Experiment 3.3

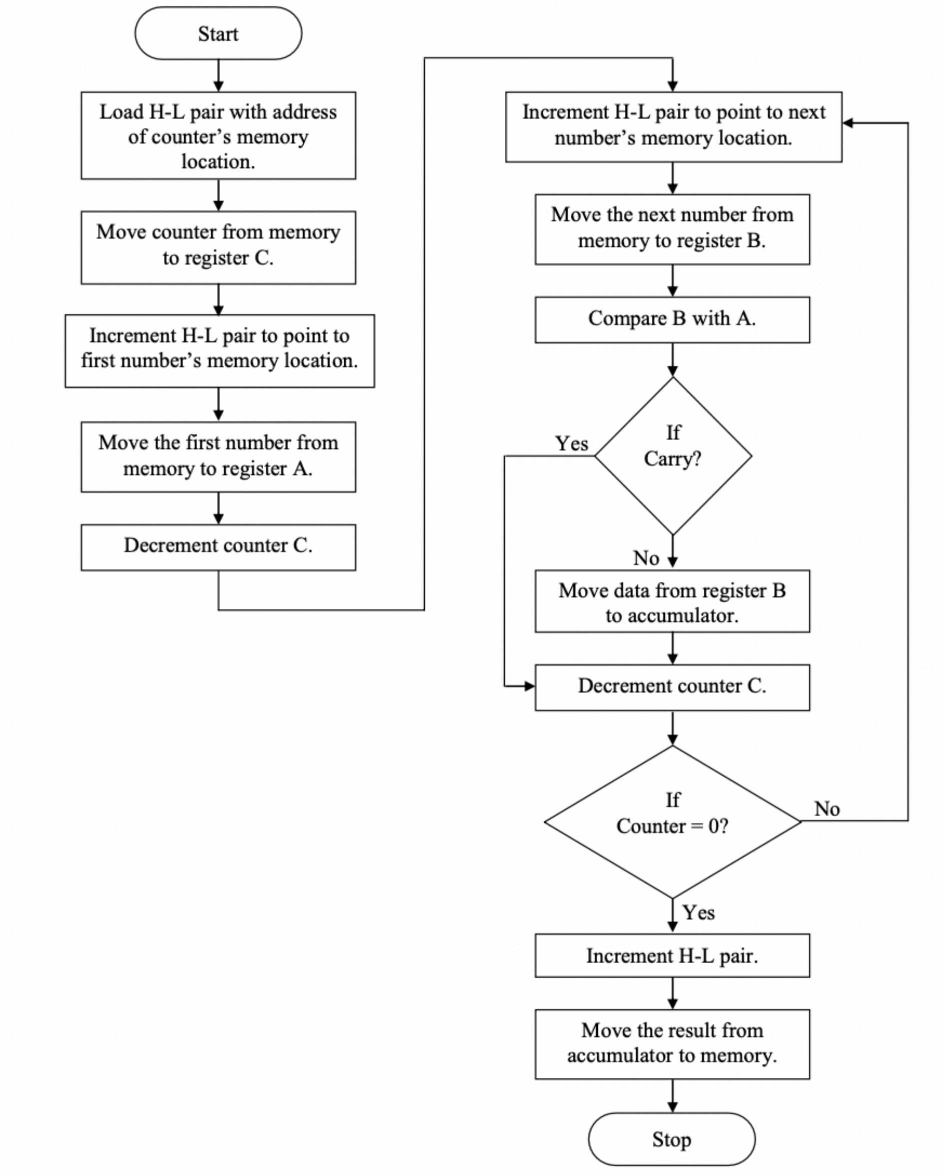
Student Name:Rajiv Paul **UID:20BCS1812**
Branch: CSE **Section/Group:607A**
Semester: 4th **Date of Performance: 03/05/2022**
Subject Name:MPI Lab **Subject Code: 22E-20CSP-253**

1) **Aim/Overview of the practical:**

a) To find the smallest number in a data array.

