***AIDS MICROPROCESSOR LAB S21 BATCH (2023-24)***

***Experiment 6(a) Title: Assembly language programming to find the GCD of two numbers***

***Name of student: Meet Raut Class Roll Number: 2201084***

***Date of Performance: 18/03/2024***

***Batch: S2-1 Timing: 3:00-5:00 Date of Submission: 18/03/2024***

***Assembly language code***

DATA\_SEG SEGMENT

NUM1 DW 85

NUM2 DW 119

GCD DW ?

DATA\_SEG ENDS

CODE\_SEG SEGMENT

ASSUME CS:CODE\_SEG,DS:DATA\_SEG

START: MOV AX,DATA\_SEG

MOV DS,AX

MOV AX, NUM1  *#move first number to AX*

MOV BX, NUM2  *#move second number to BX*

CMP AX, BX #*compare the two numbers*

JAE AGAIN

XCHG AX, BX #*move the larger number into AX*

AGAIN:

MOV DX, 00 #*initialise DX with 0*

DIV BX #*divide DX:AX by BX*

CMP DX, 0 #*verify if the remainder is zero*

JZ EXIT  *#jump if the remainder is zero*

MOV AX, BX #*move the quotient into AX*

MOV BX, DX #*move remainder into BX*

JMP AGAIN  *#repeat until GCD is found*

EXIT:

MOV GCD, BX

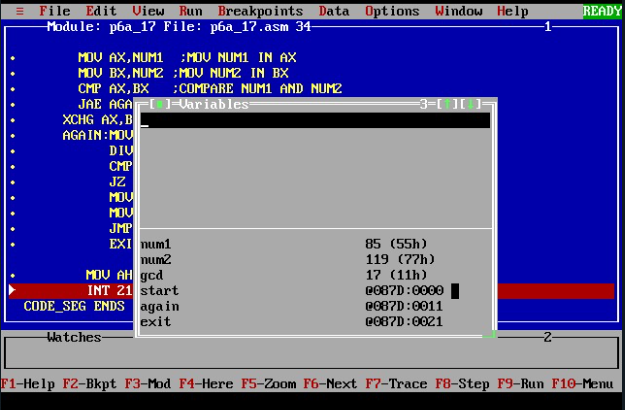
MOV AH, 4CH *;exit to DOS*

INT 21H

CODE\_SEG ENDS

END START

*Result:*

**

***Experiment 6(b) Title: Assembly language programming to find the LCM of two numbers using software tool TASM 1.4***

***Name of student: Meet Raut Class Roll Number: 2201084***

***Date of Performance: 18/03/2024***

***Batch: S2-1 Timing: 3:00-5:00 Date of Submission: 18/03/2024***

***Assembly language code***

*DATA\_SEG SEGMENT*

*NUM1 DW 85*

*NUM2 DW 119*

*GCD DW 00*

*LCM DW 00*

*DATA\_SEG ENDS*

*CODE\_SEG SEGMENT*

*ASSUME CS:CODE\_SEG, DS:DATA\_SEG*

*START:*

*MOV AX,DATA\_SEG #initialisation*

*MOV DS,AX*

*MOV AX, NUM1 # move first number to AX*

*MOV BX, NUM2 # move second number to BX*

*CMP AX, BX # compare the two numbers*

*JAE AGAIN*

*XCHG AX, BX # move the larger number into AX*

*AGAIN:*

*MOV DX, 00 #initialise DX with 0*

*DIV BX # divide DX:AX by BX*

*CMP DX, 0 # verify if the remainder is zero*

*JZ EXIT # jump if the remainder is zero*

*MOV AX, BX # move the quotient into AX*

*MOV BX, DX # move remainder into BX*

*JMP AGAIN # repeat until GCD is found*

*EXIT:*

*MOV GCD, BX #move result into GCD*

*MOV AX, NUM1 #move first number to AX*

*MOV BX, NUM2 #move second number to BX*

*MUL BX # multiply*

*MOV CX, GCD #move GCD into CX*

*MOV LCM, AX #move the result into LCM*

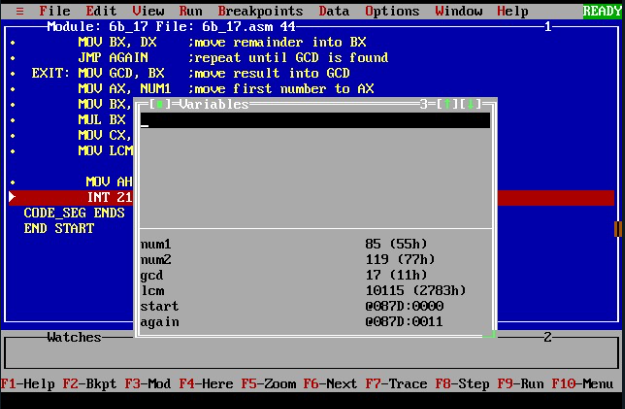
*MOV AH, 4CH #exit to DOS*

*INT 21H*

*CODE\_SEG ENDS*

*END START*

*Result:*

**

*CONCLUSION: LO 2, LO 3 mapped.*

***---------------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-----------------------------------***