Experiment 9

Title: Write a program to find LCM and GCD of two numbers

8940 Warren Fernandes SE-COMPS-B

8086 Program to find LCM

data segment
 num dw 05,04
 lcm dw 2 dup(0)
data ends
code segment
 assume cs:code,ds:data
 start: mov ax,data
 mov ds,ax
 mov dx,0h
 mov ax,num
 mov bx,num+2
 up: push ax
 push dx
 div bx
 cmp dx,0
 je exit
 pop dx
 pop ax
 add ax,num
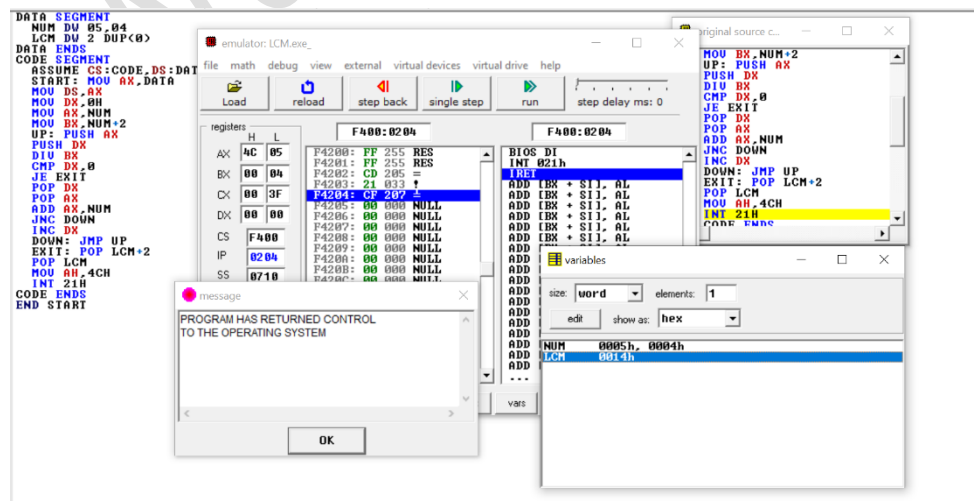 jnc down
 inc dx
 down: jmp up
 exit: pop lcm+2
 pop lcm
 mov ah,4ch
 int 21h
code ends
end start

8086 Program to find GCD

```
data SEGMENT
   n1   DB   8
   n2   DB   22
   rez  DB   ?
ENDS
stack SEGMENT
   DB   128   dup(0)
ENDS
code SEGMENT
   nzd PROC
      POP CX
      POP BX
      ciklus:
         CMP BL, 0d
         JE done
         MOV AL, BH
         MOV AH, 0d
         DIV BL
         MOV BH, BL
         MOV BL, AH
         JMP ciklus
      done:
         PUSH BX
         PUSH CX
      RET
   nzd ENDP

   start:
      MOV AX, data
      MOV DS, AX
      MOV ES, AX
      MOV AX, stack
      MOV SS, AX
      MOV BH, [n1]
      MOV BL, [n2]
      PUSH BX
      CALL nzd
      POP BX
      MOV [rez], BH
   exit:
      MOV AH, 4ch
      INT 21h
ENDS
END start
```

```
;MENT
    DB        8
    DB        22
    DB        ?

;GMENT
    128       dup(0)

;MENT
 PROC
 POP  CX
 POP  BX
ciklus:
    CMP  BL, 0d
    JE   done
    MOV  AL, BH
    MOV  AH, 0d
    DIV  BL
    MOV  BH, BL
    MOV  BL, AH
    JMP  ciklus
done:
    PUSH BX
    PUSH CX

 RET
 ENDP
·t:
 MOV  AX, data
 MOV  DS, AX
 MOV  ES, AX
 MOV  AX, stac
 MOV  SS, AX
 MOV  BH, [n1]
 MOV  BL, [n2]
 PUSH BX
 CALL nzd
 POP  BX
 MOV  [rez], B

::
 MOV  AH, 4ch
 INT  21h

·t
```

emulator: GCD.exe

file   math   debug   view   external   virtual devices   virtual drive   help

Load   reload   step back   single step   run   step delay ms: 0

registers

	H	L
AX	4C	03
BX	02	00
CX	00	2E
DX	00	00
CS	F400	
IP	0204	
SS	0711	
SP	007A	
BP	0000	
SI	0000	

0710:0000   F400:0204

```
07100: 08 008 BACK
07101: 16 022 ▬
07102: 02 002 ☻
07103: 00 000 NULL
07104: 00 000 NULL
07105: 00 000 NULL
07106: 00 000 NULL
07107: 00 000 NULL
07108: 00 000 NULL
07109: 00 000 NULL
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: 00 000 NULL
0710D: 00 000 NULL
0710E: 00 000 NULL
0710F: 00 000 NULL
07110: 00 000 NULL
07111: 00 000 NULL
```

```
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
AI
AI
AI
AI
AI
AI
AI
AI
AI
AI
AI
```

message

PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM

OK

original source c...

```
nzd ENDP
start:
 MOV  AX, data
 MOV  DS, AX
 MOV  ES, AX
 MOV  AX, stack
 MOV  SS, AX
 MOV  BH, [n1]
 MOV  BL, [n2]
 PUSH BX
 CALL nzd
 POP  BX
 MOV  [rez], BH
exit:
 MOV  AH, 4ch
 INT  21h
 ENDS
```

variables

size: byte ▾   elements: 1

edit   show as: hex ▾

N1	08h
N2	16h
REZ	02h