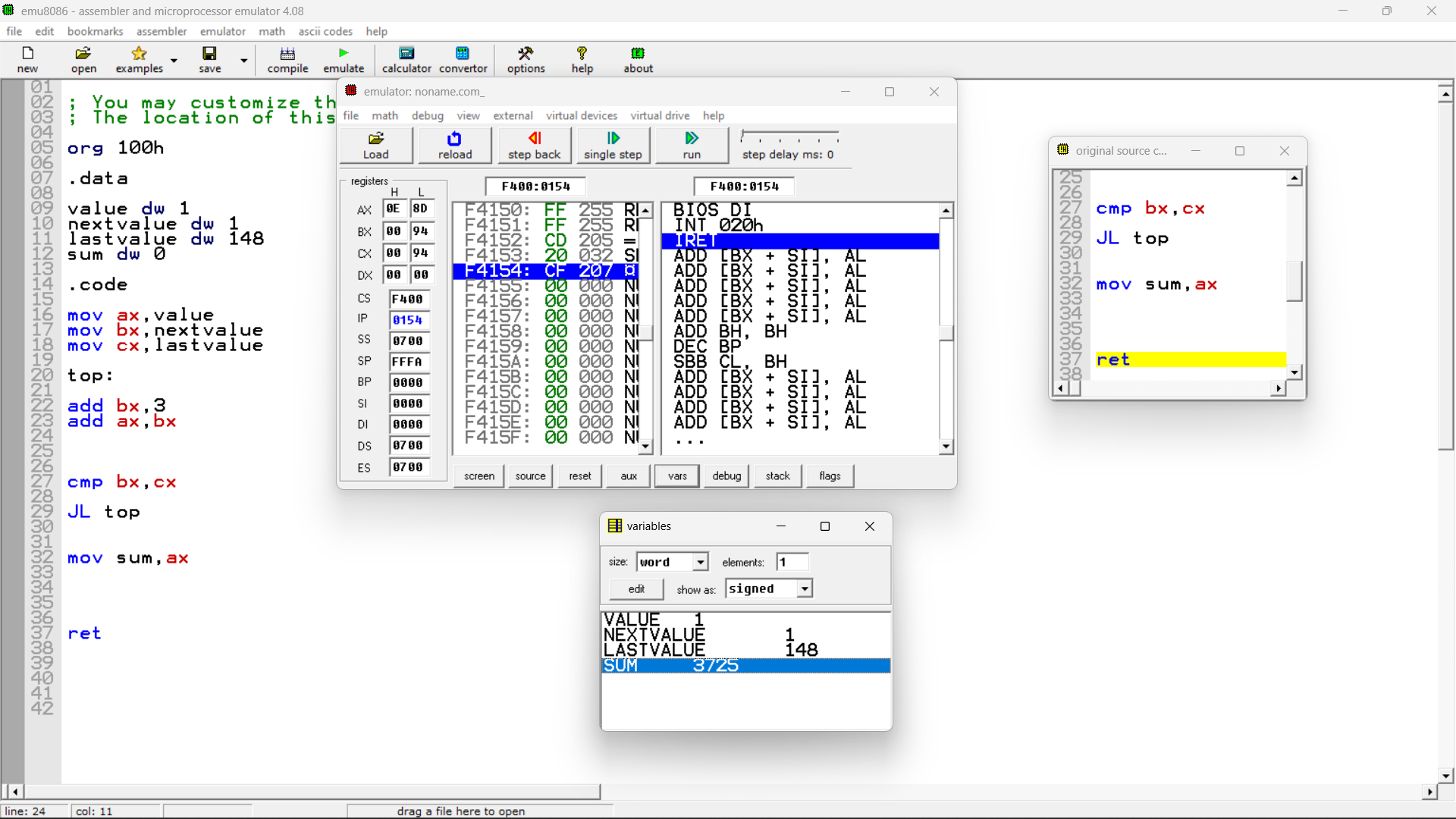
Lab 3

# Question 1

Calculate the sum using loops:

