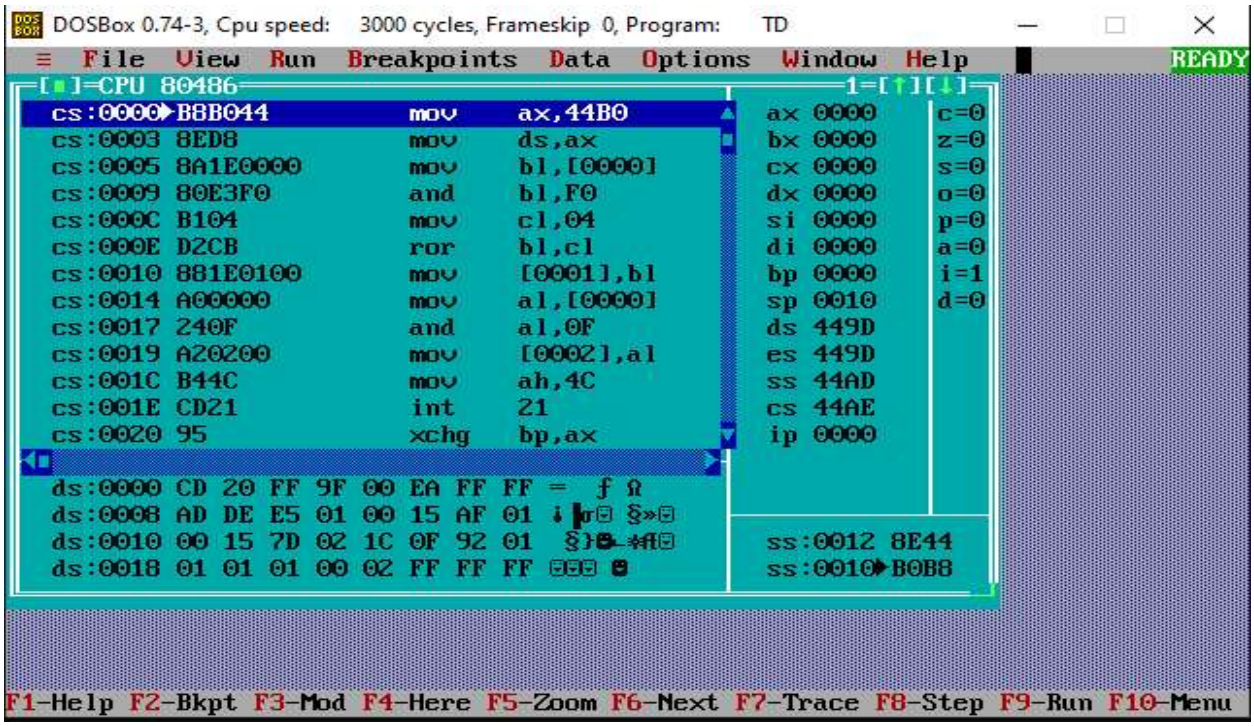
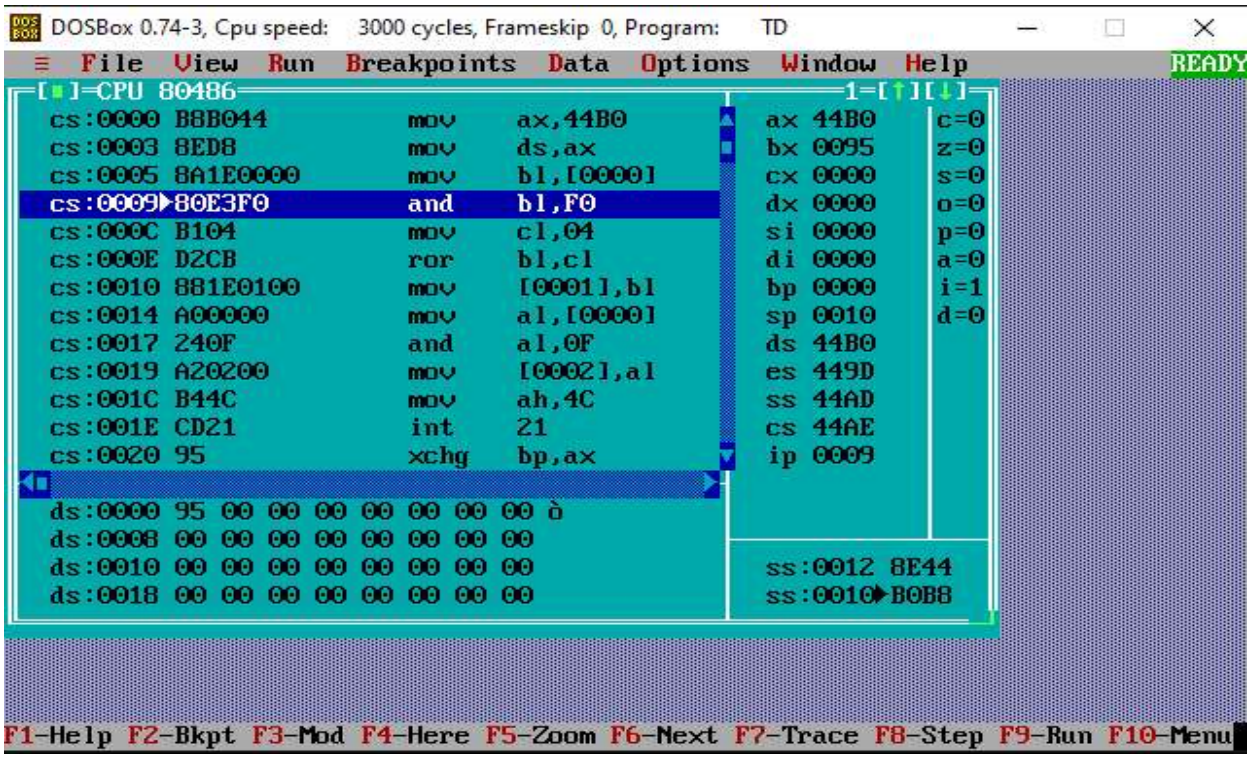
Write an ALP for addition of 16-bit BCD numbers.

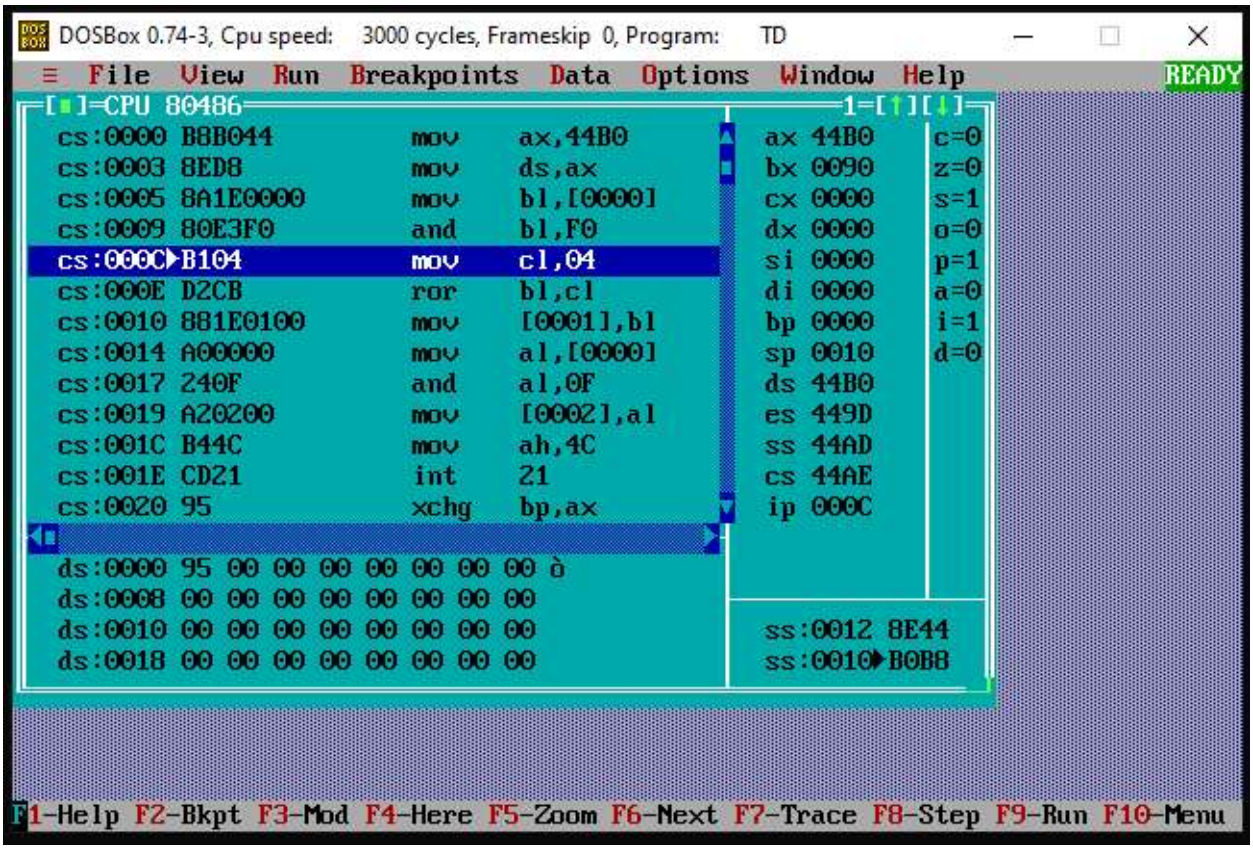
Code:

```
model small
stack 10h
data segment
  num db 95h
  num1 db 00h
  num2 db 00h
data ends
code segment
  assume cs:code, ds:data
  start:
    mov ax, data
    mov ds, ax

    mov bl, num
    and bl, 0F0h
    mov cl, 04h
    ror bl, cl

    mov num1, bl


    mov al, num
    and al, 0Fh
    mov num2, al


    mov ah, 4ch
    int 21h
code ends
end start
```

