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

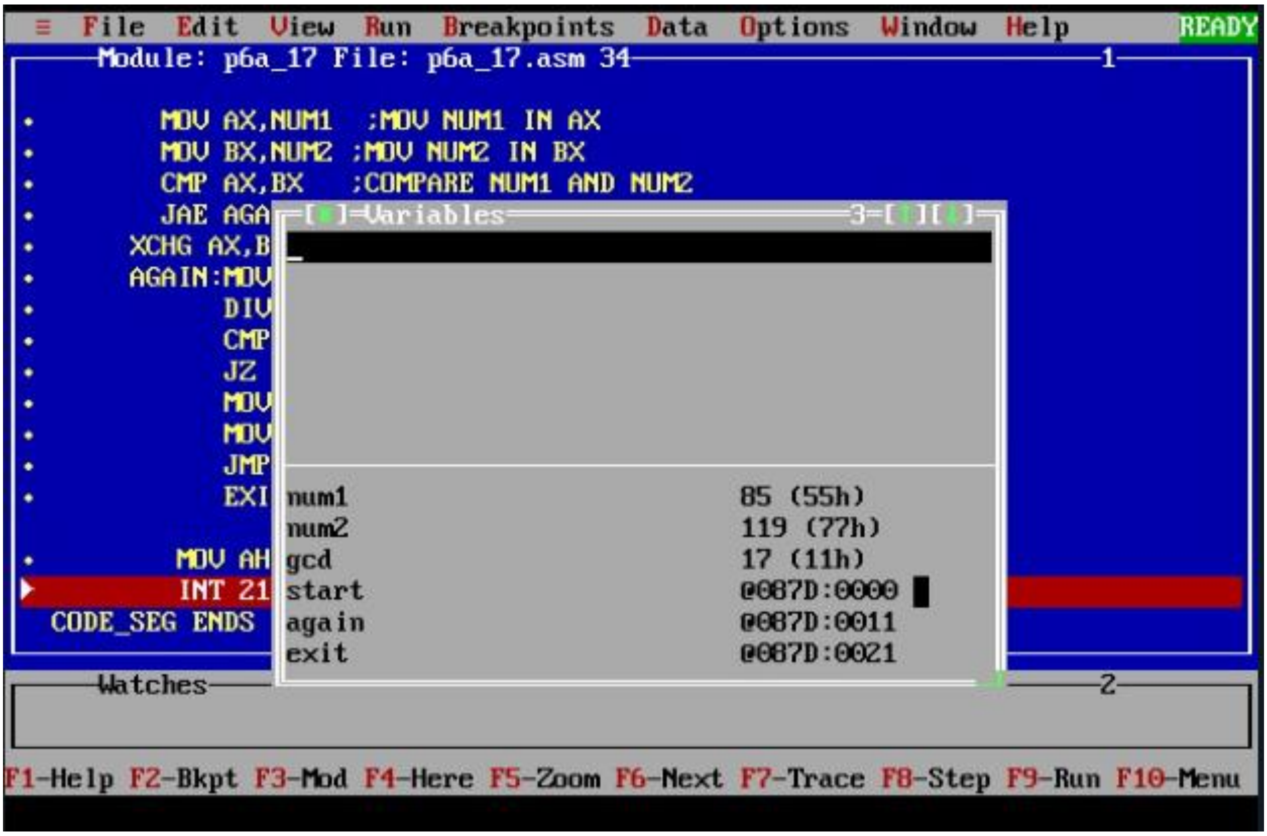
Batch: S2-1   Timing: 3:00-5:00   Date of Submission: 18/03/2024

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:



≡ File Edit View Run Breakpoints Data Options Window Help READY
Module: p6a_17 File: p6a_17.asm 34
1
[■]=Variables
3=[↑][↓]
Watches
2
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```
•         MOV AX,NUM1   ;MOV NUM1 IN AX
•         MOV BX,NUM2 ;MOV NUM2 IN BX
•         CMP AX,BX   ;COMPARE NUM1 AND NUM2
•         JAE AGA
•       XCHG AX,B
•       AGAIN:MOV
•             DIV
•             CMP
•             JZ
•             MOV
•             MOV
•             JMP
•             EXI

