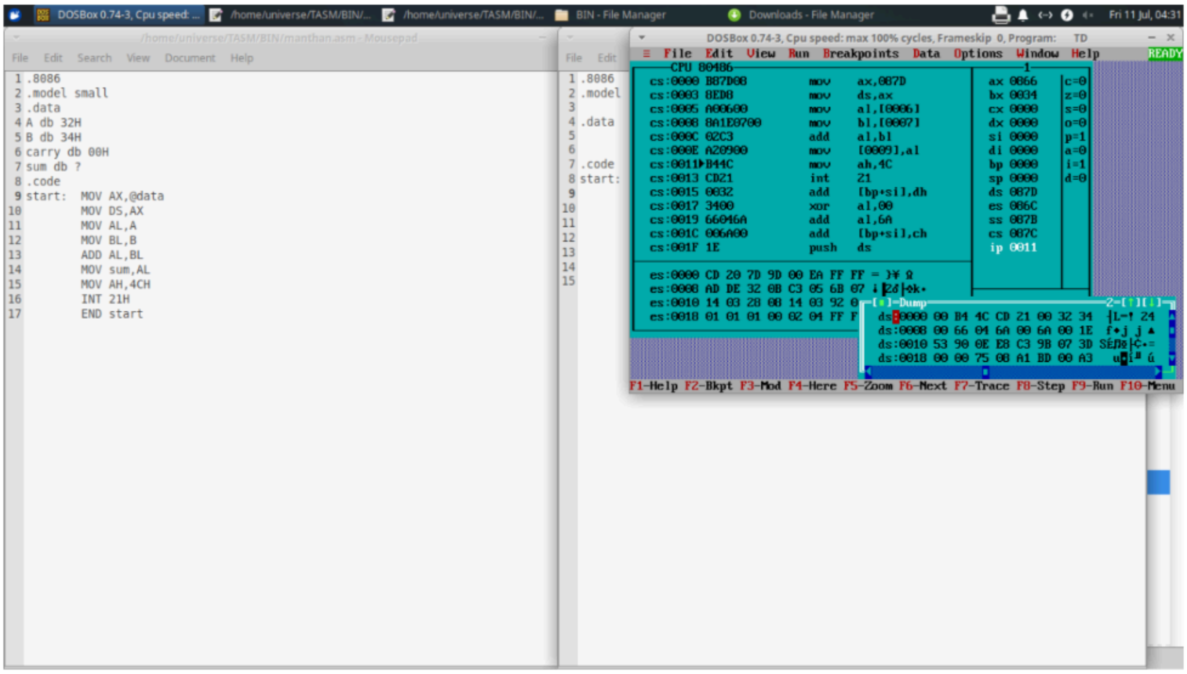
Add (all addressing Modes)



DOSBox 0.74-3, Cpu speed: ...
/home/universe/TASM/BIN/...
BIN - File Manager
Downloads - File Manager
Fri 11 Jul, 04:31
/home/universe/TASM/BIN/manthan.asm - Mousepad
File Edit Search View Document Help

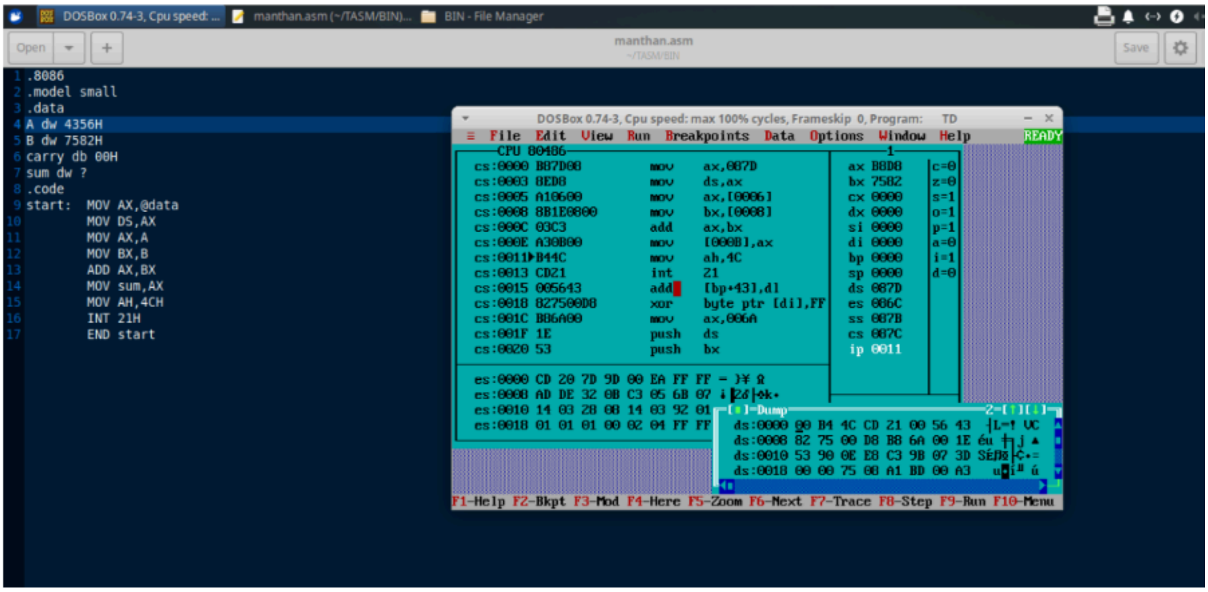
```
.8086
.model small
.data
A db 32H
B db 34H
carry db 00H
sum db ?
.code
start:  MOV AX,@data
        MOV DS,AX
        MOV AL,A
        MOV BL,B
        ADD AL,BL
        MOV sum,AL
        MOV AH,4CH
        INT 21H
        END start
```

```
.8086
.model

.data


.code
start:
```

DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
≡ File Edit View Run Breakpoints Data Options Window Help READY

```
CPU 80486
cs:0000 B87D08        mov    ax,087D          ax 0866   c=0
cs:0003 8ED8          mov    ds,ax            bx 0034   z=0
cs:0005 A00600        mov    al,[0006]        cx 0000   s=0
cs:0008 8A1E0700      mov    bl,[0007]        dx 0000   o=0
cs:000C 02C3          add    al,bl            si 0000   p=1
cs:000E A20900        mov    [0009],al        di 0000   a=0
cs:0011►B44C          mov    ah,4C            bp 0000   i=1
cs:0013 CD21          int    21               sp 0000   d=0
cs:0015 0032          add    [bp+si],dh       ds 087D
cs:0017 3400          xor    al,00            es 086C
cs:0019 66046A        add    al,6A            ss 087B
cs:001C 006A00        add    [bp+si],ch       cs 087C
cs:001F 1E            push   ds               ip 0011

es:0000 CD 20 7D 9D 00 EA FF FF = }¥ ♀
es:0008 AD DE 32 0B C3 05 6B 07 ↓▌2δ├♣k•
es:0010 14 03 28 08 14 03 92 0
es:0018 01 01 01 00 02 04 FF F

[■]-Dump
ds:0000 00 B4 4C CD 21 00 32 34  ┤L═! 24
ds:0008 00 66 04 6A 00 6A 00 1E  f♦j j ▲
ds:0010 53 90 0E E8 C3 9B 07 3D  SÉ♫Φ├¢•=
ds:0018 00 00 75 08 A1 BD 00 A3    u◘í╜ ú
```

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



DOSBox 0.74-3, Cpu speed: ...
manthan.asm (~/TASM/BIN)...
BIN - File Manager
Open
+
manthan.asm
~/TASM/BIN
Save

```
.8086
.model small
.data
A dw 4356H
B dw 7582H
carry db 00H
sum dw ?
.code
start:  MOV AX,@data
        MOV DS,AX
        MOV AX,A
        MOV BX,B
        ADD AX,BX
        MOV sum,AX
        MOV AH,4CH
        INT 21H
        END start
```

DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
≡ File Edit View Run Breakpoints Data Options Window Help READY

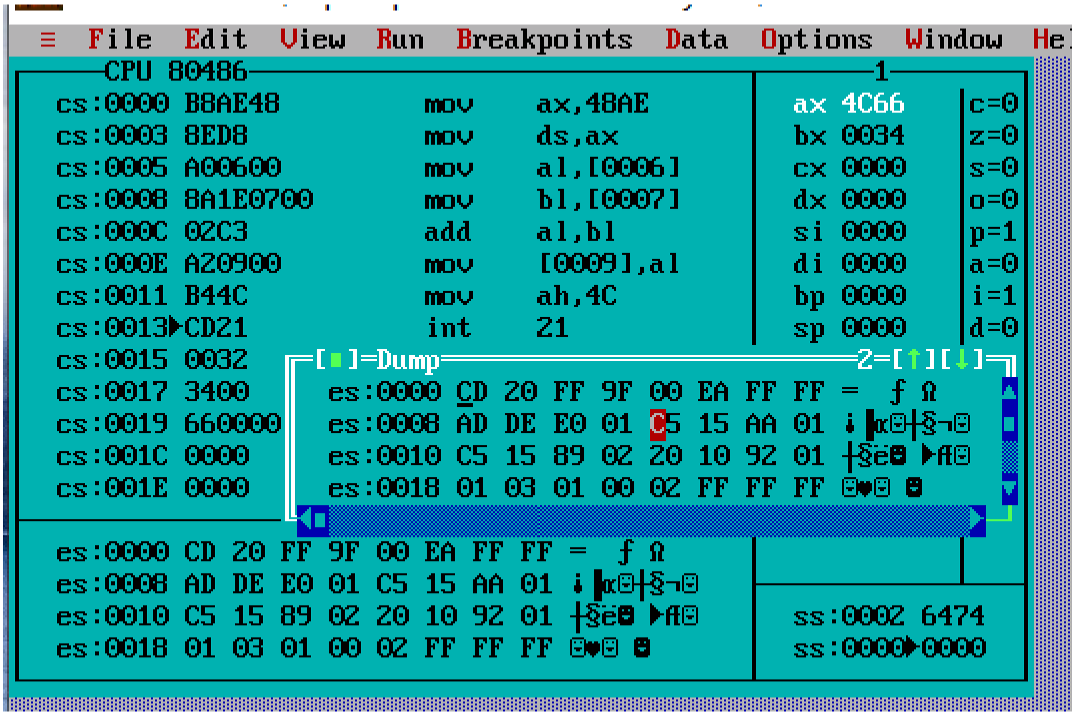
```
CPU 80486
cs:0000 B87D08        mov    ax,087D               ax B8D8   c=0
cs:0003 8ED8          mov    ds,ax                 bx 7582   z=0
cs:0005 A10600        mov    ax,[0006]             cx 0000   s=1
cs:0008 8B1E0800      mov    bx,[0008]             dx 0000   o=1
cs:000C 03C3          add    ax,bx                 si 0000   p=1
cs:000E A30B00        mov    [000B],ax             di 0000   a=0
cs:0011►B44C          mov    ah,4C                 bp 0000   i=1
cs:0013 CD21          int    21                    sp 0000   d=0
cs:0015 005643        add    [bp+43],dl            ds 087D
cs:0018 827500D8      xor    byte ptr [di],FF      es 086C
cs:001C B86A00        mov    ax,006A               ss 087B
cs:001F 1E            push   ds                    cs 087C
cs:0020 53            push   bx                    ip 0011

es:0000 CD 20 7D 9D 00 EA FF FF = }¥ ♀
es:0008 AD DE 32 0B C3 05 6B 07 ↓▌2δ├♣k•
es:0010 14 03 28 08 14 03 92 01
es:0018 01 01 01 00 02 04 FF FF

[■]-Dump
ds:0000 00 B4 4C CD 21 00 56 43  ┤L═! VC
ds:0008 82 75 00 D8 B8 6A 00 1E  éu ╪╕j ▲
ds:0010 53 90 0E E8 C3 9B 07 3D  SÉ♫Φ├¢•=
ds:0018 00 00 75 08 A1 BD 00 A3    u◘í╜ ú
```

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

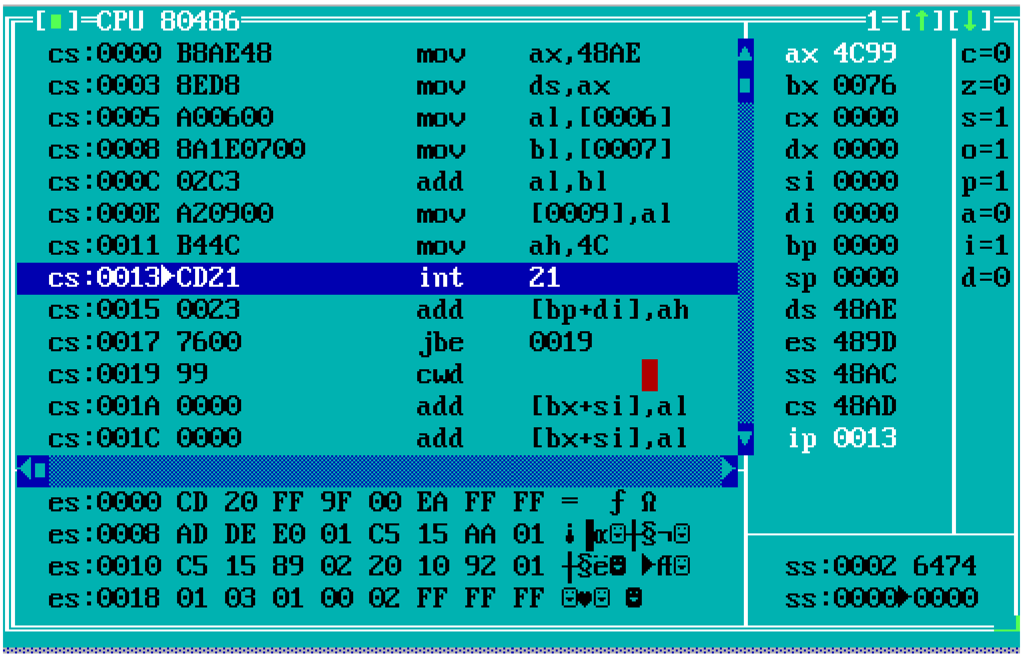
Add8:

```
.8086
.model small
.data
A db 32H
B db 34H
carry db 00H
sum db ?

.code
```

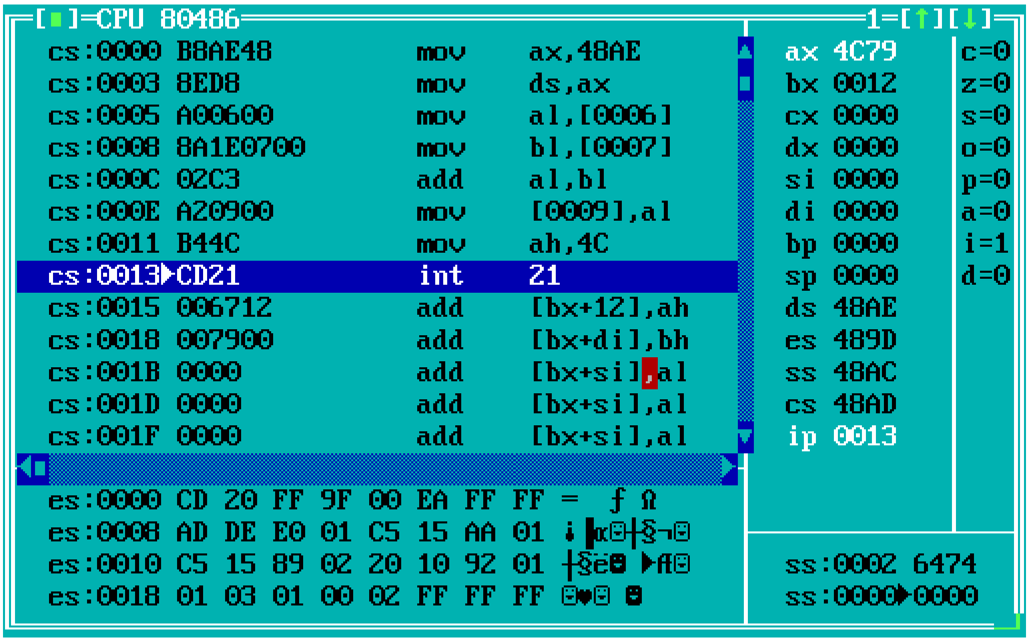
```
start:  MOV AX,@data
        MOV DS,AX
        MOV AL,A
        MOV BL,B
        ADD AL,BL
        MOV sum,AL
        MOV AH,4CH
        INT 21H
        END start
```



```
CPU 80486                                                    1
  cs:0000 B8AE48         mov    ax,48AE          ax 4C66   c=0
  cs:0003 8ED8           mov    ds,ax            bx 0034   z=0
  cs:0005 A00600         mov    al,[0006]        cx 0000   s=0
  cs:0008 8A1E0700       mov    bl,[0007]        dx 0000   o=0
  cs:000C 02C3           add    al,bl            si 0000   p=1
  cs:000E A20900         mov    [0009],al        di 0000   a=0
  cs:0011 B44C           mov    ah,4C            bp 0000   i=1
  cs:0013▶CD21           int    21               sp 0000   d=0
  cs:0015 0032     [■]=Dump=========================2=[↑][↓]
  cs:0017 3400        es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ ₧
  cs:0019 660000      es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺
  cs:001C 0000        es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺
  cs:001E 0000        es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻

  es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ ₧
  es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺       ss:0002 6474
  es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺       ss:0000▶0000
  es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```



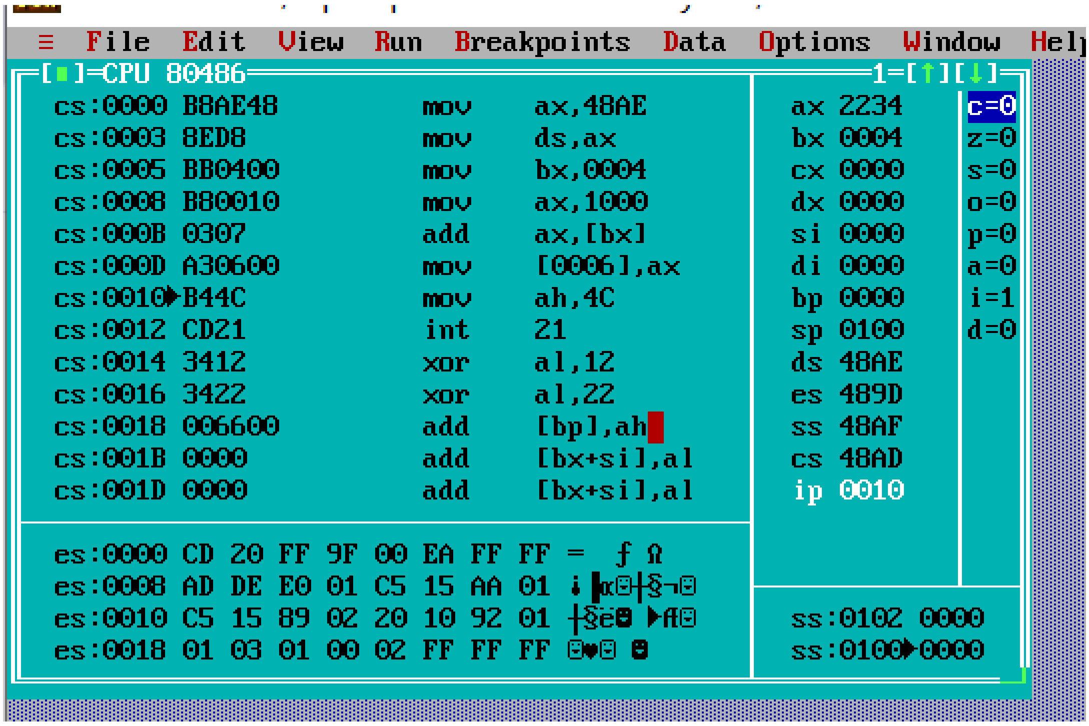
```
[■]=CPU 80486=========================================1=[↑][↓]
  cs:0000 B8AE48        mov     ax,48AE          ax 4C99   c=0
  cs:0003 8ED8          mov     ds,ax            bx 0076   z=0
  cs:0005 A00600        mov     al,[0006]        cx 0000   s=1
  cs:0008 8A1E0700      mov     bl,[0007]        dx 0000   o=1
  cs:000C 02C3          add     al,bl            si 0000   p=1
  cs:000E A20900        mov     [0009],al        di 0000   a=0
  cs:0011 B44C          mov     ah,4C            bp 0000   i=1
  cs:0013▶CD21          int     21               sp 0000   d=0
  cs:0015 0023          add     [bp+di],ah       ds 48AE
  cs:0017 7600          jbe     0019             es 489D
  cs:0019 99            cwd                      ss 48AC
  cs:001A 0000          add     [bx+si],al       cs 48AD
  cs:001C 0000          add     [bx+si],al       ip 0013

  es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ ₧
  es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺
  es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺       ss:0002 6474
  es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻          ss:0000▶0000
```