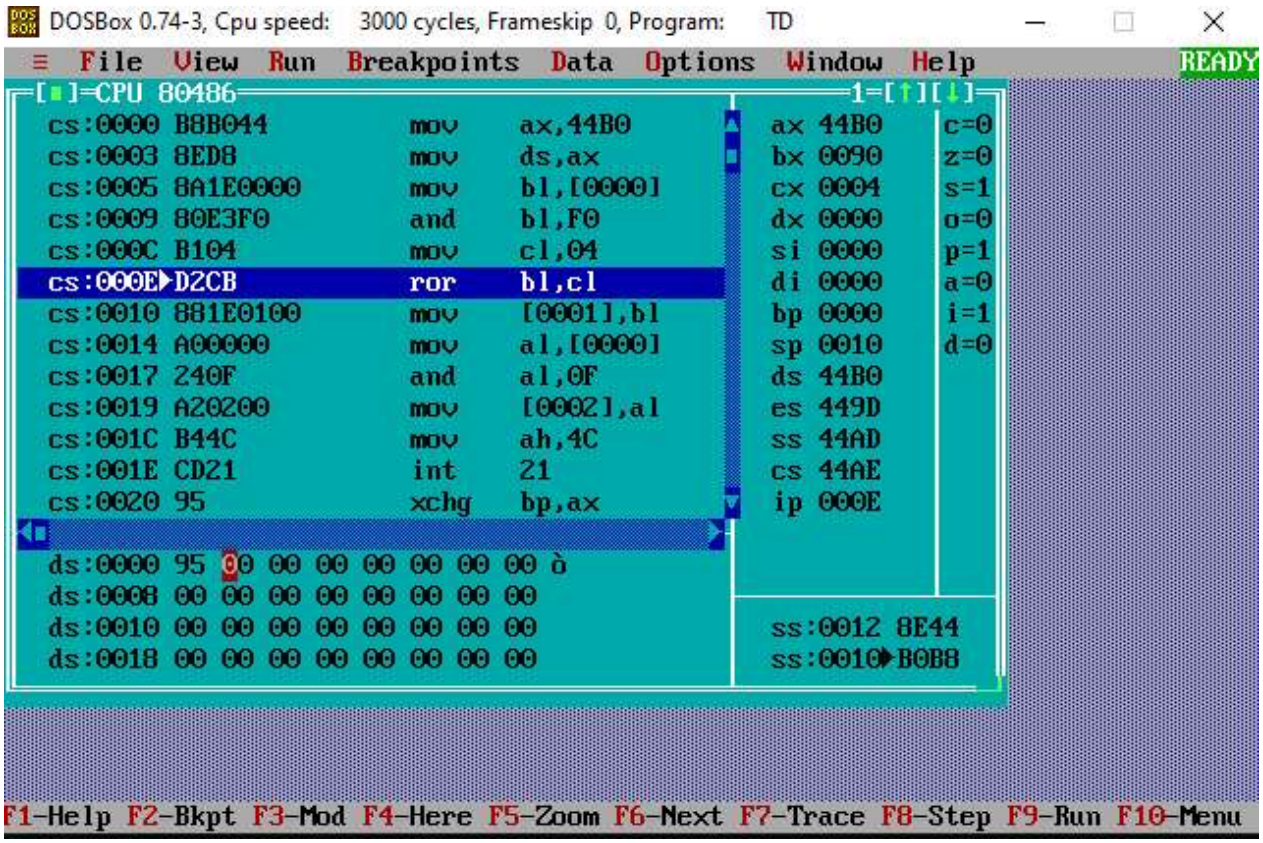




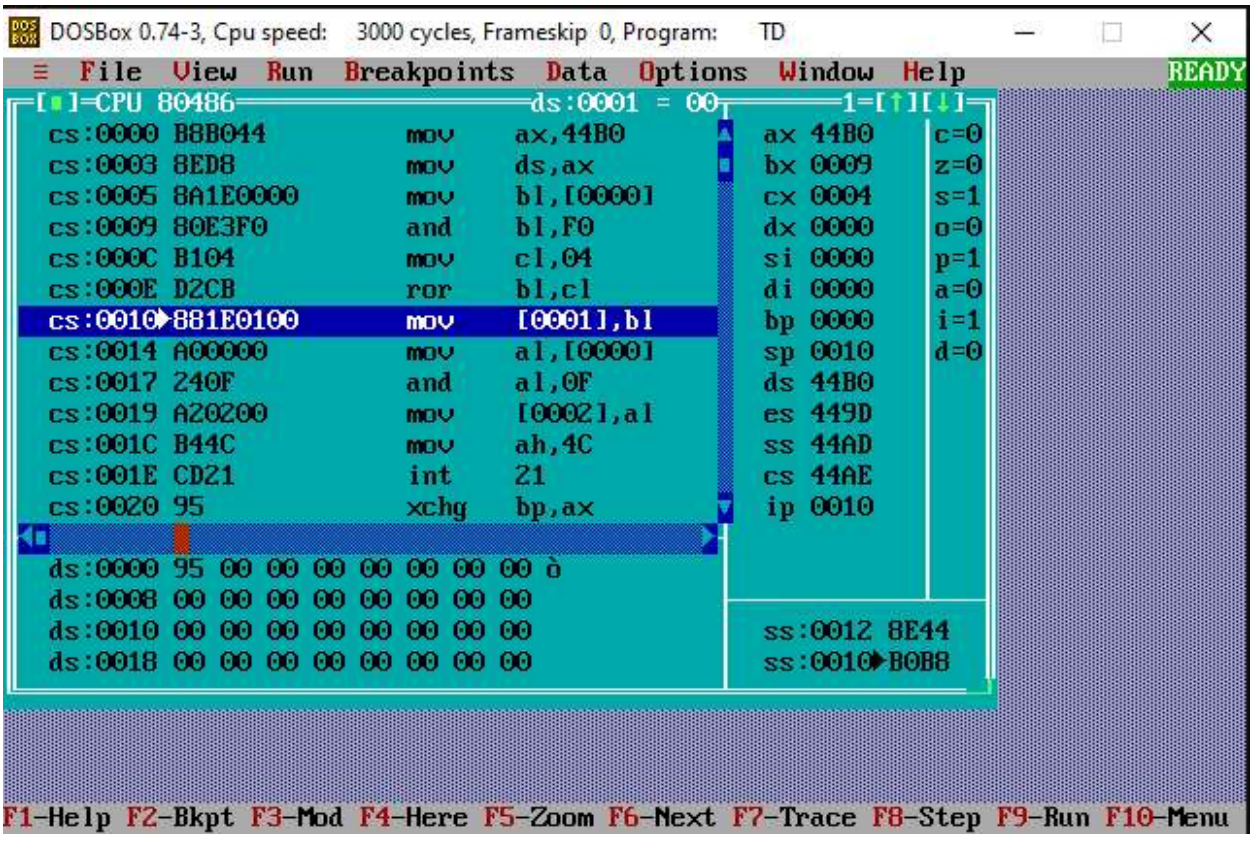
mov ax, 44B0 Loading the data segment into AX
mov ds, ax Setting up the data segment
mov bl, [0000] Copying the tens digit of 95 to BL register



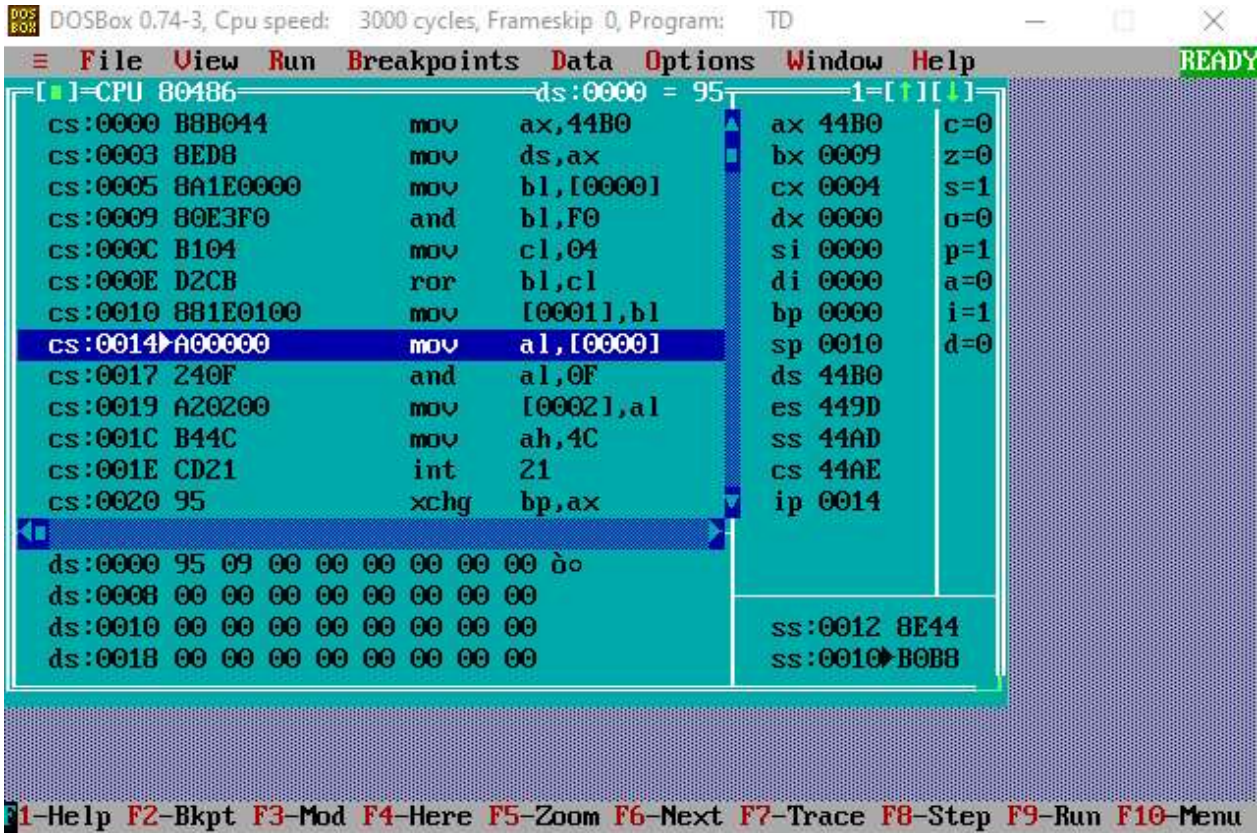
and bl, F0 Keeping only the tens digit in BL



