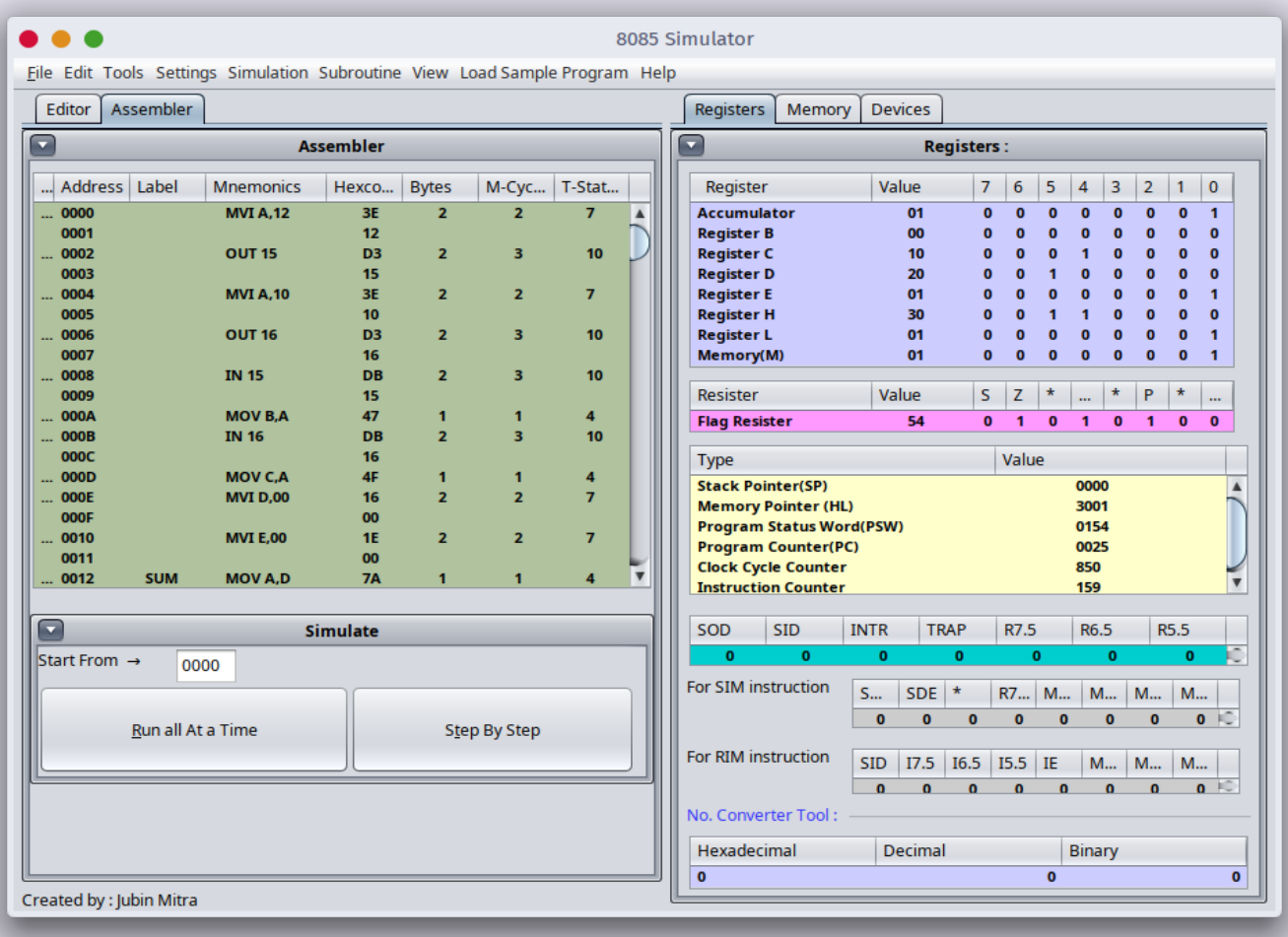
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Aim : Learning Programs using Branch Instructions like JMP, JZ, JNZ, JC etc.