Apparatus/Simulator used: 8085 simulator

Flowchart:


Start
Load H-L pair with address of counter's memory location.
Move counter from memory to register C.
Increment H-L pair to point to first number's memory location.
Move the first number from memory to register A.
Decrement counter C.
Increment H-L pair to point to next number's memory location.
Move the next number from memory to register B.
Compare B with A.
If Carry?
Yes
No
Move data from register B to accumulator.
Decrement counter C.
If Counter = 0?
No
Yes
Increment H-L pair.
Move the result from accumulator to memory.
Stop

Algorithm:

1. **LXI 4000H loads H-L pair with data from 4000H memory location.**

2. **MOV C,M moves data from memory to C register.**

3. **INX H increments the memory location of H-L pair by 1.**

4. **MOV A,M moves data from memory to accumulator.**

5. **DCR C decrements the C register by 1.**

6. **JUMP: INX H jumps to instruction increment location and increments H-L pair by 1**

7. **MOV B,M moves data from memory to B register.**

8. **CMP B compares B register with accumulator.**

9. **JC LABEL jumps to the location of the label if there is carry.**

10. **MOV A,B moves data from reg. B to accumulator.**

11. **LABEL: DCR C decrements the C register by 1.**

12. **JNZ JUMP jumps if the C register is not 0.**

13. **INX H increments the memory location of H-L pair by 1.**

14. **MOV M,A moves the data from accumulator to memory.**

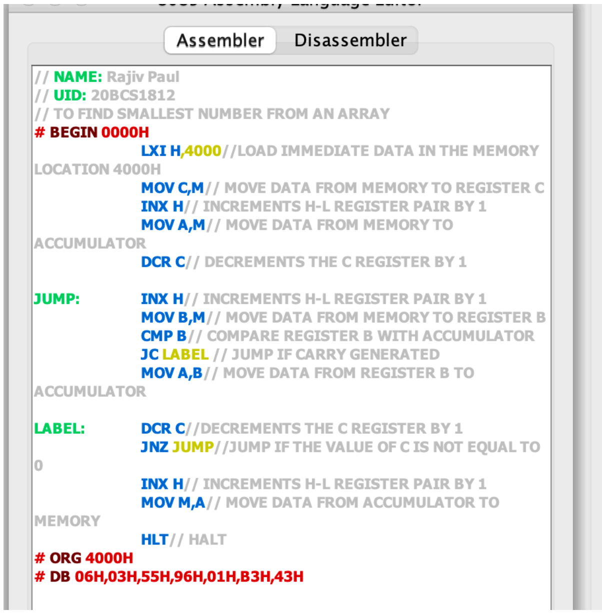
15. **HLT end of the execution.**

Steps for experiment/practical/Code:

```
# BEGIN 0000H

LXI H,4000

MOV C,M

INX H

MOV A,M

DCR C

JUMP: INX H

MOV B,M

CMP B

JC LABEL

MOV A,B

LABEL: DCR C

JNZ JUMP

INX H

MOV M,A

HLT

# ORG 4000H

# DB 06H,03H,55H,96H,01H,B3H,43H
```

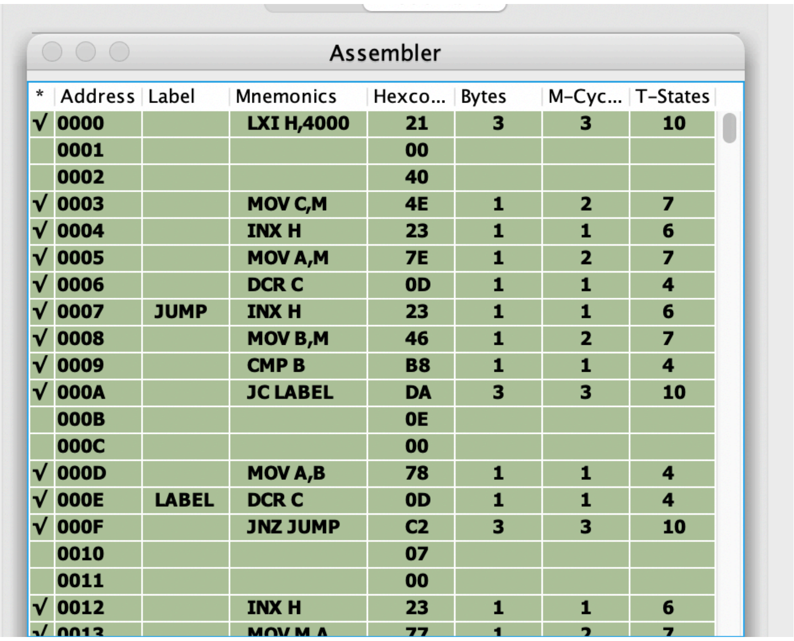
Simulation:

