**8-BIT DIVISION**

**EXP NO: 4**

**AIM:** To write an assembly language program to implement 8-bit division using 8085 processor.

**ALGORITHM:**

1. Start the program by loading a register pair with the address of memory location.
2. Move the data to a register.
3. Get the second data and load it into the accumulator.
4. Subtract the two register contents.
5. Increment the value of the carry.
6. Check whether the repeated subtraction is over.
7. Store the value of quotient and the reminder in the memory location.
8. Halt.

**PROGRAM:**

LDA 8501

MOV B, A

LDA 8500

MVI C,00

LOOP:CMP B

JC LOOP1

SUB B

INR C

JMP LOOP

STA 8503

DCR C

MOV A, C

LOOP1: STA 8502

RST 1

**INPUT:**

; Program: 8-bit Division in 8086

; Emulator: emu8086, MASM, TASM

data segment

num1 db 37 ; Dividend

num2 db 5 ; Divisor

quotient db ?

remainder db ?

data ends

code segment

assume cs:code, ds:data

start:

mov ax, data

mov ds, ax

; Load dividend in AL, clear AH

mov al, num1

mov ah, 0

; Divide AX by num2 → Quotient in AL, Remainder in AH

mov bl, num2

div bl

; Store results

mov quotient, al

mov remainder, ah

; ----- Print Quotient -----

mov al, quotient

mov ah, 0

mov bl, 10

div bl ; AX ÷ 10 → AL=quotient, AH=remainder

; Print tens digit

add al, 30h

mov dl, al

mov ah, 02h

int 21h

; Print ones digit

add ah, 30h

mov dl, ah

mov ah, 02h

int 21h

; Print space

mov dl, ' '

mov ah, 02h

int 21h

; ----- Print Remainder -----

mov al, remainder

mov ah, 0

mov bl, 10

div bl

; Print tens digit

add al, 30h

mov dl, al

mov ah, 02h

int 21h

; Print ones digit

add ah, 30h

mov dl, ah

mov ah, 02h

int 21h

; Exit program

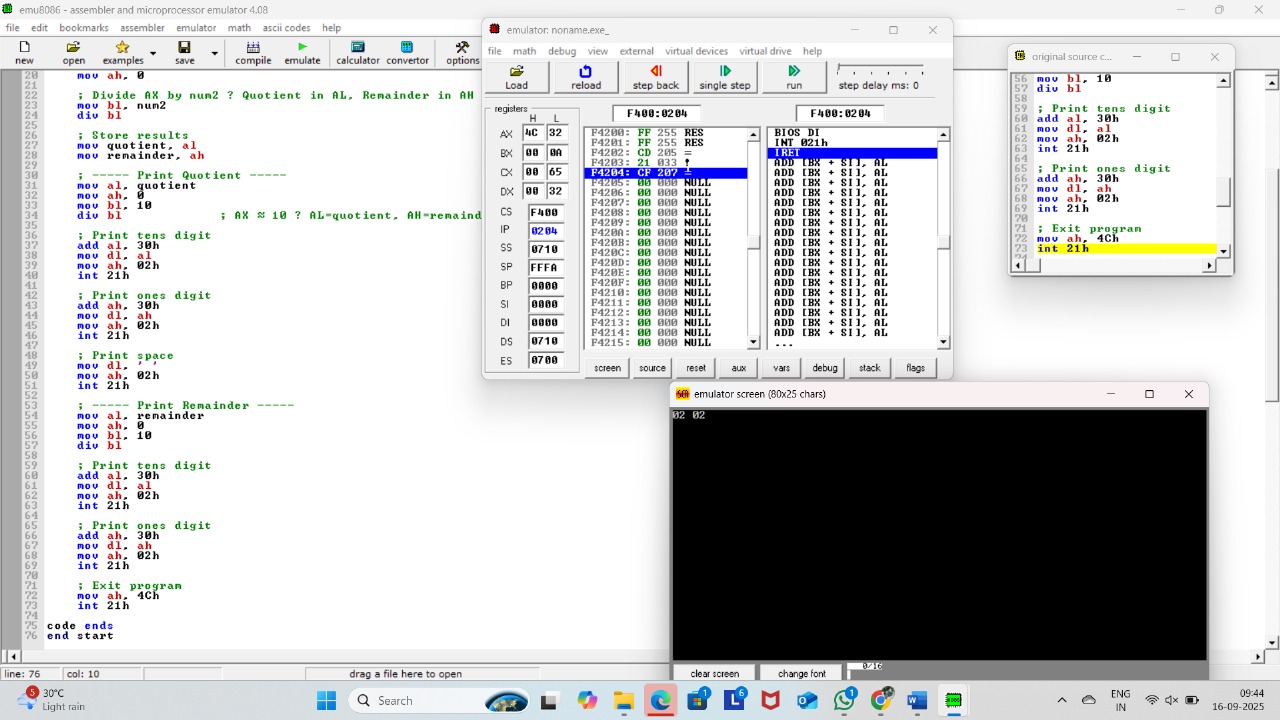
mov ah, 4Ch

int 21h

code ends

end start

**OUTPUT:**



**RESULT:** Thus the program was executed successfully using 8085 processor simulator.