•          MOV AH
►          INT 21
    CODE_SEG ENDS
```

```
num1                          85 (55h)
num2                          119 (77h)
gcd                           17 (11h)
start                         @087D:0000
again                         @087D:0011
exit                          @087D:0021
```

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

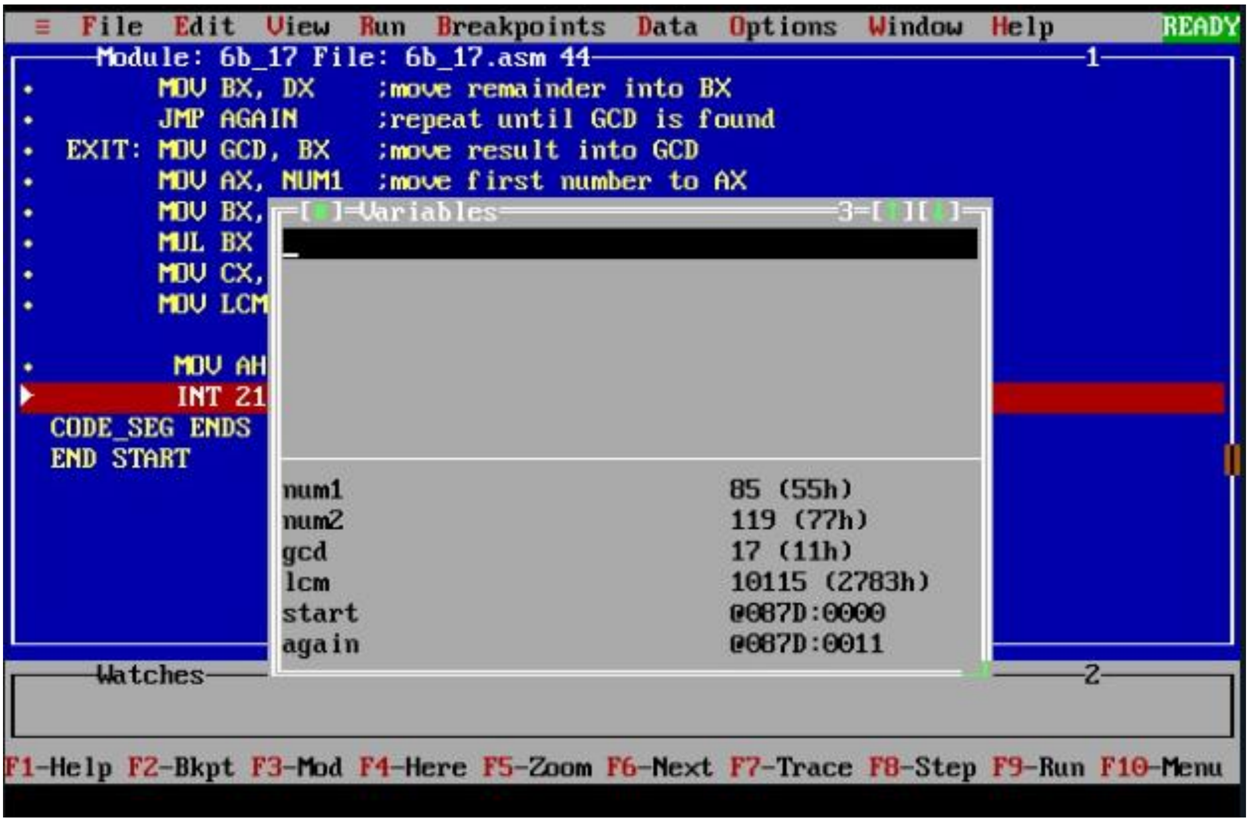
Date of Performance: 18/03/2024

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:



≡ File Edit View Run Breakpoints Data Options Window Help READY
1
[■]=Variables 3=[↑][↓]
Watches
2
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

```
Module: 6b_17 File: 6b_17.asm 44
•         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,
•         MUL BX
•         MOV CX,
•         MOV LCM

•          MOV AH
▶          INT 21
   CODE_SEG ENDS
   END START
```

```
num1                        85 (55h)
num2                        119 (77h)
gcd                         17 (11h)
lcm                         10115 (2783h)
start                       @087D:0000
again                       @087D:0011
```

CONCLUSION: LO 2, LO 3 mapped.

-----------------------------------****************************-----------------------------------