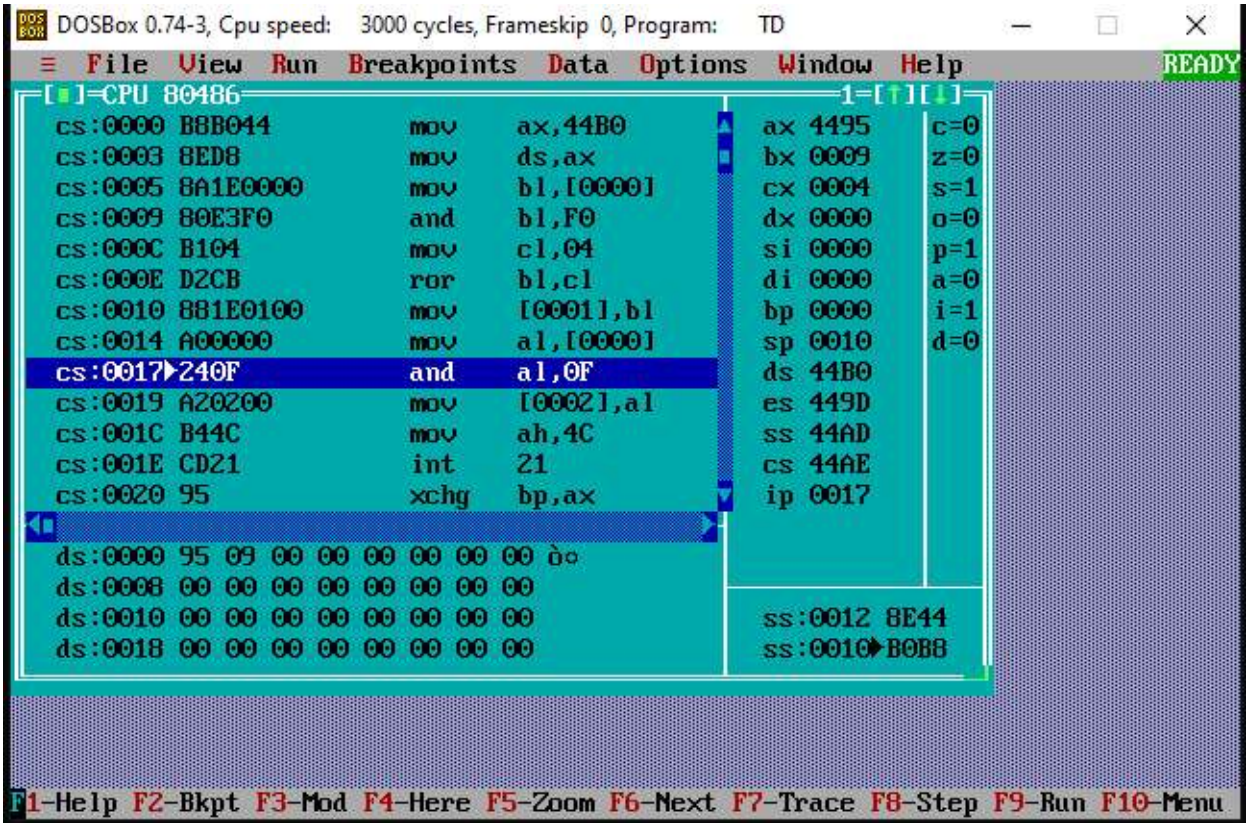
mov cl, 04 Preparing to shift the tens digit to the right



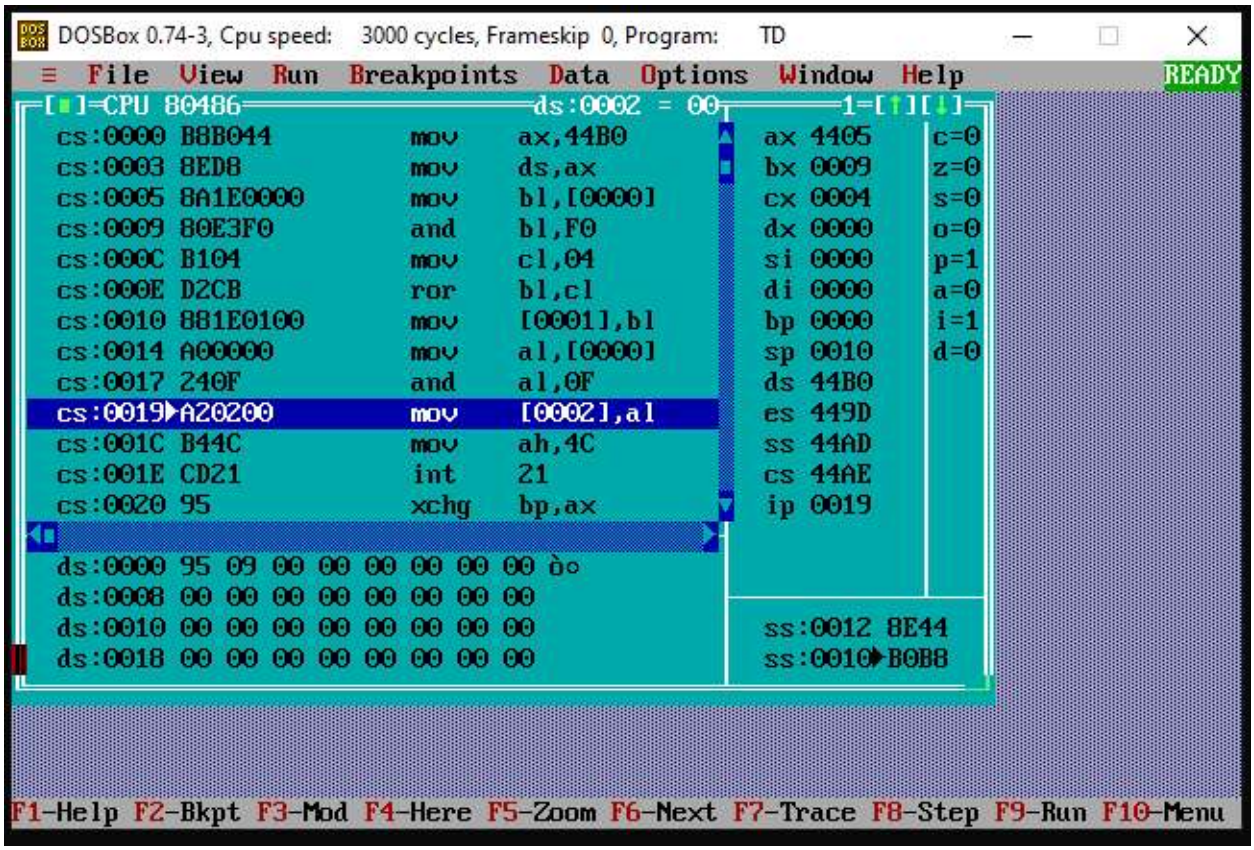
ror bl, cl Shifting the tens digit to the right



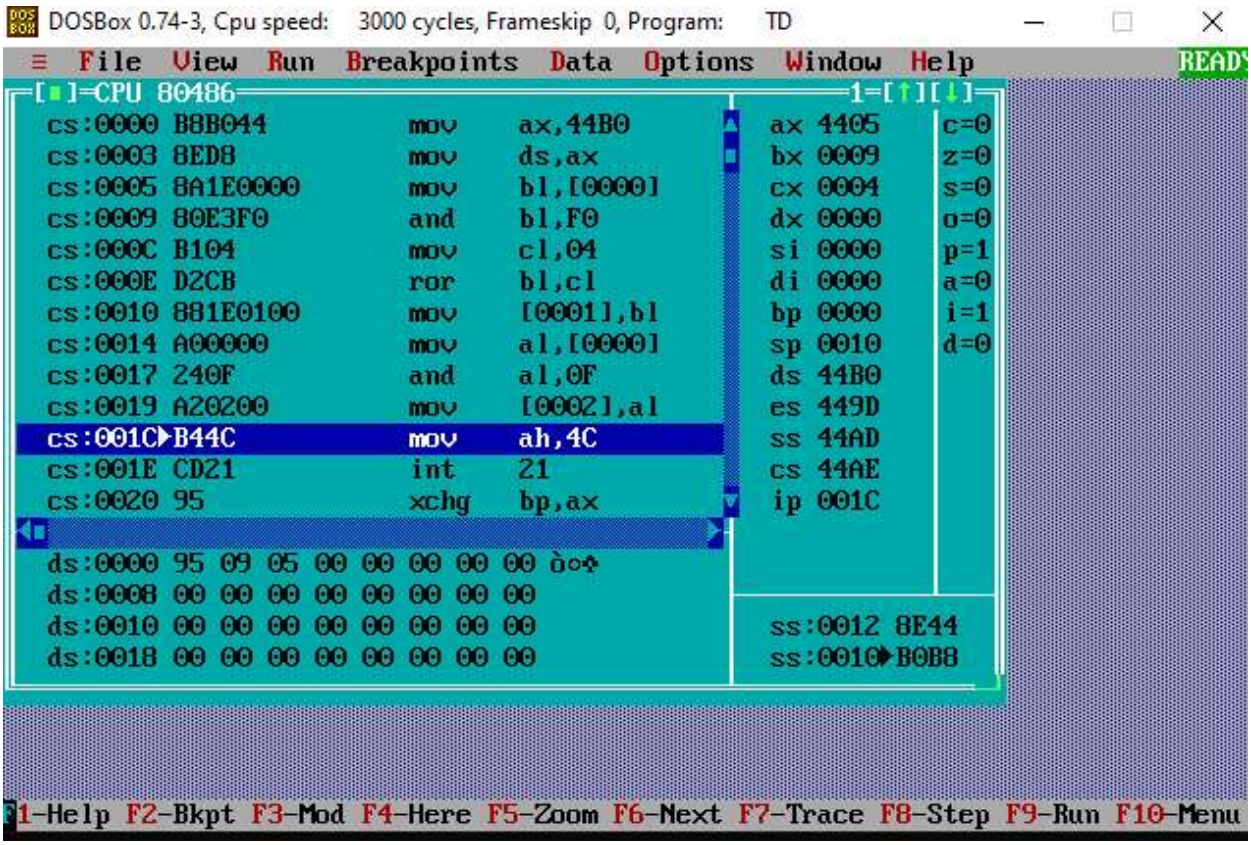
mov [0001], bl Storing the shifted tens digit in num1