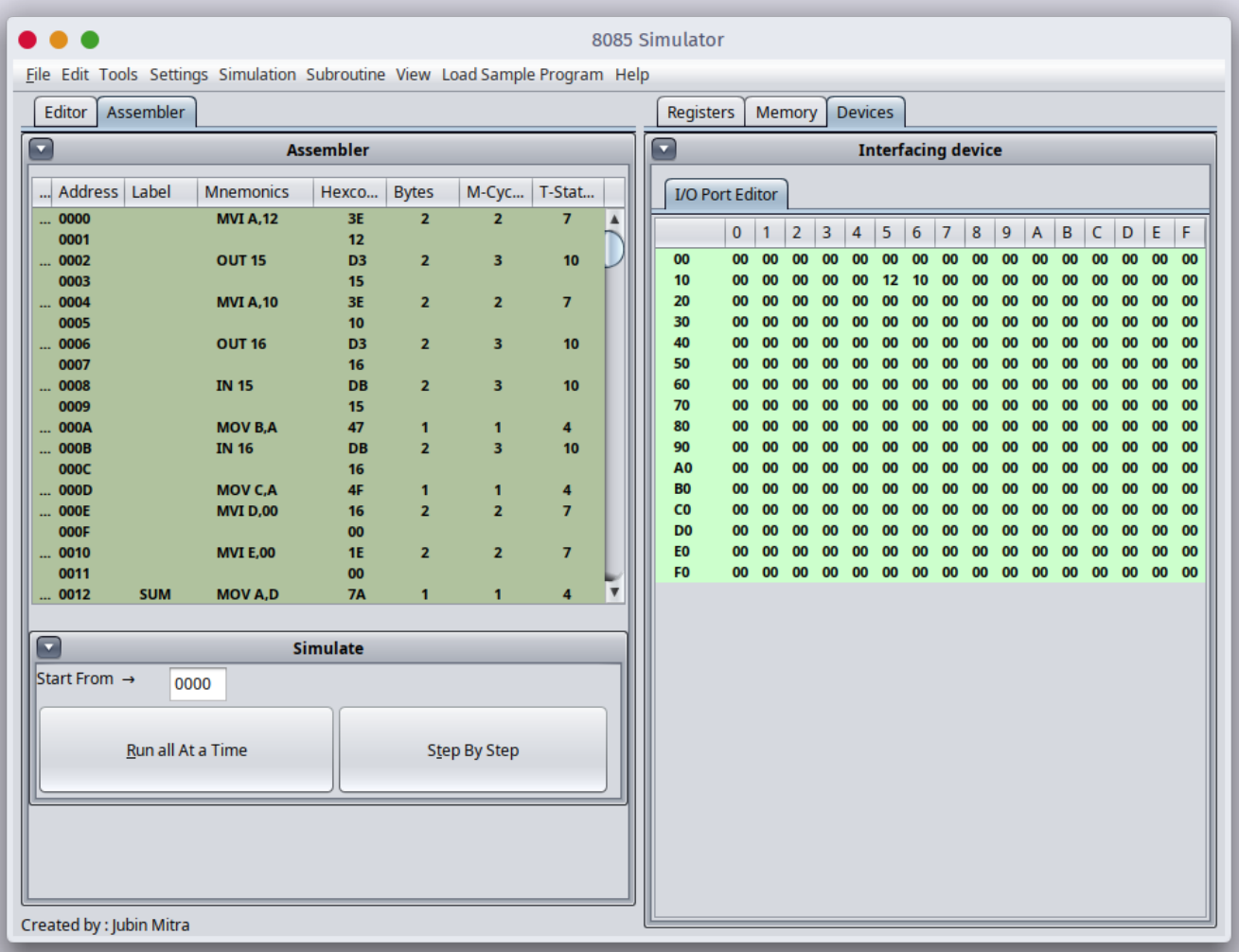
Exercise :

1. Write a Program to multiply two 8-bit numbers given as input on input ports 15H and 16H. Save the lower byte of the result on memory location 3000H and higher byte of the result on 3001H.