```
[■]=CPU 80486=========================================1=[↑][↓]=
 cs:0000 B8AE48          mov    ax,48AE       ax 4C79   c=0
 cs:0003 8ED8            mov    ds,ax         bx 0012   z=0
 cs:0005 A00600          mov    al,[0006]     cx 0000   s=0
 cs:0008 8A1E0700        mov    bl,[0007]     dx 0000   o=0
 cs:000C 02C3            add    al,bl         si 0000   p=0
 cs:000E A20900          mov    [0009],al     di 0000   a=0
 cs:0011 B44C            mov    ah,4C         bp 0000   i=1
 cs:0013►CD21            int    21            sp 0000   d=0
 cs:0015 006712          add    [bx+12],ah    ds 48AE
 cs:0018 007900          add    [bx+di],bh    es 489D
 cs:001B 0000            add    [bx+si],al    ss 48AC
 cs:001D 0000            add    [bx+si],al    cs 48AD
 cs:001F 0000            add    [bx+si],al    ip 0013

 es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01 ↓▌α☺┼§¬☺
 es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Å☺      ss:0002 6474
 es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻         ss:0000►0000
```

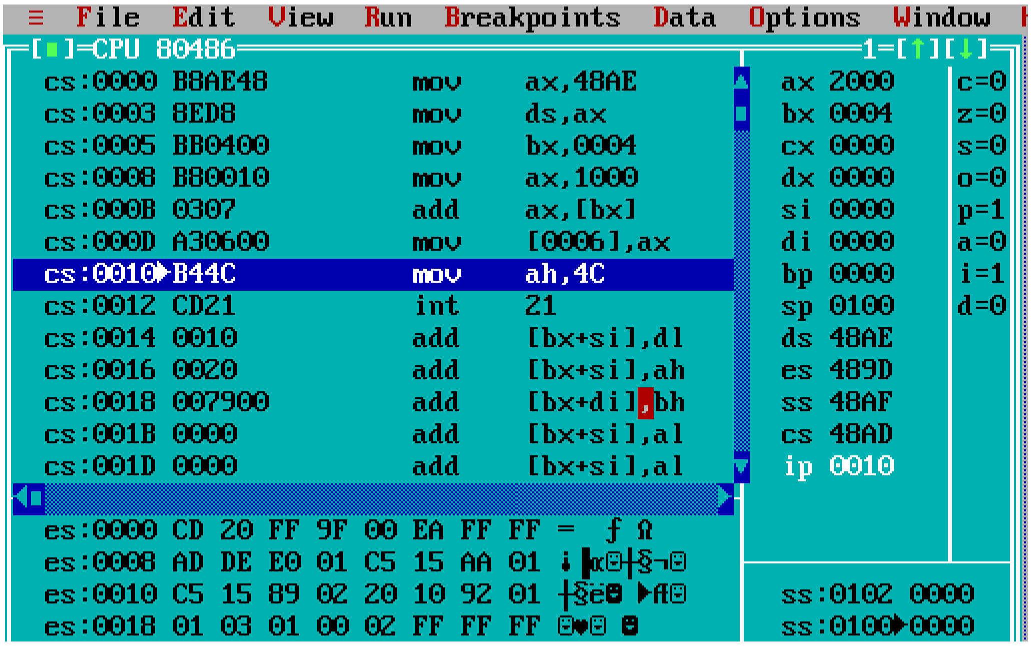
Add(Indirect):

```
.8086
.model small
.stack 100h
.data
num1 dw 1234h
result dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   lea bx,num1
   mov ax,1000h
   add ax,[bx]
   mov result,ax
   mov ah,4ch
   int 21h
end main
```



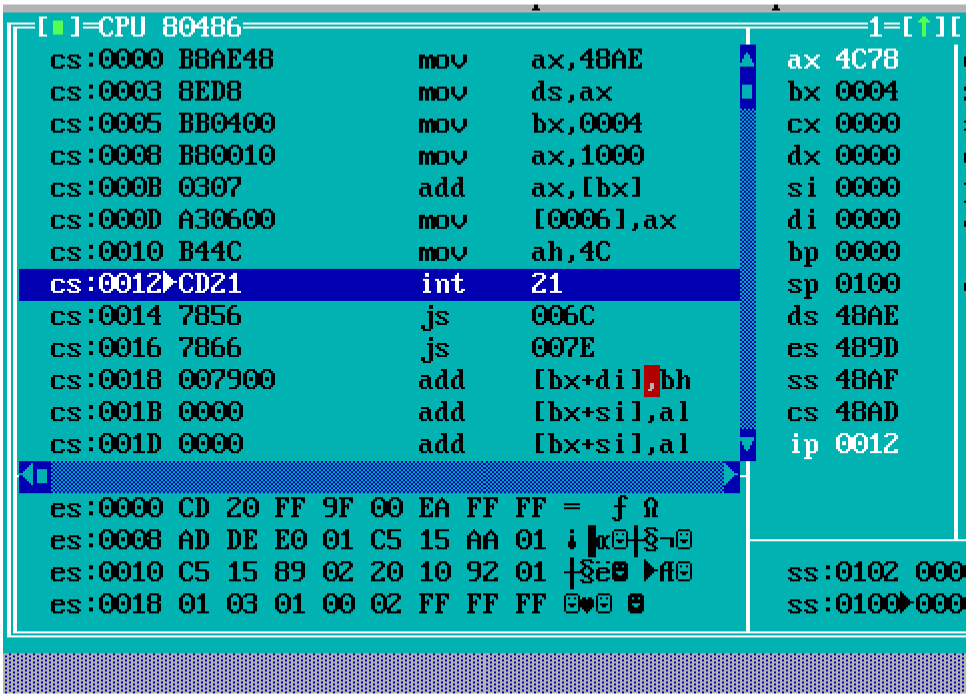
```
[■]=CPU 80486=====================================1=[↑][↓]=
   cs:0000 B8AE48         mov    ax,48AE        ax 2234   c=0
   cs:0003 8ED8           mov    ds,ax          bx 0004   z=0
   cs:0005 BB0400         mov    bx,0004        cx 0000   s=0
   cs:0008 B80010         mov    ax,1000        dx 0000   o=0
   cs:000B 0307           add    ax,[bx]        si 0000   p=0
   cs:000D A30600         mov    [0006],ax      di 0000   a=0
   cs:0010►B44C           mov    ah,4C          bp 0000   i=1
   cs:0012 CD21           int    21             sp 0100   d=0
   cs:0014 3412           xor    al,12          ds 48AE
   cs:0016 3422           xor    al,22          es 489D
   cs:0018 006600         add    [bp],ah        ss 48AF
   cs:001B 0000           add    [bx+si],al     cs 48AD
   cs:001D 0000           add    [bx+si],al     ip 0010

   es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
   es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
   es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     ss:0102 0000
   es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻        ss:0100►0000
```

```
.8086
.model small
.stack 100h
.data
num1 dw 1000h
result dw ?
.code
main:
    mov ax,@data
    mov ds,ax
    lea bx,num1
    mov ax,1000h
    add ax,[bx]
    mov result,ax
    mov ah,4ch
    int 21h
end main
```



```
cs:0000 B8AE48         mov    ax,48AE        ax 2000   c=0
cs:0003 8ED8           mov    ds,ax          bx 0004   z=0
cs:0005 BB0400         mov    bx,0004        cx 0000   s=0
cs:0008 B80010         mov    ax,1000        dx 0000   o=0
cs:000B 0307           add    ax,[bx]        si 0000   p=1
cs:000D A30600         mov    [0006],ax      di 0000   a=0
cs:0010►B44C           mov    ah,4C          bp 0000   i=1
cs:0012 CD21           int    21             sp 0100   d=0
cs:0014 0010           add    [bx+si],dl     ds 48AE
cs:0016 0020           add    [bx+si],ah     es 489D
cs:0018 007900         add    [bx+di],bh     ss 48AF
cs:001B 0000           add    [bx+si],al     cs 48AD
cs:001D 0000           add    [bx+si],al     ip 0010
es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
es:0008 AD DE E0 01 C5 15 AA 01 ¡ ▌α☺┼§¬☺
es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺       ss:0102 0000
es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻          ss:0100►0000
```

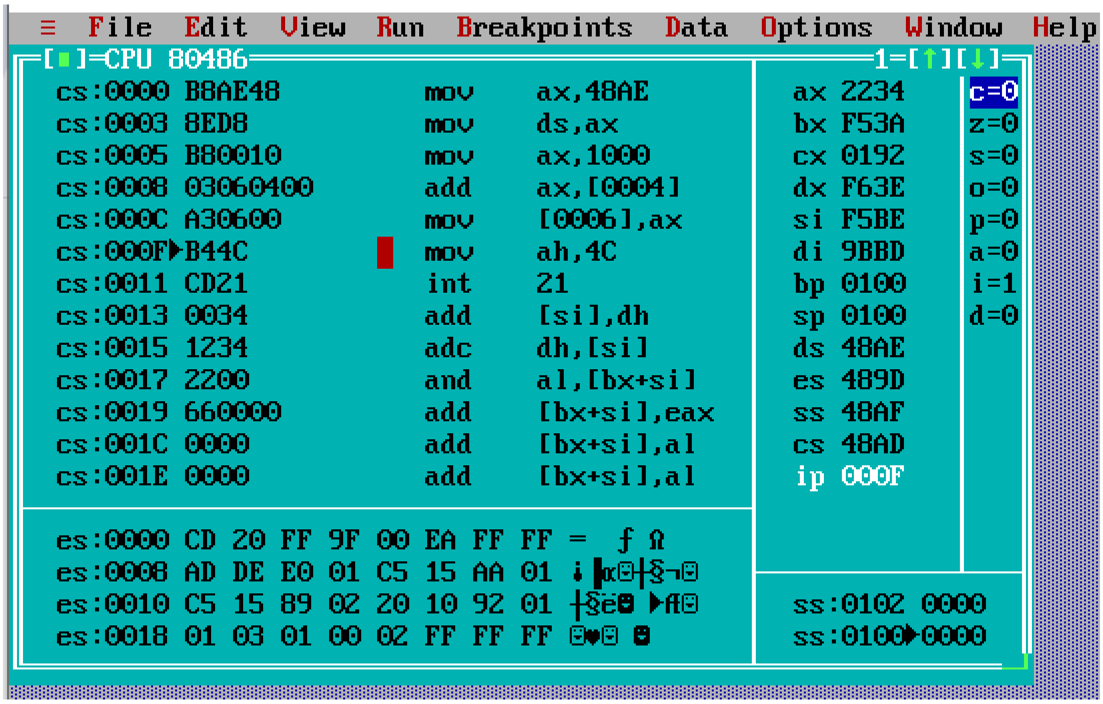
```
.8086
.model small
.stack 100h
.data
num1 dw 5678h
result dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   lea bx,num1
   mov ax,1000h
   add ax,[bx]
   mov result,ax
   mov ah,4ch
   int 21h
end main
```



```
[■]=CPU 80486                                       1=[↑][
 cs:0000 B8AE48         mov    ax,48AE          ax 4C78
 cs:0003 8ED8           mov    ds,ax            bx 0004
 cs:0005 BB0400         mov    bx,0004          cx 0000
 cs:0008 B80010         mov    ax,1000          dx 0000
 cs:000B 0307           add    ax,[bx]          si 0000
 cs:000D A30600         mov    [0006],ax        di 0000
 cs:0010 B44C           mov    ah,4C            bp 0000
 cs:0012►CD21           int    21               sp 0100
 cs:0014 7856           js     006C             ds 48AE
 cs:0016 7866           js     007E             es 489D
 cs:0018 007900         add    [bx+di],bh       ss 48AF
 cs:001B 0000           add    [bx+si],al       cs 48AD
 cs:001D 0000           add    [bx+si],al       ip 0012

 es:0000 CD 20 FF 9F 00 EA FF FF ═ ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺      ss:0102 000
 es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺      ss:0100►000
 es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

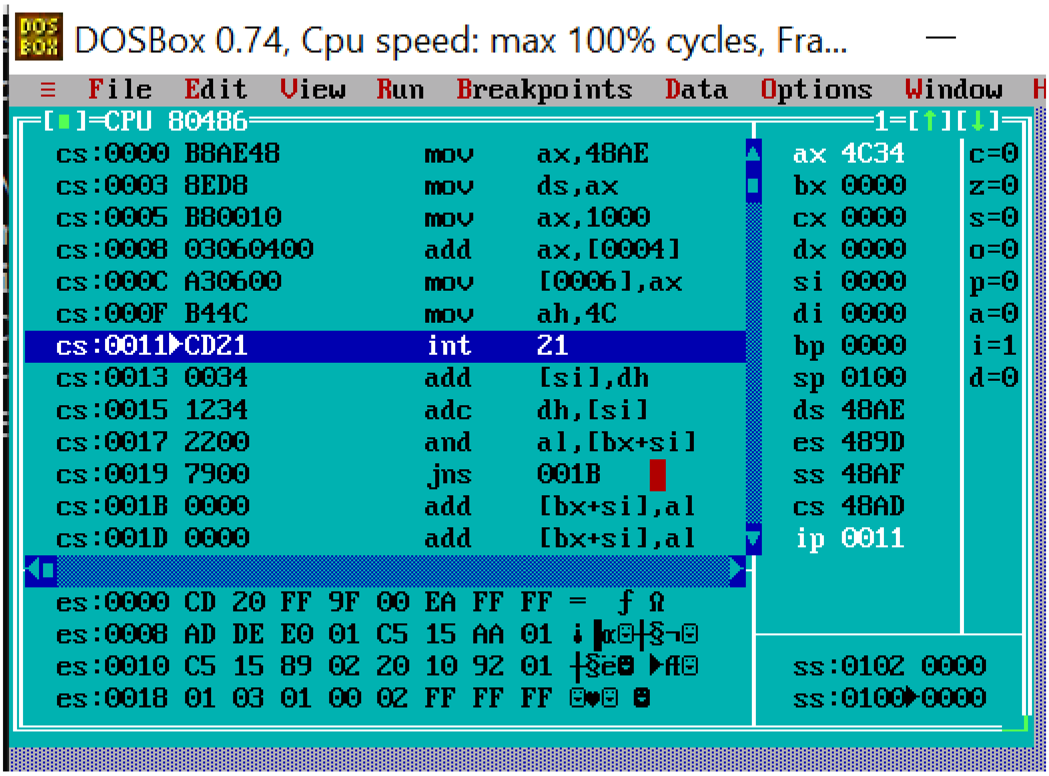
Add(Memory):

```
.8086
.model small
.stack 100h
.data
num1 dw 1234h
result dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov ax,1000h
   add ax,num1
   mov result,ax
   mov ah,4ch
   int 21h
end main
```



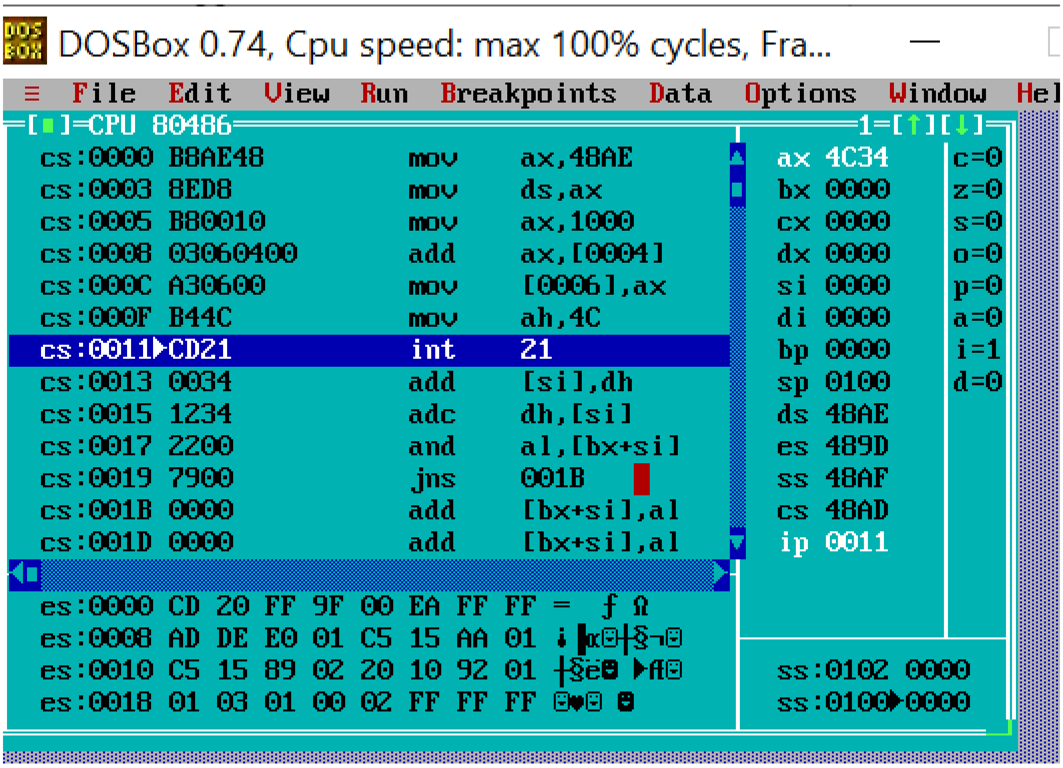
```
cs:0000 B8AE48         mov    ax,48AE      ax 2234   c=0
cs:0003 8ED8           mov    ds,ax        bx F53A   z=0
cs:0005 B80010         mov    ax,1000      cx 0192   s=0
cs:0008 03060400       add    ax,[0004]    dx F63E   o=0
cs:000C A30600         mov    [0006],ax    si F5BE   p=0
cs:000F►B44C           mov    ah,4C        di 9BBD   a=0
cs:0011 CD21           int    21           bp 0100   i=1
cs:0013 0034           add    [si],dh      sp 0100   d=0
cs:0015 1234           adc    dh,[si]      ds 48AE
cs:0017 2200           and    al,[bx+si]   es 489D
cs:0019 660000         add    [bx+si],eax  ss 48AF
cs:001C 0000           add    [bx+si],al   cs 48AD
cs:001E 0000           add    [bx+si],al   ip 000F

es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺     ss:0102 0000
es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     ss:0100►0000
es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

```
.8086
.model small
.stack 100h
.data
num1 dw 3456h
result dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov ax,1000h
   add ax,num1
   mov result,ax
   mov ah,4ch
   int 21h
end main
```


DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
≡ File Edit View Run Breakpoints Data Options Window H

```
[■]=CPU 80486                                              1=[↑][↓]
  cs:0000 B8AE48        mov    ax,48AE           ax 4C34   c=0
  cs:0003 8ED8          mov    ds,ax             bx 0000   z=0
  cs:0005 B80010        mov    ax,1000           cx 0000   s=0
  cs:0008 03060400      add    ax,[0004]         dx 0000   o=0
  cs:000C A30600        mov    [0006],ax         si 0000   p=0
  cs:000F B44C          mov    ah,4C             di 0000   a=0
  cs:0011►CD21          int    21                bp 0000   i=1
  cs:0013 0034          add    [si],dh           sp 0100   d=0
  cs:0015 1234          adc    dh,[si]           ds 48AE
  cs:0017 2200          and    al,[bx+si]        es 489D
  cs:0019 7900          jns    001B              ss 48AF
  cs:001B 0000          add    [bx+si],al        cs 48AD
  cs:001D 0000          add    [bx+si],al        ip 0011

  es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
  es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
  es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     ss:0102 0000
  es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻        ss:0100►0000
```

```
.8086
.model small
.stack 100h
.data
num1 dw 8765h
result dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov ax,1000h
   add ax,num1
   mov result,ax
   mov ah,4ch
   int 21h
end main
```

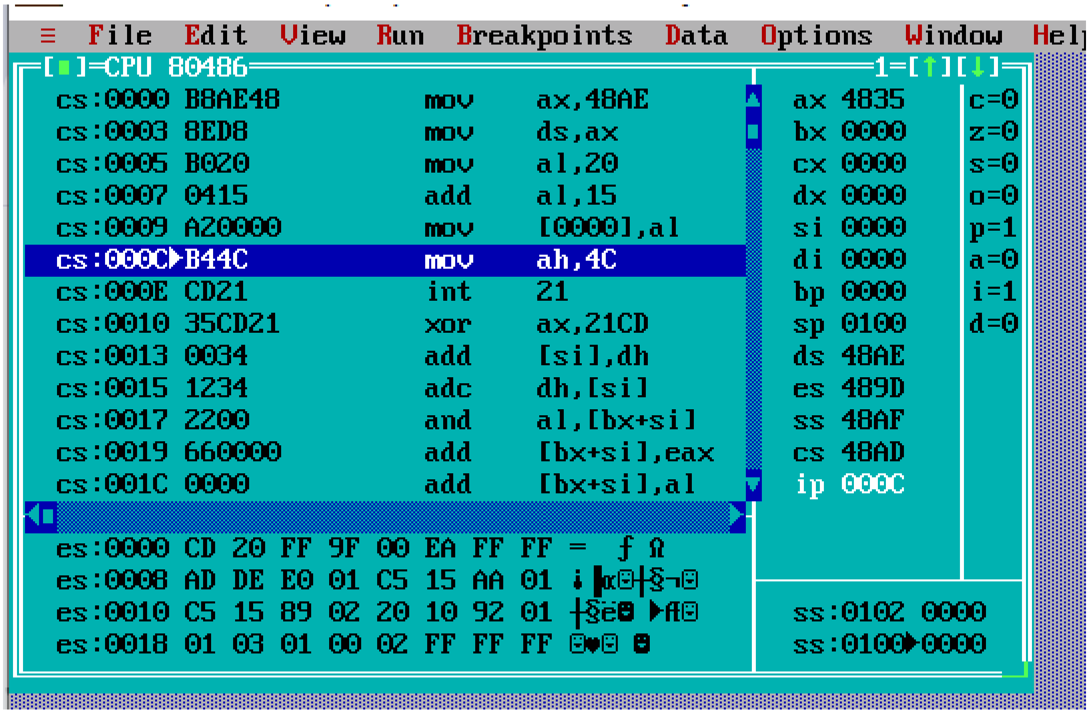


DOSBox 0.74, Cpu speed: max 100% cycles, Fra...