mov al, [0000] Copying the ones digit of 95 to AL register



and al, 0F Keeping only the ones digit in AL



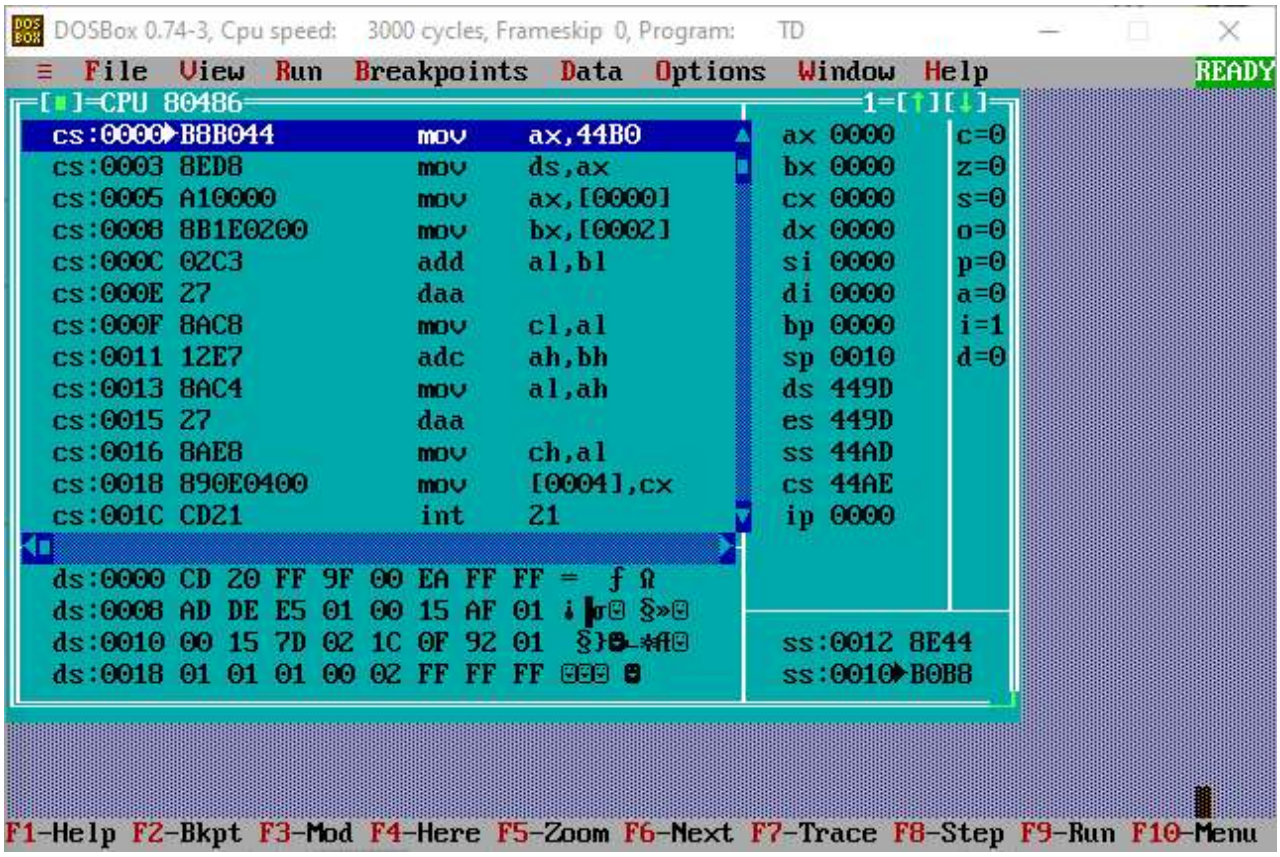
mov [0002], al Storing the ones digit in num2

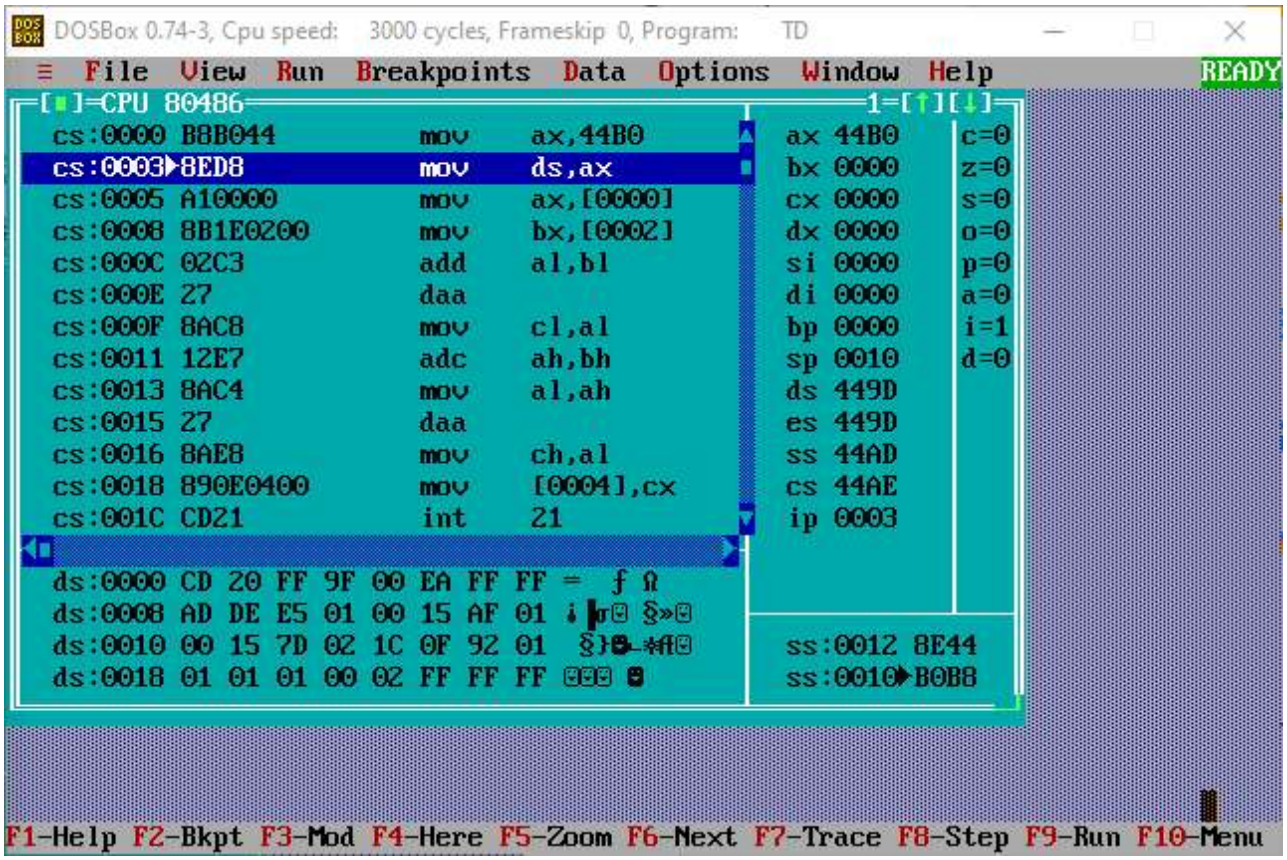
mov ah, 4C Preparing to request program termination

Write an ALP to convert a packed BCD number to an unpacked BCD number.

