.stack 100h

.data

operand1 db ?

operand2 db ?

result db ?

operator db ?

msg\_divide\_by\_zero db "Infinity", 0Dh, 0Ah, "$"

msg\_invalid\_operator db "Invalid operator", 0Dh, 0Ah, "$"

negative\_sign db "-"

ascii db 2 DUP(?)

.code

main proc

mov ax, @data

mov ds, ax

mov ah, 01h

int 21h

sub al, '0'

mov operand1, al

mov ah, 01h

int 21h

mov operator, al

mov ah, 01h

int 21h

sub al, '0'

mov operand2, al

mov dl, '='

mov ah, 02h

int 21h

cmp operator, '+'

je addition

cmp operator, '-'

je subtraction

cmp operator, '\*'

je multiplication

cmp operator, '/'

je division

jmp invalid\_operator

addition:

mov al, operand1

add al, operand2

mov result, al

jmp print\_result

subtraction:

mov al, operand1

sub al, operand2

mov result, al

jmp print\_result

multiplication:

mov al, operand1

mul operand2

mov result, al

jmp print\_result

division:

cmp operand2, 0

je divide\_by\_zero

mov al, operand1

mov bl, operand2

mov ah, 0

div bl

mov result, al

jmp print\_result

divide\_by\_zero:

mov ax, 0

mov es, ax

mov al, 75h

mov bl, 4h

mul bl

mov bx, ax

mov si, offset [infinity\_msg]

mov es:[bx], si

add bx, 2

mov ax, cs

mov es:[bx], ax

int 75h

jmp quit\_program

print\_result:

cmp result, 0

jns print\_result\_positive

mov dl, negative\_sign

mov ah, 02h

int 21h

neg result

print\_result\_positive:

MOV AL, result

MOV AH, 0

MOV BH, 0

MOV BL, 10

DIV BL

ADD AL, '0'

MOV ascii[0], AL

ADD AH, '0'

MOV ascii[1], AH

MOV BYTE PTR [ascii+2], 0Dh

MOV BYTE PTR [ascii+3], 0Ah

MOV BYTE PTR [ascii+4], '$'

MOV AH, 09h

LEA DX, ascii

INT 21h

jmp quit\_program

invalid\_operator:

mov ah, 09h

lea dx, msg\_invalid\_operator

int 21h

quit\_program:

