

Experiment 7

Name:Pratik Chavan

Div/Batch:A/A1 Roll No.07

.MODEL SMALL

.STACK 100H

.DATA

NUM1 DW 36 ; First number

NUM2 DW 24 ; Second number

GCD_RESULT DW ? ; Store GCD result

LCM_RESULT DW ? ; Store LCM result

MSG_GCD DB 'GCD: \$'

MSG_LCM DB ' LCM: \$'

NEWLINE DB 0DH, 0AH, '\$' ; New line for output formatting

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV AX, NUM1

MOV BX, NUM2

CALL GCD ; Compute GCD

MOV GCD_RESULT, AX

MOV AX, NUM1

MUL BX ; AX = NUM1 * NUM2

DIV GCD_RESULT ; AX = LCM (Product / GCD)

MOV LCM_RESULT, AX

MOV DX, OFFSET MSG_GCD

MOV AH, 09H

INT 21H ; Print "GCD: "

MOV AX, GCD_RESULT

CALL PRINT_NUM ; Print GCD

MOV DX, OFFSET NEWLINE

MOV AH, 09H

INT 21H ; Print new line

MOV DX, OFFSET MSG_LCM

MOV AH, 09H

INT 21H ; Print " LCM: "

MOV AX, LCM_RESULT

CALL PRINT_NUM ; Print LCM

MOV AH, 4CH

INT 21H ; Exit program

MAIN ENDP

; GCD Procedure (Euclidean Algorithm)

GCD PROC

CMP BX, 0

JE END_GCD

GCD_LOOP:

MOV DX, 0

DIV BX ; AX = AX / BX, Remainder in DX

MOV AX, BX

MOV BX, DX

CMP BX, 0

JNE GCD_LOOP

END_GCD:

RET

GCD ENDP

; Print Number Procedure

PRINT_NUM PROC

MOV CX, 0

NEXT_DIGIT:

MOV DX, 0

MOV BX, 10

DIV BX ; AX / 10 → Quotient in AX, Remainder in DX

PUSH DX

INC CX

TEST AX, AX

JNZ NEXT_DIGIT

PRINT_LOOP:

POP DX

ADD DL, '0'

MOV AH, 02H

INT 21H

LOOP PRINT_LOOP

RET

PRINT_NUM ENDP

END MAIN

OUTPUT:

```

CPU 80486 ds:0004 = 0000 3=[↑][↓]
cs:0003 8ED8      ♦ MOV DS, AX      ax 000C  c=0
cs:0005 A10000    ♦ MOV AX, NUM1    bx 0000  z=1
cs:0008 8B1E0200  ♦ MOV BX, NUM2    cx 0000  s=0
cs:000C E83400    ♦ CALL GCD ; Compute GCD dx 0000  o=0
cs:000F A30400    ♦ MOV GCD_RESULT, AX si 0000  p=1
cs:0012 A10000    ♦ MOV AX, NUM1    di 0000  a=0
cs:0015 F7E3      ♦ MUL BX ; AX = NUM1 * N bp 0000  i=1
cs:0017 F7360400  ♦ DIV GCD_RESULT ; AX = sp 0100  d=0
cs:001B A30600    ♦ MOV LCM_RESULT, AX ds 0884
cs:001E BA0800    ♦ MOV DX, OFFSET MSG_GCD es 086C
cs:0021 B409      ♦ MOV AH, 09H    ss 0886
cs:0023 CD21      ♦ INT 21H ; Print "GCD: cs 087C
cs:0025 A10400    ♦ MOV AX, GCD_RESULT ip 000F

es:0000 CD 20 7D 9D 00 EA FF FF = }≠ Ω
es:0008 AD DE 32 0B C5 05 6B 07 i 2δ+ok•
es:0010 15 03 28 08 15 03 93 01 Š♥(Š♥ô
es:0018 01 01 01 00 02 04 05 06 000 0+•••

ss:0102 0403
ss:0100 52FB
```