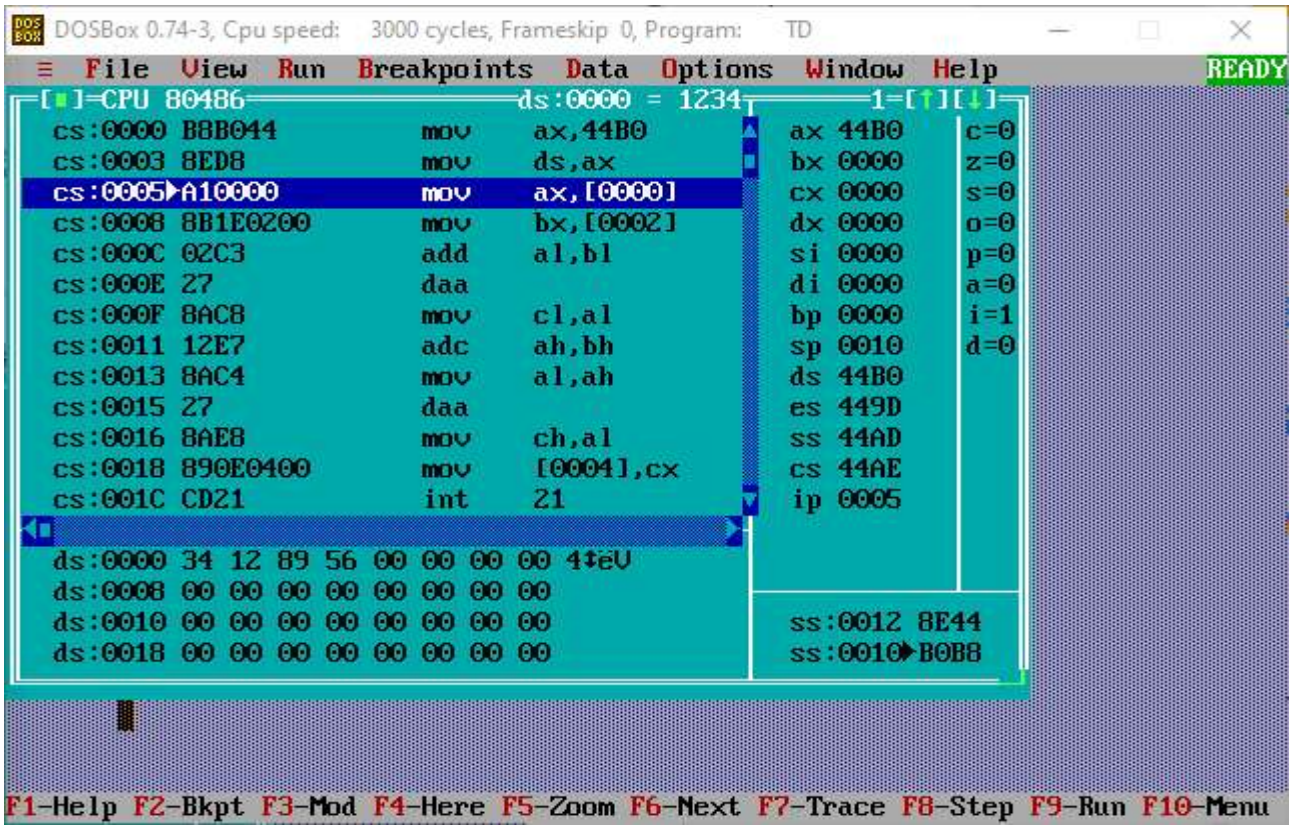
Code:

```
model small
stack 10h
data segment
  num1 dw 1234h
  num2 dw 5689h
  res dw ?
data ends
code segment
  assume cs: code, ds: data
  start:
    mov ax, data
    mov ds, ax

    mov ax, num1
    mov  bx, num2
    add al, bl
    daa
    mov cl, al
    adc ah, bh
    mov al, ah
    daa
    mov ch, al

    mov res, cx

  int 21h
code ends
end start
```

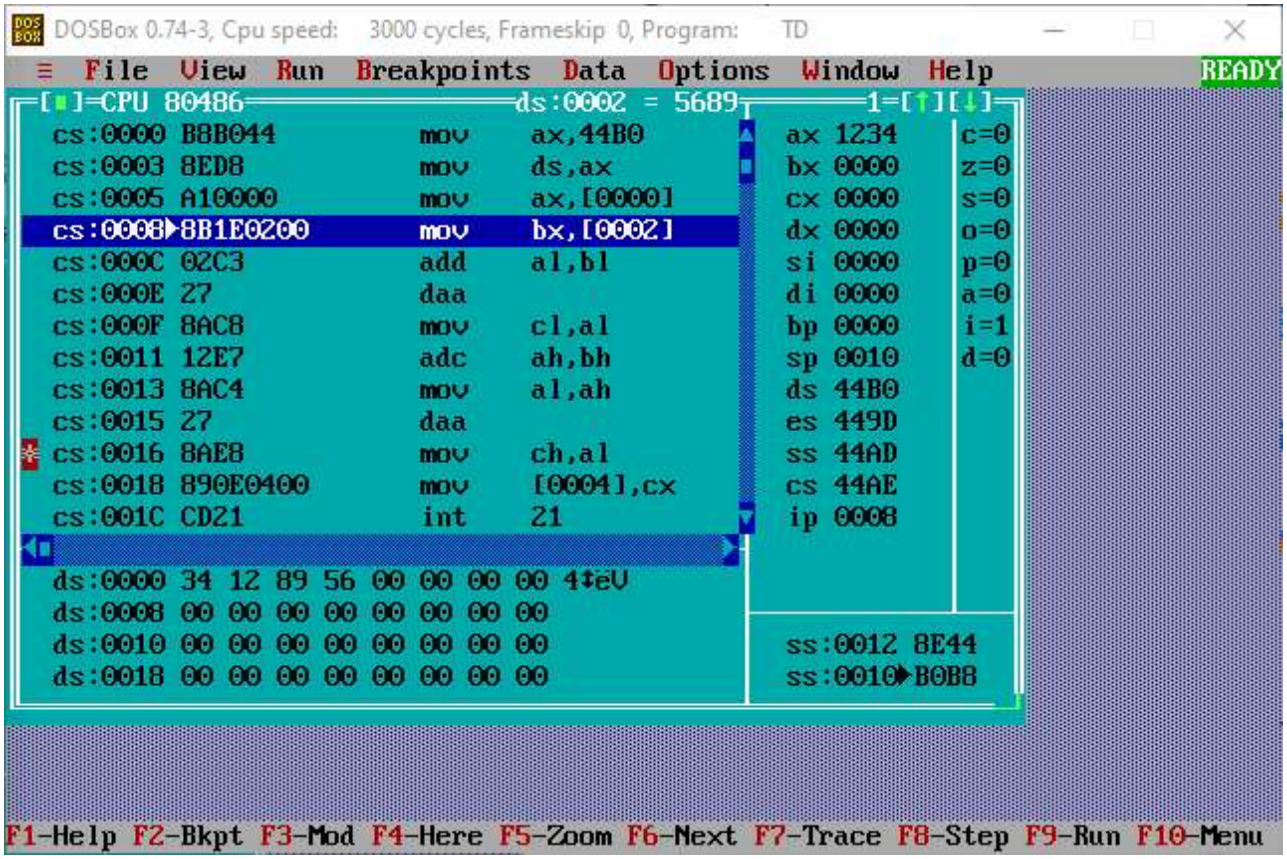




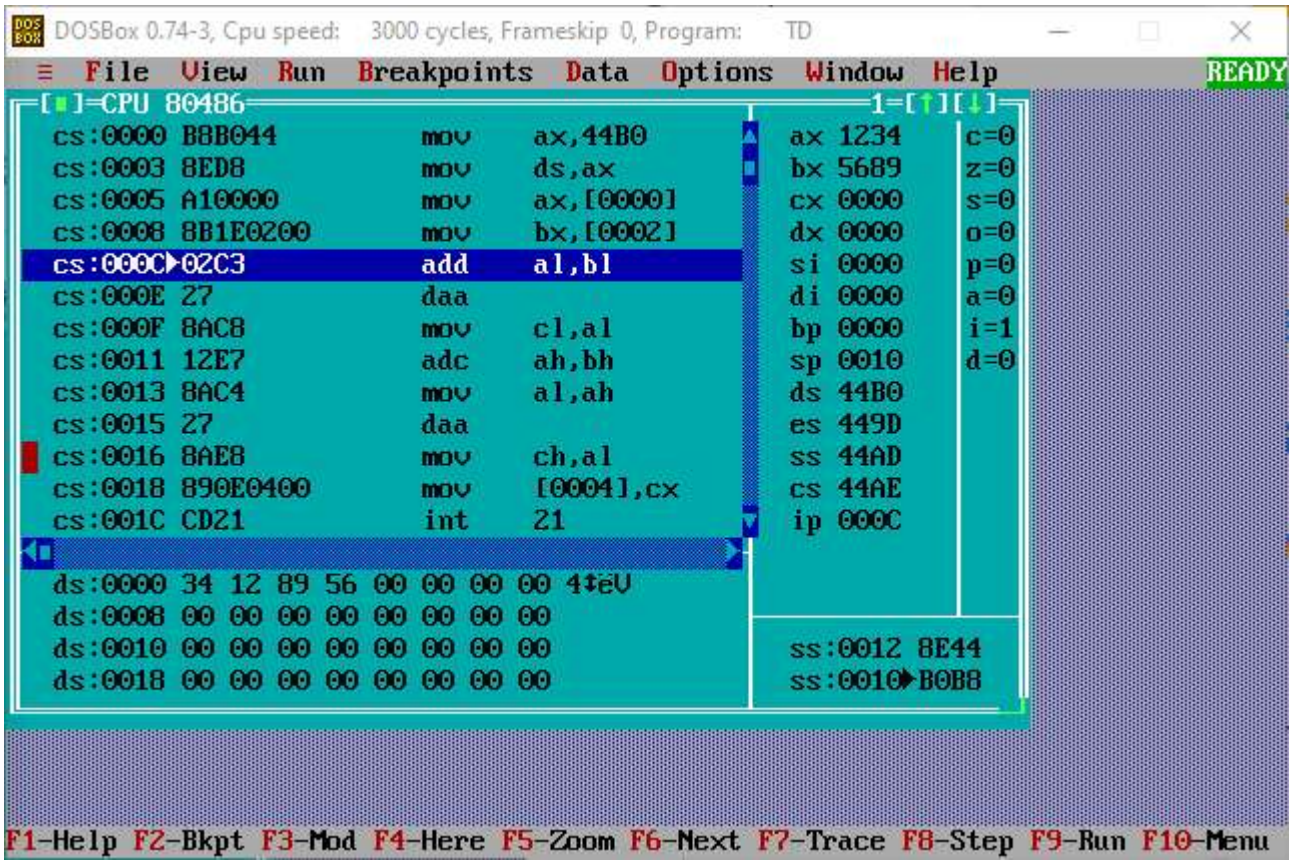
mov ax, 44B0 Load the address of the data segment into AX