1. CODE IN EDITOR WINDOW:



```
// NAME: Rajiv Paul
// UID: 20BCS1812
// TO FIND SMALLEST NUMBER FROM AN ARRAY
# BEGIN 0000H
                LXI H,4000//LOAD IMMEDIATE DATA IN THE MEMORY
LOCATION 4000H
                MOV C,M// MOVE DATA FROM MEMORY TO REGISTER C
                INX H// INCREMENTS H-L REGISTER PAIR BY 1
                MOV A,M// MOVE DATA FROM MEMORY TO
ACCUMULATOR
                DCR C// DECREMENTS THE C REGISTER BY 1

JUMP:           INX H// INCREMENTS H-L REGISTER PAIR BY 1
                MOV B,M// MOVE DATA FROM MEMORY TO REGISTER B
                CMP B// COMPARE REGISTER B WITH ACCUMULATOR
                JC LABEL // JUMP IF CARRY GENERATED
                MOV A,B// MOVE DATA FROM REGISTER B TO
ACCUMULATOR

LABEL:          DCR C//DECREMENTS THE C REGISTER BY 1
                JNZ JUMP//JUMP IF THE VALUE OF C IS NOT EQUAL TO
0
                INX H// INCREMENTS H-L REGISTER PAIR BY 1
                MOV M,A// MOVE DATA FROM ACCUMULATOR TO
MEMORY
                HLT// HALT
# ORG 4000H
# DB 06H,03H,55H,96H,01H,B3H,43H
```

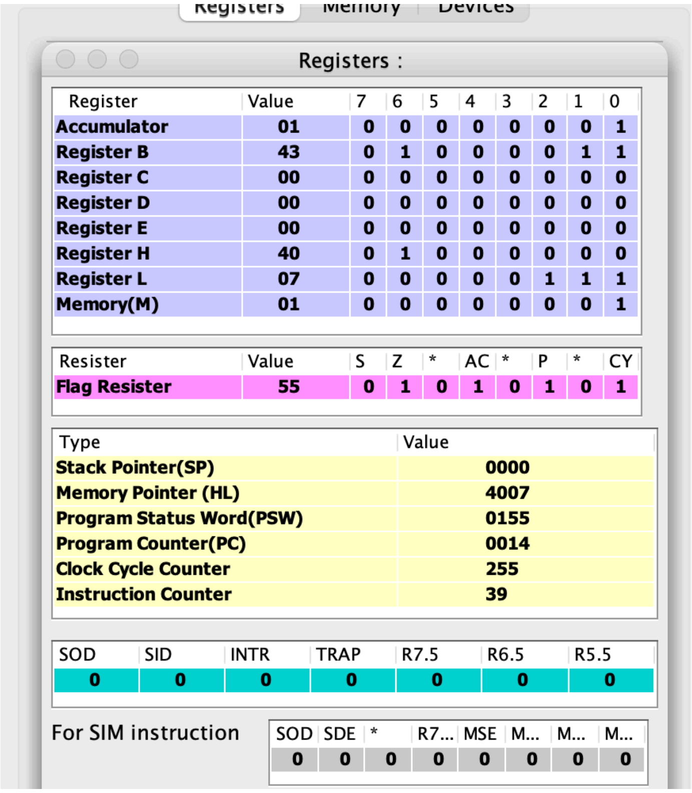
2. <u>ASSEMBLER WINDOW:</u>



Assembler

*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LXI H,4000	21	3	3	10
	0001			00			
	0002			40			
√	0003		MOV C,M	4E	1	2	7
√	0004		INX H	23	1	1	6
√	0005		MOV A,M	7E	1	2	7
√	0006		DCR C	0D	1	1	4
√	0007	JUMP	INX H	23	1	1	6
√	0008		MOV B,M	46	1	2	7
√	0009		CMP B	B8	1	1	4
√	000A		JC LABEL	DA	3	3	10
	000B			0E			
	000C			00			
√	000D		MOV A,B	78	1	1	4
√	000E	LABEL	DCR C	0D	1	1	4
√	000F		JNZ JUMP	C2	3	3	10
	0010			07			
	0011			00			
√	0012		INX H	23	1	1	6
√	0013		MOV M,A	77	1	2	7

