

**Name: Vanani Prince**

**Roll No: CE147**

**ID: 19CEUES108**

**1) Write an assembly program to find the GCD and LCM of two numbers.**

LXI H,8000H

MOV A,M

MOV D,M

INX H

MOV B,M

MOV E,M

L1: CMP B

JZ GCD

JC L2

SUB B

JMP L1

L2 : MOV C,B

MOV B,A

MOV A,C

JMP L1

GCD: MOV C,A

STA 8002H

MOV A,D

MOV B,E

MVI A,00

L3: ADD B

DCR D

JNZ L3

MOV B,A

STA 8003H

MVI D,00

MOV A,B

NEXT: CMP C

JC LOOP

SUB C

INR D

JMP NEXT

LOOP: STA 8004H

MOV A,D

STA 8005H

HLT

The screenshot shows the 8086 Assembler and Memory Editor interface. The Assembler window displays the assembly code being entered, and the Memory Editor window shows the memory contents.

Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
0000		LXI H,8000	21	3	3	10
0001			00			
0002			80			
0003		MOV A,M	7E	1	2	7
0004		MOV D,M	56	1	2	7
0005		INX H	23	1	1	6
0006		MOV B,M	46	1	2	7
0007		MOV E,M	5E	1	2	7
0008	L1	CMP B	BB	1	1	4
0009		JF C,C0	CA	3	3	10
000A			19			
000B			00			
000C		JC L2	DA	3	3	10
000D			13			
000E			00			
000F		SUB B	90	1	1	4
0010		JMP L1	C3	3	3	10
0011			0B			
0012			00			

Memory Editor window shows the memory contents:

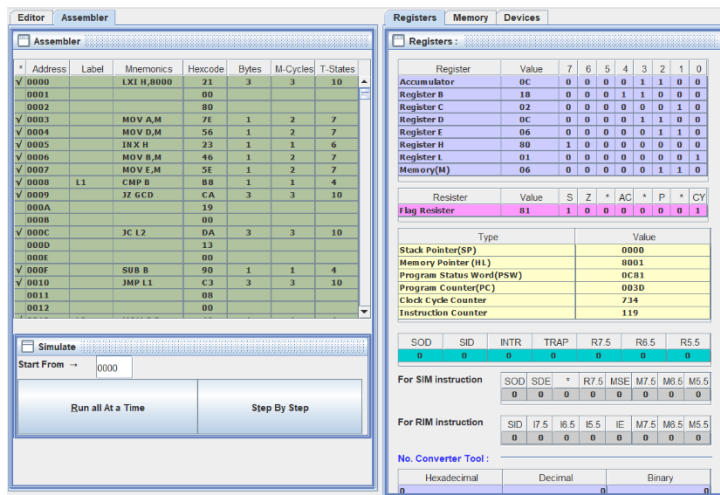
Memory Address	Value
8000	04
8001	06

The screenshot shows the 8086 Assembler and Memory Editor interface. The Assembler window displays the assembly code being entered, and the Memory Editor window shows the memory contents.

Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
0000		LXI H,8000	21	3	3	10
0001			00			
0002			80			
0003		MOV A,M	7E	1	2	7
0004		MOV D,M	56	1	2	7
0005		INX H	23	1	1	6
0006		MOV B,M	46	1	2	7
0007		MOV E,M	5E	1	2	7
0008	L1	CMP B	BB	1	1	4
0009		JF C,C0	CA	3	3	10
000A			19			
000B			00			
000C		JC L2	DA	3	3	10
000D			13			
000E			00			
000F		SUB B	90	1	1	4
0010		JMP L1	C3	3	3	10
0011			0B			
0012			00			

Memory Editor window shows the memory contents:

Memory Address	Value
0031	91
0032	14
0033	C3
0034	20
0036	32
0037	04
0038	00
0039	7A
003A	32
003B	05
003C	80
003D	76
8000	04
8001	06
8002	02
8003	18
8005	0C



## 2) Write an assembly language program to Sort an array of 10 numbers.

```
MVI C,10
```

```
DCR C
```

```
REPEAT:    MOV D,C
```

```
LXI H,4201
```

```
LOOP:     MOV A,M
```

```
INX H
```

```
CMP M
```

```
JC SKIP
```

```
MOV B,M
```

```
MOV M,A
```

```
DCX H
```

```
MOV M,B
```

```
INX H
```

```
SKIP:     DCR D
```

```
JNZ LOOP
```

```
DCR C
```

```
JNZ REPEAT
```

```
HLT
```

EditorAssembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		MVI C,10	06	2	2	7
0001			10			
✓ 0002		DCR C	0D	1	1	4
✓ 0003	REPEAT	MOV D,C	51	1	1	4
✓ 0004		LXI H,4201	21	3	3	10
0005			01			
0006			42			
✓ 0007	LOOP	MOV A,M	7E	1	2	7
✓ 0008		INX H	23	1	1	6
✓ 0009		CHP M	8E	1	2	7
✓ 000A		JC SKIP	DA	3	3	10
000B			12			
000C			00			
✓ 000D		MOV B,M	46	1	2	7
✓ 000E		MOV M,A	77	1	2	7
✓ 000F		DCX H	2B	1	1	6
✓ 0010		MOV M,B	70	1	2	7
✓ 0011		INX H	23	1	1	6
✓ 0012	SKIP	DCR D	15	1	1	4

Simulate

Start From → 0000

Run all At a TimeStep By Step

RegistersMemoryDevices

Memory Editor

Memory Range: 0000 → FFFF

Memory Address	Value
4201	0A
4202	09
4203	08
4204	07
4205	06
4206	05
4207	04
4208	03
4209	02
420A	01

☐ Show entire memory content  
☐ Show only loaded memory location  
☒ Store directly to specified memory location

EditorAssembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		MVI C,10	06	2	2	7
0001			10			
✓ 0002		DCR C	0D	1	1	4
✓ 0003	REPEAT	MOV D,C	51	1	1	4
✓ 0004		LXI H,4201	21	3	3	10
0005			01			
0006			42			
✓ 0007	LOOP	MOV A,M	7E	1	2	7
✓ 0008		INX H	23	1	1	6
✓ 0009		CHP M	8E	1	2	7
✓ 000A		JC SKIP	DA	3	3	10
000B			12			
000C			00			
✓ 000D		MOV B,M	46	1	2	7
✓ 000E		MOV M,A	77	1	2	7
✓ 000F		DCX H	2B	1	1	6
✓ 0010		MOV M,B	70	1	2	7
✓ 0011		INX H	23	1	1	6
✓ 0012	SKIP	DCR D	15	1	1	4

Simulate

Start From → 0000

Run all At a TimeStep By Step

RegistersMemoryDevices

Memory Editor

Memory Range: 0000 → FFFF

Memory Address	Value
0017	C2
0018	03
001A	76
4201	01
4202	02
4203	03
4204	04
4205	05
4206	06
4207	07
4208	08
4209	09
420A	0A

☐ Show entire memory content  
☐ Show only loaded memory location  
☒ Store directly to specified memory location

EditorAssembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0000		MVI C,10	0E	2	2	7
0001			10			
✓ 0002		DCR C	00	1	1	4
✓ 0003	REPEAT	MOV D,C	51	1	1	4
✓ 0004		LXI H,4201	21	3	3	10
0005			01			
0006			42			
✓ 0007	LOOP	MOV A,M	7E	1	2	7
✓ 0008		INX H	23	1	1	6
✓ 0009		CMP M	BE	1	2	7
✓ 000A		JC SKIP	DA	3	3	10
000B			12			
000C			00			
✓ 000D		MOV B,M	46	1	2	7
✓ 000E		MOV M,A	77	1	2	7
✓ 000F		DCX H	2B	1	1	6
✓ 0010		MOV M,B	70	1	2	7
✓ 0011		INX H	23	1	1	6
✓ 0012	SKIP	DCR D	15	1	1	4

Simulate

Start From → 0000

Run all At a TimeStep By Step

RegistersMemoryDevices

Registers

Register	Value	7	6	5	4	3	2	1	0
Accumulator	01	0	0	0	0	0	0	0	1
Register B	01	0	0	0	0	0	0	0	1
Register C	00	0	0	0	0	0	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	42	0	1	0	0	0	0	1	0
Register L	02	0	0	0	0	0	0	1	0
Memory(M)	02	0	0	0	0	0	0	1	0

Register	Value	S	Z	* AC	* P	* CY	
Flag Register	55	0	1	0	1	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer(HL)	4202
Program Status Word(PSW)	0155
Program Counter(PC)	001A
Clock Cycle Counter	7018
Instruction Counter	1000

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

For SIM Instruction

SOD	SDE	* R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0

For RIM Instruction

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool:

Hexadecimal	Decimal	Binary
0	0	0