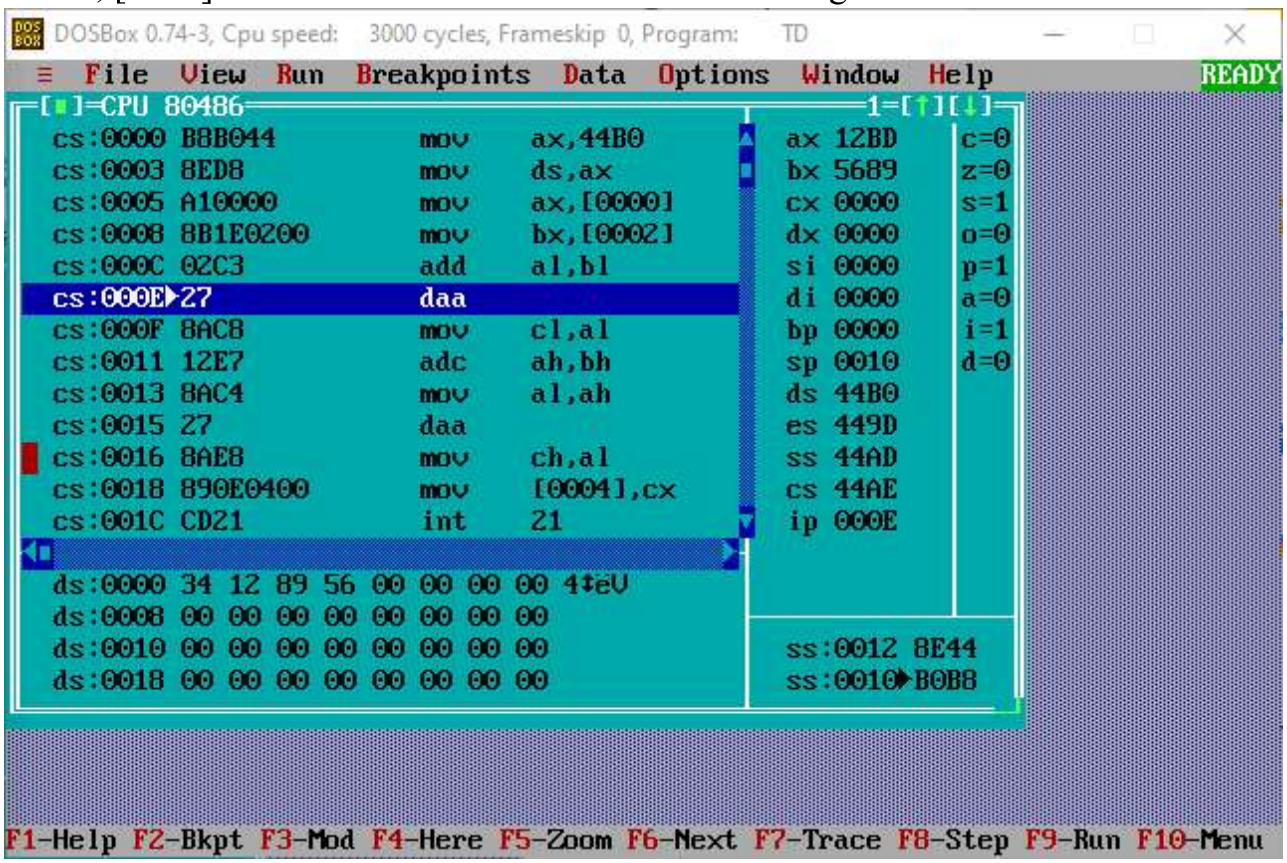
mov ds, ax Set the data segment register to the address in AX



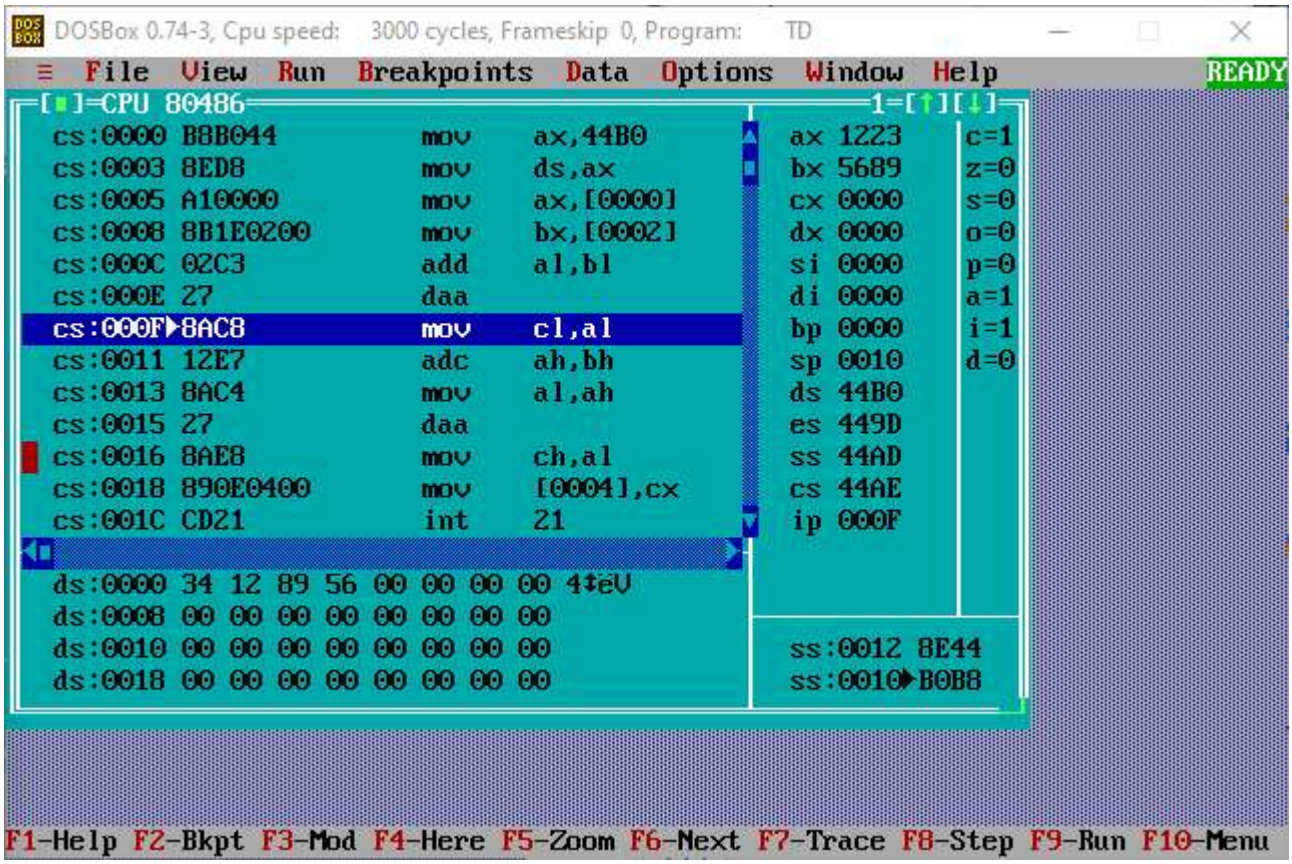
mov ax, [0000] Load the value of num1 into AX register



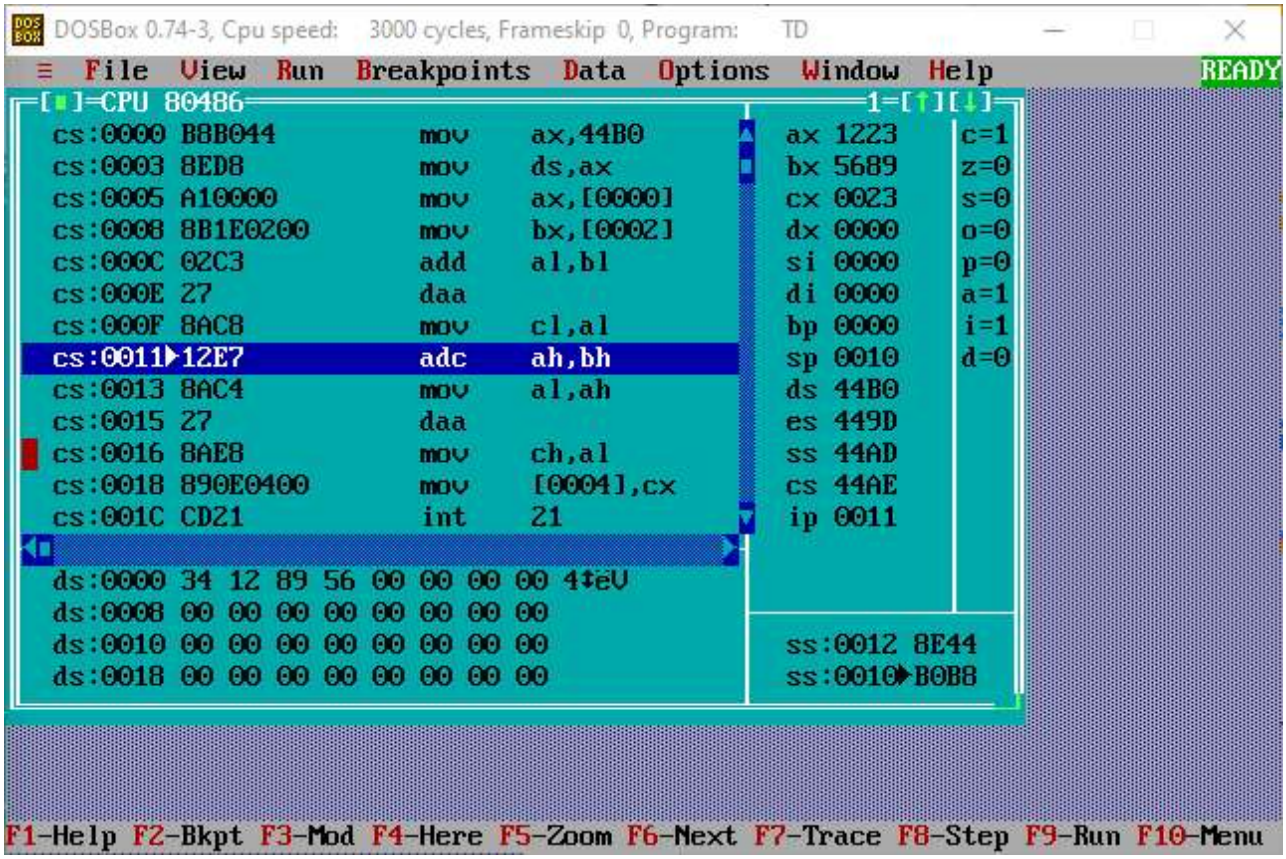
mov bx, [0002] Load the value of num2 into BX register