1+4+7...148



org 100h

.data

value dw 1

nextvalue dw 1

lastvalue dw 148

sum dw 0

.code

mov ax,value

mov bx,nextvalue

mov cx,lastvalue

top:

add bx,3

add ax,bx

cmp bx,cx

JL top

mov sum,ax

ret

# Question 2

Implement the following algo for division by repeated subtraction:

initialize quotient to 0

WHILE dividend >divisor DO

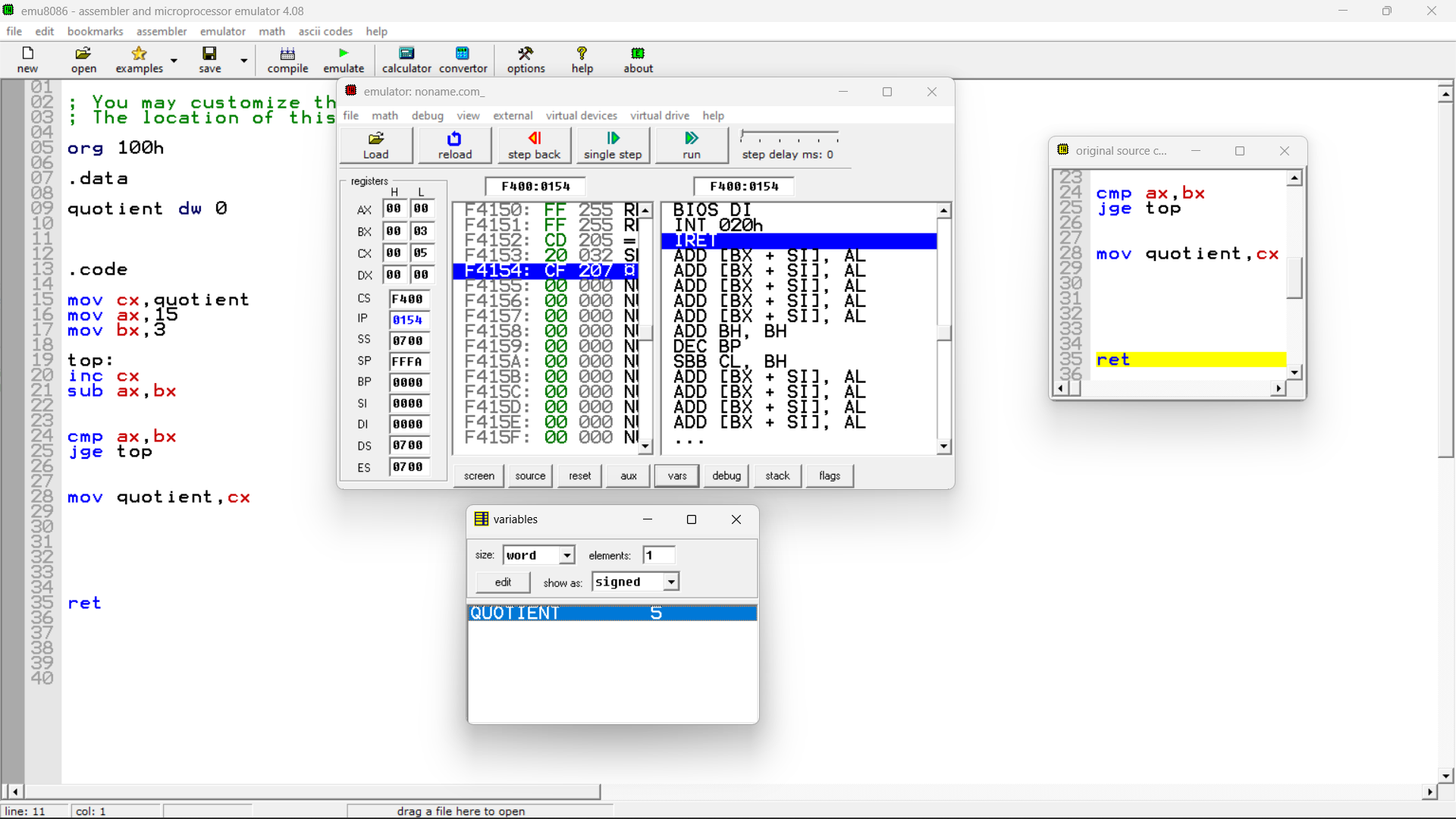
increment quotient