3. REGISTERS:



Register	Value	7	6	5	4	3	2	1	0
Accumulator	01	0	0	0	0	0	0	0	1
Register B	43	0	1	0	0	0	0	1	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	40	0	1	0	0	0	0	0	0
Register L	07	0	0	0	0	0	1	1	1
Memory(M)	01	0	0	0	0	0	0	0	1

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

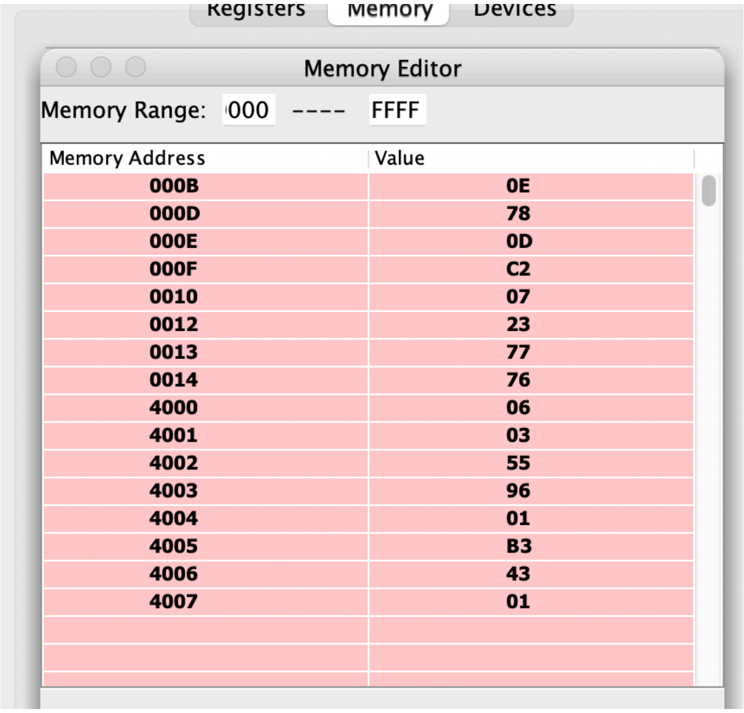
Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	4007
Program Status Word(PSW)	0155
Program Counter(PC)	0014
Clock Cycle Counter	255
Instruction Counter	39

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

For SIM instruction

SOD	SDE	*	R7...	MSE	M...	M...	M...
0	0	0	0	0	0	0	0

4. MEMORY:



Memory Range: 000 ---- FFFF

Memory Address	Value
000B	0E
000D	78
000E	0D
000F	C2
0010	07
0012	23
0013	77
0014	76
4000	06
4001	03
4002	55
4003	96
4004	01
4005	B3
4006	43
4007	01

RESULT

BEFORE EXECUTION:

```
4000H: 06H(counter)
4001H: 03H
4002H: 55H
4003H: 96H
4004H: 01H
4005H: B3H
4006H: 43H
```

AFTER EXECUTION:

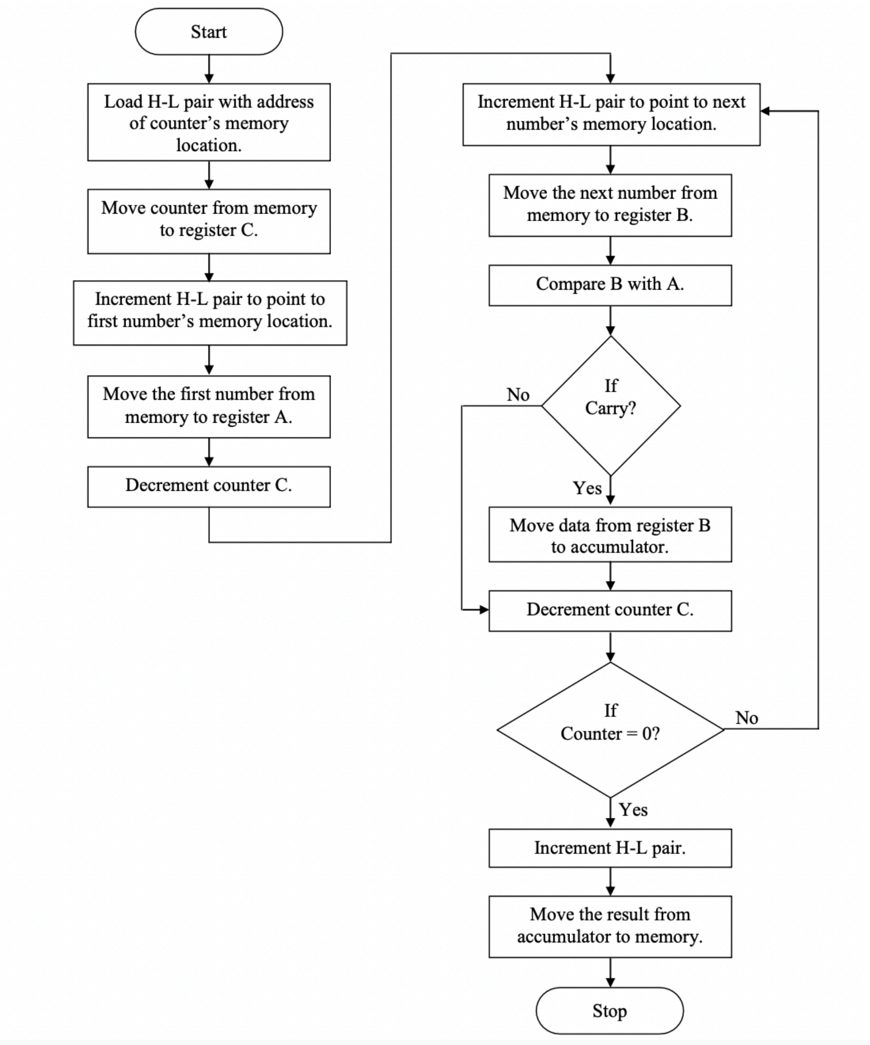
```
4007H: 01H
```

Aim/Overview of the practical:

b) To find largest number in a data array.

Apparatus/Simulator used: 8085 simulator

Flowchart:


Start
Load H-L pair with address of counter's memory location.
Move counter from memory to register C.
Increment H-L pair to point to first number's memory location.
Move the first number from memory to register A.
Decrement counter C.
Increment H-L pair to point to next number's memory location.
Move the next number from memory to register B.
Compare B with A.
If Carry?
No
Yes
Move data from register B to accumulator.
Decrement counter C.
If Counter = 0?
No
Yes
Increment H-L pair.
Move the result from accumulator to memory.
Stop

Algorithm:

1. LXI 4000H loads H-L pair with data from 4000H memory location.

2. MOV C,M moves data from memory to C register.

3. INX H increments the memory location of H-L pair by 1.

4. MOV A,M moves data from memory to accumulator.

5. DCR C decrements the C register by 1.

6. JUMP: INX H jumps to instruction increment location and increments H-L pair by 1

7. MOV B,M moves data from memory to B register.

8. CMP B compares B register with accumulator.

9. JNC LABEL jumps to the location of the label if there is carry.

10. MOV A,B moves data from reg. B to accumulator.

11. LABEL: DCR C decrements the C register by 1.

12. JNZ JUMP jumps if the C register is not 0.

13. INX H increments the memory location of H-L pair by 1.

14. MOV M,A moves the data from accumulator to memory.

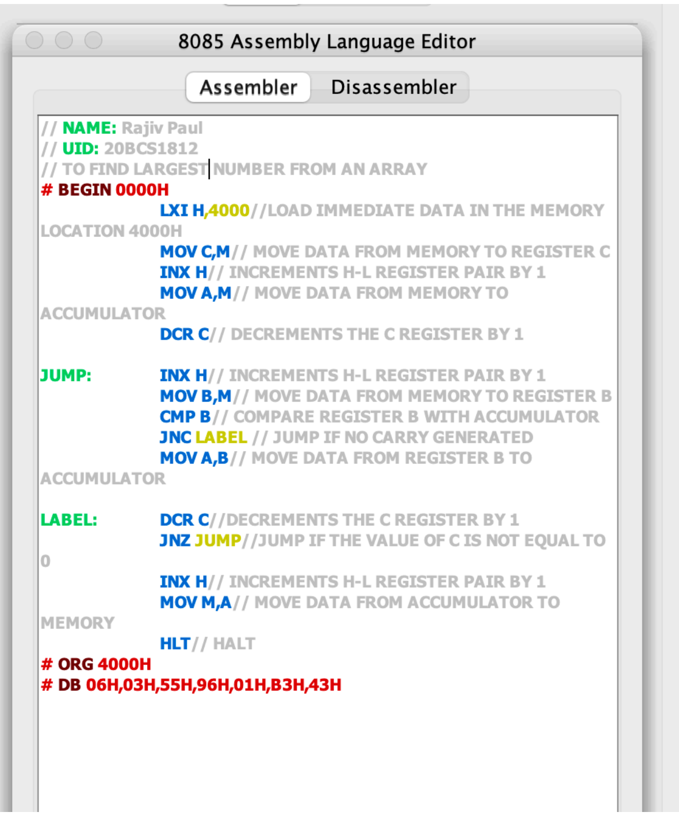
15. HLT end of the execution.

Steps for experiment/practical/Code:

```
# BEGIN 0000H

LXI H,4000

MOV C,M

INX H

MOV A,M

DCR C

JUMP: INX H

MOV B,M

CMP B

JNC LABEL

MOV A,B

LABEL: DCR C

JNZ JUMP

INX H

MOV M,A

HLT

# ORG 4000H

# DB 06H,03H,55H,96H,01H,B3H,43H
```

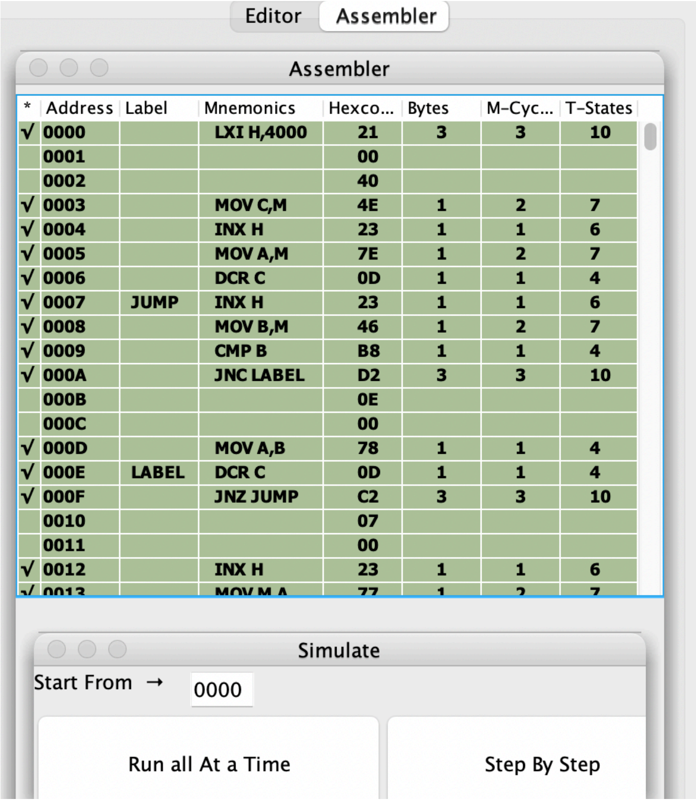
Simulation:

1. CODE IN EDITOR WINDOW:



```
// NAME: Rajiv Paul
// UID: 20BCS1812
// TO FIND LARGEST NUMBER FROM AN ARRAY
# BEGIN 0000H
                LXI H,4000//LOAD IMMEDIATE DATA IN THE MEMORY
LOCATION 4000H
                MOV C,M// MOVE DATA FROM MEMORY TO REGISTER C
                INX H// INCREMENTS H-L REGISTER PAIR BY 1
                MOV A,M// MOVE DATA FROM MEMORY TO
ACCUMULATOR
                DCR C// DECREMENTS THE C REGISTER BY 1

JUMP:           INX H// INCREMENTS H-L REGISTER PAIR BY 1
                MOV B,M// MOVE DATA FROM MEMORY TO REGISTER B
                CMP B// COMPARE REGISTER B WITH ACCUMULATOR
                JNC LABEL // JUMP IF NO CARRY GENERATED
                MOV A,B// MOVE DATA FROM REGISTER B TO
ACCUMULATOR

LABEL:          DCR C//DECREMENTS THE C REGISTER BY 1
                JNZ JUMP//JUMP IF THE VALUE OF C IS NOT EQUAL TO
0
                INX H// INCREMENTS H-L REGISTER PAIR BY 1
                MOV M,A// MOVE DATA FROM ACCUMULATOR TO
MEMORY
                HLT// HALT
# ORG 4000H
# DB 06H,03H,55H,96H,01H,B3H,43H
```

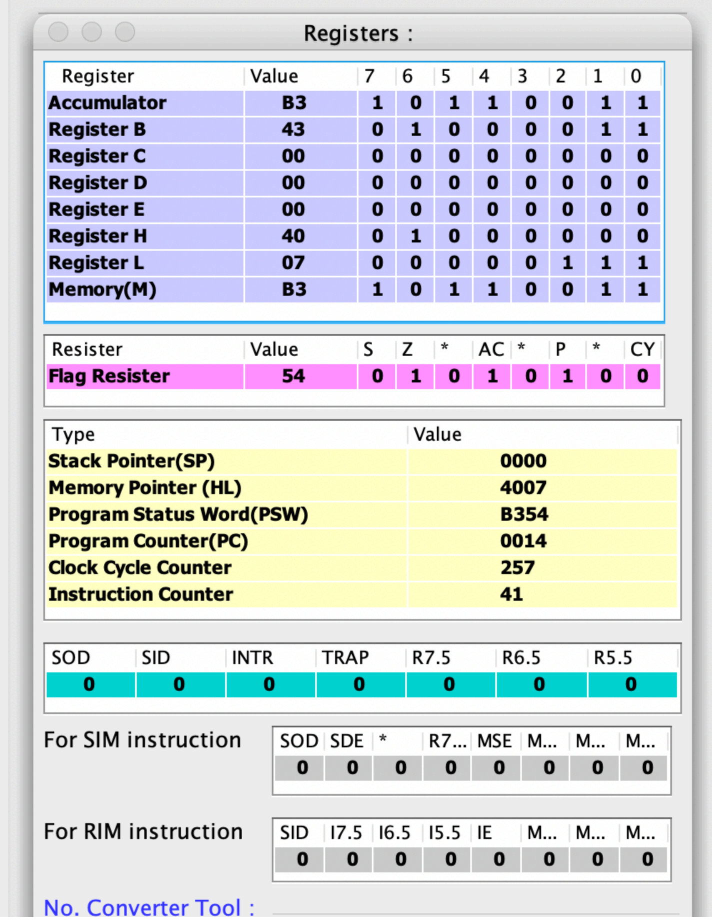
2. ASSEMBLER WINDOW:



*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LXI H,4000	21	3	3	10
	0001			00			
	0002			40			
√	0003		MOV C,M	4E	1	2	7
√	0004		INX H	23	1	1	6
√	0005		MOV A,M	7E	1	2	7
√	0006		DCR C	0D	1	1	4
√	0007	JUMP	INX H	23	1	1	6
√	0008		MOV B,M	46	1	2	7
√	0009		CMP B	B8	1	1	4
√	000A		JNC LABEL	D2	3	3	10
	000B			0E			
	000C			00			
√	000D		MOV A,B	78	1	1	4
√	000E	LABEL	DCR C	0D	1	1	4
√	000F		JNZ JUMP	C2	3	3	10
	0010			07			
	0011			00			
√	0012		INX H	23	1	1	6
√	0013		MOV M,A	77	1	2	7