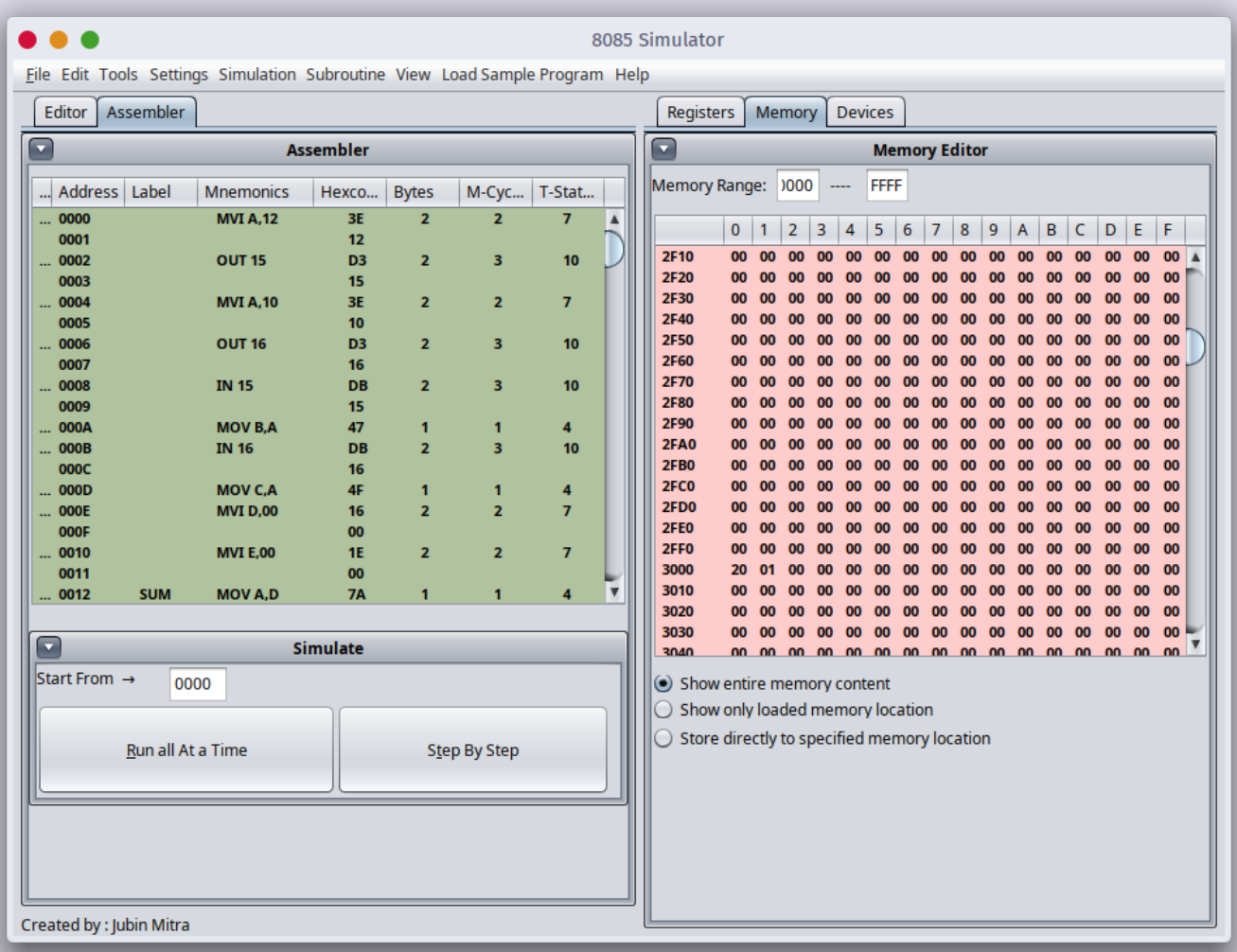
Screenshots :



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Contents and comment :

Code :

```
MVI  A, 12H     // load 12H in A
OUT  15H        // display on port 15
MVI  A, 10H     // load 10H in A
OUT  16H        // display on port 16

IN   15H        // load input from port 15
MOV  B, A       // copy From A to B
IN   16H        // load input from port 16
MOV  C, A       // copy from A to C

MVI  D, 00H     // clear D (LSB)
MVI  E, 00H     // clear E (MSB)

SUM :           // label
MOV  A, D       // copy from D to A
ADD  C          // add C to A

MOV  D, A       // copy From A to D
MOV  A, E       // copy from E to A
ACI  00H        // add carry if so
MOV  E, A       // copy from A to E