```
=[■]=CPU 80486=========================================1=[↑][↓]=
 cs:0000 B8AE48          mov    ax,48AE      ax 4C34    c=0
 cs:0003 8ED8            mov    ds,ax        bx 0000    z=0
 cs:0005 B80010          mov    ax,1000      cx 0000    s=0
 cs:0008 03060400        add    ax,[0004]    dx 0000    o=0
 cs:000C A30600          mov    [0006],ax    si 0000    p=0
 cs:000F B44C            mov    ah,4C        di 0000    a=0
 cs:0011►CD21            int    21           bp 0000    i=1
 cs:0013 0034            add    [si],dh      sp 0100    d=0
 cs:0015 1234            adc    dh,[si]      ds 48AE
 cs:0017 2200            and    al,[bx+si]   es 489D
 cs:0019 7900            jns    001B         ss 48AF
 cs:001B 0000            add    [bx+si],al   cs 48AD
 cs:001D 0000            add    [bx+si],al   ip 0011

 es:0000 CD 20 FF 9F 00 EA FF FF ═  ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺     ss:0102 0000
 es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     ss:0100►0000
 es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

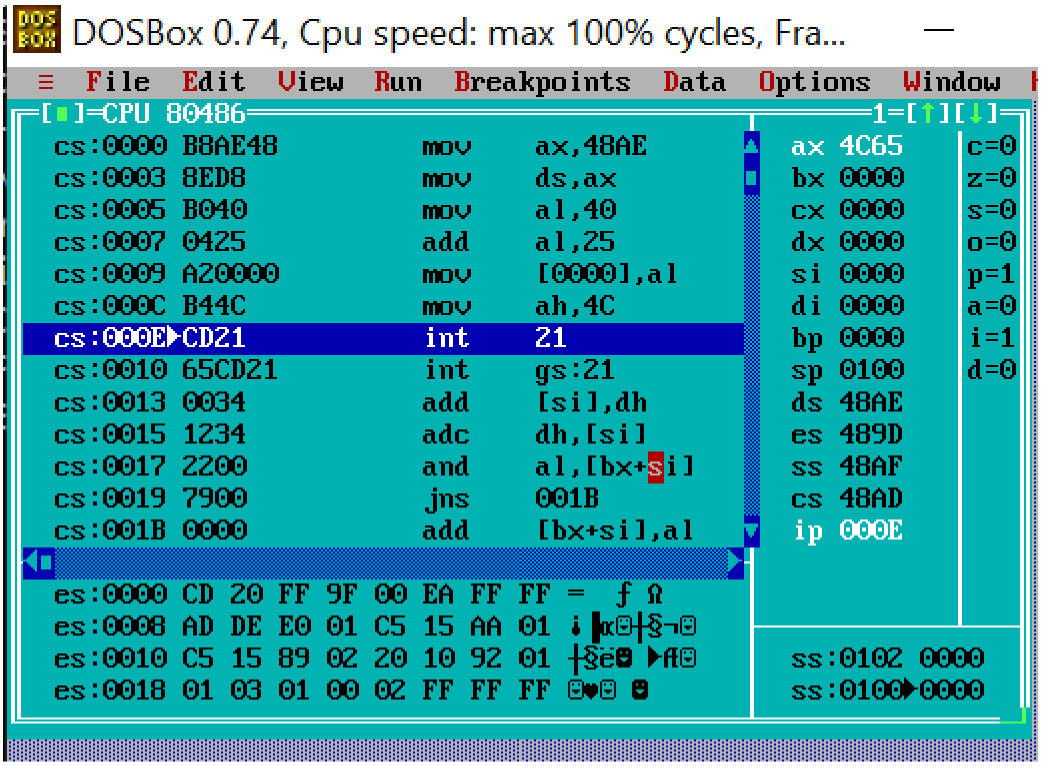
Add(Immediate):

```
.8086
.model small
.stack 100h
.data
result db ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,20h
   add al,15h
   mov result,al
   mov ah,4ch
   int 21h
end main
```



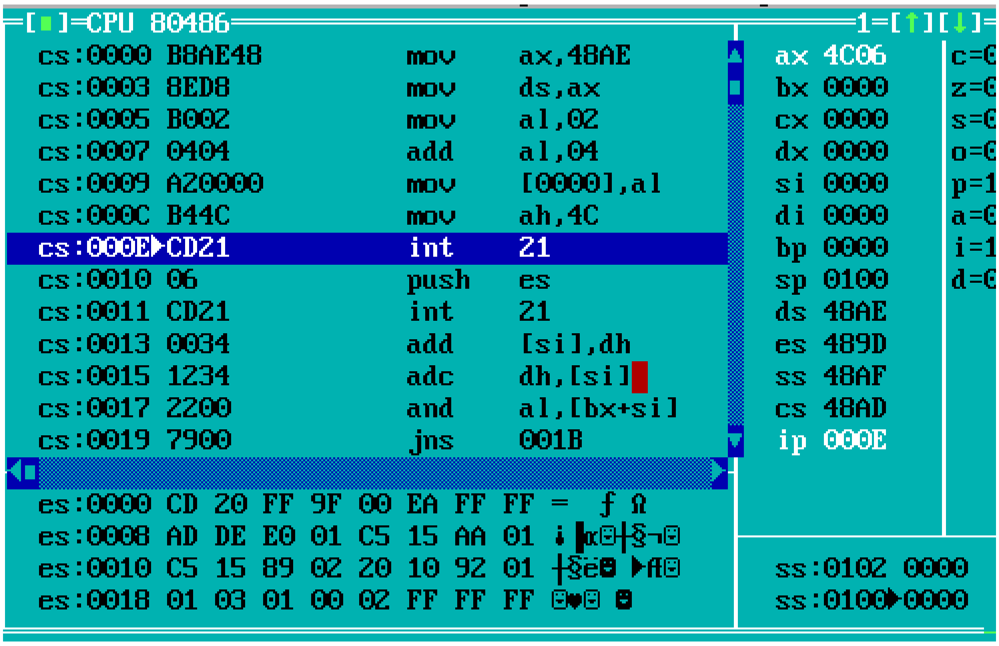
```
[■]=CPU 80486=================================1=[↑][↓]=
 cs:0000 B8AE48        mov   ax,48AE        ax 4835   c=0
 cs:0003 8ED8          mov   ds,ax          bx 0000   z=0
 cs:0005 B020          mov   al,20          cx 0000   s=0
 cs:0007 0415          add   al,15          dx 0000   o=0
 cs:0009 A20000        mov   [0000],al      si 0000   p=1
 cs:000C►B44C          mov   ah,4C          di 0000   a=0
 cs:000E CD21          int   21             bp 0000   i=1
 cs:0010 35CD21        xor   ax,21CD        sp 0100   d=0
 cs:0013 0034          add   [si],dh        ds 48AE
 cs:0015 1234          adc   dh,[si]        es 489D
 cs:0017 2200          and   al,[bx+si]     ss 48AF
 cs:0019 660000        add   [bx+si],eax    cs 48AD
 cs:001C 0000          add   [bx+si],al     ip 000C

 es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
 es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     ss:0102 0000
 es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻        ss:0100►0000
```

```
.8086
.model small
.stack 100h
.data
result db ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,40h
   add al,25h
   mov result,al
   mov ah,4ch
   int 21h
end main
```


DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
≡ File Edit View Run Breakpoints Data Options Window

```
[■]=CPU 80486=====================================1=[↑][↓]=
 cs:0000 B8AE48         mov    ax,48AE        ax 4C65   c=0
 cs:0003 8ED8           mov    ds,ax          bx 0000   z=0
 cs:0005 B040           mov    al,40          cx 0000   s=0
 cs:0007 0425           add    al,25          dx 0000   o=0
 cs:0009 A20000         mov    [0000],al      si 0000   p=1
 cs:000C B44C           mov    ah,4C          di 0000   a=0
 cs:000E►CD21           int    21             bp 0000   i=1
 cs:0010 65CD21         int    gs:21          sp 0100   d=0
 cs:0013 0034           add    [si],dh        ds 48AE
 cs:0015 1234           adc    dh,[si]        es 489D
 cs:0017 2200           and    al,[bx+si]     ss 48AF
 cs:0019 7900           jns    001B           cs 48AD
 cs:001B 0000           add    [bx+si],al     ip 000E

 es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
 es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     ss:0102 0000
 es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻        ss:0100►0000
```

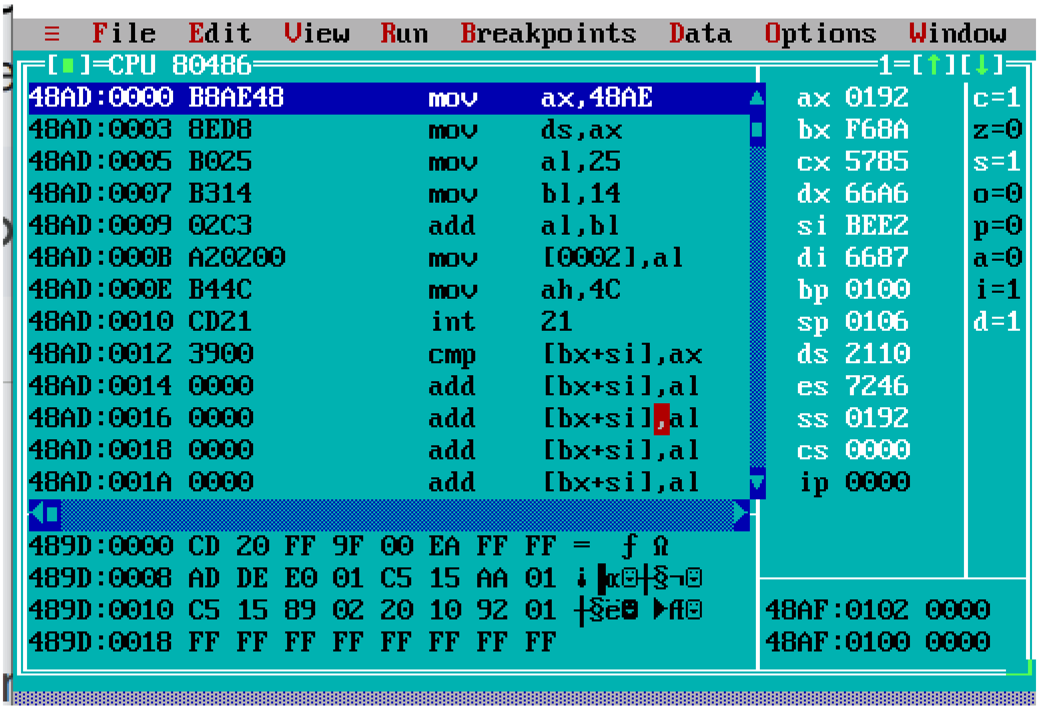
```
.8086
.model small
.stack 100h
.data
result db ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,2h
   add al,4h
   mov result,al
   mov ah,4ch
   int 21h
end main
```



```
=[■]=CPU 80486=================================================1=[↑][↓]=
  cs:0000 B8AE48           mov    ax,48AE         ax 4C06     c=0
  cs:0003 8ED8             mov    ds,ax           bx 0000     z=0
  cs:0005 B002             mov    al,02           cx 0000     s=0
  cs:0007 0404             add    al,04           dx 0000     o=0
  cs:0009 A20000           mov    [0000],al       si 0000     p=1
  cs:000C B44C             mov    ah,4C           di 0000     a=0
  cs:000E►CD21             int    21              bp 0000     i=1
  cs:0010 06               push   es              sp 0100     d=0
  cs:0011 CD21             int    21              ds 48AE
  cs:0013 0034             add    [si],dh         es 489D
  cs:0015 1234             adc    dh,[si]         ss 48AF
  cs:0017 2200             and    al,[bx+si]      cs 48AD
  cs:0019 7900             jns    001B            ip 000E

  es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
  es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
  es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺    ss:0102 0000
  es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻       ss:0100►0000
```

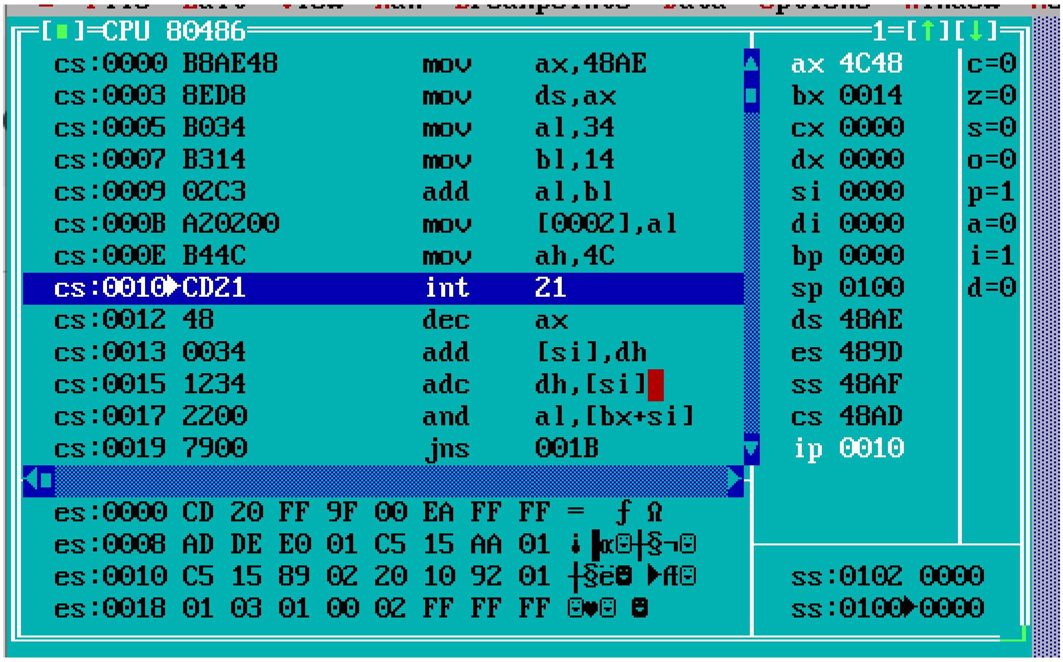
Add(Register Immediate):

```
.8086
.model small
.stack 100h
.data
result db ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,25h
   mov bl,14h
   add al,bl
   mov result,al
   mov ah,4ch
   int 21h
end main
```



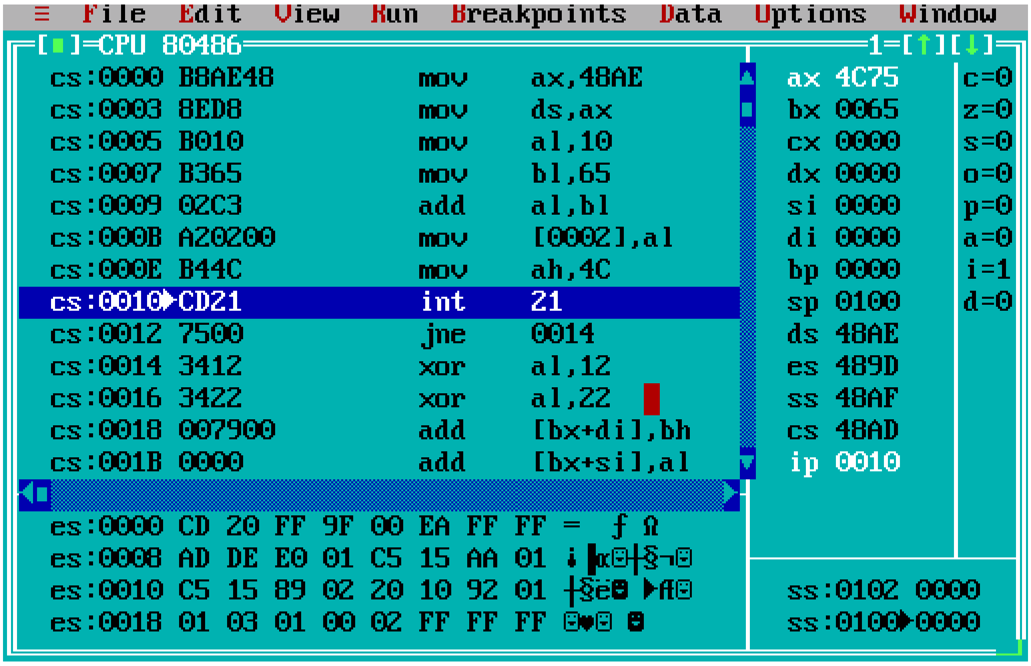
```
48AD:0000 B8AE48         mov     ax,48AE      ax 0192   c=1
48AD:0003 8ED8           mov     ds,ax        bx F68A   z=0
48AD:0005 B025           mov     al,25        cx 5785   s=1
48AD:0007 B314           mov     bl,14        dx 66A6   o=0
48AD:0009 02C3           add     al,bl        si BEE2   p=0
48AD:000B A20200         mov     [0002],al    di 6687   a=0
48AD:000E B44C           mov     ah,4C        bp 0100   i=1
48AD:0010 CD21           int     21           sp 0106   d=1
48AD:0012 3900           cmp     [bx+si],ax   ds 2110
48AD:0014 0000           add     [bx+si],al   es 7246
48AD:0016 0000           add     [bx+si],al   ss 0192
48AD:0018 0000           add     [bx+si],al   cs 0000
48AD:001A 0000           add     [bx+si],al   ip 0000

489D:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
489D:0008 AD DE E0 01 C5 15 AA 01 ¡ α☺┼§¬☺     48AF:0102 0000
489D:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     48AF:0100 0000
489D:0018 FF FF FF FF FF FF FF FF
```

```
.8086
.model small
.stack 100h
.data
result db ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,34h
   mov bl,14h
   add al,bl
   mov result,al
   mov ah,4ch
   int 21h
end main
```



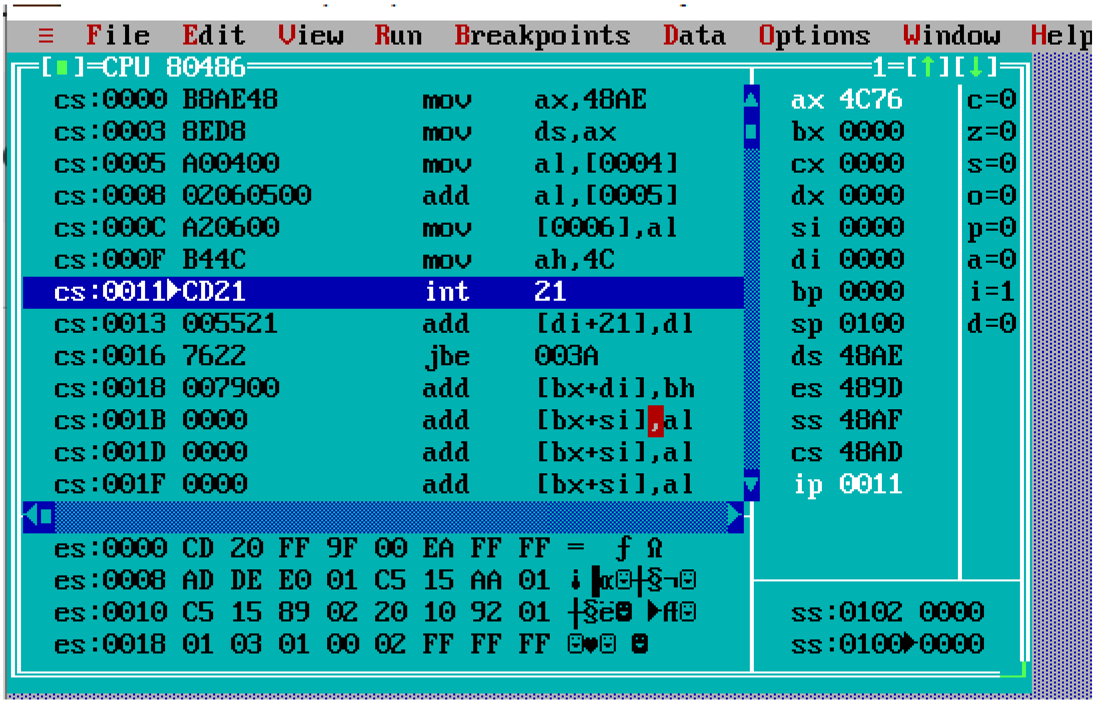
```
[■]=CPU 80486================================================1=[↑][↓]=
 cs:0000 B8AE48         mov    ax,48AE            ax 4C48   c=0
 cs:0003 8ED8           mov    ds,ax              bx 0014   z=0
 cs:0005 B034           mov    al,34              cx 0000   s=0
 cs:0007 B314           mov    bl,14              dx 0000   o=0
 cs:0009 02C3           add    al,bl              si 0000   p=1
 cs:000B A20200         mov    [0002],al          di 0000   a=0
 cs:000E B44C           mov    ah,4C              bp 0000   i=1
 cs:0010►CD21           int    21                 sp 0100   d=0
 cs:0012 48             dec    ax                 ds 48AE
 cs:0013 0034           add    [si],dh            es 489D
 cs:0015 1234           adc    dh,[si]            ss 48AF
 cs:0017 2200           and    al,[bx+si]         cs 48AD
 cs:0019 7900           jns    001B               ip 0010

 es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺       ss:0102 0000
 es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺       ss:0100►0000
 es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