mov ah, 4Ch

int 21h

main endp

infinity\_msg PROC

mov ah, 09h

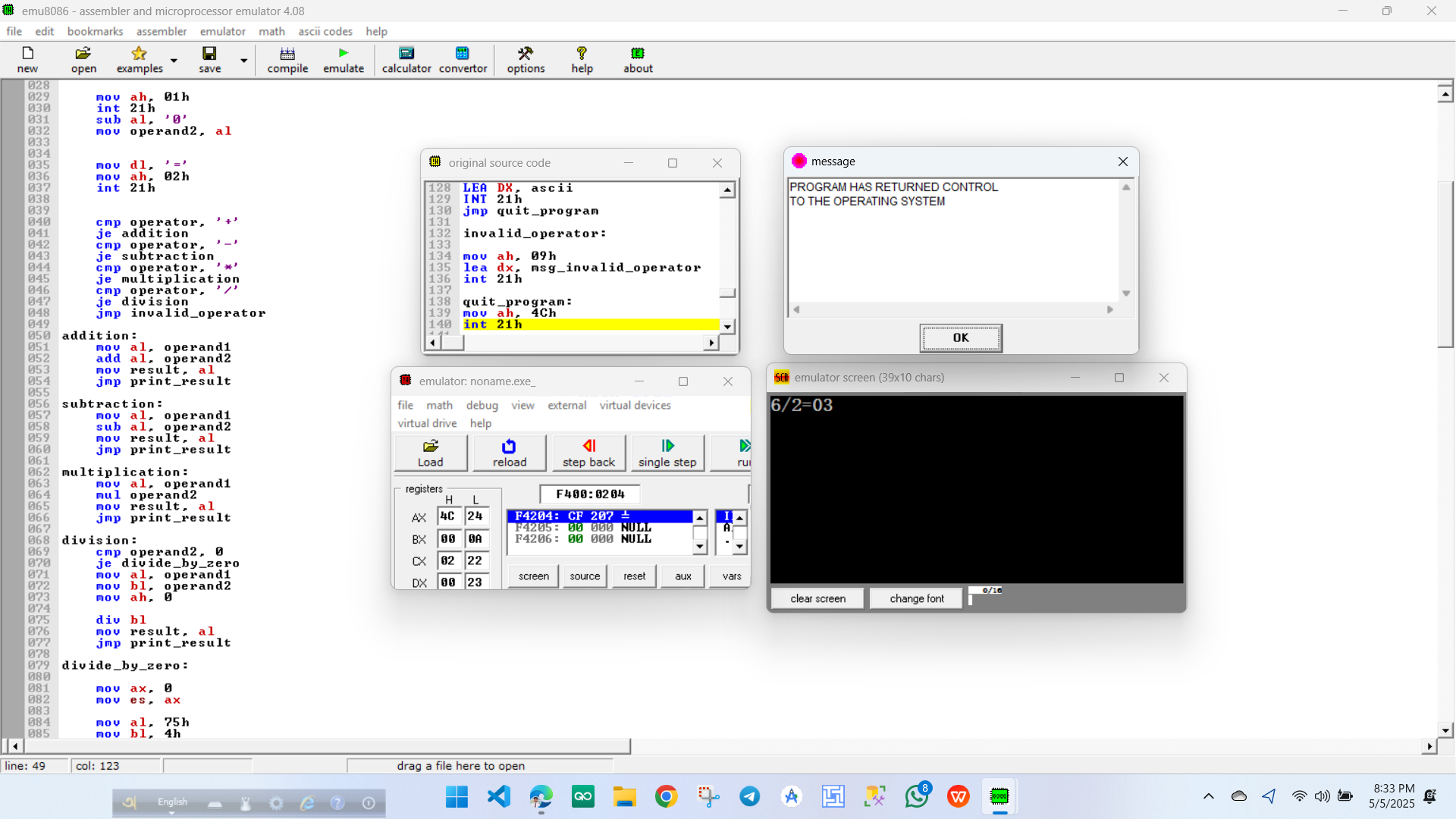