DCR  B          // decrement B (counter)
JNZ  SUM        // repeat SUM till B get 0

LXI  H, 3000H   // load 3000 in HL
MOV  M, D       // load LSB to 3000 H
LXI  H, 3001H   // load 3000l in HL
MOV  M, E       // load MSB to 3001 H

HLT             // end
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0000		MVI A, 12	3E	2	2	7
0001			12			
0002		OUT 15	D3	2	3	10
0003			15			
0004		MVI A, 10	3E	2	2	7
0005			10			
0006		OUT 16	D3	2	3	10
0007			16			
0008		IN 15	DB	2	3	10
0009			15			
000A		MOV B, A	47	1	1	4
000B		IN 16	DB	2	3	10
000C			16			
000D		MOV C, A	4F	1	1	4
000E		MVI D, 00	16	2	2	7
000F			00			
0010		MVI E, 00	1E	2	2	7
0011			00			
0012	SUM	MOV A, D	7A	1	1	4
0013		ADD C	81	1	1	4
0014		MOV D, A	57	1	1	4
0015		MOV A, E	78	1	1	4
0016		ACI 00	CE	2	2	7
0017			00			
0018		MOV E, A	5F	1	1	4
0019		DCR B	05	1	1	4
001A		JNZ SUM	C2	3	3	10
001B			12			
001C			00			
001D		LXI H, 3000	21	3	3	10
001E			00			

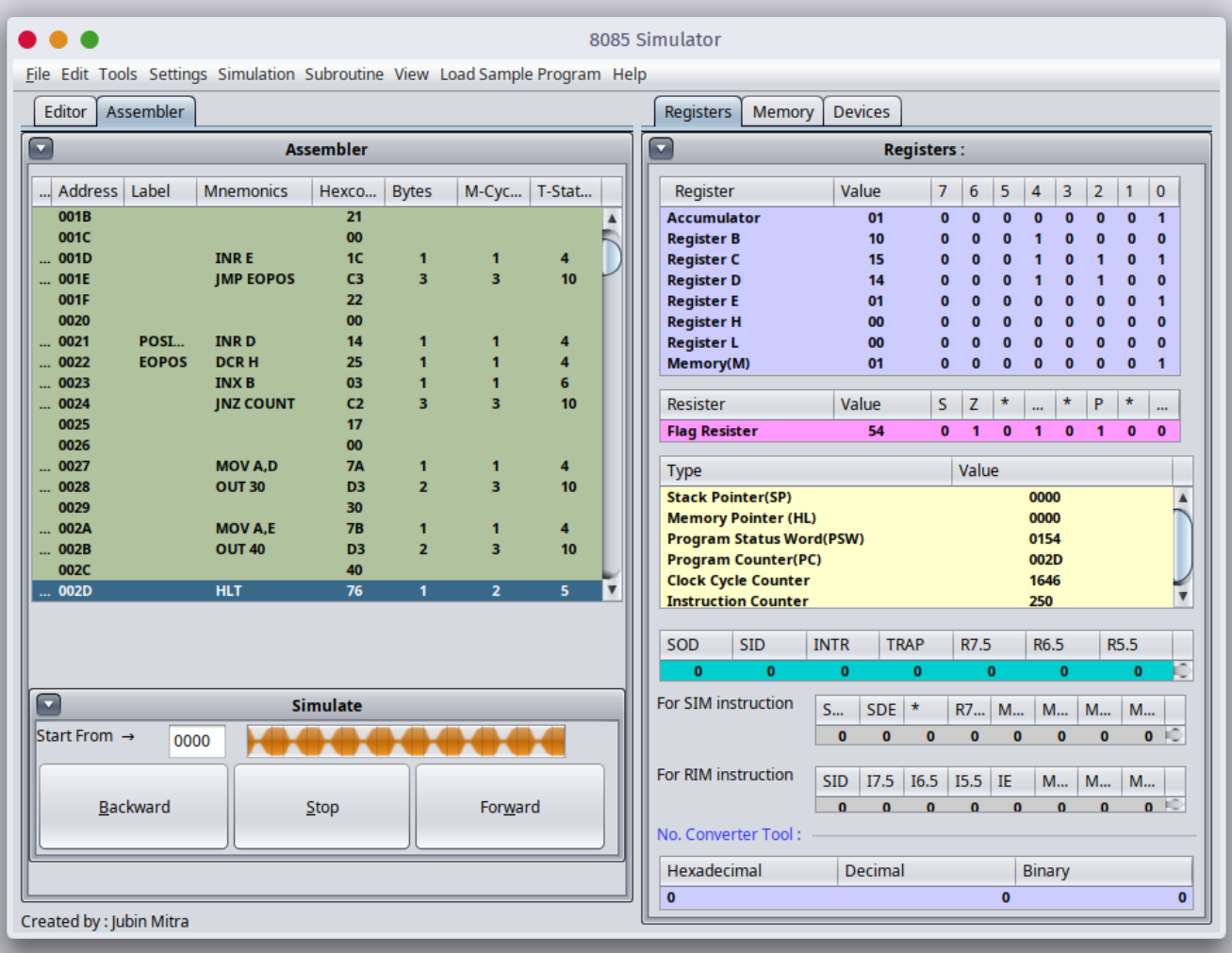
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
001F			30			
0020		MOV M,D	72	1	2	7
0021		LXI H,3001	21	3	3	10
0022			01			
0023			30			
0024		MOV M,E	73	1	2	7
0025		HLT	76	1	2	5

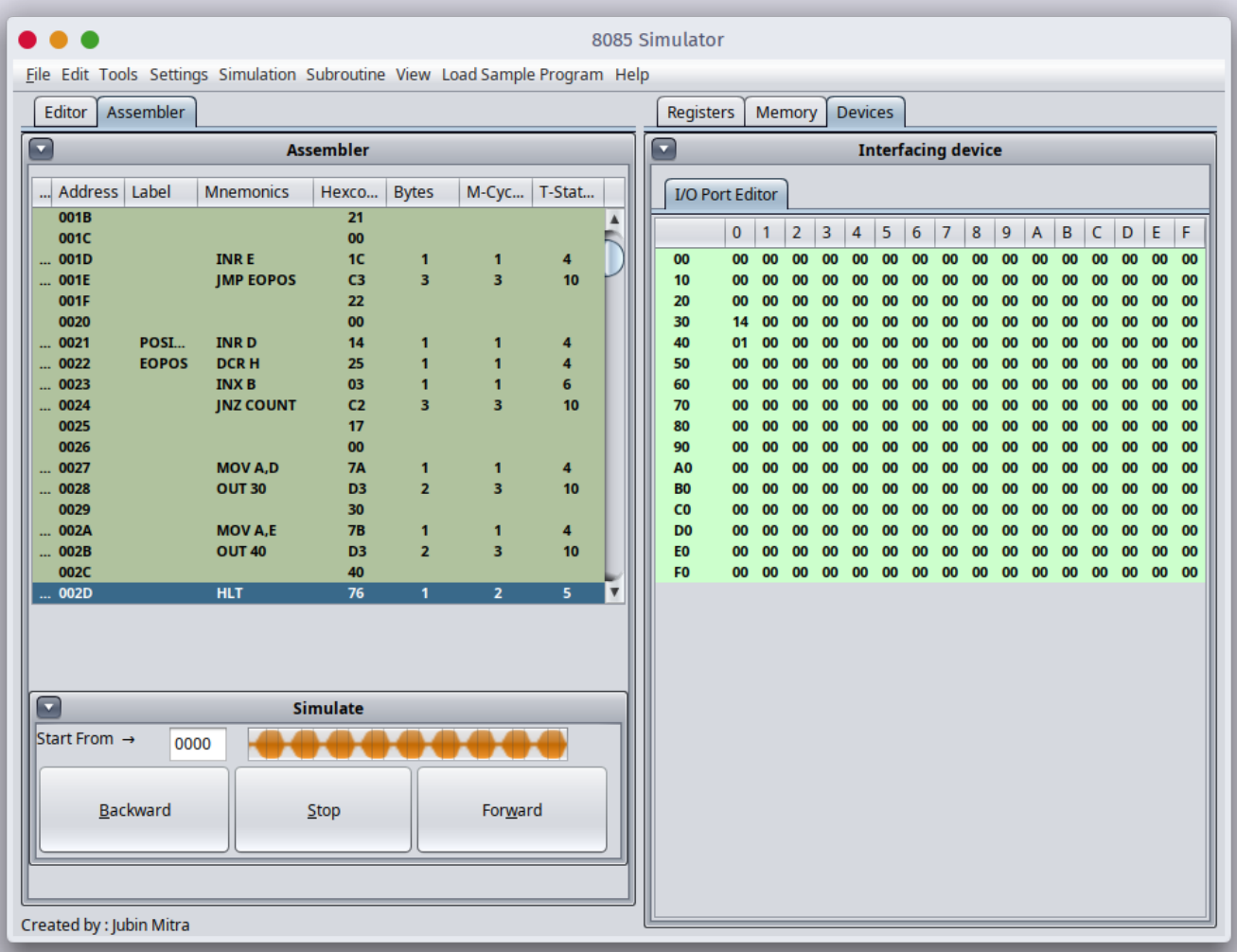
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

2. Write a Program to count positive and negative numbers out of 20 numbers stored in memory. (Assume appropriate memory location in your program and load 20 different bytes in memory using assembler directives). Display the count of positive nos on output port 30 H and negative numbers on 40 H.