```
.8086
.model small
.stack 100h
.data
result db ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,10h
   mov bl,65h
   add al,bl
   mov result,al
   mov ah,4ch
   int 21h
end main
```



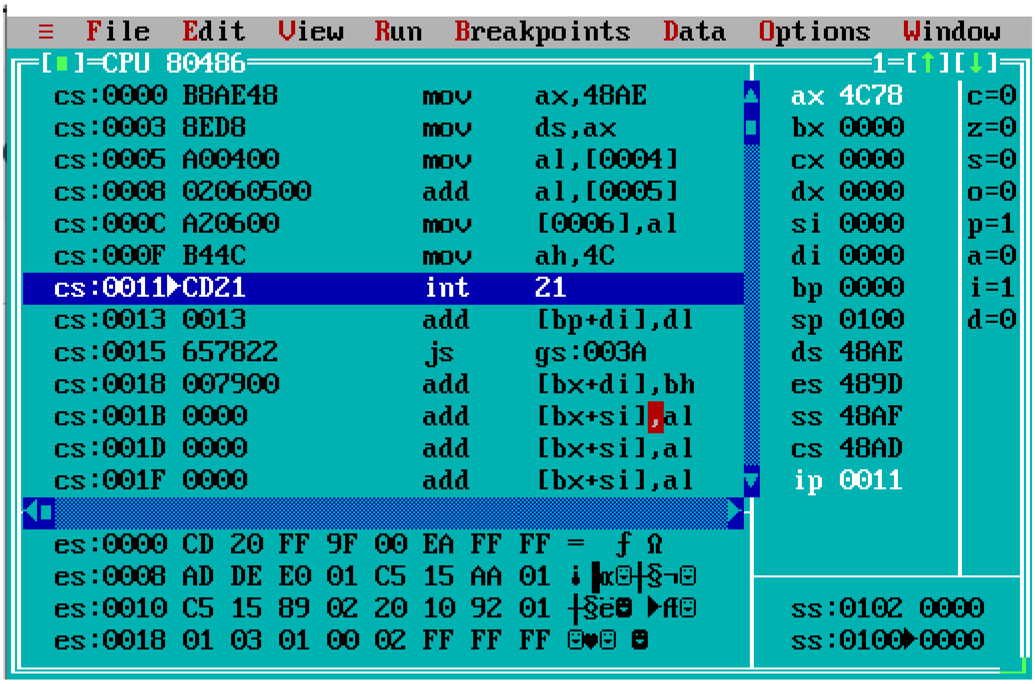
```
cs:0000 B8AE48        mov    ax,48AE       ax 4C75   c=0
cs:0003 8ED8          mov    ds,ax         bx 0065   z=0
cs:0005 B010          mov    al,10         cx 0000   s=0
cs:0007 B365          mov    bl,65         dx 0000   o=0
cs:0009 02C3          add    al,bl         si 0000   p=0
cs:000B A20200        mov    [0002],al     di 0000   a=0
cs:000E B44C          mov    ah,4C         bp 0000   i=1
cs:0010►CD21          int    21            sp 0100   d=0
cs:0012 7500          jne    0014          ds 48AE
cs:0014 3412          xor    al,12         es 489D
cs:0016 3422          xor    al,22         ss 48AF
cs:0018 007900        add    [bx+di],bh    cs 48AD
cs:001B 0000          add    [bx+si],al    ip 0010

es:0000 CD 20 FF 9F 00 EA FF FF = ƒ ß
es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     ss:0102 0000
es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻        ss:0100►0000
```

```
.8086
.model small
.stack 100h
.data
num1 db 55h
num2 db 21h
result db ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,num1
   add al,num2
   mov result,al
   mov ah,4ch
   int 21h
end main
```



```
cs:0000 B8AE48         mov    ax,48AE        ax 4C76   c=0
cs:0003 8ED8           mov    ds,ax          bx 0000   z=0
cs:0005 A00400         mov    al,[0004]      cx 0000   s=0
cs:0008 02060500       add    al,[0005]      dx 0000   o=0
cs:000C A20600         mov    [0006],al      si 0000   p=0
cs:000F B44C           mov    ah,4C          di 0000   a=0
cs:0011►CD21           int    21             bp 0000   i=1
cs:0013 005521         add    [di+21],dl     sp 0100   d=0
cs:0016 7622           jbe    003A           ds 48AE
cs:0018 007900         add    [bx+di],bh     es 489D
cs:001B 0000           add    [bx+si],al     ss 48AF
cs:001D 0000           add    [bx+si],al     cs 48AD
cs:001F 0000           add    [bx+si],al     ip 0011

es:0000 CD 20 FF 9F 00 EA FF FF ═  ƒ Ω
es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺       ss:0102 0000
es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺       ss:0100►0000
es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

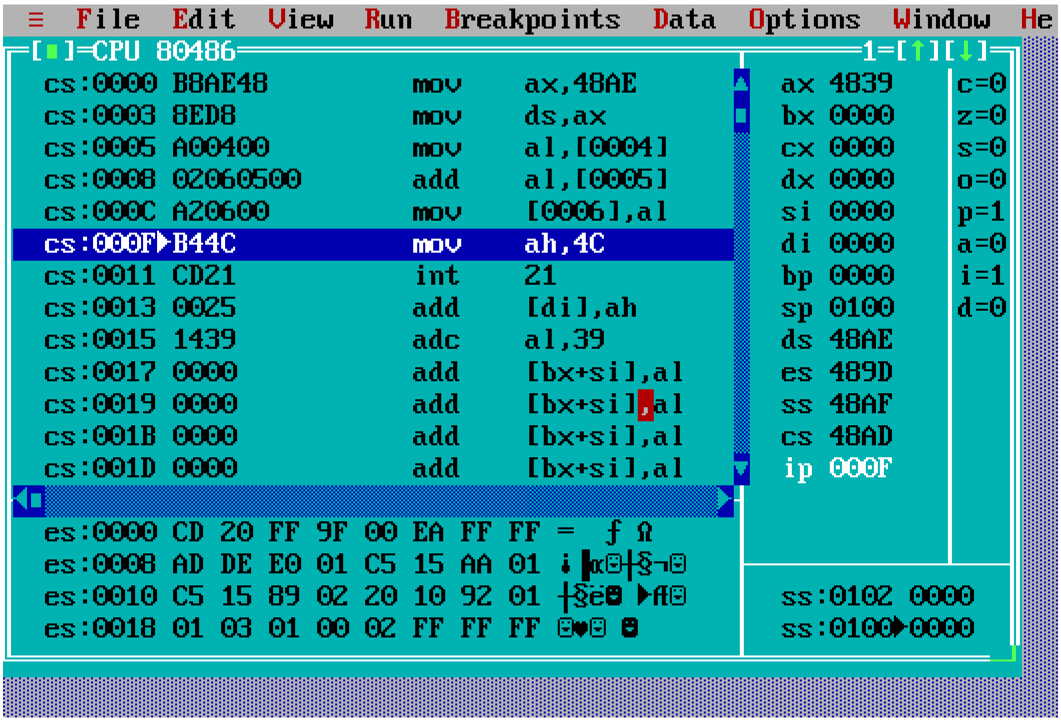
```
.8086
.model small
.stack 100h
.data
num1 db 13h
num2 db 65h
result db ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,num1
   add al,num2
   mov result,al
   mov ah,4ch
   int 21h
end main
```



```
cs:0000 B8AE48      mov   ax,48AE        ax 4C78   c=0
cs:0003 8ED8        mov   ds,ax          bx 0000   z=0
cs:0005 A00400      mov   al,[0004]      cx 0000   s=0
cs:0008 02060500    add   al,[0005]      dx 0000   o=0
cs:000C A20600      mov   [0006],al      si 0000   p=1
cs:000F B44C        mov   ah,4C          di 0000   a=0
cs:0011►CD21        int   21             bp 0000   i=1
cs:0013 0013        add   [bp+di],dl     sp 0100   d=0
cs:0015 657822      js    gs:003A        ds 48AE
cs:0018 007900      add   [bx+di],bh     es 489D
cs:001B 0000        add   [bx+si],al     ss 48AF
cs:001D 0000        add   [bx+si],al     cs 48AD
cs:001F 0000        add   [bx+si],al     ip 0011

es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
es:0008 AD DE E0 01 C5 15 AA 01 ¡ ▌α☺┼§¬☺      ss:0102 0000
es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺       ss:0100►0000
es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

Add(Direct):

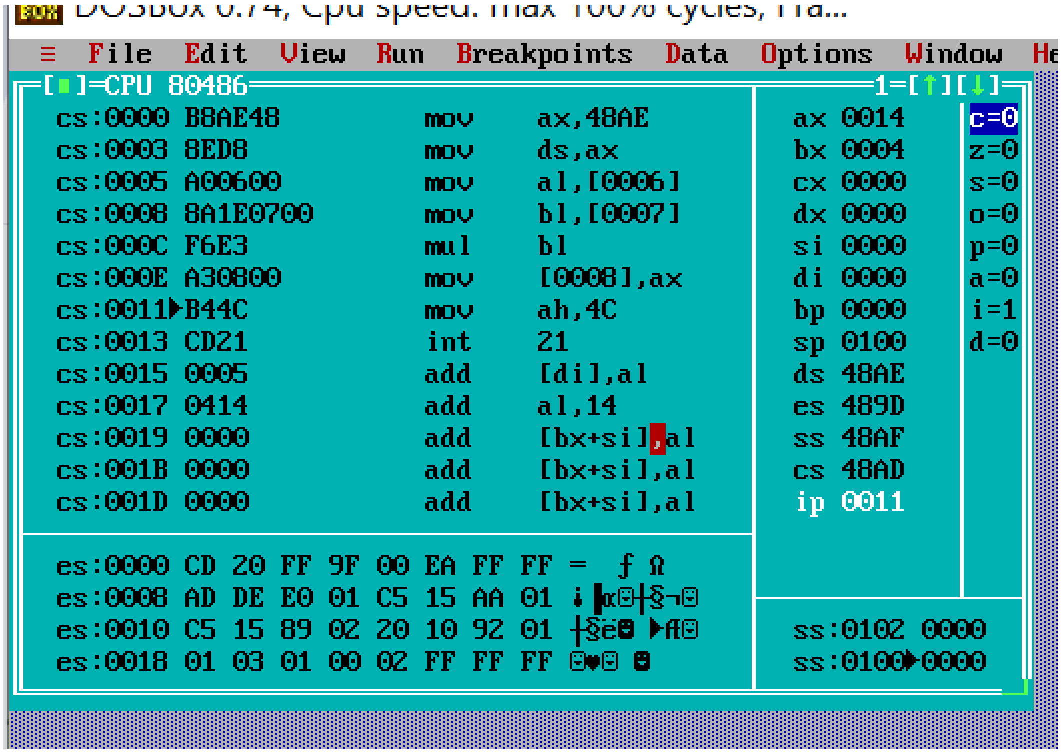
```
.8086
.model small
.stack 100h
.data
num1 db 25h
num2 db 14h
result db ?
.code
main:
    mov ax,@data
    mov ds,ax
    mov al,num1
    add al,num2
    mov result,al
    mov ah,4ch
    int 21h
end main
```



```
[■]=CPU 80486=======================================1=[↑][↓]=
  cs:0000 B8AE48         mov    ax,48AE          ax 4839   c=0
  cs:0003 8ED8           mov    ds,ax            bx 0000   z=0
  cs:0005 A00400         mov    al,[0004]        cx 0000   s=0
  cs:0008 02060500       add    al,[0005]        dx 0000   o=0
  cs:000C A20600         mov    [0006],al        si 0000   p=1
  cs:000F►B44C           mov    ah,4C            di 0000   a=0
  cs:0011 CD21           int    21               bp 0000   i=1
  cs:0013 0025           add    [di],ah          sp 0100   d=0
  cs:0015 1439           adc    al,39            ds 48AE
  cs:0017 0000           add    [bx+si],al       es 489D
  cs:0019 0000           add    [bx+si],al       ss 48AF
  cs:001B 0000           add    [bx+si],al       cs 48AD
  cs:001D 0000           add    [bx+si],al       ip 000F

  es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
  es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺        ss:0102 0000
  es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺        ss:0100►0000
  es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

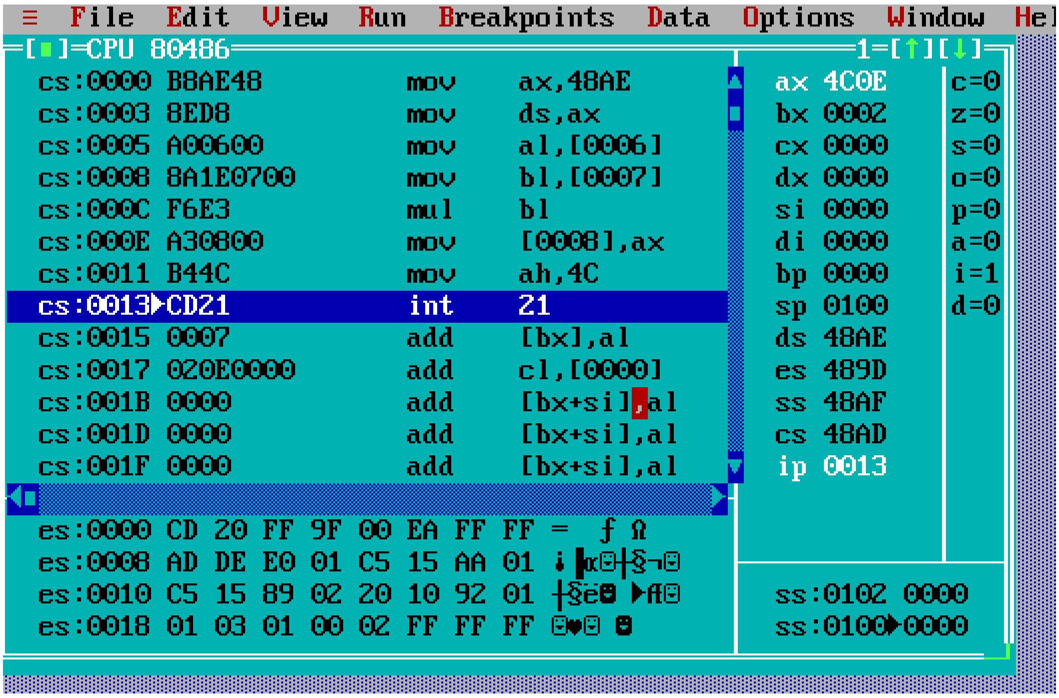
Multiplication(8bit):

```
.8086
.model small
.stack 100h
.data
num1 db 5
num2 db 4
result dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,num1
   mov bl,num2
   mul bl
   mov result,ax
   mov ah,4ch
   int 21h
end main
```


DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
≡ File Edit View Run Breakpoints Data Options Window He

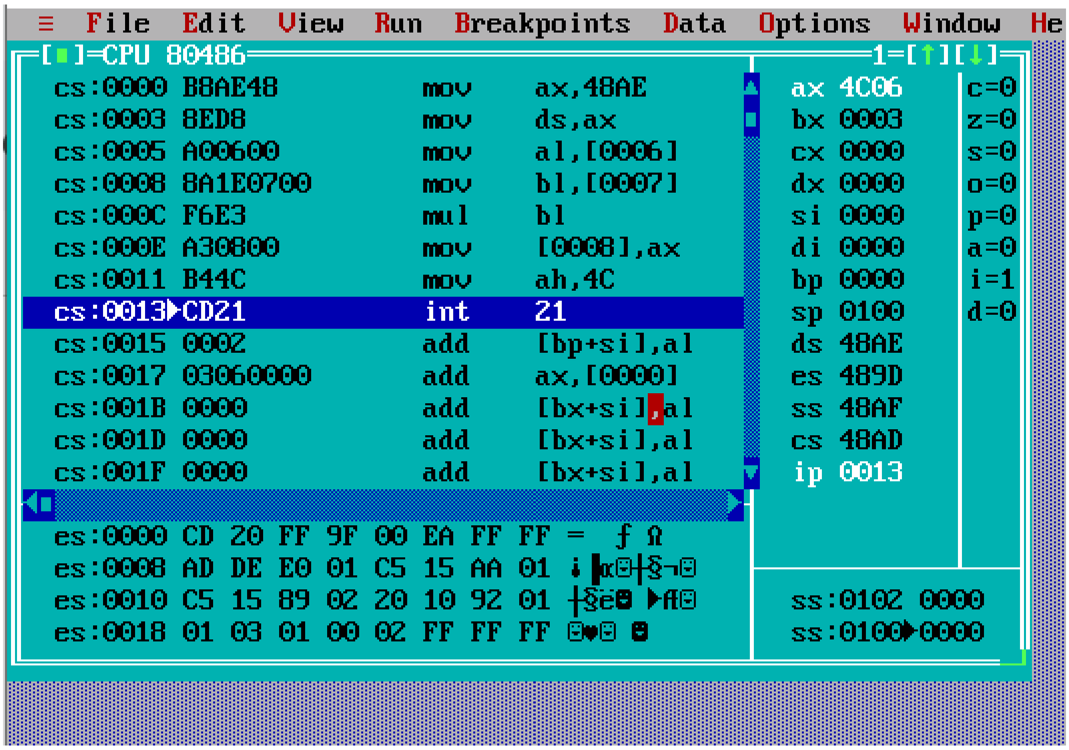
```
[■]=CPU 80486=========================================1=[↑][↓]=
 cs:0000 B8AE48        mov    ax,48AE        ax 0014   c=0
 cs:0003 8ED8          mov    ds,ax          bx 0004   z=0
 cs:0005 A00600        mov    al,[0006]      cx 0000   s=0
 cs:0008 8A1E0700      mov    bl,[0007]      dx 0000   o=0
 cs:000C F6E3          mul    bl             si 0000   p=0
 cs:000E A30800        mov    [0008],ax      di 0000   a=0
 cs:0011►B44C          mov    ah,4C          bp 0000   i=1
 cs:0013 CD21          int    21             sp 0100   d=0
 cs:0015 0005          add    [di],al        ds 48AE
 cs:0017 0414          add    al,14          es 489D
 cs:0019 0000          add    [bx+si],al     ss 48AF
 cs:001B 0000          add    [bx+si],al     cs 48AD
 cs:001D 0000          add    [bx+si],al     ip 0011

 es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺     ss:0102 0000
 es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     ss:0100►0000
 es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

```
.8086
.model small
.stack 100h
.data
num1 db 7
num2 db 2
result dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,num1
   mov bl,num2
   mul bl
   mov result,ax
   mov ah,4ch
   int 21h
end main
```

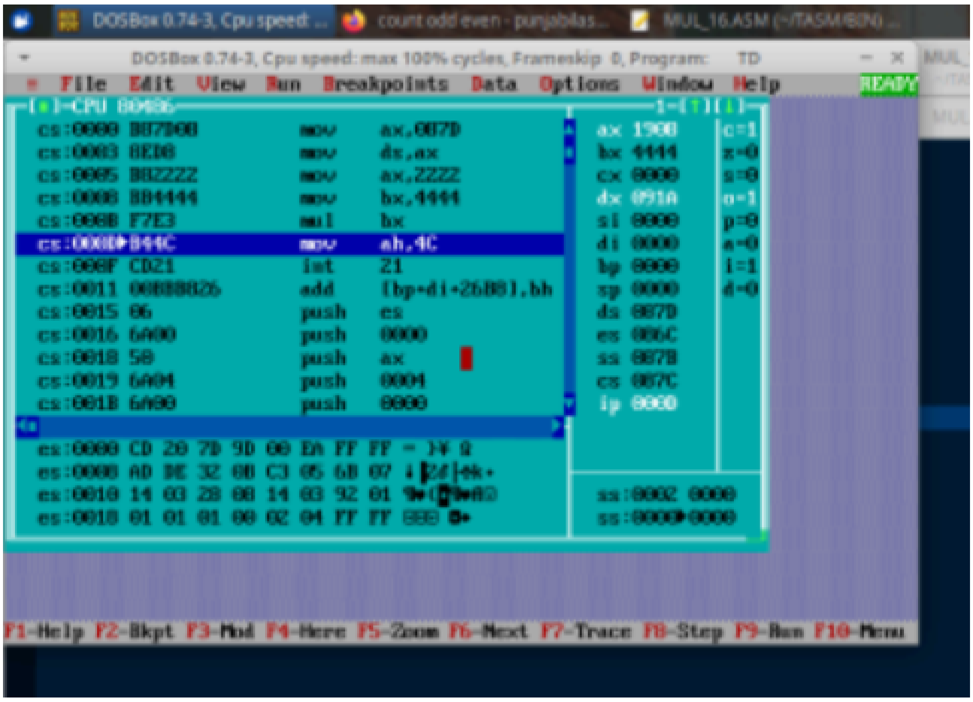


```
cs:0000 B8AE48         mov    ax,48AE      ax 4C0E   c=0
cs:0003 8ED8           mov    ds,ax        bx 0002   z=0
cs:0005 A00600         mov    al,[0006]    cx 0000   s=0
cs:0008 8A1E0700       mov    bl,[0007]    dx 0000   o=0
cs:000C F6E3           mul    bl           si 0000   p=0
cs:000E A30800         mov    [0008],ax    di 0000   a=0
cs:0011 B44C           mov    ah,4C        bp 0000   i=1
cs:0013▶CD21           int    21           sp 0100   d=0
cs:0015 0007           add    [bx],al      ds 48AE
cs:0017 020E0000       add    cl,[0000]    es 489D
cs:001B 0000           add    [bx+si],al   ss 48AF
cs:001D 0000           add    [bx+si],al   cs 48AD
cs:001F 0000           add    [bx+si],al   ip 0013

es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ▶Æ☺     ss:0102 0000
es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻        ss:0100▶0000
```

```
.8086
.model small
.stack 100h
.data
num1 db 2
num2 db 3
result dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov al,num1
   mov bl,num2
   mul bl
   mov result,ax
   mov ah,4ch
   int 21h
end main
```



```
cs:0000 B8AE48        mov     ax,48AE          ax 4C06   c=0
cs:0003 8ED8          mov     ds,ax            bx 0003   z=0
cs:0005 A00600        mov     al,[0006]        cx 0000   s=0
cs:0008 8A1E0700      mov     bl,[0007]        dx 0000   o=0
cs:000C F6E3          mul     bl               si 0000   p=0
cs:000E A30800        mov     [0008],ax        di 0000   a=0
cs:0011 B44C          mov     ah,4C            bp 0000   i=1
cs:0013►CD21          int     21               sp 0100   d=0
cs:0015 0002          add     [bp+si],al       ds 48AE
cs:0017 03060000      add     ax,[0000]        es 489D
cs:001B 0000          add     [bx+si],al       ss 48AF
cs:001D 0000          add     [bx+si],al       cs 48AD
cs:001F 0000          add     [bx+si],al       ip 0013

es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺       ss:0102 0000
es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺       ss:0100►0000
es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