lea dx, msg\_divide\_by\_zero

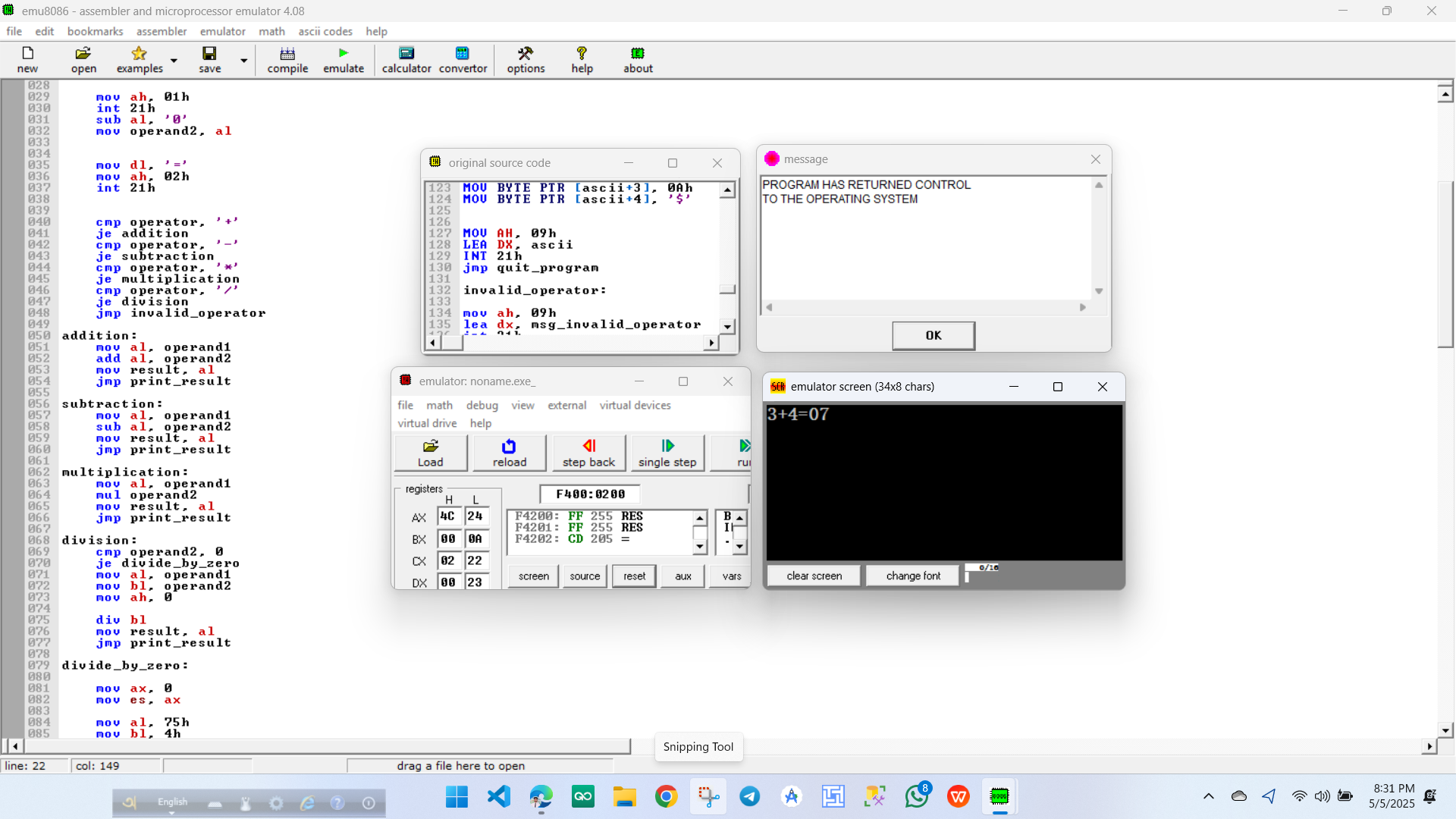
int 21h

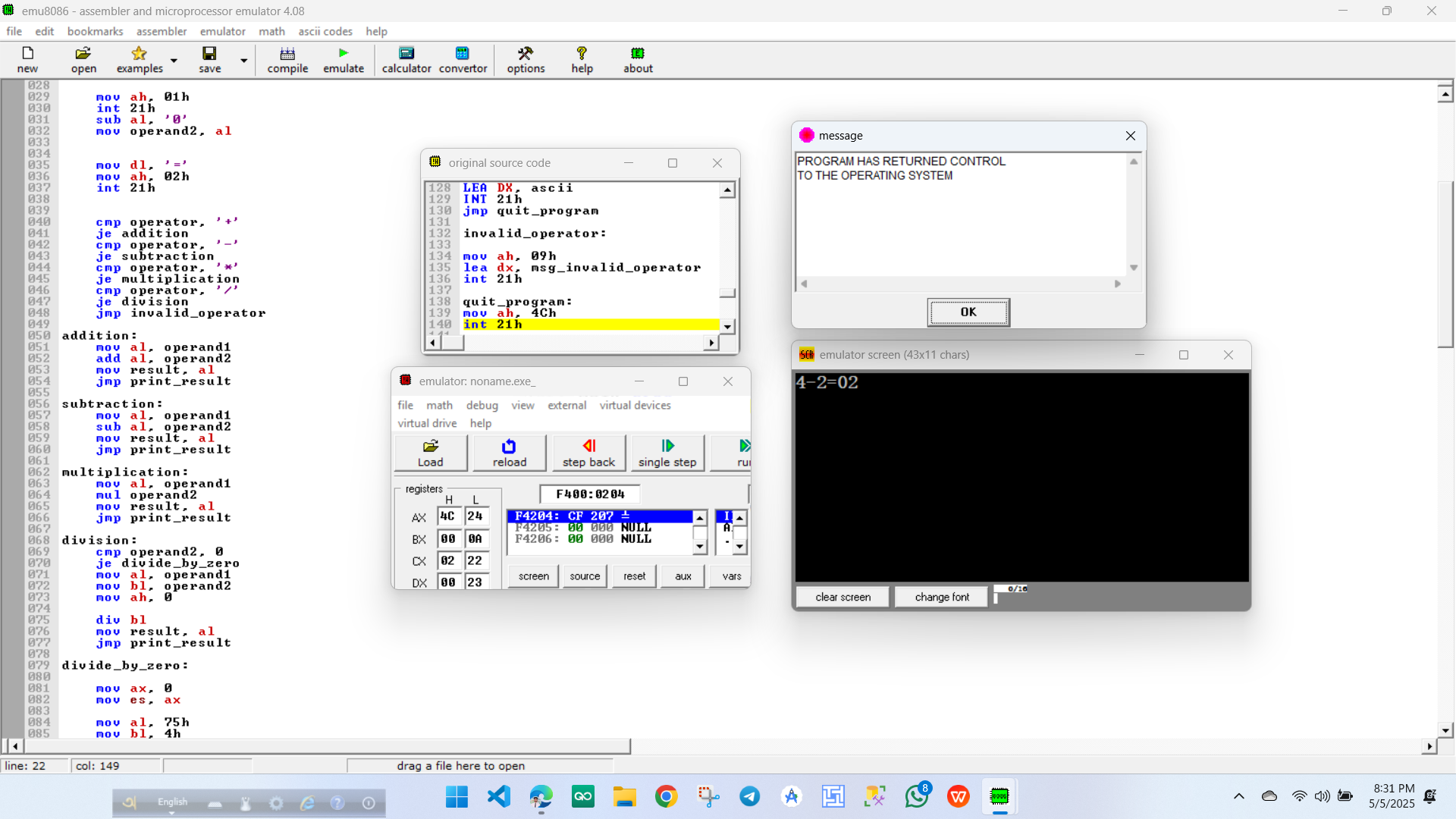