Screenshots :



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Contents and comment :

Code :

```
LXI B, 1000H    // memory location start
MVI H, 14H      // counter
MVI A, 80H      // load data in A
DLOAD:          // label
STAX B          // store A in BC pointer
INX B           // increment BC pair
INR A           // increment A
DCR H           // decrement counter
JNZ DLOAD       // repeat DLOAD till H

LXI B, 1000H    // load 1000 in BC
MVI H, 15H      // counter
MVI D, 00H      // +ve count
MVI E, 00H      // -ve count

COUNT:          // label
LDAX B          // load data in B
ADI 00H         // update flag if so
JP POSITIVE     // If +ve then jump
INR E           // increment -ve count
JMP EOPOS       // jump to end of +ve
POSITIVE:       // label
INR D           // increment D
EOPOS:          // label
DCR H           // decrement counter
INX B           // increment BC
JNZ COUNT       // repeat till H->0

MOV A, D        // load +ve count
OUT 30H         // display on port 30
MOV A, E        // load -ve count
OUT 40H         // display on port 40
HLT             // end
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0000		LXI B, 1000	01	3	3	10
0001			00			
0002			10			
0003		MVI H, 14	26	2	2	7
0004			14			
0005		MVI A, 80	3E	2	2	7
0006			80			
0007	DLOAD	STAX B	02	1	2	7
0008		INX B	03	1	1	6
0009		INR A	3C	1	1	4
000A		DCR H	25	1	1	4
000B		JNZ DLOAD	C2	3	3	10
000C			07			
000D			00			
000E		LXI B, 1000	01	3	3	10
000F			00			
0010			10			
0011		MVI H, 15	26	2	2	7
0012			15			
0013		MVI D, 00	16	2	2	7
0014			00			
0015		MVI E, 00	1E	2	2	7
0016			00			
0017	COUNT	LDAX B	0A	1	2	7
0018		ADI 00	C6	2	2	7
0019			00			
001A		JP POSITIVE	F2	3	3	10
001B			21			
001C			00			
001D		INR E	1C	1	1	4
001E		JMP EOPOS	C3	3	3	10

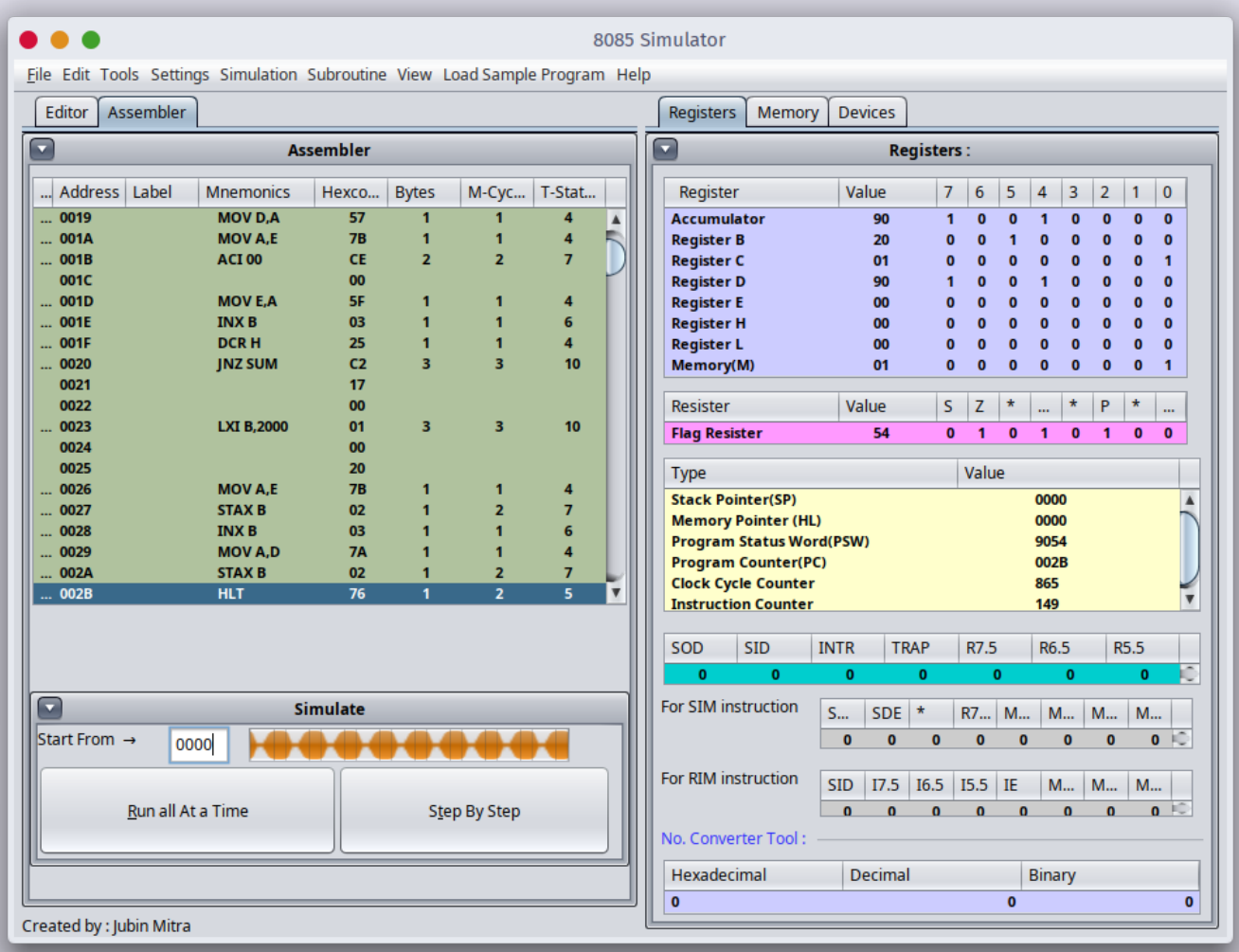
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
001F			22			
0020			00			
0021	POSITIVE	INR D	14	1	1	4
0022	EOPOS	DCR H	25	1	1	4
0023		INX B	03	1	1	4
0024		JNZ COUNT	C2	3	3	10
0025			17			
0026			00			
0027		MOV A, D	7A	1	1	4
0028		OUT 30	D3	2	3	10
0029			30			
002A		MOV A, E	78	1	1	4
002B		OUT 40	D3	2	3	10
002C			40			
002D		HLT	76	1	2	5

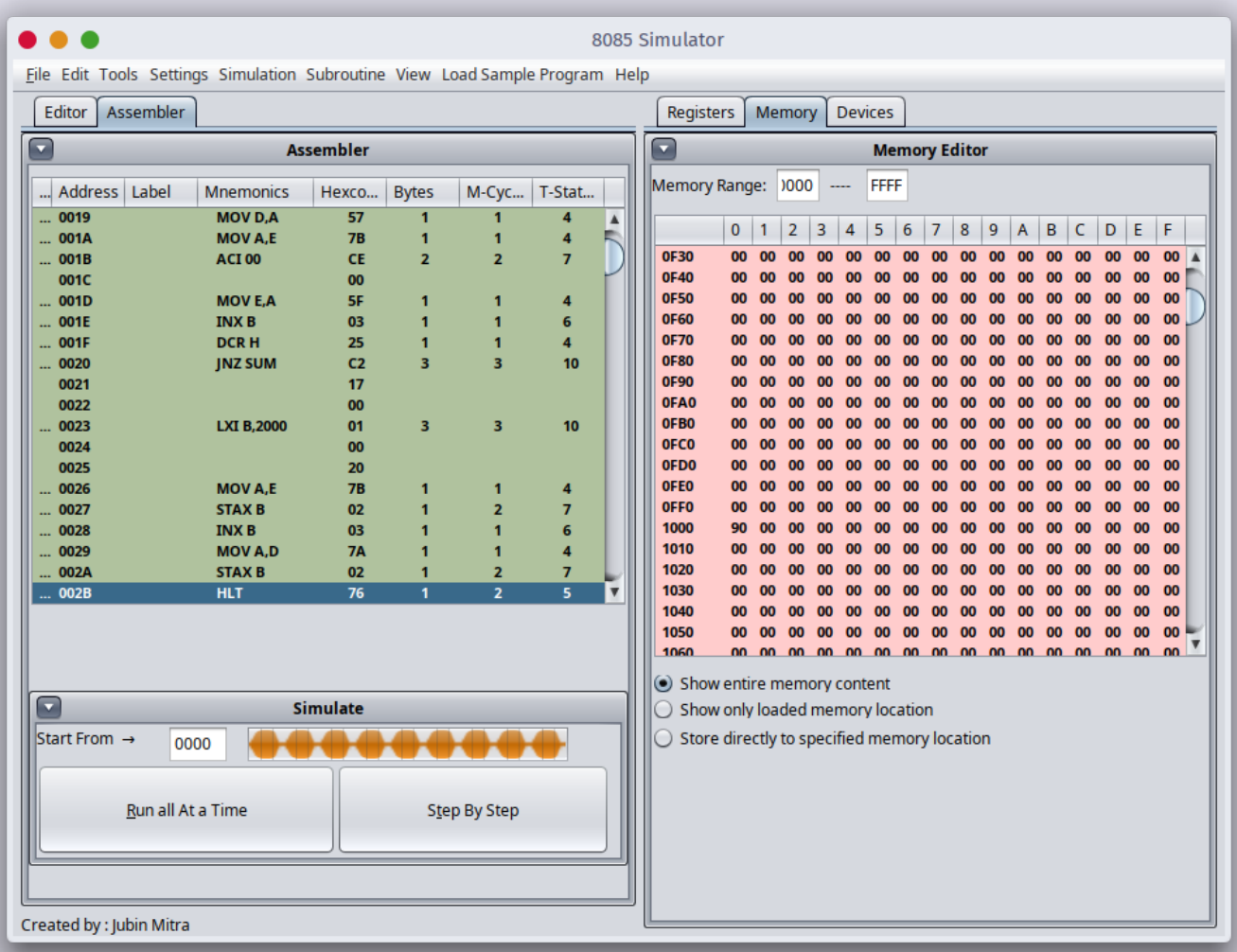
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

3. Write a program to add 10 bytes stored in a string starting from 1000H. Store result at 2000H (LSB), 2001H (MSB).

Screenshots :



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Contents and comment :

```
Code : LXI B, 1000H   // mem addr. start
       MVI H, 0AH     // counter
       MVI A, 12H     // load 12 in A
       DATA:          // label
       STAX B         // store A in BC pointer
       INX B          // increment BC
       INR A          // increment A
       DCR H          // decrement H
       JNZ DATA       // repeat till H->0

       LXI B, 1000H   // load 1000 in BC
       MVI H, 0AH     // counter