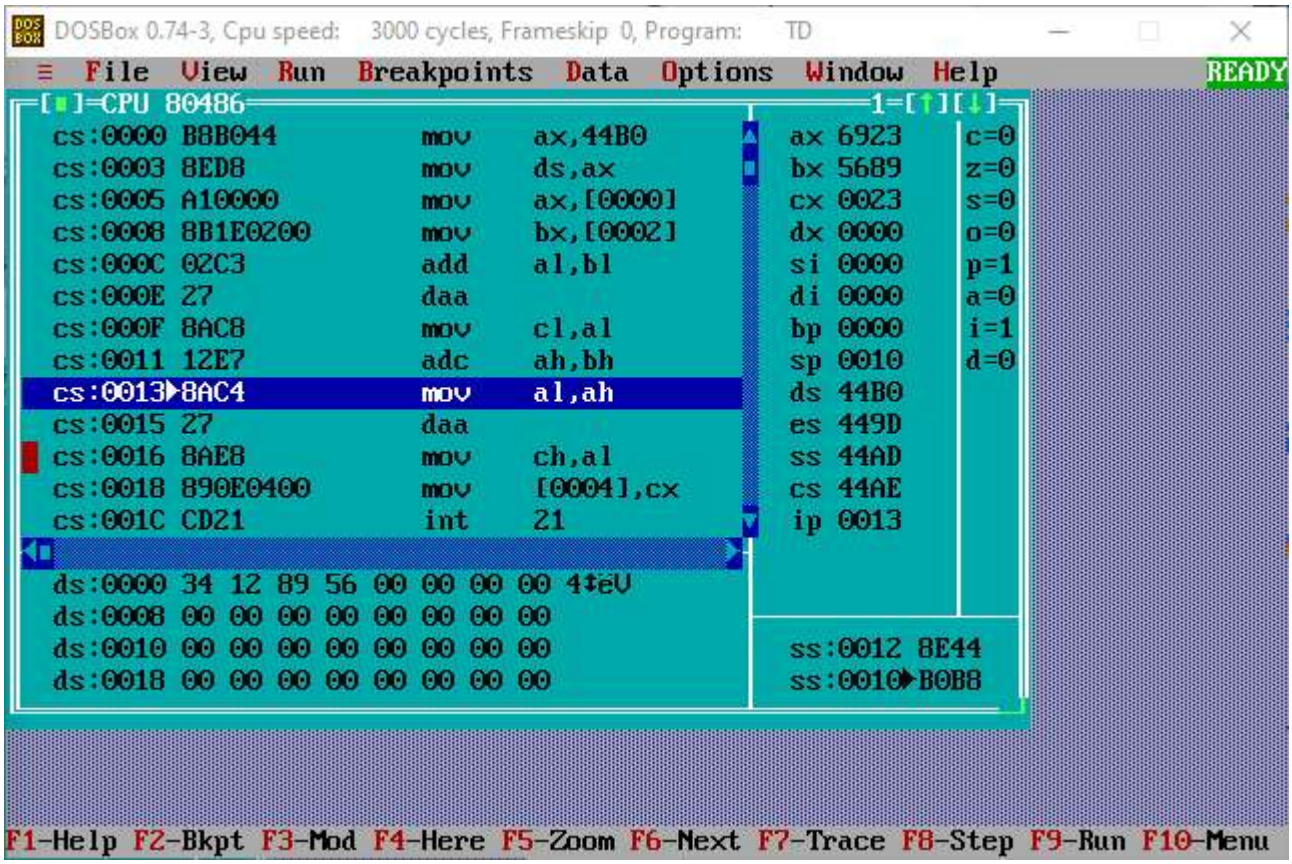
add al, bl Add the low bytes of num1 and num2, storing the result in AL



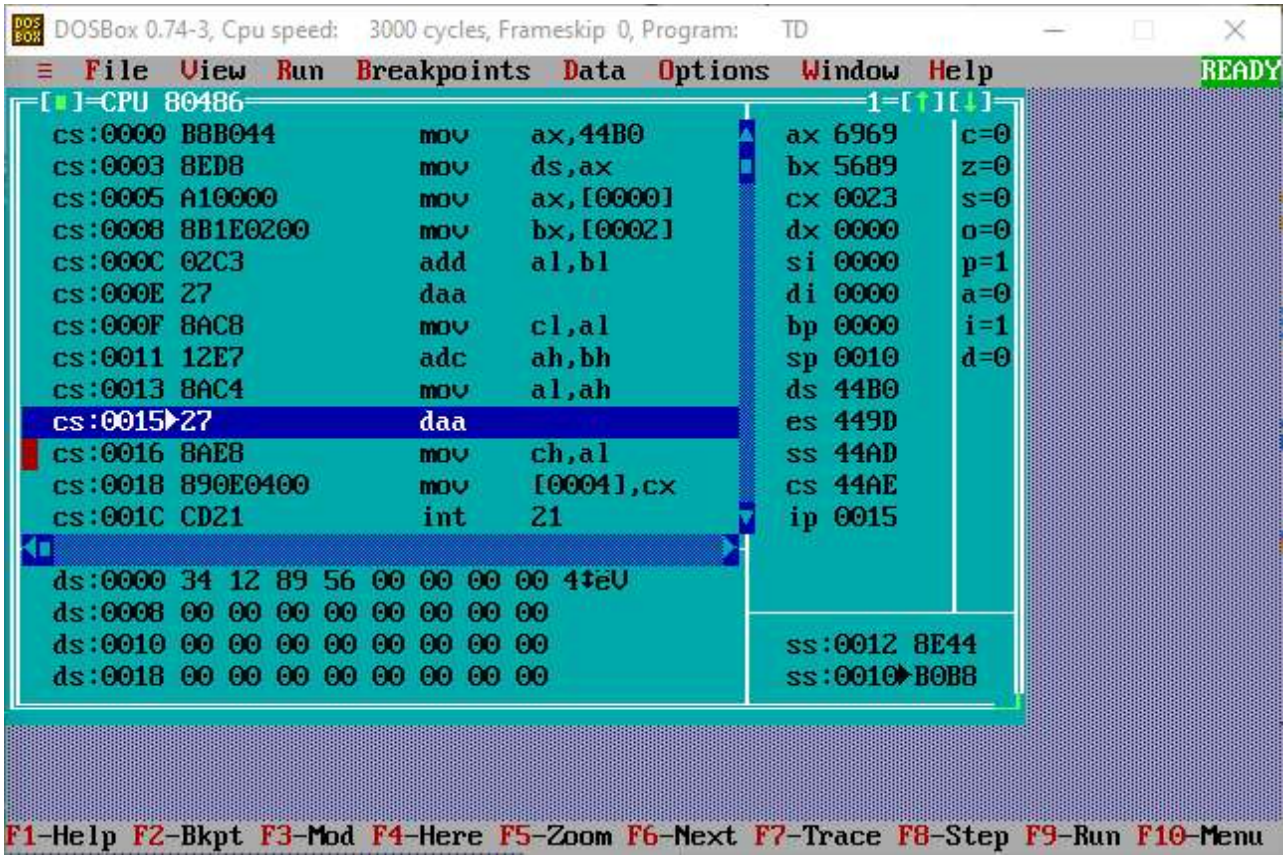
daa Adjust AL after addition to handle decimal carry



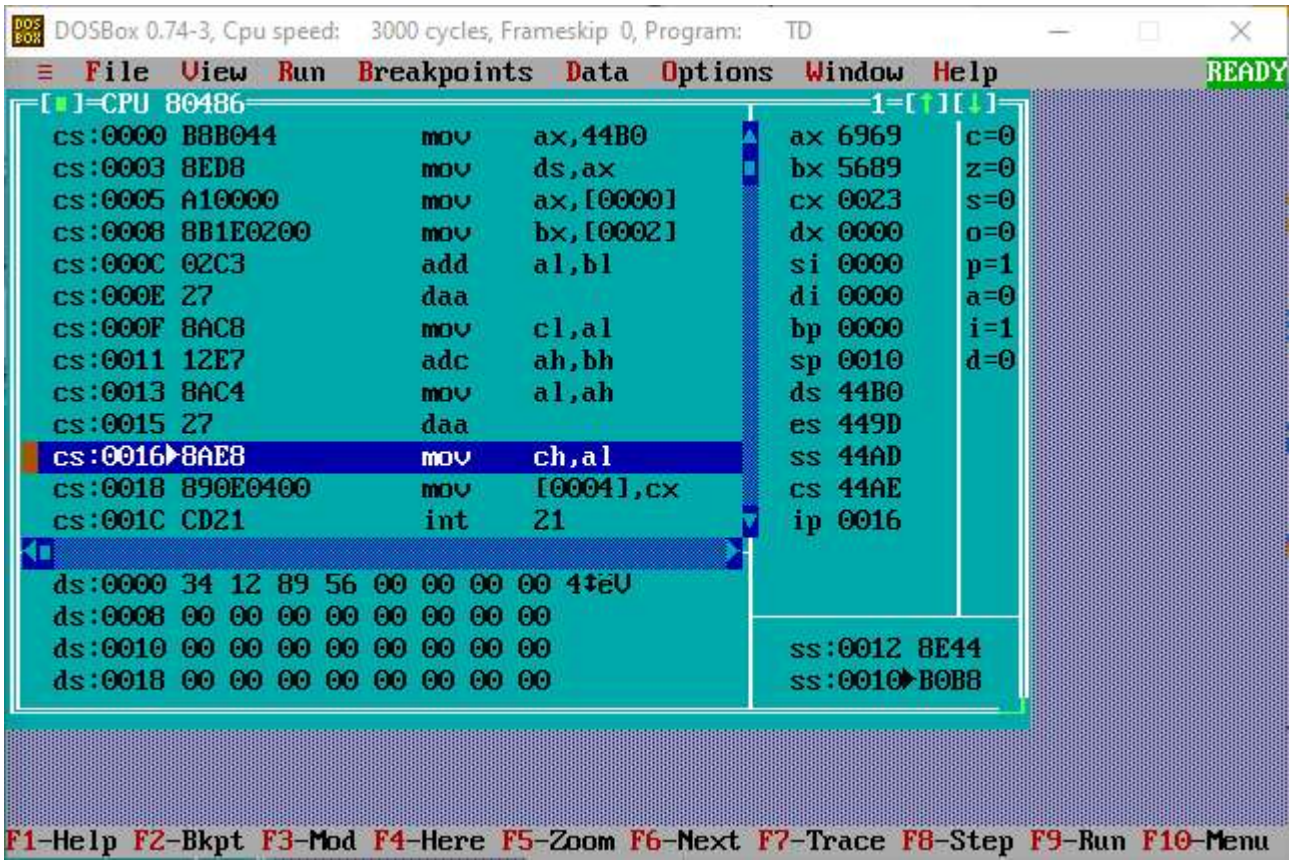
mov cl, al Move the adjusted low byte to CL register