IRET

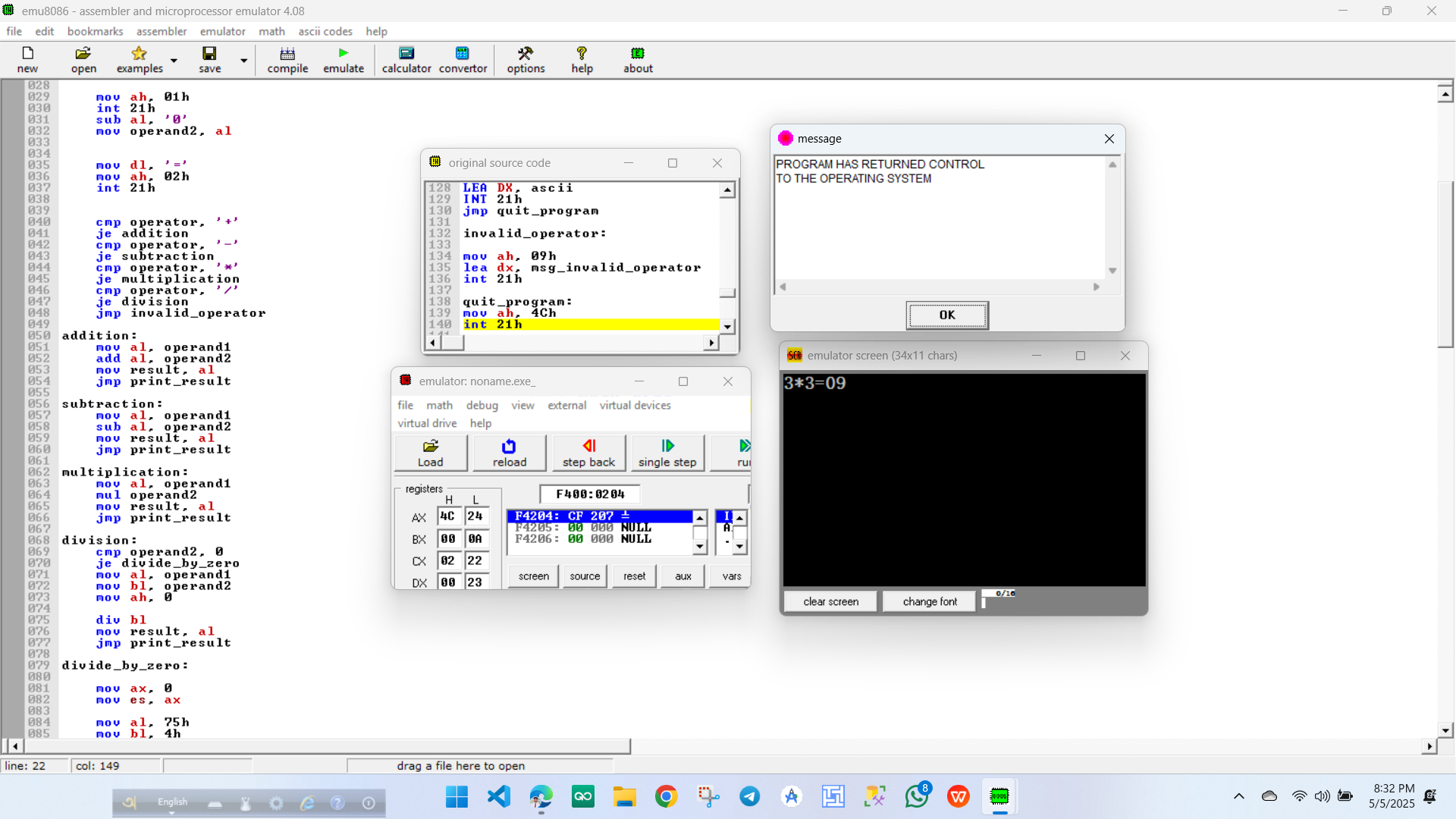
infinity\_msg ENDP

end main







****