Start From → 0000

3. REGISTERS:



Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	B3	1	0	1	1	0	0	1	1
Register B	43	0	1	0	0	0	0	1	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	40	0	1	0	0	0	0	0	0
Register L	07	0	0	0	0	0	1	1	1
Memory(M)	B3	1	0	1	1	0	0	1	1

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	54	0	1	0	1	0	1	0	0

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	4007
Program Status Word(PSW)	B354
Program Counter(PC)	0014
Clock Cycle Counter	257
Instruction Counter	41

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

For SIM instruction

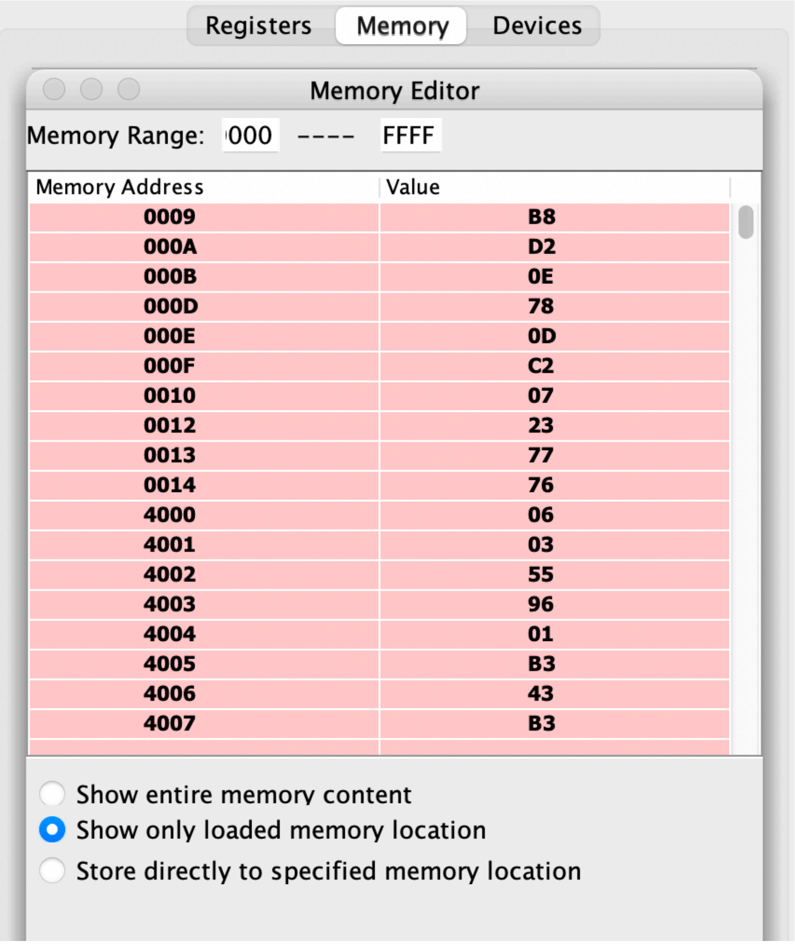
SOD	SDE	*	R7...	MSE	M...	M...	M...
0	0	0	0	0	0	0	0

For RIM instruction

SID	I7.5	I6.5	I5.5	IE	M...	M...	M...
0	0	0	0	0	0	0	0

No. Converter Tool :

4. MEMORY:



Memory Range: 000 ---- FFFF

Memory Address	Value
0009	B8
000A	D2
000B	0E
000D	78
000E	0D
000F	C2
0010	07
0012	23
0013	77
0014	76
4000	06
4001	03
4002	55
4003	96
4004	01
4005	B3
4006	43
4007	B3

○ Show entire memory content
● Show only loaded memory location
○ Store directly to specified memory location

RESULT

BEFORE EXECUTION:

4000H: 06H(counter)
4001H: 03H
4002H: 55H
4003H: 96H
4004H: 01H
4005H: B3H
4006H: 43H

AFTER EXECUTION:

4007H: B3H

Learning outcomes (What I have learnt):

1.Learnt about 8085 simulator

2.Learnt how to find largest number in a data array.

3.Learnt how to find smallest number in a data array.

4.Learnt about CMP and its function

5.Learnt about difference between JNC,JNZ and JC.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			