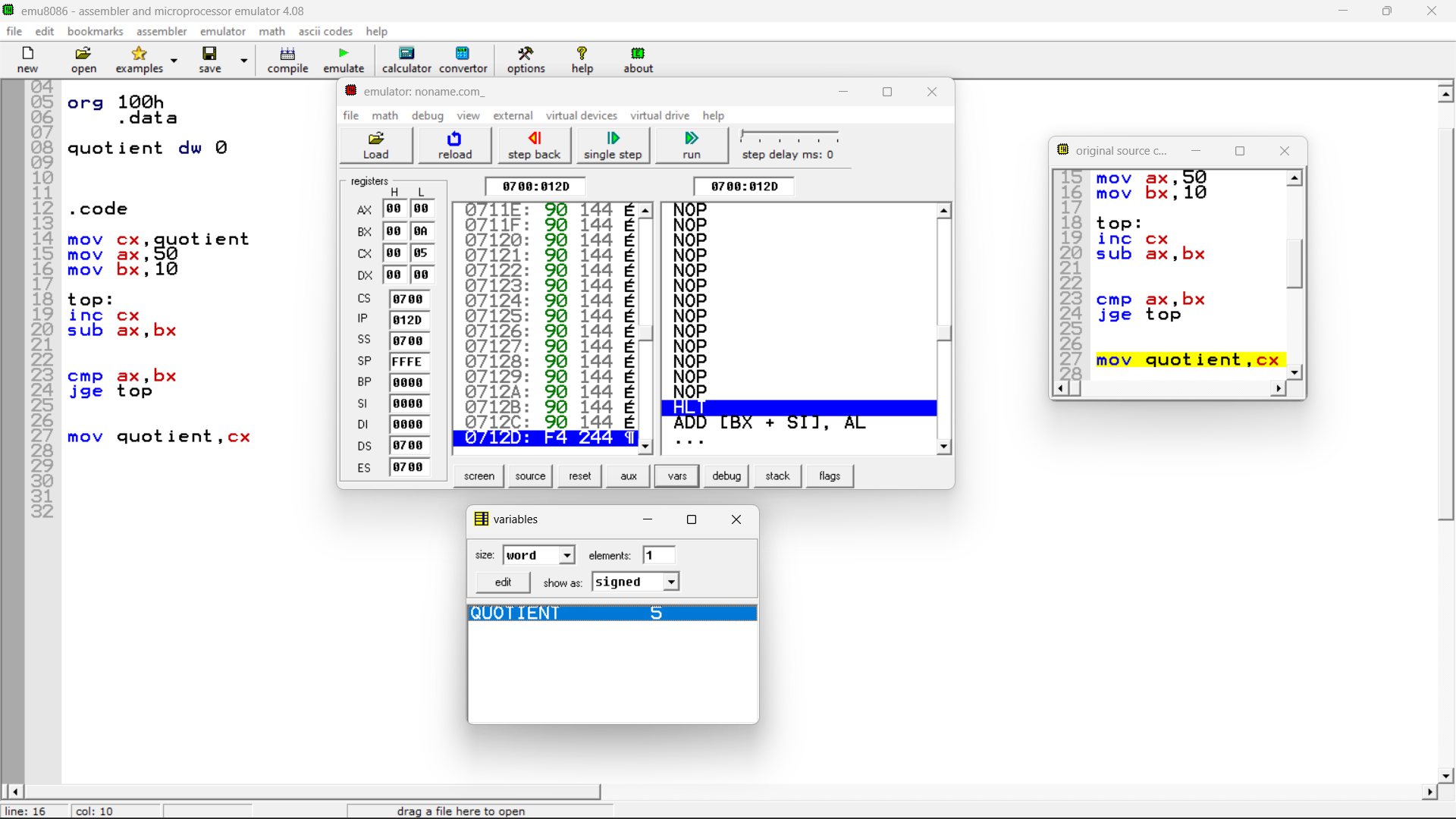
subtract divisor from dividend

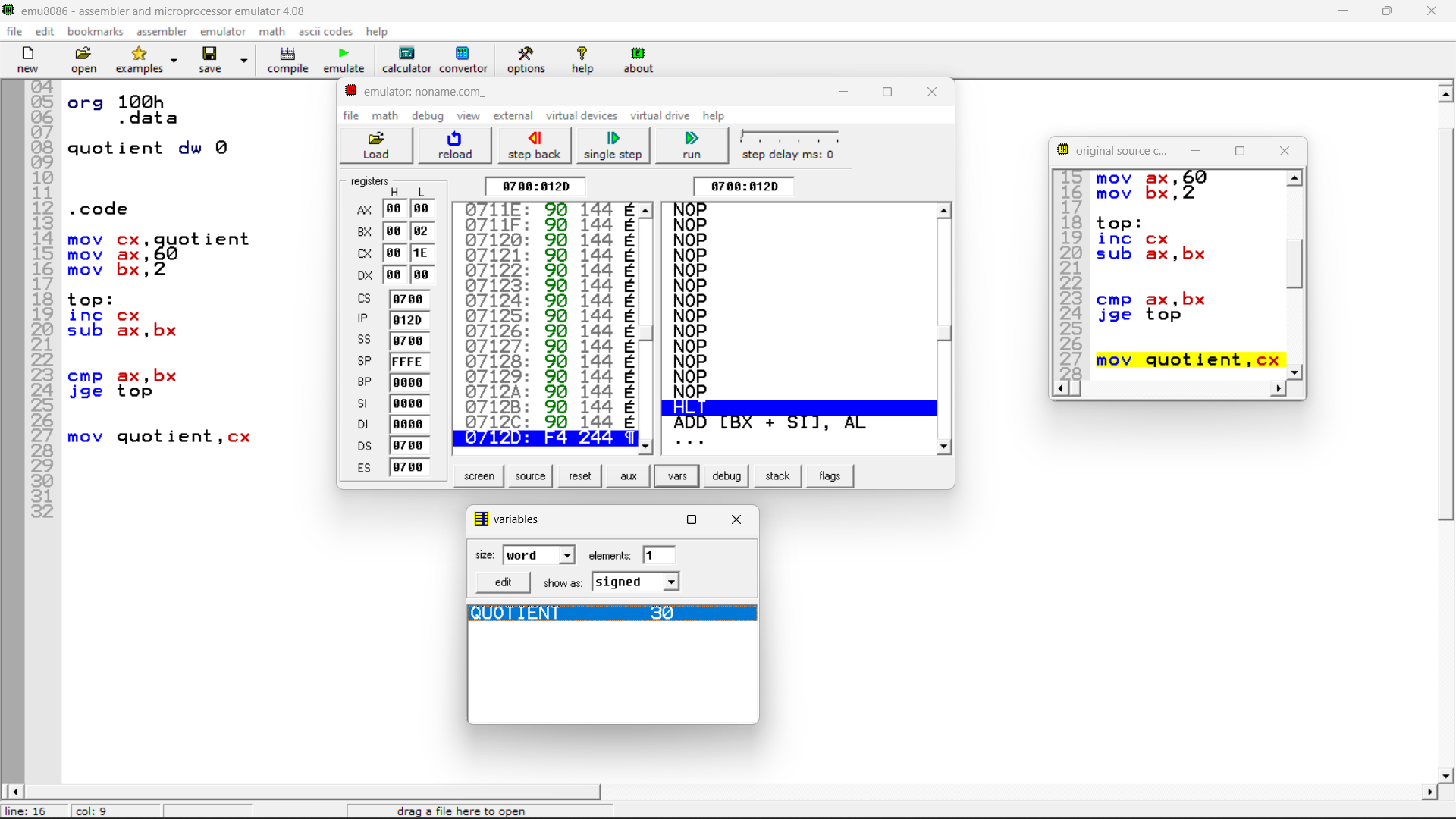
END\_ WHILE

Write a sequence of instructions to divide.AX by BX, and put the

quotient in CX.