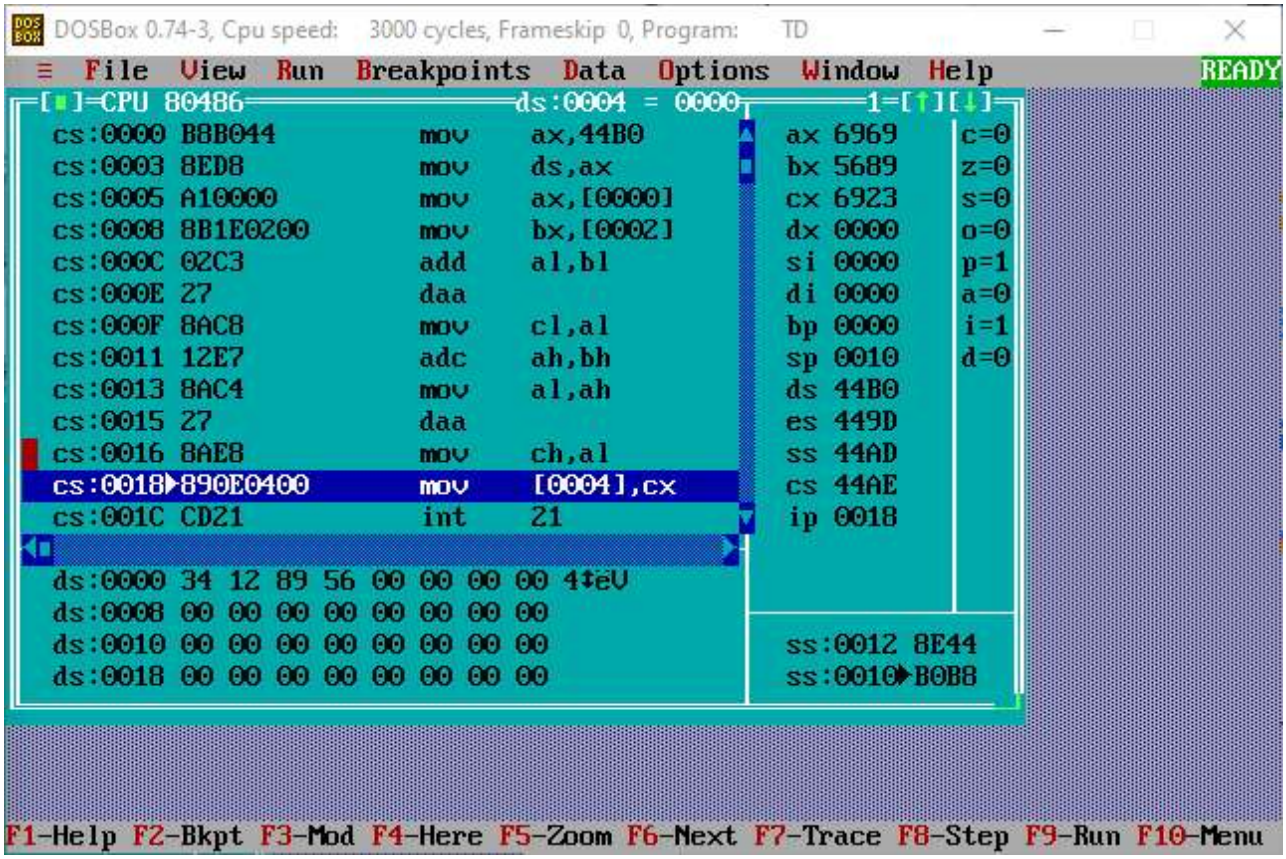
adc ah, bh Add the high bytes of num1 and num2 along with carry to AH



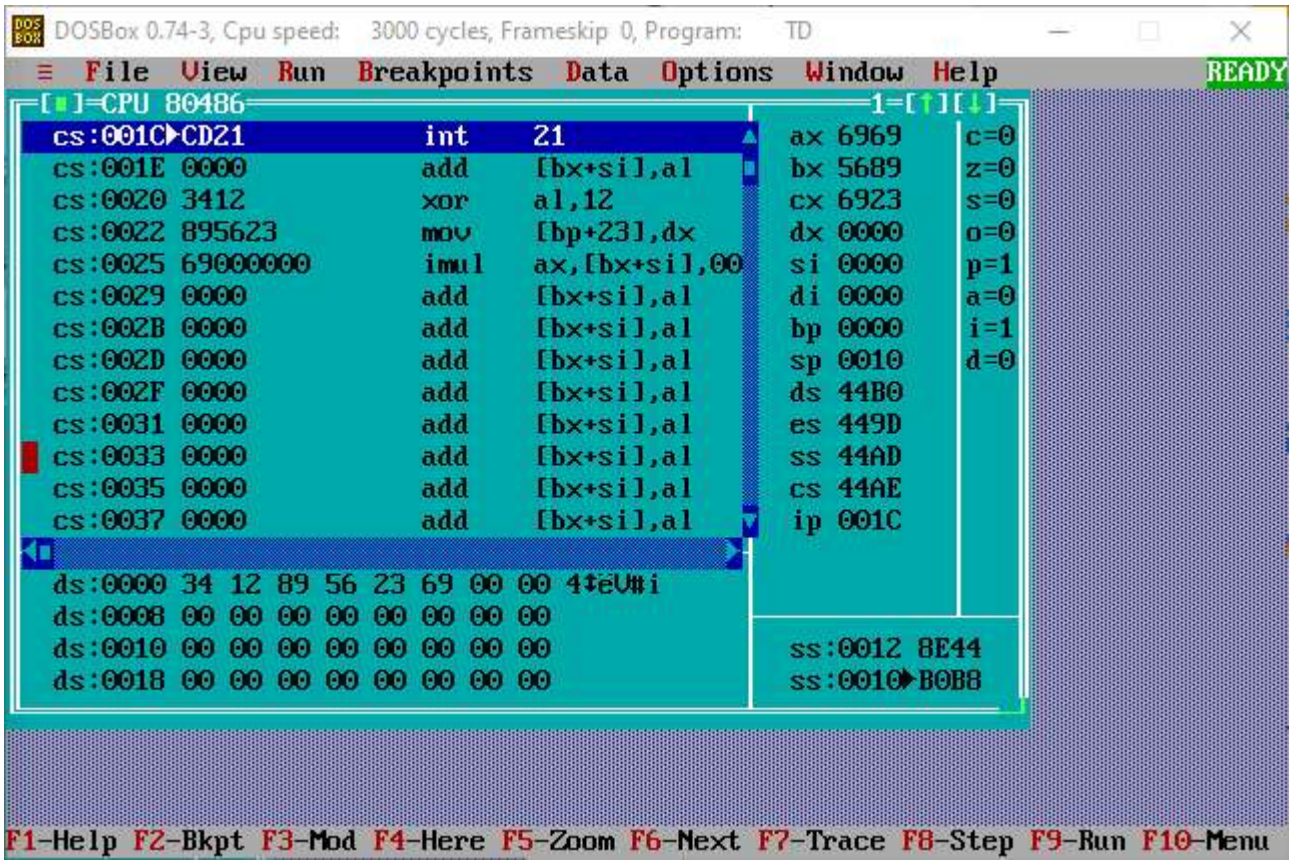
mov al, ah Move the adjusted high byte to AL register



daa Adjust AL after addition to handle decimal carry



mov ch, al Move the final adjusted high byte to CH register


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
File View Run Breakpoints Data Options Window Help
READY
CPU 80486
cs:001C CD21 int 21
cs:001E 0000 add [bx+si],al
cs:0020 3412 xor al,12
cs:0022 895623 mov [bp+23],dx
cs:0025 69000000 imul ax,[bx+si],00
cs:0029 0000 add [bx+si],al
cs:002B 0000 add [bx+si],al
cs:002D 0000 add [bx+si],al
cs:002F 0000 add [bx+si],al
cs:0031 0000 add [bx+si],al
cs:0033 0000 add [bx+si],al
cs:0035 0000 add [bx+si],al
cs:0037 0000 add [bx+si],al
ds:0000 34 12 89 56 23 69 00 00 4‡ëV#i
ds:0008 00 00 00 00 00 00 00 00
ds:0010 00 00 00 00 00 00 00 00
ds:0018 00 00 00 00 00 00 00 00
ax 6969 c=0
bx 5689 z=0
cx 6923 s=0
dx 0000 o=0
si 0000 p=1
di 0000 a=0
bp 0000 i=1
sp 0010 d=0
ds 44B0
es 449D
ss 44AD
cs 44AE
ip 001C
ss:0012 8E44
ss:0010 B0B8
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu