ADIT LUHADIA

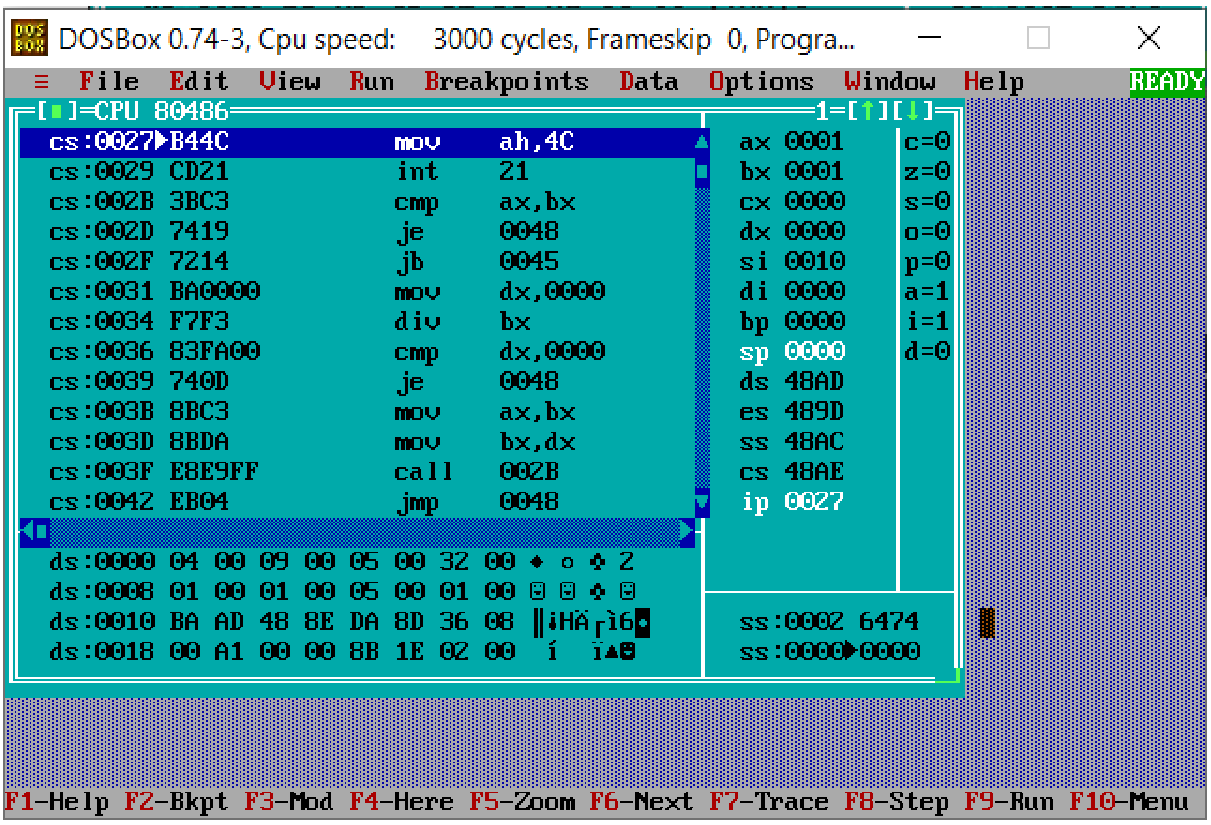
190911112

IT A

COMP LAB WEEK 3

1. GCD of four unsigned words using procedure

```
; GCD of four unsigned words using procedure
DATA SEGMENT
    NUM DW 4,9,5,50
    GCD DW 3 DUP(?)
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
    START:
        MOV DX,DATA
        MOV DS,DX

        LEA SI,GCD
        MOV AX,NUM ; AX=0004
        MOV BX,NUM+2 ; BX=0009
        CALL GCDPROC ; gcd of 9,4

        MOV AX,NUM+4 ; AX=0005
        MOV BX,NUM+6 ; BX=0050
        CALL GCDPROC ; gcd of 5,50

        ;FINAL GCD
        MOV AX,GCD
        MOV BX,GCD+2
        CALL GCDPROC

        MOV AH,4CH
        INT 21H

        GCDPROC:
            CMP AX,BX
            JZ EXIT
            JB BIGAX; if AX IS BELOW BX, WE WANT TO MAKE AX BIGGER

            ;AX=0090 BX=0040
            MOV DX,0
            DIV BX
            CMP DX,0
            JZ EXIT
            MOV AX,BX
            MOV BX,DX

            CALL GCDPROC
            JMP EXIT
            BIGAX:
                XCHG AX,BX
                JMP GCDPROC

            EXIT: MOV [SI],BX ; GCD FOUND
                  INC SI
                  INC SI
            RET

CODE ENDS
END START
```

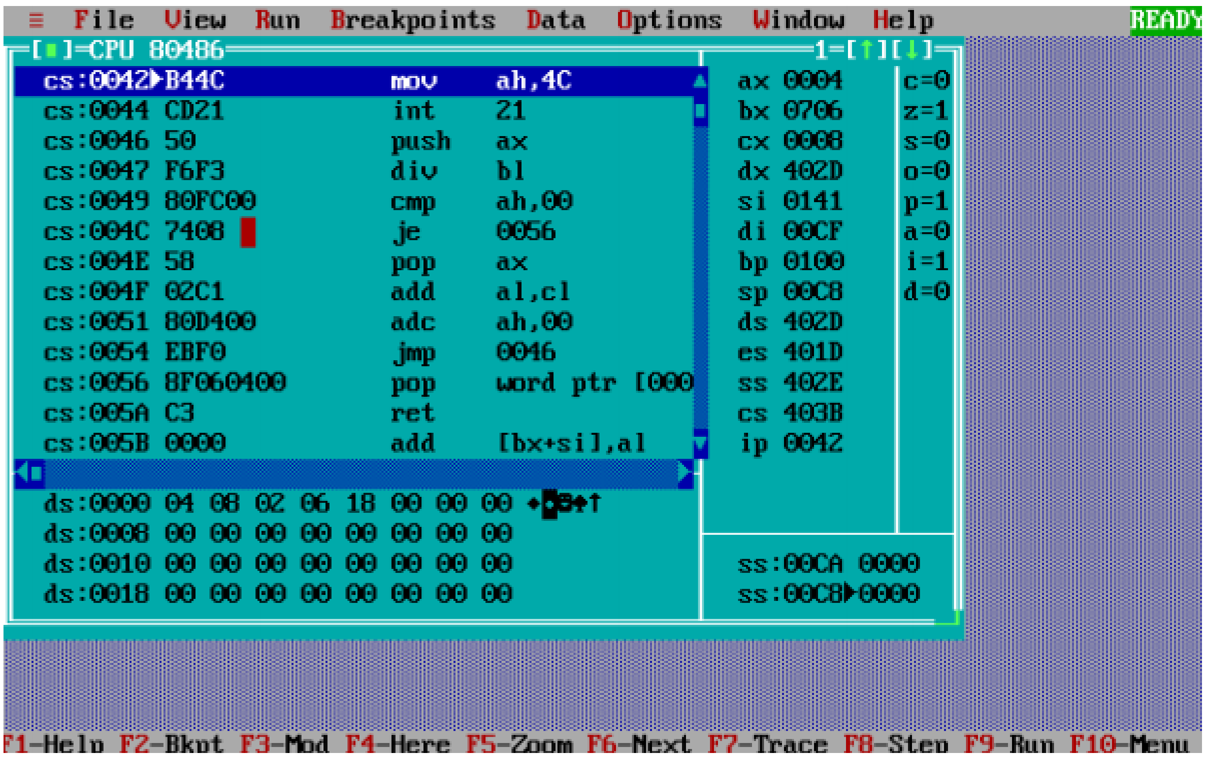


```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
 ≡  File  Edit  View  Run  Breakpoints  Data  Options  Window  Help       READY
[■]=CPU 80486=====================================1=[↑][↓]=
 cs:0027▶B44C          mov    ah,4C        ax 0001   c=0
 cs:0029 CD21          int    21           bx 0001   z=0
 cs:002B 3BC3          cmp    ax,bx        cx 0000   s=0
 cs:002D 7419          je     0048         dx 0000   o=0
 cs:002F 7214          jb     0045         si 0010   p=0
 cs:0031 BA0000        mov    dx,0000      di 0000   a=1
 cs:0034 F7F3          div    bx           bp 0000   i=1
 cs:0036 83FA00        cmp    dx,0000      sp 0000   d=0
 cs:0039 740D          je     0048         ds 48AD
 cs:003B 8BC3          mov    ax,bx        es 489D
 cs:003D 8BDA          mov    bx,dx        ss 48AC
 cs:003F E8E9FF        call   002B         cs 48AE
 cs:0042 EB04          jmp    0048         ip 0027

 ds:0000 04 00 09 00 05 00 32 00 ♦ ○ ♣ 2
 ds:0008 01 00 01 00 05 00 01 00 ☺ ☺ ♣ ☺    ss:0002 6474
 ds:0010 BA AD 48 8E DA 8D 36 08 ║¡HÄ┌ì6◘   ss:0000▶0000
 ds:0018 00 A1 00 00 8B 1E 02 00  í  ï▲☻
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```

2. LCM of four unsigned bytes using function which finds LCM of two unsigned bytes.

```
; LCM of four unsigned bytes using function which finds LCM of two unsigned by
tes.
DATA SEGMENT
        NUM DB 25, 15, 21, 30 ; 4 unsigned numbers
        LCM1 DW ?
        LCM DW ?
```

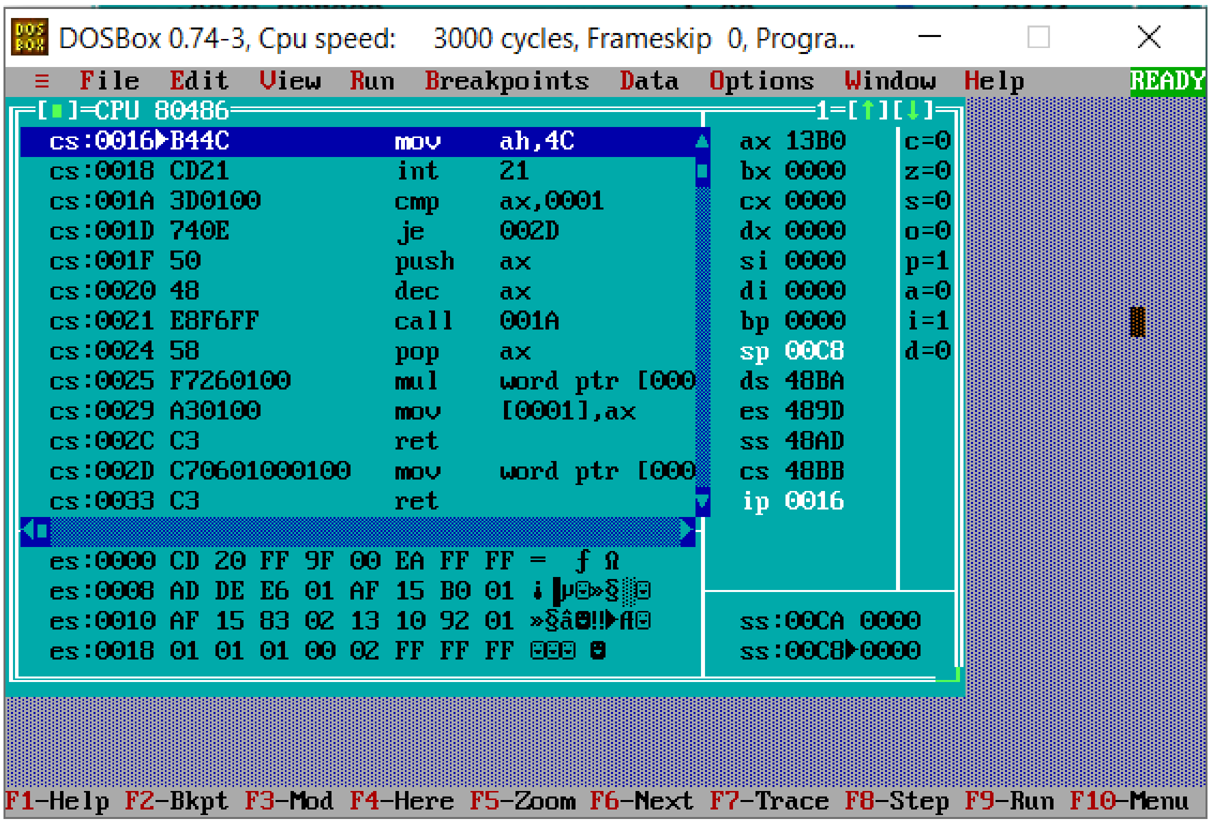
```
DATA ENDS
STACK SEGMENT
        DW 100 DUP(?)
        TOS LABEL WORD
STACK ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:STACK
START:  MOV AX, DATA
        MOV DS, AX
        MOV AX, STACK
        MOV SS, AX
        LEA SP, TOS
        MOV AH, 0
        MOV AL, NUM
        MOV BL, NUM+1
        CALL LCMPROC
        MOV AX, LCM
        MOV LCM1, AX
        MOV AH, 0
        MOV AL, NUM+2
        MOV BH, 0
        MOV BL, NUM+3
        CALL LCMPROC
        MOV AX, LCM1
        MOV BX, LCM
        CALL LCMPROC
        MOV AH, 4CH
        INT 21H
LCMPROC PROC
BACK:   PUSH AX
        DIV BL
        CMP AH, 0
        JZ DOWN
        POP AX
        ADD AL, NUM
        ADC AH, 0
        JMP BACK
DOWN:   POP LCM
        RET
LCMPROC ENDP
CODE ENDS
END START
```



```
 ≡  File  View  Run  Breakpoints  Data  Options  Window  Help                    READY
=[■]=CPU 80486=========================================1=[↑][↓]=
  cs:0042►B44C           mov    ah,4C           ▲ ax 0004   c=0
  cs:0044 CD21           int    21                bx 0706   z=1
  cs:0046 50             push   ax                cx 0008   s=0
  cs:0047 F6F3           div    bl                dx 402D   o=0
  cs:0049 80FC00         cmp    ah,00             si 0141   p=1
  cs:004C 7408 █         je     0056              di 00CF   a=0
  cs:004E 58             pop    ax                bp 0100   i=1
  cs:004F 02C1           add    al,cl             sp 00C8   d=0
  cs:0051 80D400         adc    ah,00             ds 402D
  cs:0054 EBF0           jmp    0046              es 401D
  cs:0056 8F060400       pop    word ptr [000     ss 402E
  cs:005A C3             ret                      cs 403B
  cs:005B 0000           add    [bx+si],al      ▼ ip 0042
◄■                                              ►
  ds:0000 04 08 02 06 18 00 00 00 ♦◘☻♠↑
  ds:0008 00 00 00 00 00 00 00 00
  ds:0010 00 00 00 00 00 00 00 00             ss:00CA 0000
  ds:0018 00 00 00 00 00 00 00 00             ss:00C8►0000

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```

3. Factorial of unsigned byte using recursion

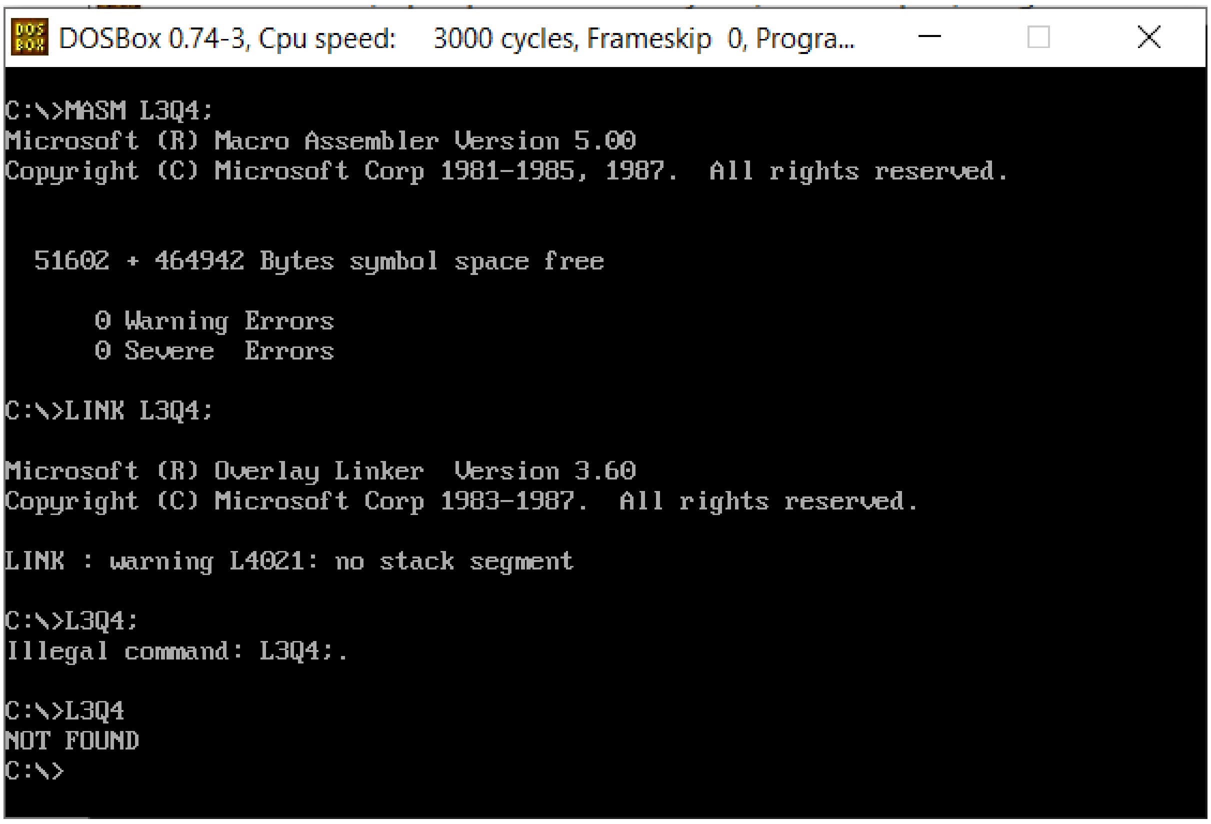
```
; Factorial of unsigned byte using recursion
STACK SEGMENT
        STK DW 100 DUP(?)
        TOS LABEL WORD
STACK ENDS

DATA SEGMENT
        NUM DB 07
        RES DW ?
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:STACK
START:  MOV AX, DATA
        MOV DS, AX
        MOV AX, STACK
        MOV SS, AX
        LEA SP, TOS
        MOV AL, NUM
        MOV AH, 0
        CALL FACT
        MOV AH, 4CH
        INT 21H
FACT PROC
        CMP AX, 01H
        JE EXIT
        PUSH AX
        DEC AX
        CALL FACT
        POP AX
        MUL RES
        MOV RES, AX
        RET
EXIT:   MOV RES, 01
        RET
FACT ENDP
CODE ENDS
END START
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
≡ File Edit View Run Breakpoints Data Options Window Help READY
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```
[■]=CPU 80486                                      1=[↑][↓]
 cs:0016►B44C           mov    ah,4C           ax 13B0    c=0
 cs:0018 CD21           int    21              bx 0000    z=0
 cs:001A 3D0100         cmp    ax,0001         cx 0000    s=0
 cs:001D 740E           je     002D            dx 0000    o=0
 cs:001F 50             push   ax              si 0000    p=1
 cs:0020 48             dec    ax              di 0000    a=0
 cs:0021 E8F6FF         call   001A            bp 0000    i=1
 cs:0024 58             pop    ax              sp 00C8    d=0
 cs:0025 F7260100       mul    word ptr [000   ds 48BA
 cs:0029 A30100         mov    [0001],ax       es 489D
 cs:002C C3             ret                    ss 48AD
 cs:002D C70601000100   mov    word ptr [000   cs 48BB
 cs:0033 C3             ret                    ip 0016

 es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
 es:0008 AD DE E6 01 AF 15 B0 01 ¡▐µ☺»§░☺
 es:0010 AF 15 83 02 13 10 92 01 »§â☻‼►Æ☺       ss:00CA 0000
 es:0018 01 01 01 00 02 FF FF FF ☺☺☺ ☻          ss:00C8►0000
```

4. Linear search in an array of 10 unsigned Words

```
; Linear search in an array of 10 unsigned Words
PRINTSTR MACRO MSG
        LEA DX, MSG
        MOV AH, 09H
        INT 21H
ENDM

DATA SEGMENT
        ARRAY DW 19H, 20, 21H, 22H, 10H, 11H, 25H, 62H, 12H,09H
        M2 DB 'FOUND $'
        M3 DB 'NOT FOUND $'
        ELE DW 45H
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:  MOV AX, DATA
        MOV DS, AX
        MOV CX, 10
        MOV SI, 0
        MOV AX, ELE
BACK1:  CMP AX, ARRAY[SI]
        JZ DOWN
        INC SI
        INC SI
        LOOP BACK1
        PRINTSTR M3
        JMP EXIT1
DOWN:   PRINTSTR M2
EXIT1:  MOV AH, 4CH
        INT 21H
CODE ENDS
END START
```

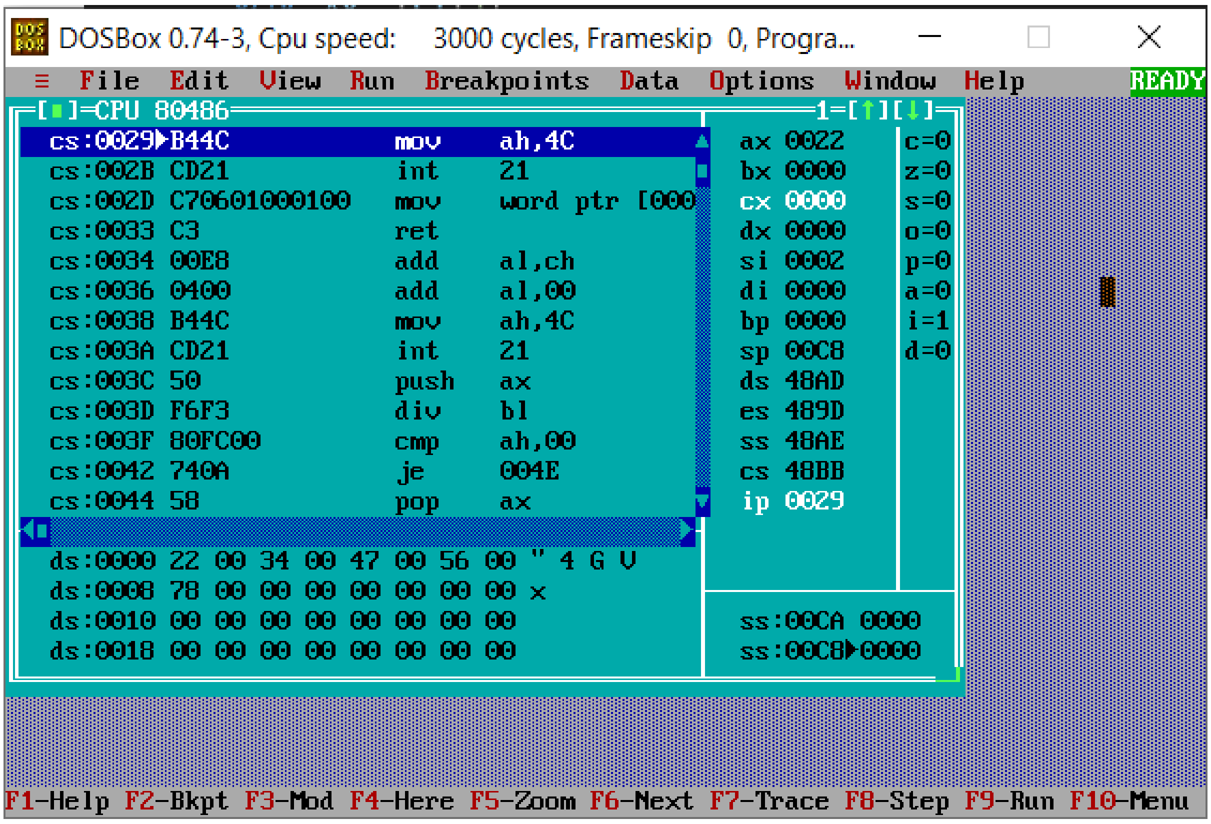


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
C:\>MASM L3Q4;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.


  51602 + 464942 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>LINK L3Q4;

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

LINK : warning L4021: no stack segment

C:\>L3Q4;
Illegal command: L3Q4;.

C:\>L3Q4
NOT FOUND
C:\>
```

5. Sorting of Signed Words Using Bubble sort

```
; Sorting of Signed Words Using Bubble sort
DATA SEGMENT
        ARRAY DW 34H, 78H, 56H, 47H, 22H
DATA ENDS

STACK SEGMENT
        DW 100 DUP(?)
        TOS LABEL WORD
STACK ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA, SS:STACK
START:  MOV AX, DATA
        MOV DS, AX
        MOV AX, STACK
        MOV SS, AX
        LEA SP, TOS
        MOV CX, 4
BACK1:  PUSH CX
        LEA SI, ARRAY
BACK:   MOV AX, [SI]
        CMP AX, [SI+2]
        JC DOWN
        XCHG AX, [SI+2]
        MOV [SI], AX
DOWN:   INC SI
        INC SI
        LOOP BACK
        POP CX
        LOOP BACK1
        MOV AH, 4CH
        INT 21H
CODE ENDS
END START
```



```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Progra...
 ≡  File  Edit  View  Run  Breakpoints  Data  Options  Window  Help         READY
 [■]=CPU 80486=======================================================1=[↑][↓]=
  cs:0029▶B44C              mov    ah,4C              ax 0022   c=0
  cs:002B CD21              int    21                 bx 0000   z=0
  cs:002D C70601000100      mov    word ptr [000      cx 0000   s=0
  cs:0033 C3                ret                       dx 0000   o=0
  cs:0034 00E8              add    al,ch              si 0002   p=0
  cs:0036 0400              add    al,00              di 0000   a=0
  cs:0038 B44C              mov    ah,4C              bp 0000   i=1
  cs:003A CD21              int    21                 sp 00C8   d=0
  cs:003C 50                push   ax                 ds 48AD
  cs:003D F6F3              div    bl                 es 489D
  cs:003F 80FC00            cmp    ah,00              ss 48AE
  cs:0042 740A              je     004E               cs 48BB
  cs:0044 58                pop    ax                 ip 0029

  ds:0000 22 00 34 00 47 00 56 00 " 4 G V
  ds:0008 78 00 00 00 00 00 00 00 x                   ss:00CA 0000
  ds:0010 00 00 00 00 00 00 00 00                     ss:00C8▶0000
  ds:0018 00 00 00 00 00 00 00 00

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```