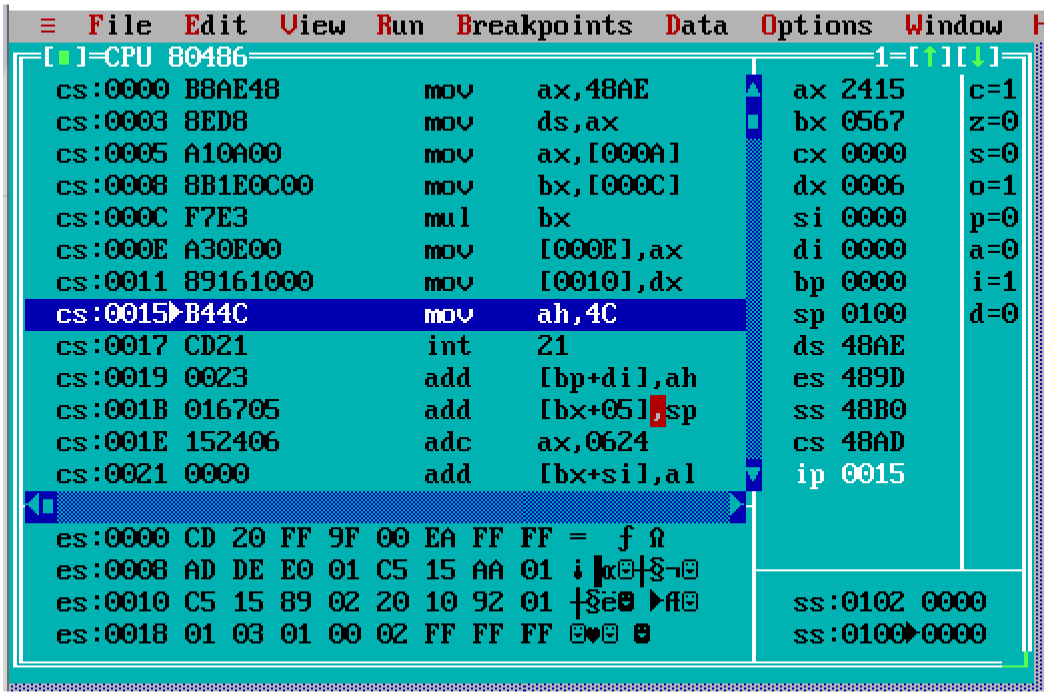

DOSBox 0.74-3, Cpu speed ...
count odd even - punjablas...
MUL_16.ASM (~/TASM/BIN) ...
DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
READY
[■]=CPU 80486
1=[↑][↓]
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```
cs:0000 B87D08        mov     ax,087D          ax 1908   c=1
cs:0003 8ED8          mov     ds,ax            bx 4444   z=0
cs:0005 B82222        mov     ax,2222          cx 0000   s=0
cs:0008 BB4444        mov     bx,4444          dx 091A   o=1
cs:000B F7E3          mul     bx               si 0000   p=0
cs:000D►B44C          mov     ah,4C            di 0000   a=0
cs:000F CD21          int     21               bp 0000   i=1
cs:0011 00BB8826      add     [bp+di+26B8],bh  sp 0000   d=0
cs:0015 06            push    es               ds 087D
cs:0016 6A00          push    0000             es 086C
cs:0018 50            push    ax               ss 087B
cs:0019 6A04          push    0004             cs 087C
cs:001B 6A00          push    0000             ip 000D

es:0000 CD 20 7D 9D 00 EA FF FF
es:0008 AD DE 32 0B C3 05 6B 07                ss:0002 0000
es:0010 14 03 28 0B 14 03 92 01                ss:0000►0000
es:0018 01 01 01 00 02 04 FF FF
```

Multiplication(16bit):

```
.model small
.stack 100h
.data
num1 dw 123h
num2 dw 567h
result1 dw ?
```

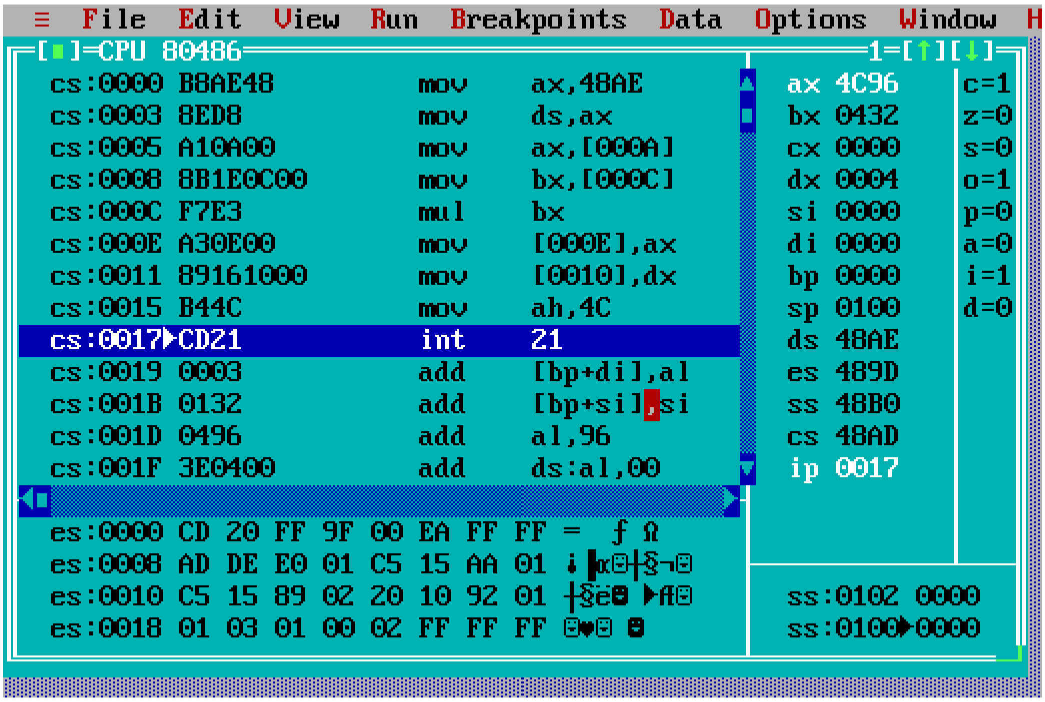
```
result2 dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov ax,num1
   mov bx,num2
   mul bx
   mov result1,ax
   mov result2,dx
   mov ah,4ch
   int 21h
end main
```



```
[■]=CPU 80486=========================================1=[↑][↓]=
 cs:0000 B8AE48          mov    ax,48AE         ax 2415   c=1
 cs:0003 8ED8            mov    ds,ax           bx 0567   z=0
 cs:0005 A10A00          mov    ax,[000A]       cx 0000   s=0
 cs:0008 8B1E0C00        mov    bx,[000C]       dx 0006   o=1
 cs:000C F7E3            mul    bx              si 0000   p=0
 cs:000E A30E00          mov    [000E],ax       di 0000   a=0
 cs:0011 89161000        mov    [0010],dx       bp 0000   i=1
 cs:0015►B44C            mov    ah,4C           sp 0100   d=0
 cs:0017 CD21            int    21              ds 48AE
 cs:0019 0023            add    [bp+di],ah      es 489D
 cs:001B 016705          add    [bx+05],sp      ss 48B0
 cs:001E 152406          adc    ax,0624         cs 48AD
 cs:0021 0000            add    [bx+si],al      ip 0015

 es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺        ss:0102 0000
 es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺        ss:0100►0000
 es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

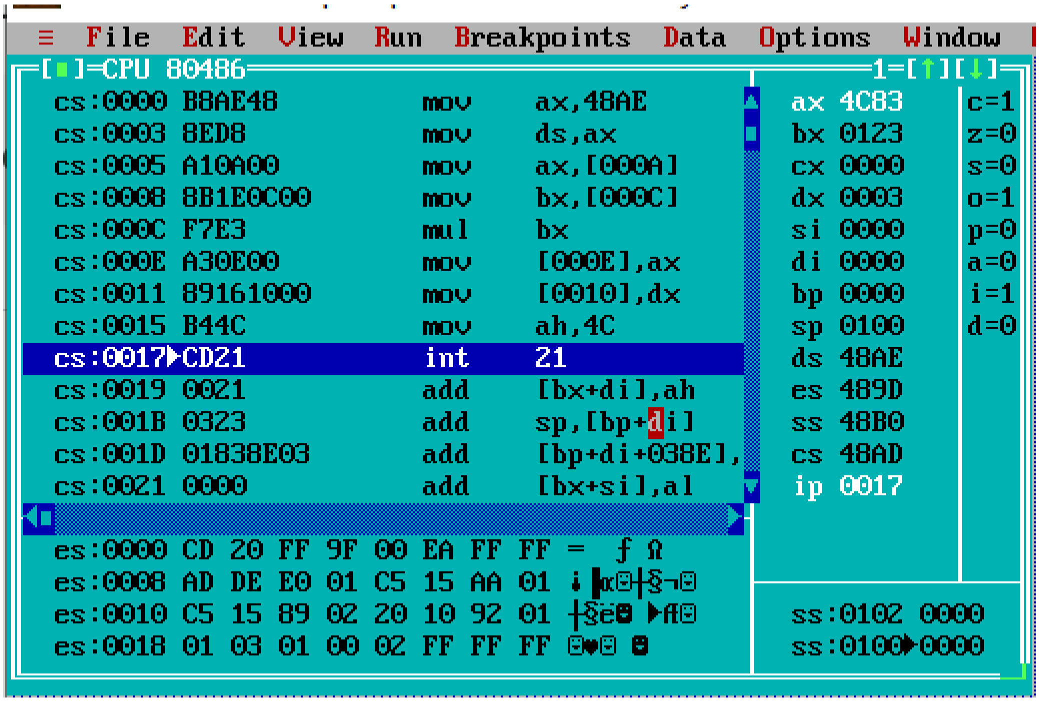
```
.model small
.stack 100h
.data
num1 dw 103h
num2 dw 432h
result1 dw ?
result2 dw ?
.code
main:
   mov ax,@data
   mov ds,ax
   mov ax,num1
   mov bx,num2
   mul bx
   mov result1,ax
```

```
    mov result2,dx
    mov ah,4ch
    int 21h
end main
```

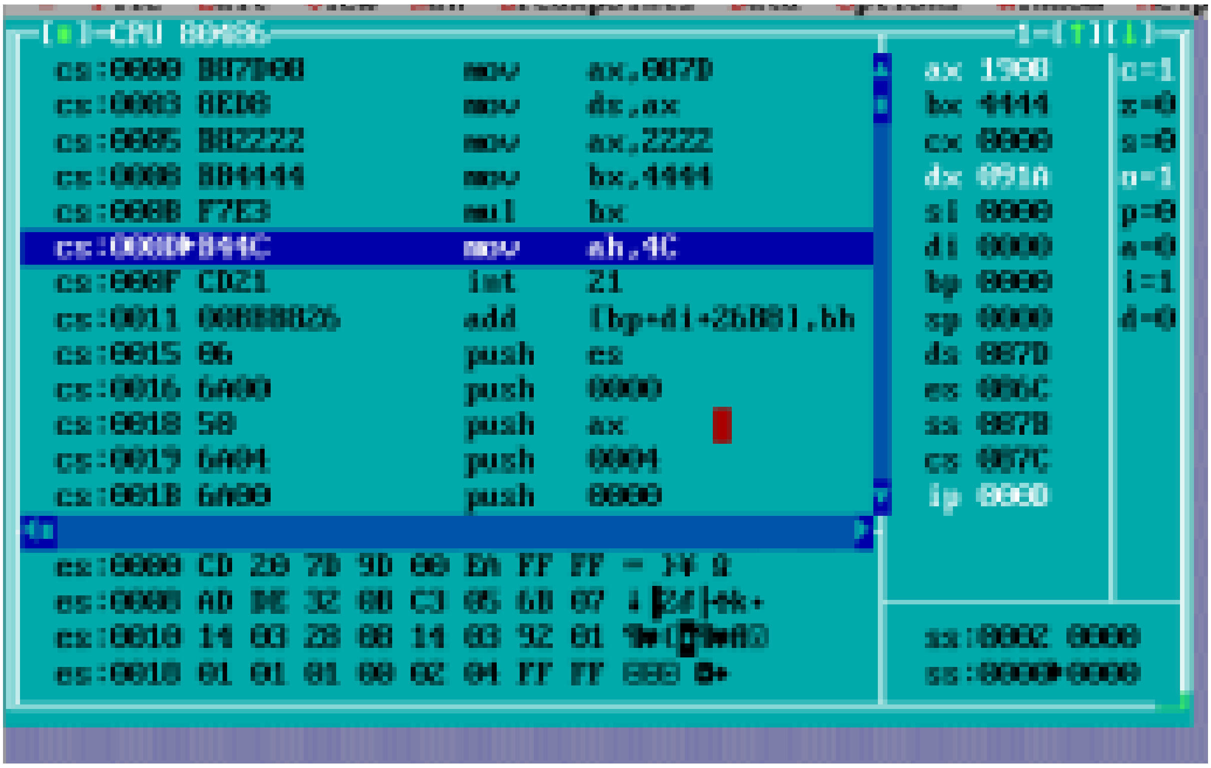


```
  cs:0000 B8AE48         mov    ax,48AE      ax 4C96   c=1
  cs:0003 8ED8           mov    ds,ax        bx 0432   z=0
  cs:0005 A10A00         mov    ax,[000A]    cx 0000   s=0
  cs:0008 8B1E0C00       mov    bx,[000C]    dx 0004   o=1
  cs:000C F7E3           mul    bx           si 0000   p=0
  cs:000E A30E00         mov    [000E],ax    di 0000   a=0
  cs:0011 89161000       mov    [0010],dx    bp 0000   i=1
  cs:0015 B44C           mov    ah,4C        sp 0100   d=0
  cs:0017►CD21           int    21           ds 48AE
  cs:0019 0003           add    [bp+di],al   es 489D
  cs:001B 0132           add    [bp+si],si   ss 48B0
  cs:001D 0496           add    al,96        cs 48AD
  cs:001F 3E0400         add    ds:al,00     ip 0017

  es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
  es:0008 AD DE E0 01 C5 15 AA 01 ¡ ▌α☺┼§¬☺    ss:0102 0000
  es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺    ss:0100►0000
  es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻
```

```
.model small
.stack 100h
.data
num1 dw 321h
num2 dw 123h
result1 dw ?
result2 dw ?
.code
main:
    mov ax,@data
    mov ds,ax
    mov ax,num1
    mov bx,num2
    mul bx
    mov result1,ax
    mov result2,dx
    mov ah,4ch
    int 21h
end main
```



```
[■]=CPU 80486                                          1=[↑][↓]
 cs:0000 B8AE48      mov    ax,48AE          ax 4C83   c=1
 cs:0003 8ED8        mov    ds,ax            bx 0123   z=0
 cs:0005 A10A00      mov    ax,[000A]        cx 0000   s=0
 cs:0008 8B1E0C00    mov    bx,[000C]        dx 0003   o=1
 cs:000C F7E3        mul    bx               si 0000   p=0
 cs:000E A30E00      mov    [000E],ax        di 0000   a=0
 cs:0011 89161000    mov    [0010],dx        bp 0000   i=1
 cs:0015 B44C        mov    ah,4C            sp 0100   d=0
 cs:0017►CD21        int    21               ds 48AE
 cs:0019 0021        add    [bx+di],ah       es 489D
 cs:001B 0323        add    sp,[bp+di]       ss 48B0
 cs:001D 01838E03    add    [bp+di+038E],    cs 48AD
 cs:0021 0000        add    [bx+si],al       ip 0017

 es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01           ss:0102 0000
 es:0010 C5 15 89 02 20 10 92 01           ss:0100►0000
 es:0018 01 03 01 00 02 FF FF FF
```

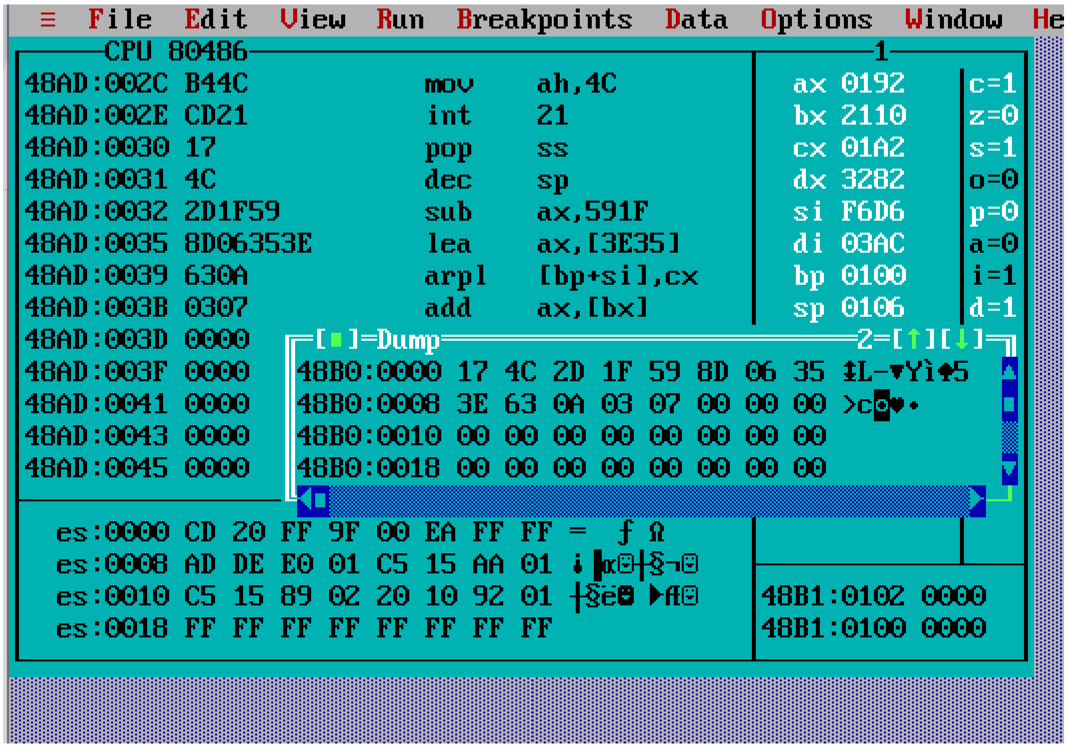


```
[■]=CPU 80486                                          1=[↑][↓]
 cs:0000 B87D08      mov    ax,087D          ax 1908   c=1
 cs:0003 8ED8        mov    ds,ax            bx 4444   z=0
 cs:0005 B82222      mov    ax,2222          cx 0000   s=0
 cs:0008 BB4444      mov    bx,4444          dx 091A   o=1
 cs:000B F7E3        mul    bx               si 0000   p=0
 cs:000D►B44C        mov    ah,4C            di 0000   a=0
 cs:000F CD21        int    21               bp 0000   i=1
 cs:0011 00BBB826    add    [bp+di+26B8],bh  sp 0000   d=0
 cs:0015 06          push   es               ds 087D
 cs:0016 6A00        push   0000             es 086C
 cs:0018 50          push   ax               ss 087B
 cs:0019 6A04        push   0004             cs 087C
 cs:001B 6A00        push   0000             ip 000D

 es:0000 CD 20 7D 9D 00 EA FF FF
 es:0008 AD DE 32 00 C3 05 6B 07           ss:0002 0000
 es:0010 14 03 28 08 14 03 92 01           ss:0000►0000
 es:0018 01 01 01 00 02 04 FF FF
```

EVEN ODD:

```
.8086
.model small
.stack 100h
.data
```

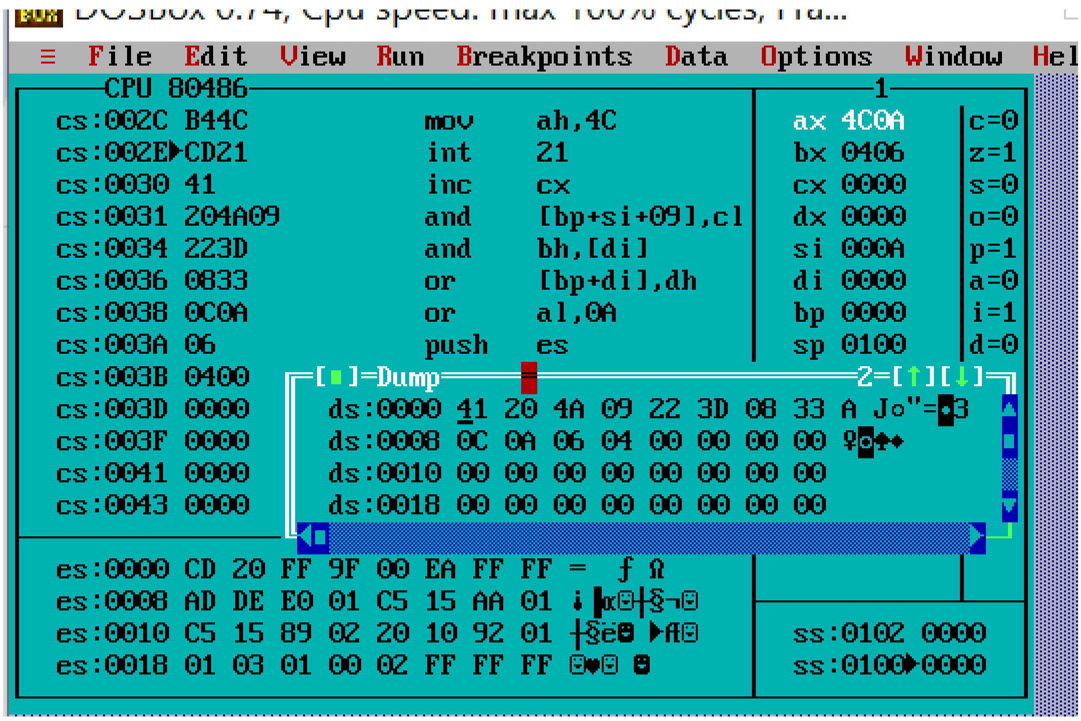
```
arr db 23,76,45,31,89,141,6,53,62,99
arraysize db 10
even_count db 0
odd_count db 0
.code
start:
    mov ax,@data
    mov ds,ax
    mov cl,arraysize
    mov si,0
    mov bl,0
    mov bh,0
check_loop:
    mov al,arr[si]
    test al,01h
    jnz is_odd
    inc bl
    jmp next_num
is_odd:
    inc bh
next_num:
    inc si
    dec cl
    jnz check_loop
    mov even_count,bl
    mov odd_count,bh
    mov ah,4ch
    int 21h
end start
```



```
CPU 80486                                                   1
48AD:002C B44C          mov    ah,4C           ax 0192   c=1
48AD:002E CD21          int    21              bx 2110   z=0
48AD:0030 17            pop    ss              cx 01A2   s=1
48AD:0031 4C            dec    sp              dx 3282   o=0
48AD:0032 2D1F59        sub    ax,591F         si F6D6   p=0
48AD:0035 8D06353E      lea    ax,[3E35]       di 03AC   a=0
48AD:0039 630A          arpl   [bp+si],cx      bp 0100   i=1
48AD:003B 0307          add    ax,[bx]         sp 0106   d=1
48AD:003D 0000    [■]=Dump                           2=[↑][↓]
48AD:003F 0000    48B0:0000 17 4C 2D 1F 59 8D 06 35 ‡L-▼Yì♠5
48AD:0041 0000    48B0:0008 3E 63 0A 03 07 00 00 00 >c◙♥•
48AD:0043 0000    48B0:0010 00 00 00 00 00 00 00 00
48AD:0045 0000    48B0:0018 00 00 00 00 00 00 00 00

  es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
  es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺
  es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     48B1:0102 0000
  es:0018 FF FF FF FF FF FF FF FF              48B1:0100 0000
```