.data

quotient dw 0

.code

mov cx,quotient

mov ax,15

mov bx,3

top:

inc cx

sub ax,bx

cmp ax,bx

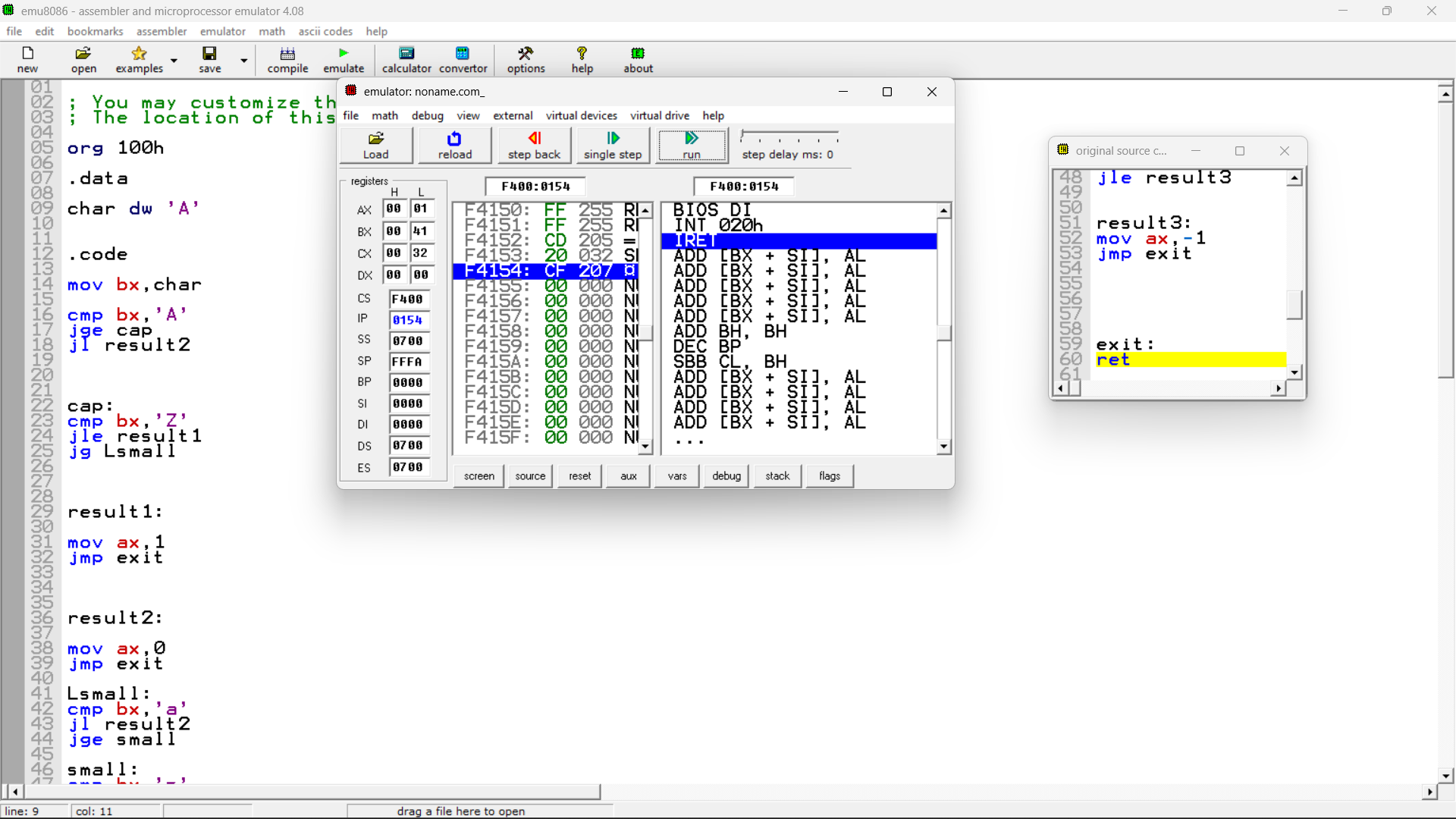
jge top

mov quotient,cx

ret

# Question 3

Read a character from CX and if it's upper case alphabet then put AX=1 , if it’s a lower case alphabet then put AX=-1 else put AX=0



org 100h

.data

char dw 'A'

.code

mov bx,char

cmp bx,'A'

jge cap

jl result2

cap:

cmp bx,'Z'

jle result1

jg Lsmall

result1:

mov ax,1

jmp exit

result2:

mov ax,0

jmp exit

Lsmall:

cmp bx,'a'

jl result2

jge small

small:

cmp bx,'z'

jle result3

result3:

mov ax,-1

jmp exit

exit:

ret