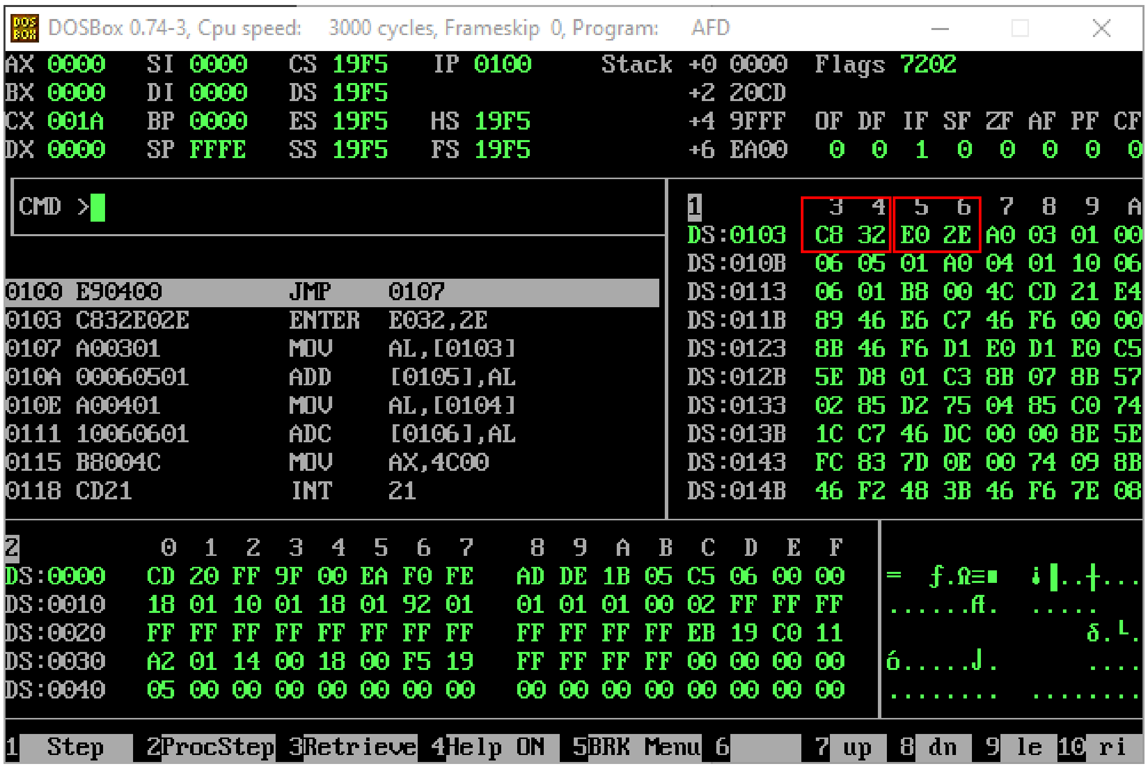
Addition



```
13000 + 12000 =
25,000

HEX   61A8
DEC   25,000
OCT   60 650
BIN   0110 0001 1010 1000
```

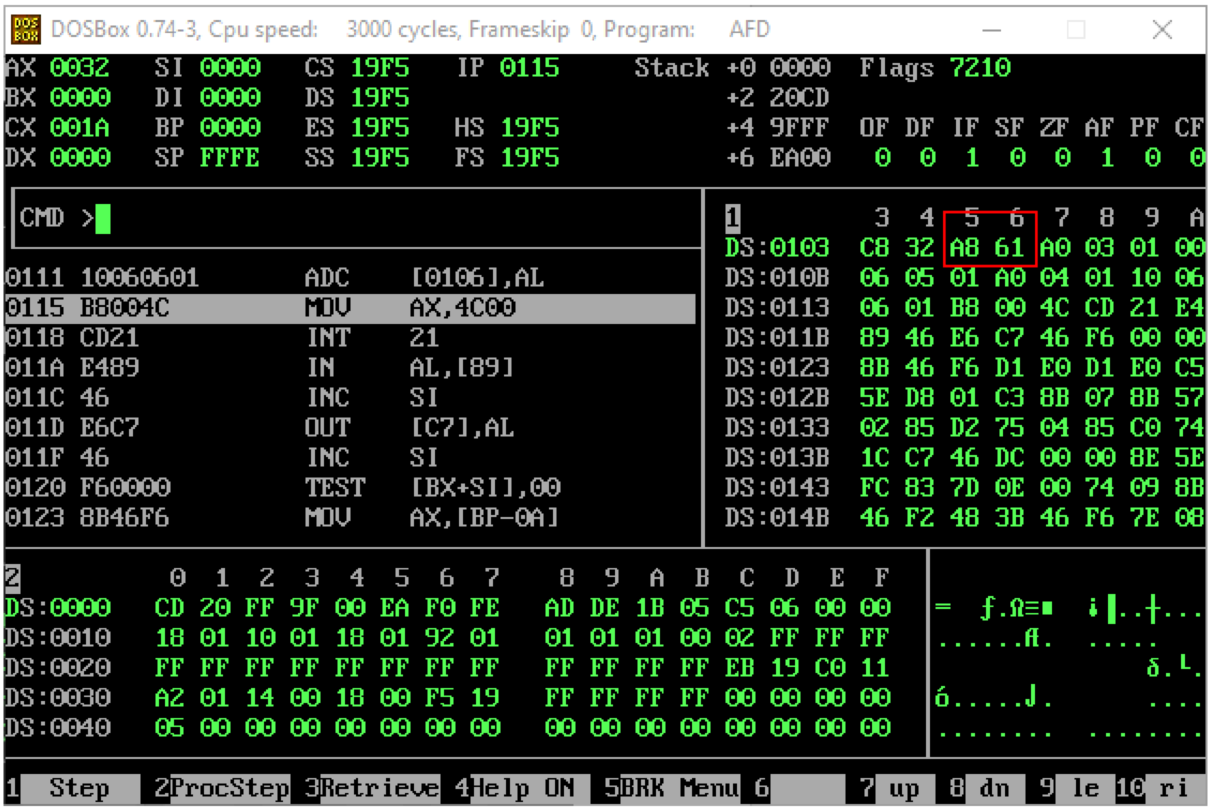
Now through AFD Showing

Before Addition: -



```
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 001A   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  C8 32 E0 2E A0 03 01 00
                                                DS:010B  06 05 01 A0 04 01 10 06
0100 E90400          JMP    0107                DS:0113  06 01 B8 00 4C CD 21 E4
0103 C832E02E        ENTER  E032,2E             DS:011B  89 46 E6 C7 46 F6 00 00
0107 A00301          MOV    AL,[0103]           DS:0123  8B 46 F6 D1 E0 D1 E0 C5
010A 00060501        ADD    [0105],AL           DS:012B  5E D8 01 C3 8B 07 8B 57
010E A00401          MOV    AL,[0104]           DS:0133  02 85 D2 75 04 85 C0 74
0111 10060601        ADC    [0106],AL           DS:013B  1C C7 46 DC 00 00 8E 5E
0115 B8004C          MOV    AX,4C00             DS:0143  FC 83 7D 0E 00 74 09 8B
0118 CD21            INT    21                  DS:014B  46 F2 48 3B 46 F6 7E 08

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
```

After Addition: -

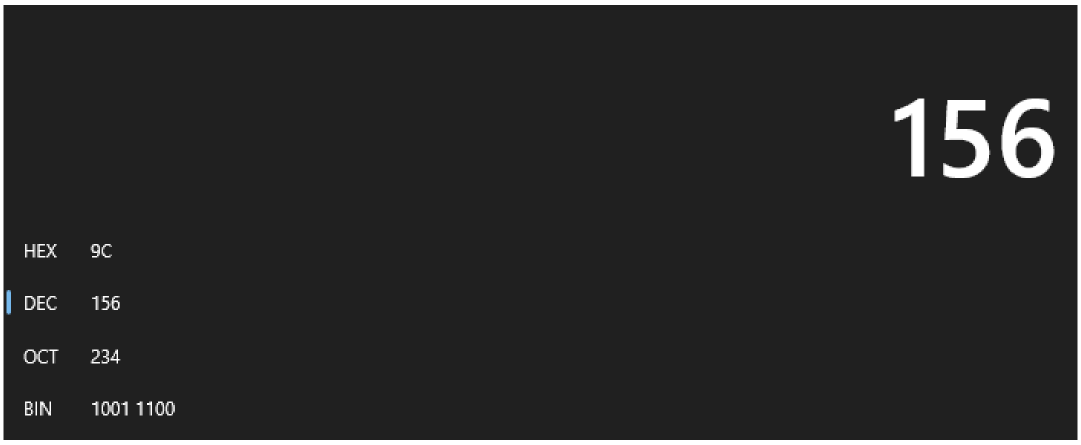


```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Program:    AFD
AX 0032   SI 0000   CS 19F5   IP 0115      Stack +0 0000  Flags 7210
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 001A   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF  DF  IF  SF  ZF  AF  PF  CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0   0   1   0   0   1   0   0

 CMD >                                          1         3  4  5  6  7  8  9  A
                                                DS:0103  C8 32 A8 61 A0 03 01 00
0111 10060601        ADC    [0106],AL           DS:010B  06 05 01 A0 04 01 10 06
0115 B8004C          MOV    AX,4C00             DS:0113  06 01 B8 00 4C CD 21 E4
0118 CD21            INT    21                  DS:011B  89 46 E6 C7 46 F6 00 00
011A E489            IN     AL,[89]             DS:0123  8B 46 F6 D1 E0 D1 E0 C5
011C 46              INC    SI                  DS:012B  5E D8 01 C3 8B 07 8B 57
011D E6C7            OUT    [C7],AL             DS:0133  02 85 D2 75 04 85 C0 74
011F 46              INC    SI                  DS:013B  1C C7 46 DC 00 00 8E 5E
0120 F60000          TEST   [BX+SI],00          DS:0143  FC 83 7D 0E 00 74 09 8B
0123 8B46F6          MOV    AX,[BP-0A]          DS:014B  46 F2 48 3B 46 F6 7E 08

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 02 FF FF 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
```

Task 3

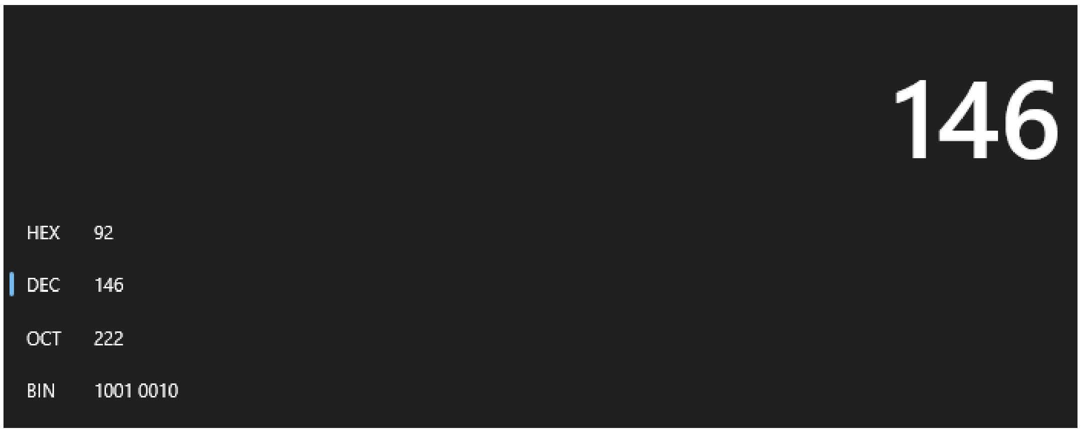
Apply the concept of extended shifting and addition for the multiplication of two 8-bit numbers of your own choice.

Multiplicand: -



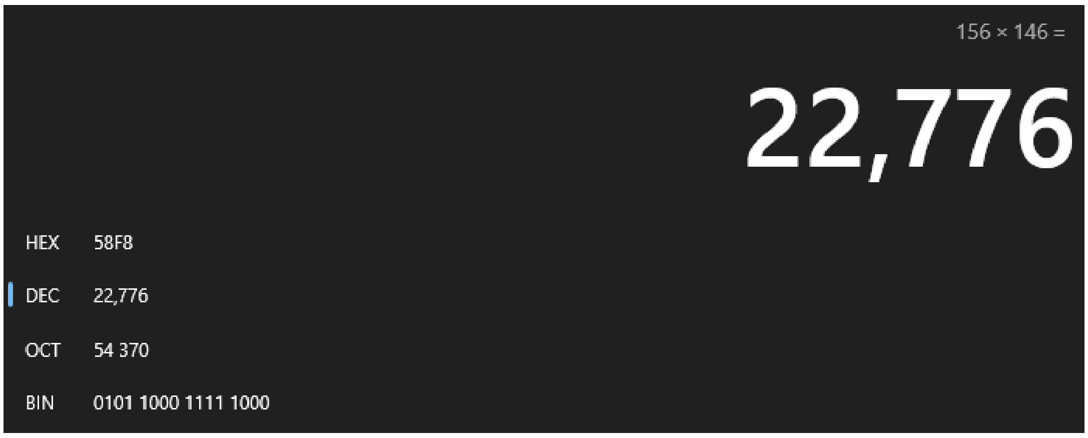
```
156

HEX  9C
DEC  156
OCT  234
BIN  1001 1100
```

Multiplier: -



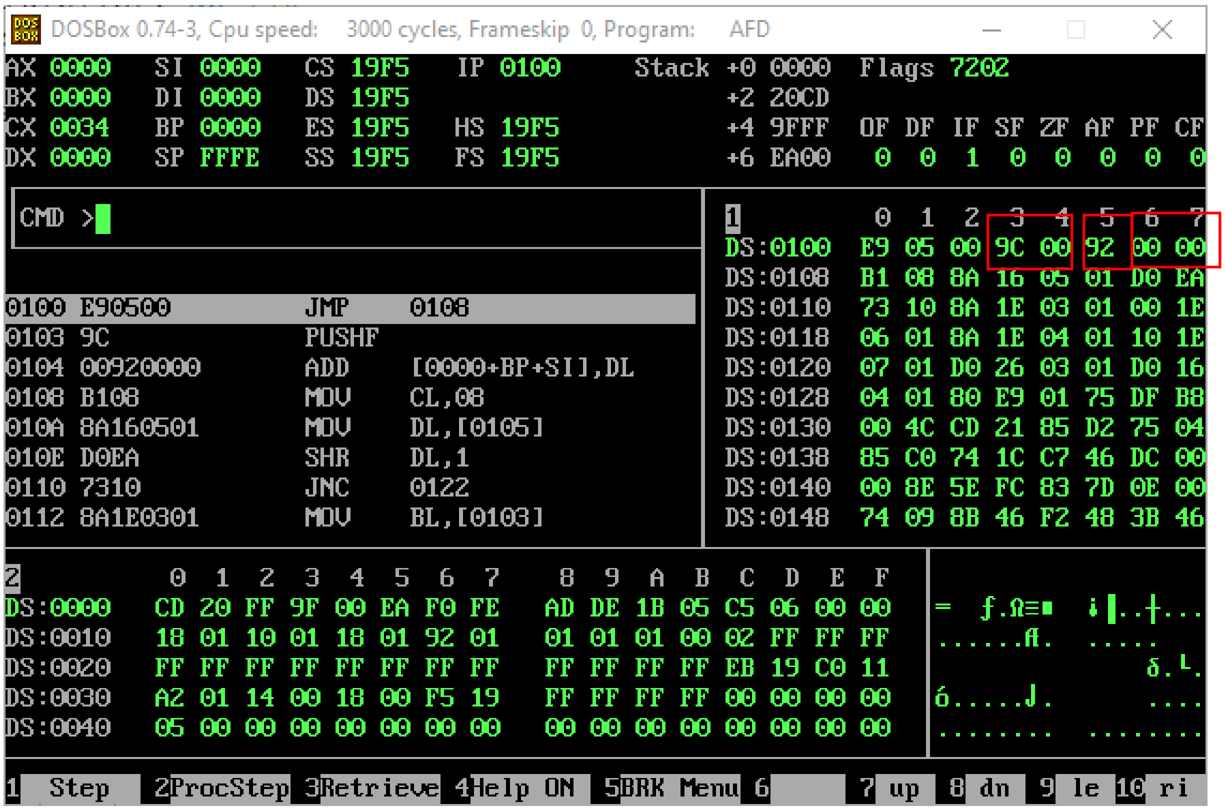
```
146

HEX 92
DEC 146
OCT 222
BIN 1001 0010
```

Result: -



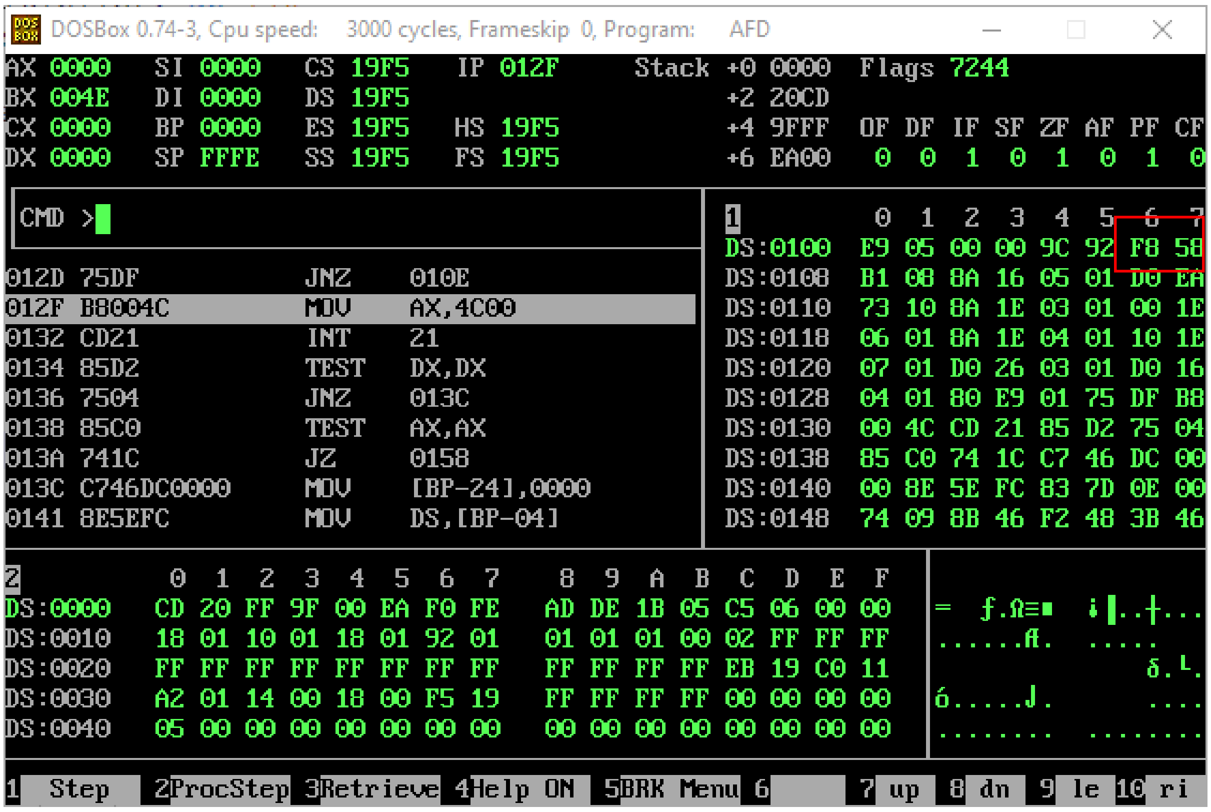
```
156 × 146 =
22,776

HEX 58F8
DEC 22,776
OCT 54 370
BIN 0101 1000 1111 1000
```

Initial Values By Looking In AFD : -



```
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 0034   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:0100   E9 05 00 9C 00 92 00 00
                                                DS:0108   B1 08 8A 16 05 01 D0 EA
0100 E90500          JMP     0108               DS:0110   73 10 8A 1E 03 01 00 1E
0103 9C              PUSHF                      DS:0118   06 01 8A 1E 04 01 10 1E
0104 00920000        ADD     [0000+BP+SI],DL    DS:0120   07 01 D0 26 03 01 D0 16
0108 B108            MOV     CL,08              DS:0128   04 01 80 E9 01 75 DF B8
010A 8A160501        MOV     DL,[0105]          DS:0130   00 4C CD 21 85 D2 75 04
010E D0EA            SHR     DL,1               DS:0138   85 C0 74 1C C7 46 DC 00
0110 7310            JNC     0122               DS:0140   00 8E 5E FC 83 7D 0E 00
0112 8A1E0301        MOV     BL,[0103]          DS:0148   74 09 8B 46 F2 48 3B 46

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
```

Result: -

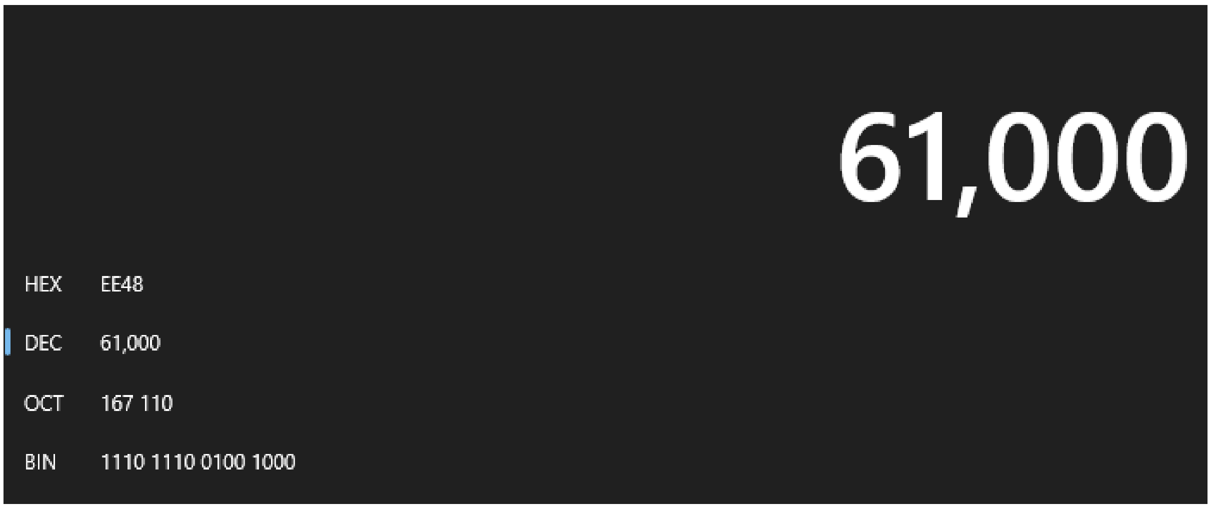


```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Program:    AFD
AX 0000   SI 0000   CS 19F5   IP 012F      Stack +0 0000  Flags 7244
BX 004E   DI 0000   DS 19F5                      +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   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:0100   E9 05 00 00 9C 92 F8 58
012D 75DF            JNZ     010E               DS:0108   B1 08 8A 16 05 01 D0 EA
012F B8004C          MOV     AX,4C00            DS:0110   73 10 8A 1E 03 01 00 1E
0132 CD21            INT     21                 DS:0118   06 01 8A 1E 04 01 10 1E
0134 85D2            TEST    DX,DX              DS:0120   07 01 D0 26 03 01 D0 16
0136 7504            JNZ     013C               DS:0128   04 01 80 E9 01 75 DF B8
0138 85C0            TEST    AX,AX              DS:0130   00 4C CD 21 85 D2 75 04
013A 741C            JZ      0158               DS:0138   85 C0 74 1C C7 46 DC 00
013C C746DC0000      MOV     [BP-24],0000       DS:0140   00 8E 5E FC 83 7D 0E 00
0141 8E5EFC          MOV     DS,[BP-04]         DS:0148   74 09 8B 46 F2 48 3B 46

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
```

Task 4

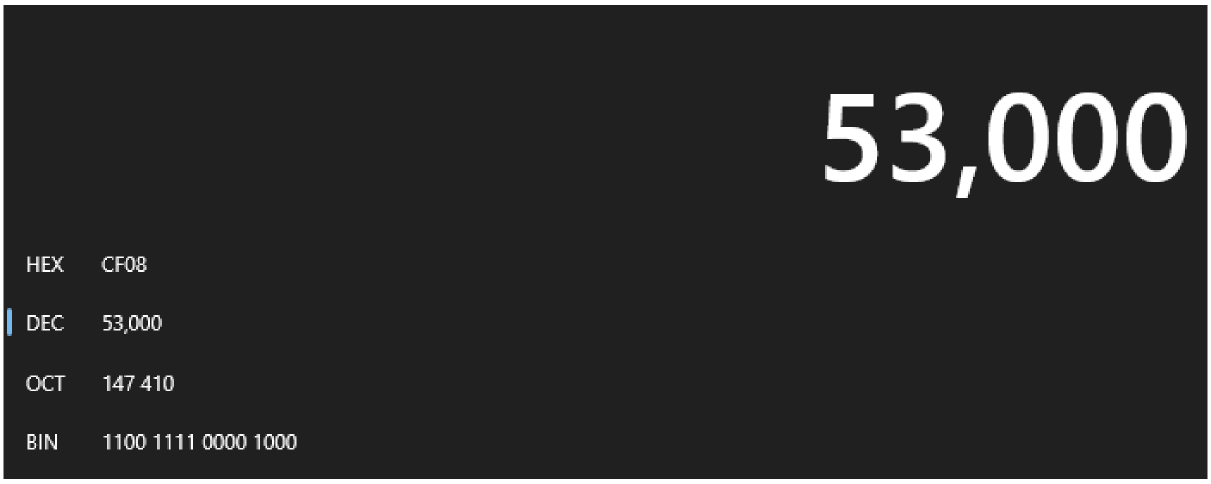
Repeat task 3 for two 16-bit numbers of your own choice.

Multiplicand: -



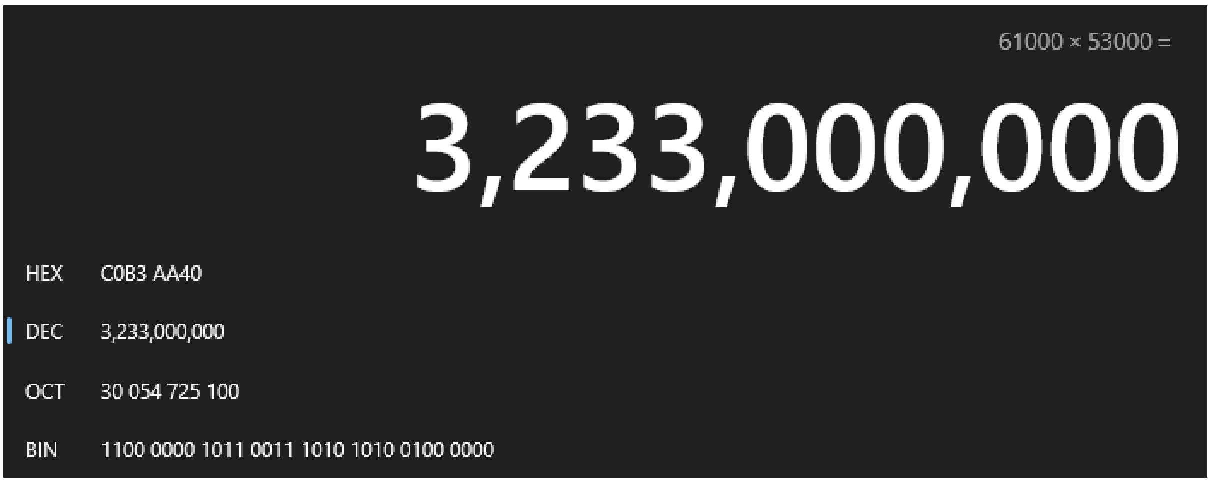
```
61,000

HEX   EE48
DEC   61,000
OCT   167 110
BIN   1110 1110 0100 1000
```

Multiplier: -



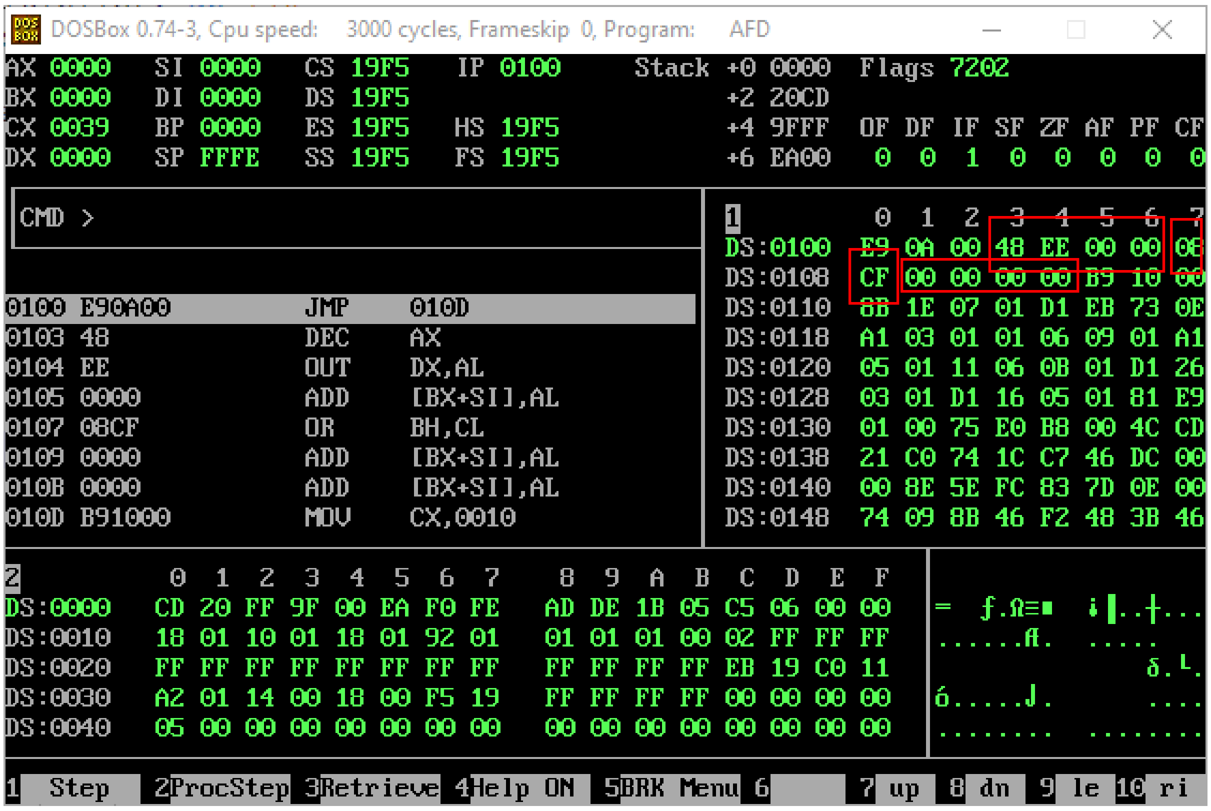
```
53,000

HEX   CF08
DEC   53,000
OCT   147 410
BIN   1100 1111 0000 1000
```

Result: -



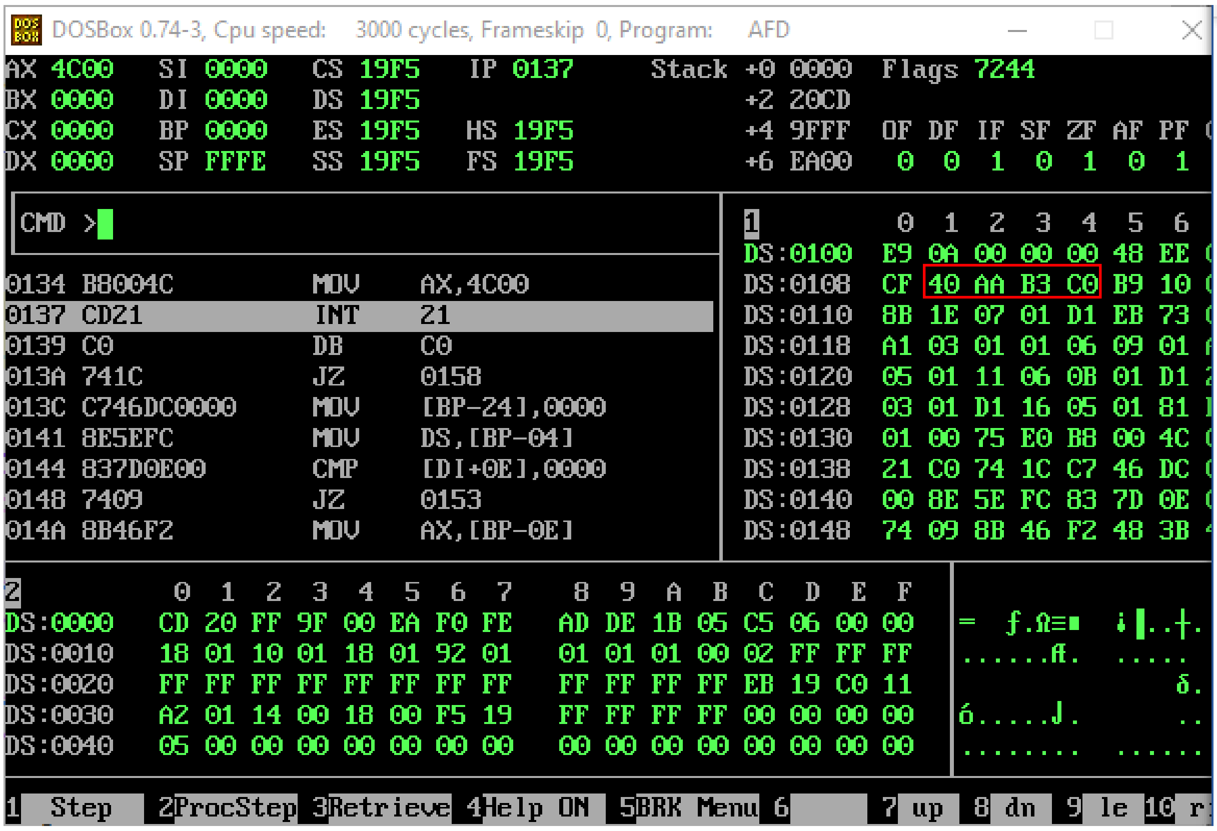
```
61000 × 53000 =
3,233,000,000

HEX   C0B3 AA40
DEC   3,233,000,000
OCT   30 054 725 100
BIN   1100 0000 1011 0011 1010 1010 0100 0000
```

Initial Values By Looking At The AFD: -



```
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 0039   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:0100   E9 0A 00 48 EE 00 00 0E
                                                DS:0108   CF 00 00 00 00 B9 10 00
0100 E90A00         JMP     010D                DS:0110   8B 1E 07 01 D1 EB 73 0E
0103 48             DEC     AX                  DS:0118   A1 03 01 01 06 09 01 A1
0104 EE             OUT     DX,AL               DS:0120   05 01 11 06 0B 01 D1 26
0105 0000           ADD     [BX+SI],AL          DS:0128   03 01 D1 16 05 01 81 E9
0107 08CF           OR      BH,CL               DS:0130   01 00 75 E0 B8 00 4C CD
0109 0000           ADD     [BX+SI],AL          DS:0138   21 C0 74 1C C7 46 DC 00
010B 0000           ADD     [BX+SI],AL          DS:0140   00 8E 5E FC 83 7D 0E 00
010D B91000         MOV     CX,0010             DS:0148   74 09 8B 46 F2 48 3B 46

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 02 FF FF 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
```

Result: -



```
DOSBox 0.74-3, Cpu speed:   3000 cycles, Frameskip 0, Program:   AFD
AX 4C00   SI 0000   CS 19F5   IP 0137     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
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00    0  0  1  0  1  0  1

 CMD >                                          1          0  1  2  3  4  5  6
                                                DS:0100   E9 0A 00 00 00 48 EE
                                                DS:0108   CF 40 AA B3 C0 B9 10
0134 B8004C         MOV     AX,4C00             DS:0110   8B 1E 07 01 D1 EB 73
0137 CD21           INT     21                  DS:0118   A1 03 01 01 06 09 01
0139 C0             DB      C0                  DS:0120   05 01 11 06 0B 01 D1
013A 741C           JZ      0158                DS:0128   03 01 D1 16 05 01 81
013C C746DC0000     MOV     [BP-24],0000        DS:0130   01 00 75 E0 B8 00 4C
0141 8E5EFC         MOV     DS,[BP-04]          DS:0138   21 C0 74 1C C7 46 DC
0144 837D0E00       CMP     [DI+0E],0000        DS:0140   00 8E 5E FC 83 7D 0E
0148 7409           JZ      0153                DS:0148   74 09 8B 46 F2 48 3B
014A 8B46F2         MOV     AX,[BP-0E]

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 02 FF FF 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 r
```