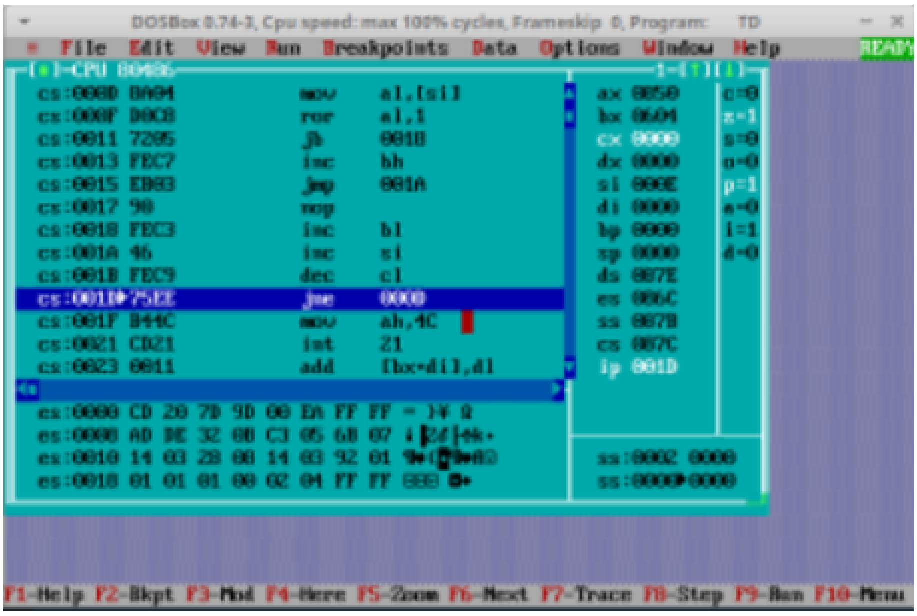
```
.8086
.model small
.stack 100h
.data
arr db 65,32,74,9,34,61,8,51,12
arraysize db 10
even_count db 0
odd_count db 0
.code
start:
   mov ax,@data
   mov ds,ax
   mov cl,arraysize
   mov si,0
   mov bl,0
   mov bh,0
check_loop:
   mov al,arr[si]
   test al,01h
   jnz is_odd
   inc bl
   jmp next_num
is_odd:
   inc bh
next_num:
   inc si
   dec cl
```

```
    jnz check_loop
    mov even_count,bl
    mov odd_count,bh
    mov ah,4ch
    int 21h
end start
```



DOSBox 0.74, Cpu speed: max 100% cycles, Fra...

```
CPU 80486                                                      1
 cs:002C B44C          mov    ah,4C                 ax 4C0A   c=0
 cs:002E▶CD21          int    21                    bx 0406   z=1
 cs:0030 41            inc    cx                    cx 0000   s=0
 cs:0031 204A09        and    [bp+si+09],cl         dx 0000   o=0
 cs:0034 223D          and    bh,[di]               si 000A   p=1
 cs:0036 0833          or     [bp+di],dh            di 0000   a=0
 cs:0038 0C0A          or     al,0A                 bp 0000   i=1
 cs:003A 06            push   es                    sp 0100   d=0
 cs:003B 0400      [■]=Dump                       2=[↑][↓]
 cs:003D 0000        ds:0000 41 20 4A 09 22 3D 08 33 A Jo"=■3
 cs:003F 0000        ds:0008 0C 0A 06 04 00 00 00 00 ♀◙♠♦
 cs:0041 0000        ds:0010 00 00 00 00 00 00 00 00
 cs:0043 0000        ds:0018 00 00 00 00 00 00 00 00

 es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
 es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
 es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺           ss:0102 0000
 es:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻              ss:0100▶0000
```



DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD

≡ File Edit View Run Breakpoints Data Options Window Help READY

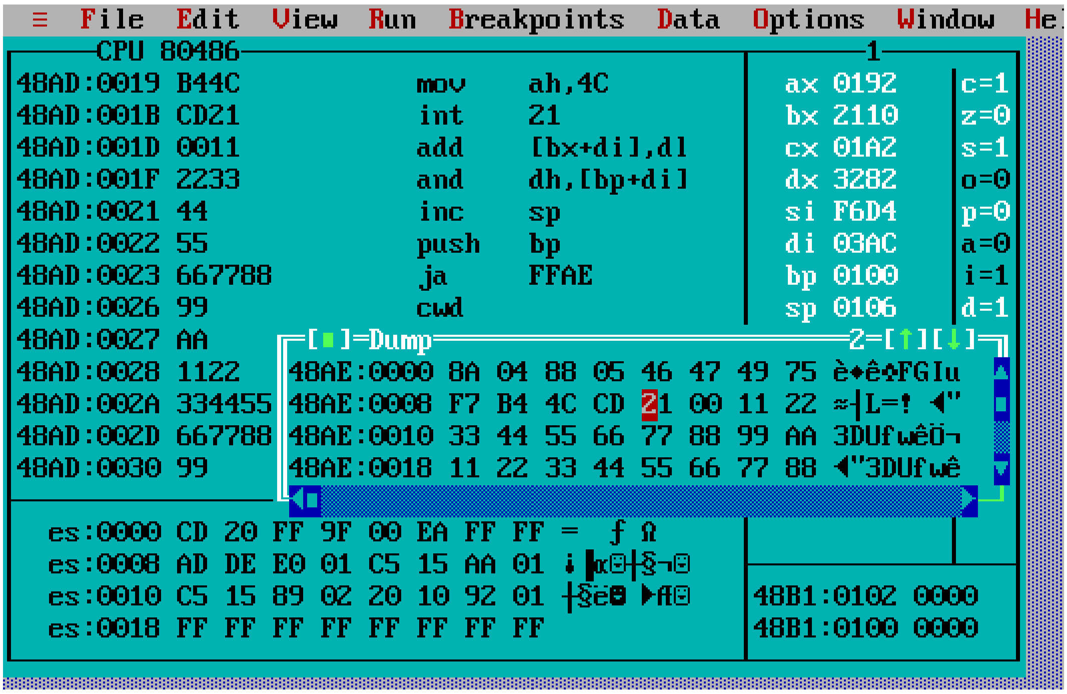
```
[■]=CPU 80486                                    1=[↑][↓]
 cs:000D 8A04          mov    al,[si]            ax 0050   c=0
 cs:000F D0C8          ror    al,1               bx 0604   z=1
 cs:0011 7205          jb     0018               cx 0000   s=0
 cs:0013 FEC7          inc    bh                 dx 0000   o=0
 cs:0015 EB03          jmp    001A               si 000E   p=1
 cs:0017 90            nop                       di 0000   a=0
 cs:0018 FEC3          inc    bl                 bp 0000   i=1
 cs:001A 46            inc    si                 sp 0000   d=0
 cs:001B FEC9          dec    cl                 ds 087E
 cs:001D▶75EE          jne    000D               es 086C
 cs:001F B44C          mov    ah,4C              ss 087B
 cs:0021 CD21          int    21                 cs 087C
 cs:0023 0011          add    [bx+di],dl         ip 001D

 es:0000 CD 20 7D 9D 00 EA FF FF = }¥ Ω
 es:0008 AD DE 32 0B C3 05 6B 07 ¡▌2♂├♣k•
 es:0010 14 03 2B 0B 14 03 92 01 ¶♥+♂¶♥Æ☺        ss:0002 0000
 es:0018 01 01 01 00 02 04 FF FF ☺☺☺ ☻♦          ss:0000▶0000
```

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

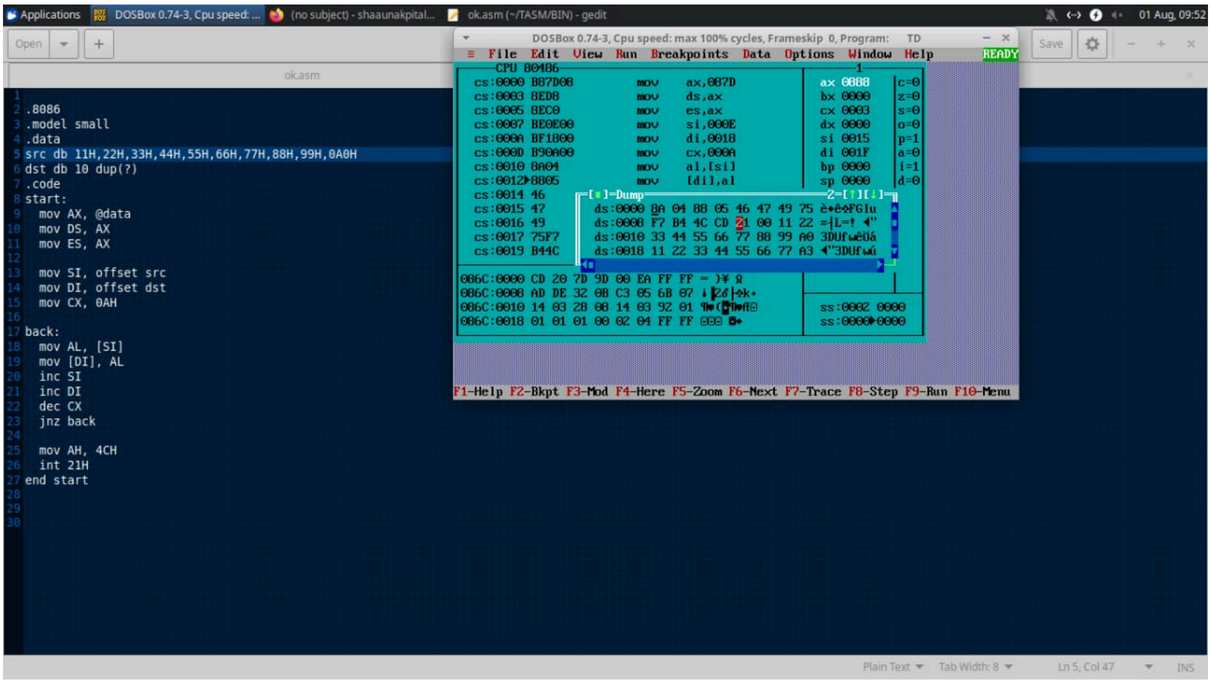
BLOCK TRANSFER:

```
.model small
```

```
.stack 100h
.data
src db 11h,22h,33h,44h,55h,66h,77h,88h,99h,0AAh
dst db 10 dup(?)
.code
start:
    mov ax, @data
    mov ds, ax
    mov es, ax

    mov si, offset src
    mov di, offset dst
    mov cx, 0Ah

back:
    mov al, [si]
    mov [di], al
    inc si
    inc di
    dec cx
    jnz back

    mov ah, 4Ch
    int 21h
end start
```



```
CPU 80486                                                      1
48AD:0019 B44C            mov    ah,4C                ax 0192   c=1
48AD:001B CD21            int    21                   bx 2110   z=0
48AD:001D 0011            add    [bx+di],dl           cx 01A2   s=1
48AD:001F 2233            and    dh,[bp+di]           dx 3282   o=0
48AD:0021 44              inc    sp                   si F6D4   p=0
48AD:0022 55              push   bp                   di 03AC   a=0
48AD:0023 667788          ja     FFAE                 bp 0100   i=1
48AD:0026 99              cwd                         sp 0106   d=1
48AD:0027 AA       [■]=Dump==========================2=[↑][↓]
48AD:0028 1122     48AE:0000 8A 04 88 05 46 47 49 75 è♦ê♣FGIu
48AD:002A 334455   48AE:0008 F7 B4 4C CD 21 00 11 22 ≈┤L=! ◄"
48AD:002D 667788   48AE:0010 33 44 55 66 77 88 99 AA 3DUfwêÖ¬
48AD:0030 99       48AE:0018 11 22 33 44 55 66 77 88 ◄"3DUfwê

  es:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
  es:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺
  es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺    48B1:0102 0000
  es:0018 FF FF FF FF FF FF FF FF             48B1:0100 0000
```

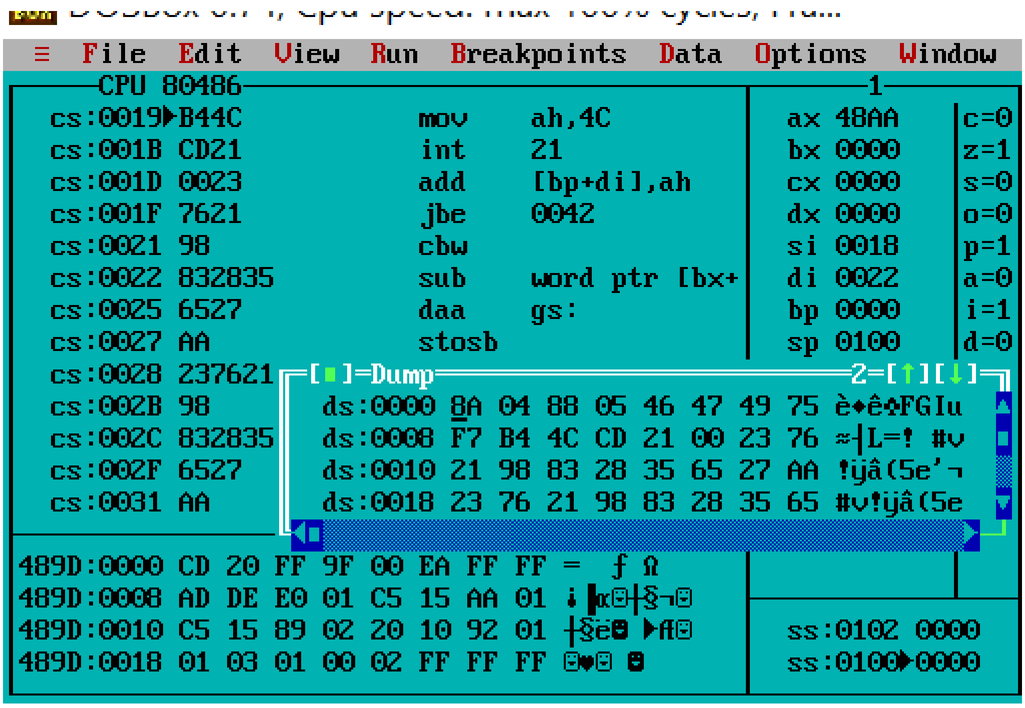

Applications
DOSBox 0.74-3, Cpu speed: ...
(no subject) - shaaunakpital...
ok.asm (~/TASM/BIN) - gedit
01 Aug, 09:52
Open
+
ok.asm
Save
DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
≡ File Edit View Run Breakpoints Data Options Window Help
READY
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
Plain Text
Tab Width: 8
Ln 5, Col 47
INS

```

.8086
.model small
.data
src db 11H,22H,33H,44H,55H,66H,77H,88H,99H,0A0H
dst db 10 dup(?)
.code
start:
  mov AX, @data
  mov DS, AX
  mov ES, AX

  mov SI, offset src
  mov DI, offset dst
  mov CX, 0AH

back:
  mov AL, [SI]
  mov [DI], AL
  inc SI
  inc DI
  dec CX
  jnz back

  mov AH, 4CH
  int 21H
end start
```

```
CPU 80486                                                 1
 cs:0000 B87D08        mov    ax,087D           ax 0888   c=0
 cs:0003 8ED8          mov    ds,ax             bx 0000   z=0
 cs:0005 8EC0          mov    es,ax             cx 0003   s=0
 cs:0007 BE0E00        mov    si,000E           dx 0000   o=0
 cs:000A BF1800        mov    di,0018           si 0015   p=1
 cs:000D B90A00        mov    cx,000A           di 001F   a=0
 cs:0010 8A04          mov    al,[si]           bp 0000   i=1
 cs:0012▶8805          mov    [di],al           sp 0000   d=0
 cs:0014 46     [■]=Dump======================2=[↑][↓]
 cs:0015 47       ds:0000 8A 04 88 05 46 47 49 75 è♦ê♣FGIu
 cs:0016 49       ds:0008 F7 B4 4C CD 21 00 11 22 ≈┤L=! ◄"
 cs:0017 75F7     ds:0010 33 44 55 66 77 88 99 A0 3DUfwêÖá
 cs:0019 B44C     ds:0018 11 22 33 44 55 66 77 A3 ◄"3DUfwú

086C:0000 CD 20 7D 9D 00 EA FF FF = }¥ Ω
086C:0008 AD DE 32 0B C3 05 6B 07 ¡▐2δ├♣k•
086C:0010 14 03 28 08 14 03 92 01 ¶♥(◘¶♥Æ☺      ss:0002 0000
086C:0018 01 01 01 00 02 04 FF FF ☺☺☺ ☻♦        ss:0000▶0000
```

.model small
.stack 100h
.data
src db 23H,76H,21H,98H,83H,28H,35H,65H,27H,0AAh
dst db 10 dup(?)
.code
start:
    mov ax, @data
    mov ds, ax
    mov es, ax

```
   mov si, offset src
   mov di, offset dst
   mov cx, 0Ah

back:
   mov al, [si]
   mov [di], al
   inc si
   inc di
   dec cx
   jnz back

   mov ah, 4Ch
   int 21h
end start
```

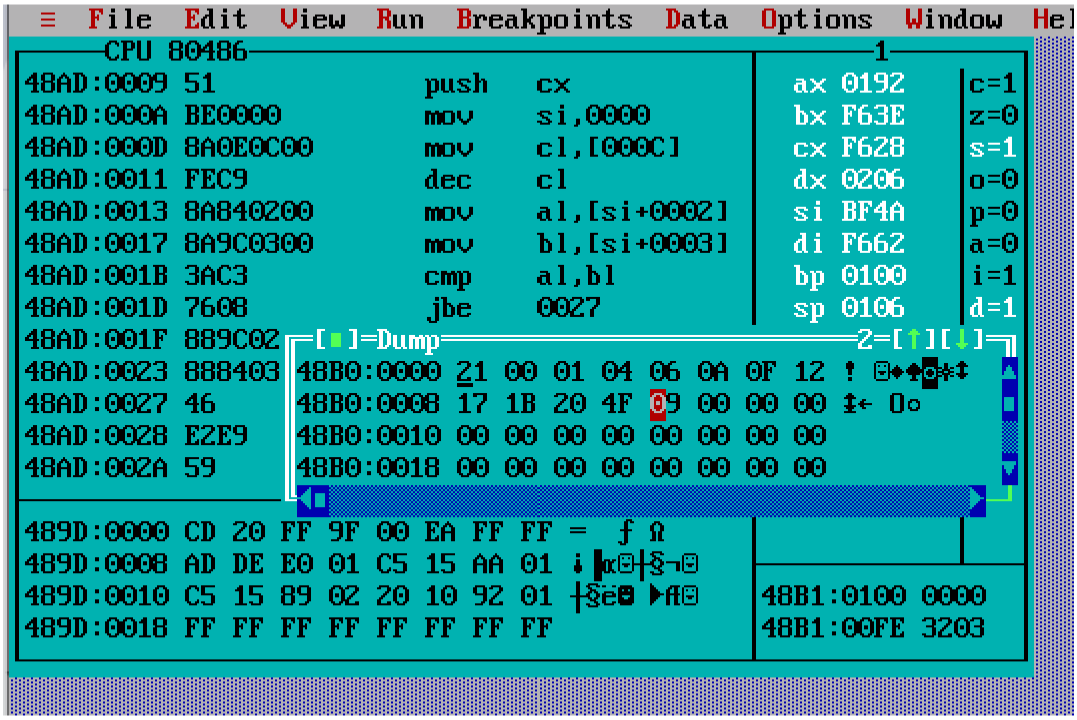


```
CPU 80486                                                    1
  cs:0019▶B44C          mov     ah,4C              ax 48AA   c=0
  cs:001B CD21          int     21                 bx 0000   z=1
  cs:001D 0023          add     [bp+di],ah         cx 0000   s=0
  cs:001F 7621          jbe     0042               dx 0000   o=0
  cs:0021 98            cbw                        si 0018   p=1
  cs:0022 832835        sub     word ptr [bx+      di 0022   a=0
  cs:0025 6527          daa     gs:                bp 0000   i=1
  cs:0027 AA            stosb                      sp 0100   d=0
  cs:0028 237621 [■]=Dump                                   2=[↑][↓]
  cs:002B 98       ds:0000 8A 04 88 05 46 47 49 75 è♦ê♠FGIu
  cs:002C 832835   ds:0008 F7 B4 4C CD 21 00 23 76 ≈┤L=! #v
  cs:002F 6527     ds:0010 21 98 83 28 35 65 27 AA !ÿâ(5e'¬
  cs:0031 AA       ds:0018 23 76 21 98 83 28 35 65 #v!ÿâ(5e

489D:0000 CD 20 FF 9F 00 EA FF FF = ƒ Ω
489D:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺
489D:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺         ss:0102 0000
489D:0018 01 03 01 00 02 FF FF FF ☺♥☺ ☻           ss:0100▶0000
```

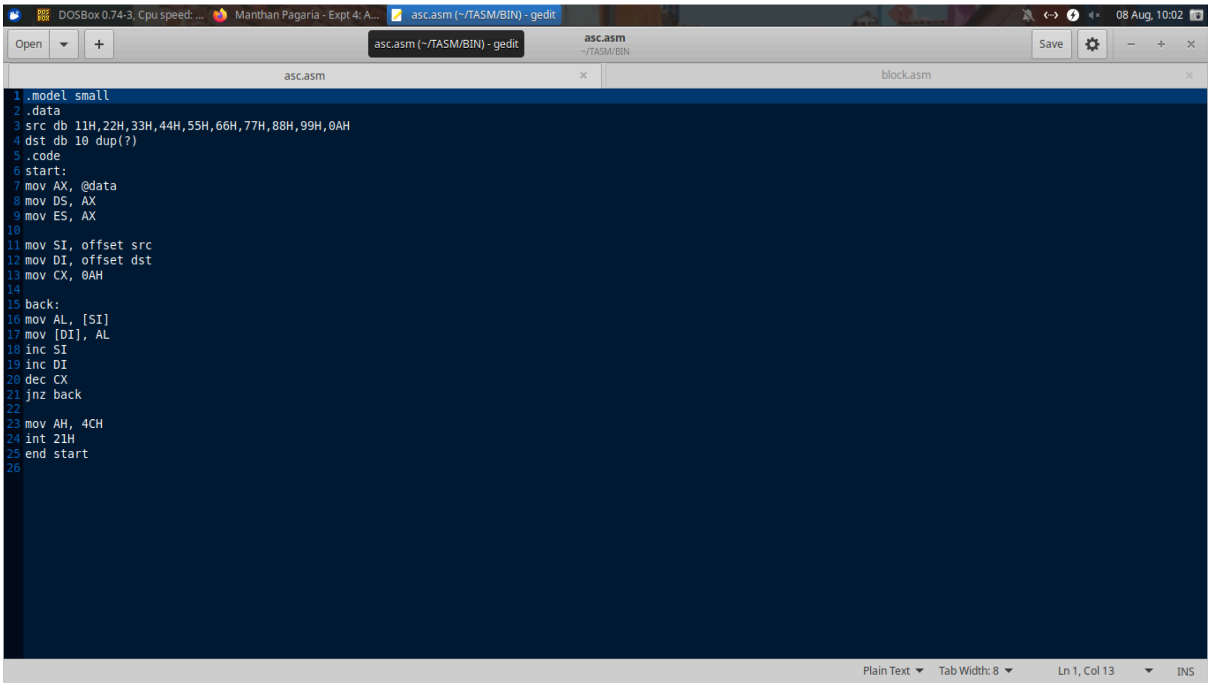
ASCENDING:

```
.8086
.model small
.stack
.data
arr db 22H,11H,44H,33H,66H,55H,88H,77H,99H,0AH

.code
start:  mov AX, @data
       mov DS, AX
       mov CX, 09H
```

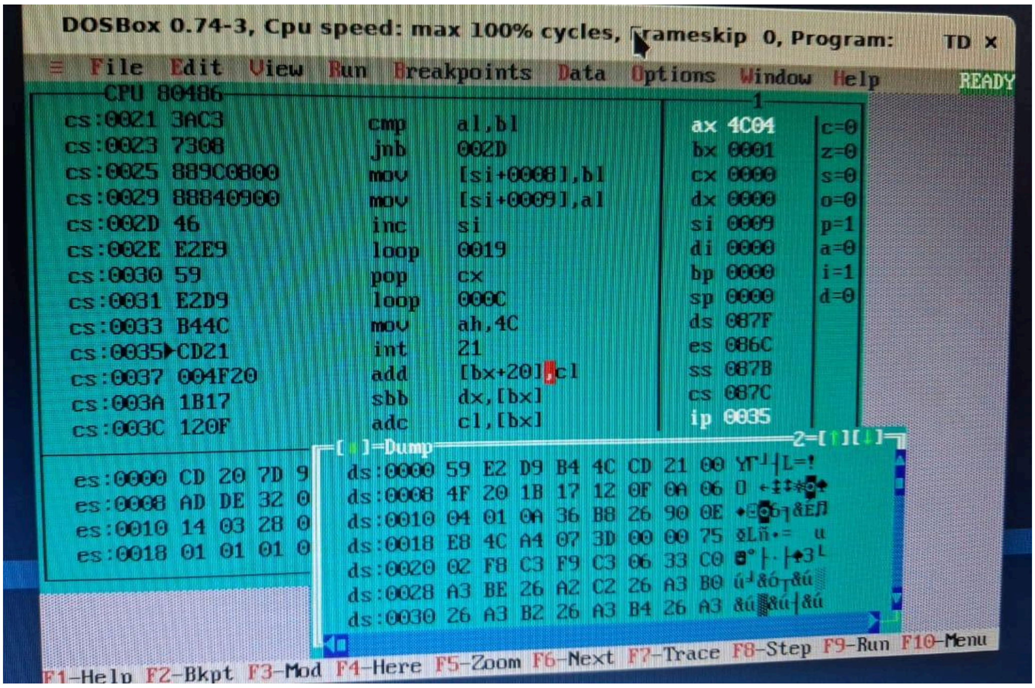
```
outer:  mov SI, 00H
        mov DX, CX

inner:  mov AL, arr[SI]
        mov BL, arr[SI+1]
        cmp AL, BL
        jbe skip
        mov arr[SI], BL
        mov arr[SI+1], AL
skip:   inc SI
        dec DX
        jnz inner
        dec CX
        jnz outer

        mov AH, 4CH
        int 21H
end start
```



```
48AD:0009 51            push   cx                  ax 0192   c=1
48AD:000A BE0000        mov    si,0000             bx F63E   z=0
48AD:000D 8A0E0C00      mov    cl,[000C]           cx F628   s=1
48AD:0011 FEC9          dec    cl                  dx 0206   o=0
48AD:0013 8A840200      mov    al,[si+0002]        si BF4A   p=0
48AD:0017 8A9C0300      mov    bl,[si+0003]        di F662   a=0
48AD:001B 3AC3          cmp    al,bl               bp 0100   i=1
48AD:001D 7608          jbe    0027                sp 0106   d=1
48AD:001F 889C02
48AD:0023 888403  48B0:0000 21 00 01 04 06 0A 0F 12 ! ☺♦♠◙☼↕
48AD:0027 46      48B0:0008 17 1B 20 4F 09 00 00 00 ↨← O○
48AD:0028 E2E9    48B0:0010 00 00 00 00 00 00 00 00
48AD:002A 59      48B0:0018 00 00 00 00 00 00 00 00

489D:0000 CD 20 FF 9F 00 EA FF FF ═  ƒ Ω
489D:0008 AD DE E0 01 C5 15 AA 01 ¡▐α☺┼§¬☺
489D:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺   48B1:0100 0000
489D:0018 FF FF FF FF FF FF FF FF            48B1:00FE 3203
```

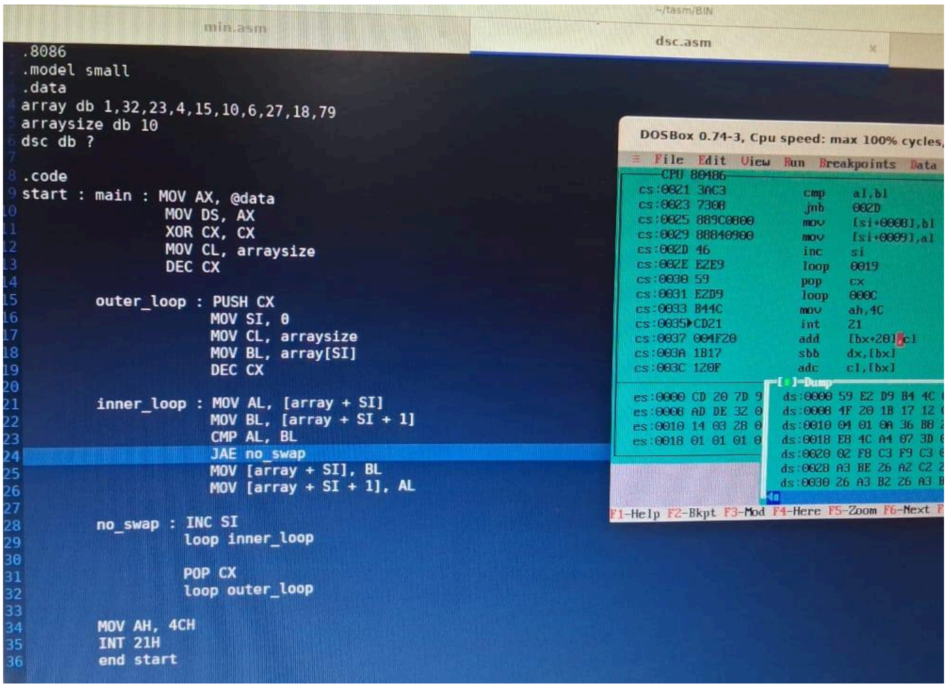


```
.model small
.data
src db 11H,22H,33H,44H,55H,66H,77H,88H,99H,0AH
dst db 10 dup(?)
.code
start:
mov AX, @data
mov DS, AX
mov ES, AX

mov SI, offset src
mov DI, offset dst
mov CX, 0AH

back:
mov AL, [SI]
mov [DI], AL
inc SI
inc DI
dec CX
jnz back

mov AH, 4CH
int 21H
end start
```


DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
≡ File Edit View Run Breakpoints Data Options Window Help READY
CPU 80486
1
[ ]=Dump
2=[↑][↓]
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```
cs:0021 3AC3         cmp   al,bl             ax 4C04   c=0
cs:0023 7308         jnb   002D              bx 0001   z=0
cs:0025 889C0800     mov   [si+0008],bl      cx 0000   s=0
cs:0029 88840900     mov   [si+0009],al      dx 0000   o=0
cs:002D 46           inc   si                si 0009   p=1
cs:002E E2E9         loop  0019              di 0000   a=0
cs:0030 59           pop   cx                bp 0000   i=1
cs:0031 E2D9         loop  000C              sp 0000   d=0
cs:0033 B44C         mov   ah,4C             ds 087F
cs:0035▶CD21         int   21                es 086C
cs:0037 004F20       add   [bx+20],cl        ss 087B
cs:003A 1B17         sbb   dx,[bx]           cs 087C
cs:003C 120F         adc   cl,[bx]           ip 0035

es:0000 CD 20 7D 9
es:0008 AD DE 32 0
es:0010 14 03 28 0
es:0018 01 01 01 0

ds:0000 59 E2 D9 B4 4C CD 21 00
ds:0008 4F 20 1B 17 12 0F 0A 06
ds:0010 04 01 0A 36 B8 26 90 0E
ds:0018 E8 4C A4 07 3D 00 00 75
ds:0020 02 F8 C3 F9 C3 06 33 C0
ds:0028 A3 BE 26 A2 C2 26 A3 B0
ds:0030 26 A3 B2 26 A3 B4 26 A3
```

DESCENDING:


min.asm
~/tasm/BIN
dsc.asm
DOSBox 0.74-3, Cpu speed: max 100% cycles,
File Edit View Run Breakpoints Data
CPU 80486
Dump
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next

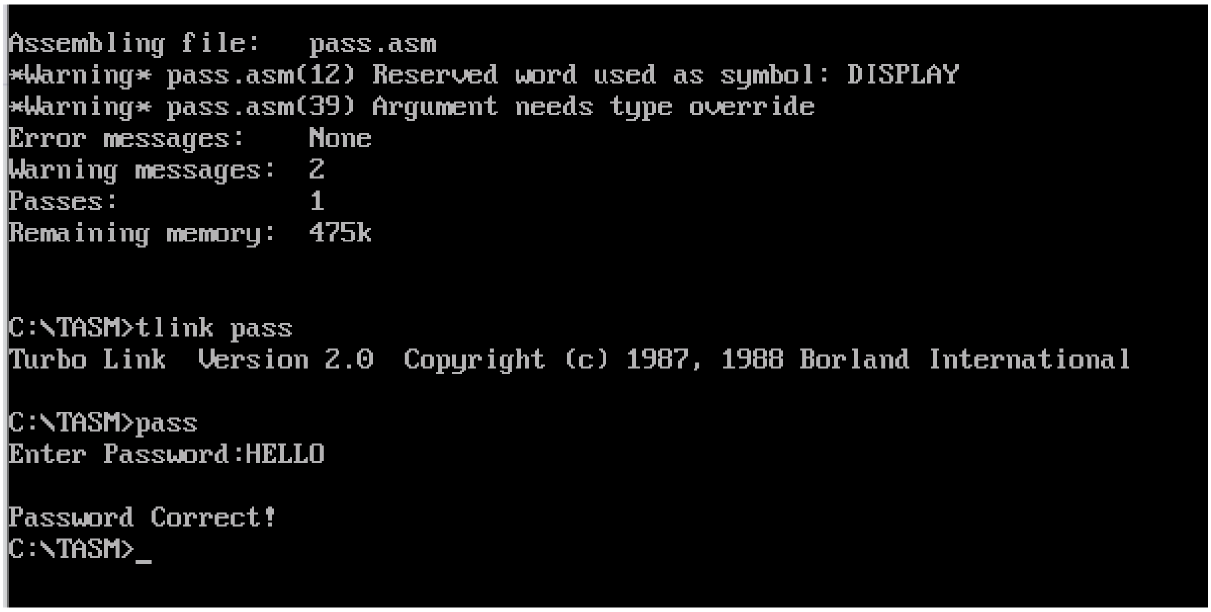
```
.8086
.model small
.data
array db 1,32,23,4,15,10,6,27,18,79
arraysize db 10
dsc db ?

.code
start : main : MOV AX, @data
               MOV DS, AX
               XOR CX, CX
               MOV CL, arraysize
               DEC CX

        outer_loop : PUSH CX
                     MOV SI, 0
                     MOV CL, arraysize
                     MOV BL, array[SI]
                     DEC CX

        inner_loop : MOV AL, [array + SI]
                     MOV BL, [array + SI + 1]
                     CMP AL, BL
                     JAE no_swap
                     MOV [array + SI], BL
                     MOV [array + SI + 1], AL

        no_swap : INC SI
                  loop inner_loop

                  POP CX
                  loop outer_loop

        MOV AH, 4CH
        INT 21H
        end start
```

```
cs:0021 3AC3          cmp    al,bl
cs:0023 7308          jnb    002D
cs:0025 889C0800      mov    [si+0008],bl
cs:0029 88840900      mov    [si+0009],al
cs:002D 46            inc    si
cs:002E E2E9          loop   0019
cs:0030 59            pop    cx
cs:0031 E2D9          loop   000C
cs:0033 B44C          mov    ah,4C
cs:0035▸CD21          int    21
cs:0037 004F20        add    [bx+20],cl
cs:003A 1B17          sbb    dx,[bx]
cs:003C 120F          adc    cl,[bx]

es:0000 CD 20 7D 9     ds:0000 59 E2 D9 B4 4C
es:0008 AD DE 32 0     ds:0008 4F 20 1B 17 12
es:0010 14 03 28 0     ds:0010 04 01 0A 36 B8
es:0018 01 01 01 0     ds:0018 E8 4C A4 07 3D
                       ds:0020 02 F8 C3 F9 C3
                       ds:0028 A3 BE 26 A2 C2
                       ds:0030 26 A3 B2 26 A3
```

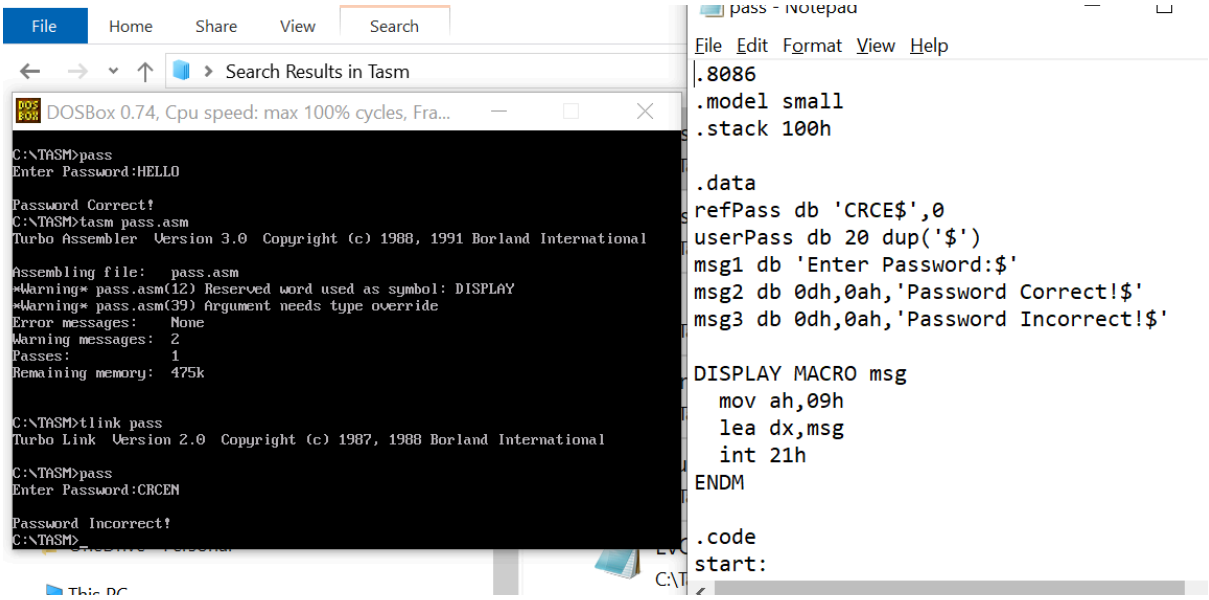
Password verification:

```
.8086
.model small
.stack 100h

.data
refPass db 'HELLO$',0
userPass db 20 dup('$')
msg1 db 'Enter Password:$'
msg2 db 0dh,0ah,'Password Correct!$'
msg3 db 0dh,0ah,'Password Incorrect!$'

DISPLAY MACRO msg
  mov ah,09h
  lea dx,msg
  int 21h
ENDM

.code
start:
  mov ax,@data
  mov ds,ax
  mov es,ax
  cld

  DISPLAY msg1

  lea di,userPass
```

```
read_char:
  mov ah,1
  int 21h
  cmp al,0dh
  je compare_pass
  mov [di],al
  inc di
  jmp read_char

compare_pass:
  mov [di],'$'
  lea si,refPass
  lea di,userPass
  mov cx,5
  repe cmpsb
  je correct

incorrect:
  DISPLAY msg3
  jmp done

correct:
  DISPLAY msg2

done:
  mov ah,4Ch
  int 21h
end start
```

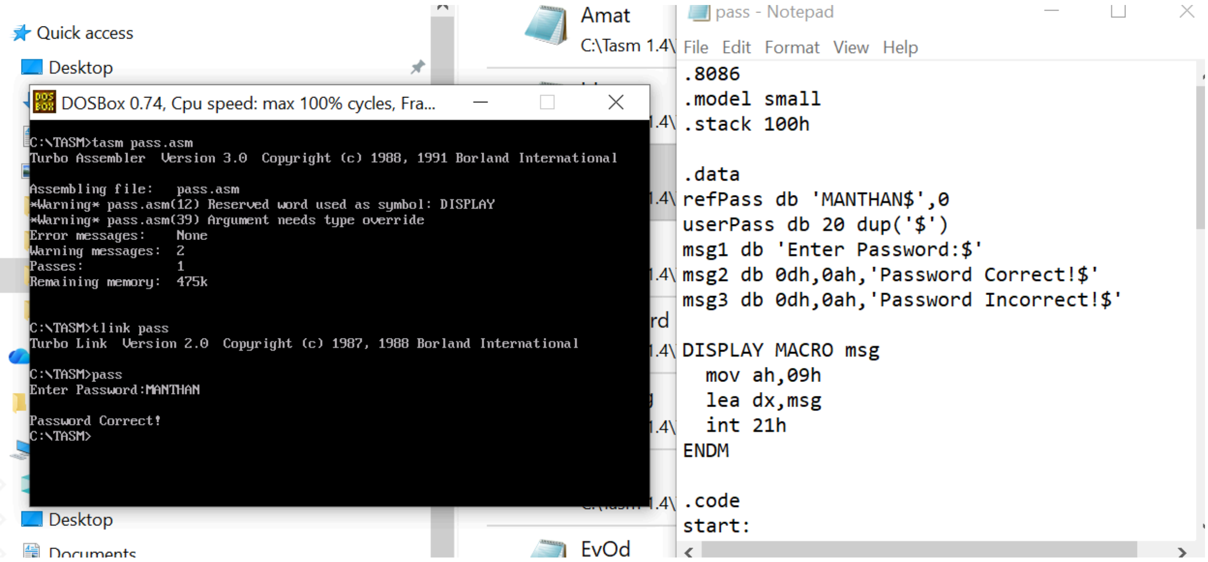


```
Assembling file:   pass.asm
*Warning* pass.asm(12) Reserved word used as symbol: DISPLAY
*Warning* pass.asm(39) Argument needs type override
Error messages:    None
Warning messages:  2
Passes:            1
Remaining memory:  475k


C:\TASM>tlink pass
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>pass
Enter Password:HELLO

Password Correct!
C:\TASM>_
```


File
Home
Share
View
Search
Search Results in Tasm
DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
pass - Notepad
File Edit Format View Help

```
C:\TASM>pass
Enter Password:HELLO

Password Correct!
C:\TASM>tasm pass.asm
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   pass.asm
*Warning* pass.asm(12) Reserved word used as symbol: DISPLAY
*Warning* pass.asm(39) Argument needs type override
Error messages:    None
Warning messages:  2
Passes:            1
Remaining memory:  475k


C:\TASM>tlink pass
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>pass
Enter Password:CRCEN

Password Incorrect!
C:\TASM>_
```

```
.8086
.model small
.stack 100h

.data
refPass db 'CRCE$',0
userPass db 20 dup('$')
msg1 db 'Enter Password:$'
msg2 db 0dh,0ah,'Password Correct!$'
msg3 db 0dh,0ah,'Password Incorrect!$'

DISPLAY MACRO msg
  mov ah,09h
  lea dx,msg
  int 21h
ENDM

.code
start:
```


Quick access
Desktop
Amat
C:\Tasm 1.4\
DOSBox 0.74, Cpu speed: max 100% cycles, Fra...
pass - Notepad
File Edit Format View Help
Desktop
Documents
EvOd

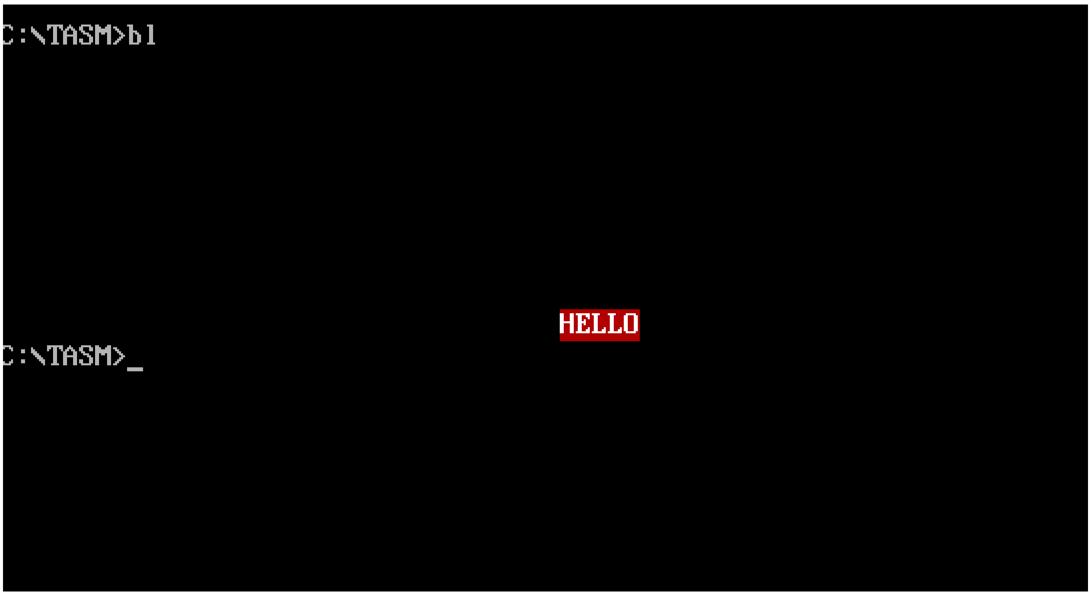
```
C:\TASM>tasm pass.asm
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   pass.asm
*Warning* pass.asm(12) Reserved word used as symbol: DISPLAY
*Warning* pass.asm(39) Argument needs type override
Error messages:    None
Warning messages:  2
Passes:            1
Remaining memory:  475k


C:\TASM>tlink pass
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>pass
Enter Password:MANTHAN

Password Correct!
C:\TASM>
```

```
.8086
.model small
.stack 100h

.data
refPass db 'MANTHAN$',0
userPass db 20 dup('$')
msg1 db 'Enter Password:$'
msg2 db 0dh,0ah,'Password Correct!$'
msg3 db 0dh,0ah,'Password Incorrect!$'

DISPLAY MACRO msg
  mov ah,09h
  lea dx,msg
  int 21h
ENDM

.code
start:
```

BLINKING:

```
.model small
.stack 100h
.data
msg db 'HELLO$'
.code
start:
   mov ax, @data
   mov ds, ax

   mov dh, 10
   mov dl, 35
   mov ah, 02h
   int 10h
```

```
    mov si, offset msg
    mov bl, 0CFh
    mov bh, 0
a
next:
    mov al, [si]
    cmp al, '$'
    je done

    mov ah, 09h
    mov cx, 1
    int 10h

    inc si
    inc dl
    mov ah, 02h
    int 10h
    jmp next

done:
    mov ah, 4Ch
    int 21h
end start
```

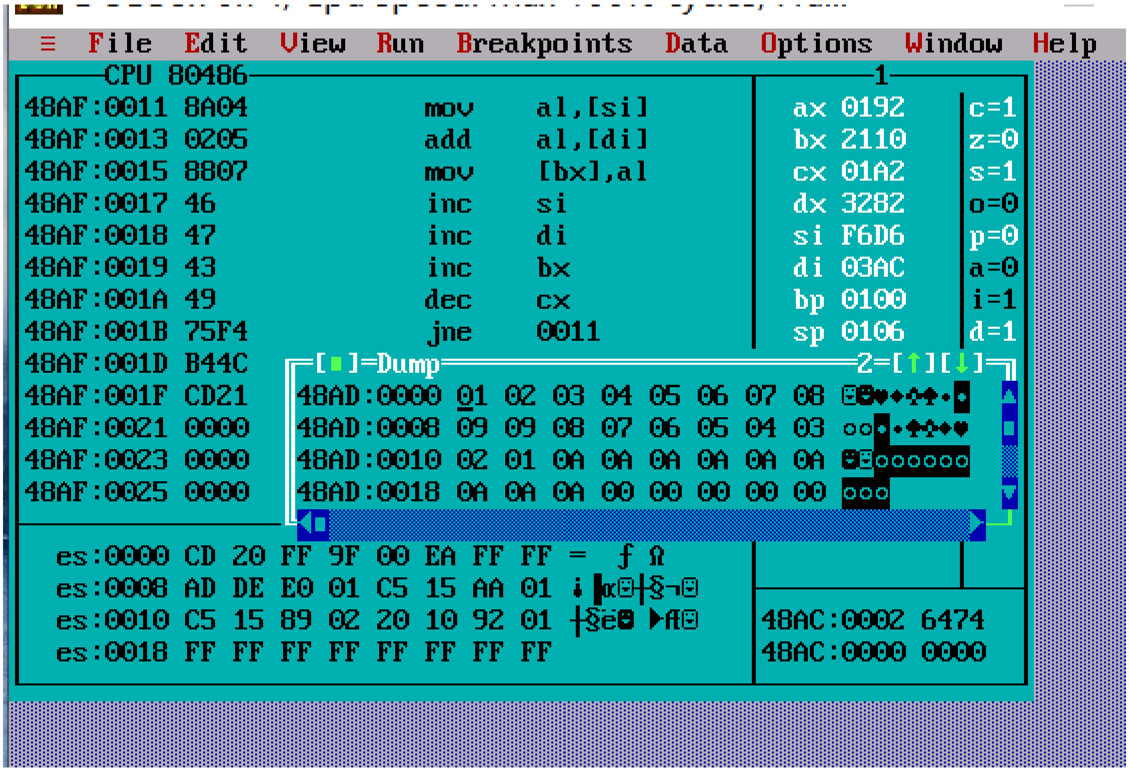


```
C:\TASM>bl




                                       HELLO
C:\TASM>_
```

MATRIX ADDITION:

```
DATA SEGMENT
M1 DB 01H,02H,03H,04H,05H,06H,07H,08H,09H
M2 DB 09H,08H,07H,06H,05H,04H,03H,02H,01H
RES DB 09 DUP(?)
DATA ENDS
```

```
CODE SEGMENT
ASSUME CS:CODE, DS:DATA

START:
MOV AX,DATA
MOV DS,AX

LEA SI,M1
LEA DI,M2
LEA BX,RES
MOV CX,09

AGAIN:
MOV AL,[SI]
ADD AL,[DI]
MOV [BX],AL
INC SI
INC DI
INC BX
DEC CX
JNZ AGAIN

MOV AH,4CH
INT 21H

CODE ENDS
END START
```



```
CPU 80486                                                  1
48AF:0011 8A04          mov    al,[si]           ax 0192   c=1
48AF:0013 0205          add    al,[di]           bx 2110   z=0
48AF:0015 8807          mov    [bx],al           cx 01A2   s=1
48AF:0017 46            inc    si                dx 3282   o=0
48AF:0018 47            inc    di                si F6D6   p=0
48AF:0019 43            inc    bx                di 03AC   a=0
48AF:001A 49            dec    cx                bp 0100   i=1
48AF:001B 75F4          jne    0011              sp 0106   d=1
48AF:001D B44C    [■]=Dump=====================2=[↑][↓]
48AF:001F CD21    48AD:0000 01 02 03 04 05 06 07 08 ☺☻♥♦♣♠•◘
48AF:0021 0000    48AD:0008 09 09 08 07 06 05 04 03 ○○◘•♠♣♦♥
48AF:0023 0000    48AD:0010 02 01 0A 0A 0A 0A 0A 0A ☻☺◙◙◙◙◙◙
48AF:0025 0000    48AD:0018 0A 0A 0A 00 00 00 00 00 ◙◙◙

  es:0000 CD 20 FF 9F 00 EA FF FF ═  ƒ Ω
  es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
  es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     48AC:0002 6474
  es:0018 FF FF FF FF FF FF FF FF             48AC:0000 0000
```

Matrix Multiply:
DATA SEGMENT
M1 DB 1,2,3,4,5,6,7,8,9
M2 DB 9,8,7,6,5,4,3,2,1
RES DB 9 DUP(?)
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA

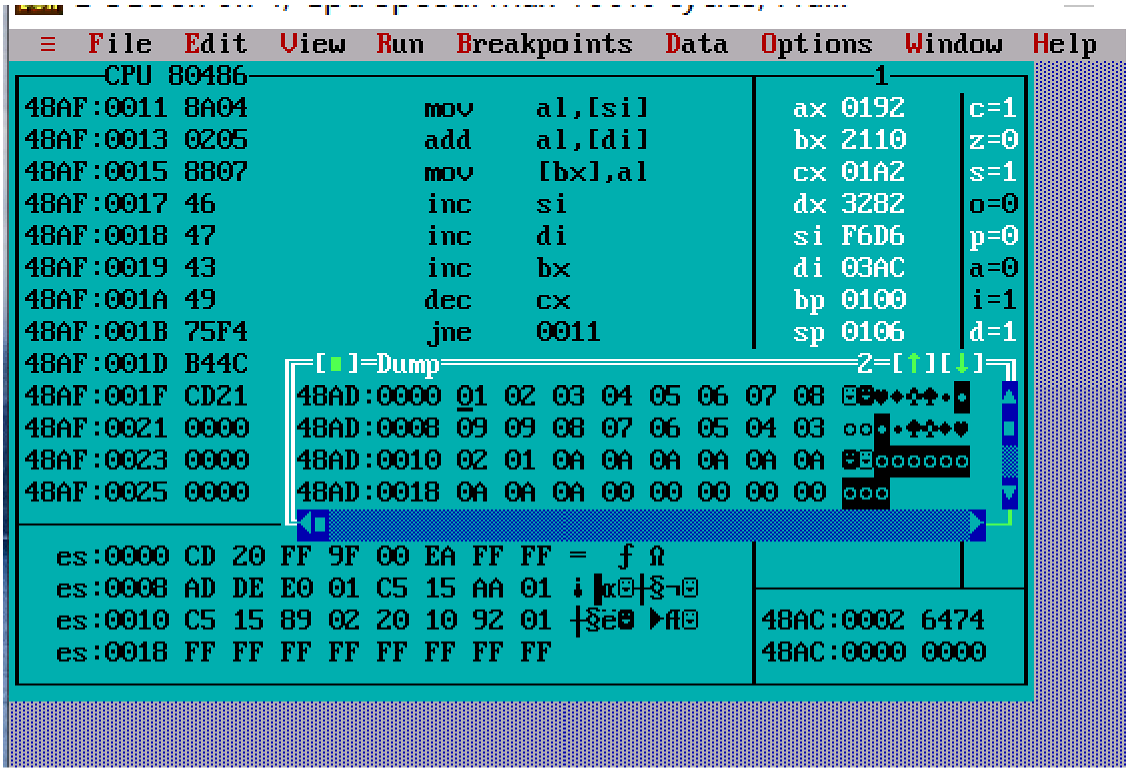
START:
MOV AX,DATA
MOV DS,AX

LEA SI,M1
LEA DI,M2
LEA BX,RES
MOV CH,03

ROW_LOOP:
MOV CL,03
PUSH SI
PUSH DI
MOV DH,03

COL_LOOP:

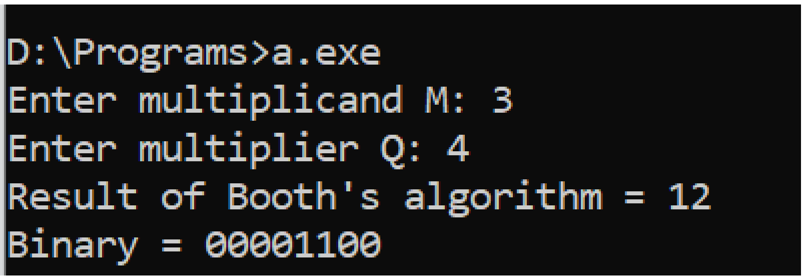
```
PUSH CX
PUSH SI
MOV CL,03
MOV AL,00H
MOV BL,00H

INNER_LOOP:
MOV AH,[SI]
MOV BH,[DI]
MUL BH
ADD BL,AL
INC SI
ADD DI,03
DEC CL
JNZ INNER_LOOP

MOV [BX],BL
INC BX
POP SI
POP CX
INC SI
DEC DH
JNZ COL_LOOP

POP DI
POP SI
ADD SI,03
DEC CHa
JNZ ROW_LOOP

MOV AH,4CH
INT 21H

CODE ENDS
END START
```



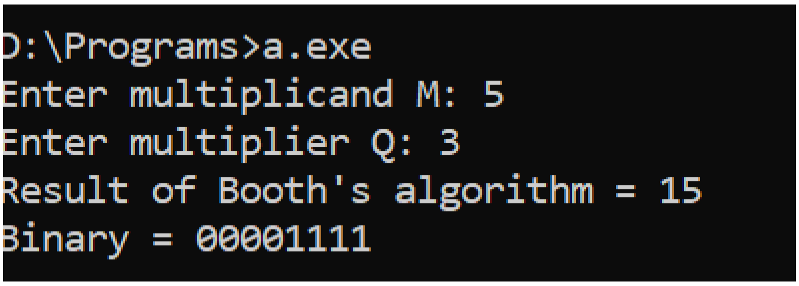
```
CPU 80486                                              1
48AF:0011 8A04          mov    al,[si]        ax 0192   c=1
48AF:0013 0205          add    al,[di]        bx 2110   z=0
48AF:0015 8807          mov    [bx],al        cx 01A2   s=1
48AF:0017 46            inc    si             dx 3282   o=0
48AF:0018 47            inc    di             si F6D6   p=0
48AF:0019 43            inc    bx             di 03AC   a=0
48AF:001A 49            dec    cx             bp 0100   i=1
48AF:001B 75F4          jne    0011           sp 0106   d=1
48AF:001D B44C   [■]=Dump                            2=[↑][↓]
48AF:001F CD21   48AD:0000 01 02 03 04 05 06 07 08 ☺☻♥♦♣♠•◘
48AF:0021 0000   48AD:0008 09 09 08 07 06 05 04 03 ○○◘•♠♣♦♥
48AF:0023 0000   48AD:0010 02 01 0A 0A 0A 0A 0A 0A ☻☺◙◙◙◙◙◙
48AF:0025 0000   48AD:0018 0A 0A 0A 00 00 00 00 00 ◙◙◙

   es:0000 CD 20 FF 9F 00 EA FF FF =  ƒ Ω
   es:0008 AD DE E0 01 C5 15 AA 01 ¡▌α☺┼§¬☺
   es:0010 C5 15 89 02 20 10 92 01 ┼§ë☻ ►Æ☺     48AC:0002 6474
   es:0018 FF FF FF FF FF FF FF FF              48AC:0000 0000
```

BOOTHS:

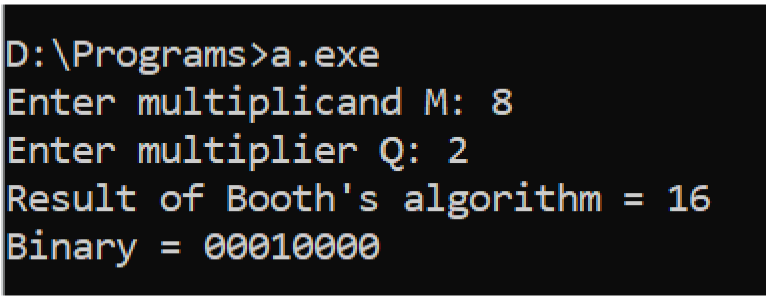
```
#include <stdio.h>

void toBinary(int num, int bits) {
    for (int i = bits - 1; i >= 0; i--)
        printf("%d", (num >> i) & 1);
}

int main() {
    int m, q, a = 0, qn = 0, n = 4;

    printf("Enter multiplicand M: ");
    scanf("%d", &m);
    printf("Enter multiplier Q: ");
    scanf("%d", &q);

    for (int i = 0; i < n; i++) {
        int q0 = q & 1;
        if (q0 == 1 && qn == 0)
            a -= m;
        else if (q0 == 0 && qn == 1)
            a += m;
        qn = q0;
        int sign = (a >> (n - 1)) & 1;
        q = (q >> 1) | ((a & 1) << (n - 1));
        a = (a >> 1) | (sign << (n - 1));
    }

    int result = a * (1 << n) + q;
    printf("Result of Booth's algorithm = %d\n", result);
    printf("Binary = ");
    toBinary(result, n * 2);
    printf("\n");

    return 0;
}
```



```
D:\Programs>a.exe
Enter multiplicand M: 3
Enter multiplier Q: 4
Result of Booth's algorithm = 12
Binary = 00001100
```



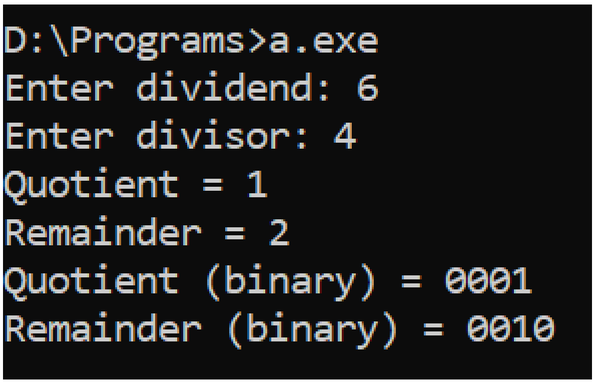
```
D:\Programs>a.exe
Enter multiplicand M: 5
Enter multiplier Q: 3
Result of Booth's algorithm = 15
Binary = 00001111
```



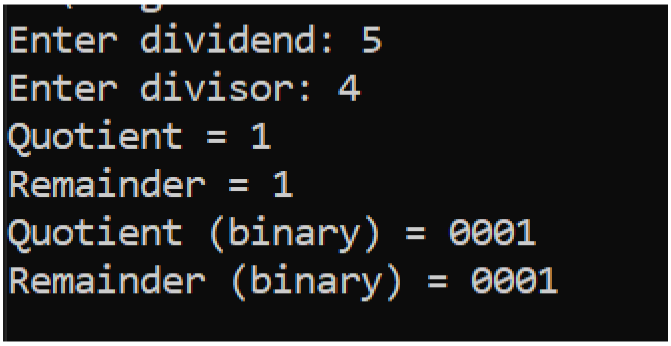
```
D:\Programs>a.exe
Enter multiplicand M: 8
Enter multiplier Q: 2
Result of Booth's algorithm = 16
Binary = 00010000
```

RESTORING:

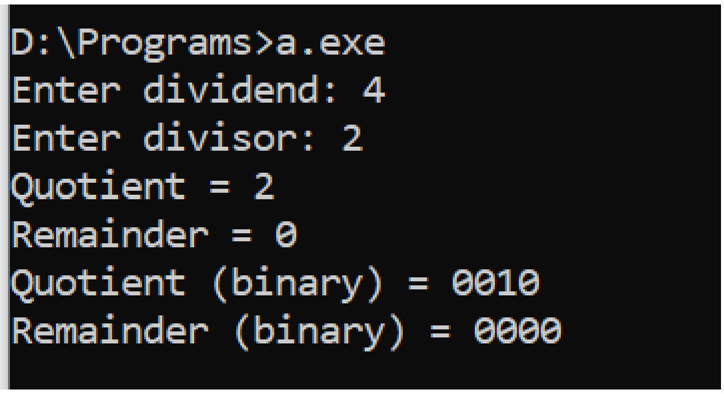
```
#include <stdio.h>

void toBinary(int num, int bits) {
    for (int i = bits - 1; i >= 0; i–)
        printf("%d", (num >> i) & 1);
}

int main() {
    int dividend, divisor;
    printf("Enter dividend: ");
    scanf("%d", &dividend);
    printf("Enter divisor: ");
    scanf("%d", &divisor);

    int n = 4;
    int A = 0;
    int Q = dividend;
    int M = divisor;

    for (int i = 0; i < n; i++) {
        A = (A << 1) | ((Q & 8) >> 3);
        Q = (Q << 1) & 0xF;
        A = A - M;
        if (A < 0) {
            A = A + M;
            Q &= 0xE;
        } else {
            Q |= 1;
        }
    }

    printf("Quotient = %d\n", Q);
    printf("Remainder = %d\n", A);
    printf("Quotient (binary) = ");
    toBinary(Q, n);
    printf("\nRemainder (binary) = ");
    toBinary(A, n);
    printf("\n");

    return 0;
}
```



```
D:\Programs>a.exe
Enter dividend: 6
Enter divisor: 4
Quotient = 1
Remainder = 2
Quotient (binary) = 0001
Remainder (binary) = 0010
```



```
Enter dividend: 5
Enter divisor: 4
Quotient = 1
Remainder = 1
Quotient (binary) = 0001
Remainder (binary) = 0001
```



```
D:\Programs>a.exe
Enter dividend: 4
Enter divisor: 2
Quotient = 2
Remainder = 0
Quotient (binary) = 0010
Remainder (binary) = 0000
```