       MVI D, 00H     // clear D (LSB)
       MVI E, 00H     // clear E (MSB)

       SUM:           // label
       LDAX B         // load from mem.
       ADD D          // add D (LSB)
       MOV D, A       // copy from A to D
       MOV A, E       // copy from E to A
       ACI 00H        // add if carry
       MOV E, A       // copy from A to E
       INX B          // increment BC
       DCR H          // decrement H
       JNZ SUM        // repeat till H->0

       LXI B, 2000H   // mem. addr. start
       MOV A, E       // copy from E to A
       STAX B         // store LSB
       INX B          // increment BC
       MOV A, D       // copy from D to A
       STAX B         // store MSB
       HLT            // end
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0000		LXI B,1000	01	3	3	10
0001			00			
0002			10			
0003		MVI H,0A	26	2	2	7
0004			0A			
0005		MVI A,12	3E	2	2	7
0006			12			
0007	DATA	STAX B	02	1	2	7
0008		INX B	03	1	1	6
0009		INR A	3C	1	1	4
000A		DCR M	25	1	1	4
000B		JNZ DATA	C2	3	3	10
000C			07			
000D			00			
000E		LXI B,1000	01	3	3	10
000F			00			
0010			10			
0011		MVI H,0A	26	2	2	7
0012			0A			
0013		MVI D,00	16	2	2	7
0014			00			
0015		MVI E,00	1E	2	2	7
0016			00			
0017	SUM	LDAX B	0A	1	2	7
0018		ADD D	82	1	1	4
0019		MOV D,A	57	1	1	4
001A		MOV A,E	7B	1	1	4
001B		ACI 00	CE	2	2	7
001C			00			
001D		MOV E,A	5F	1	1	4
001E		INX B	03	1	1	6

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

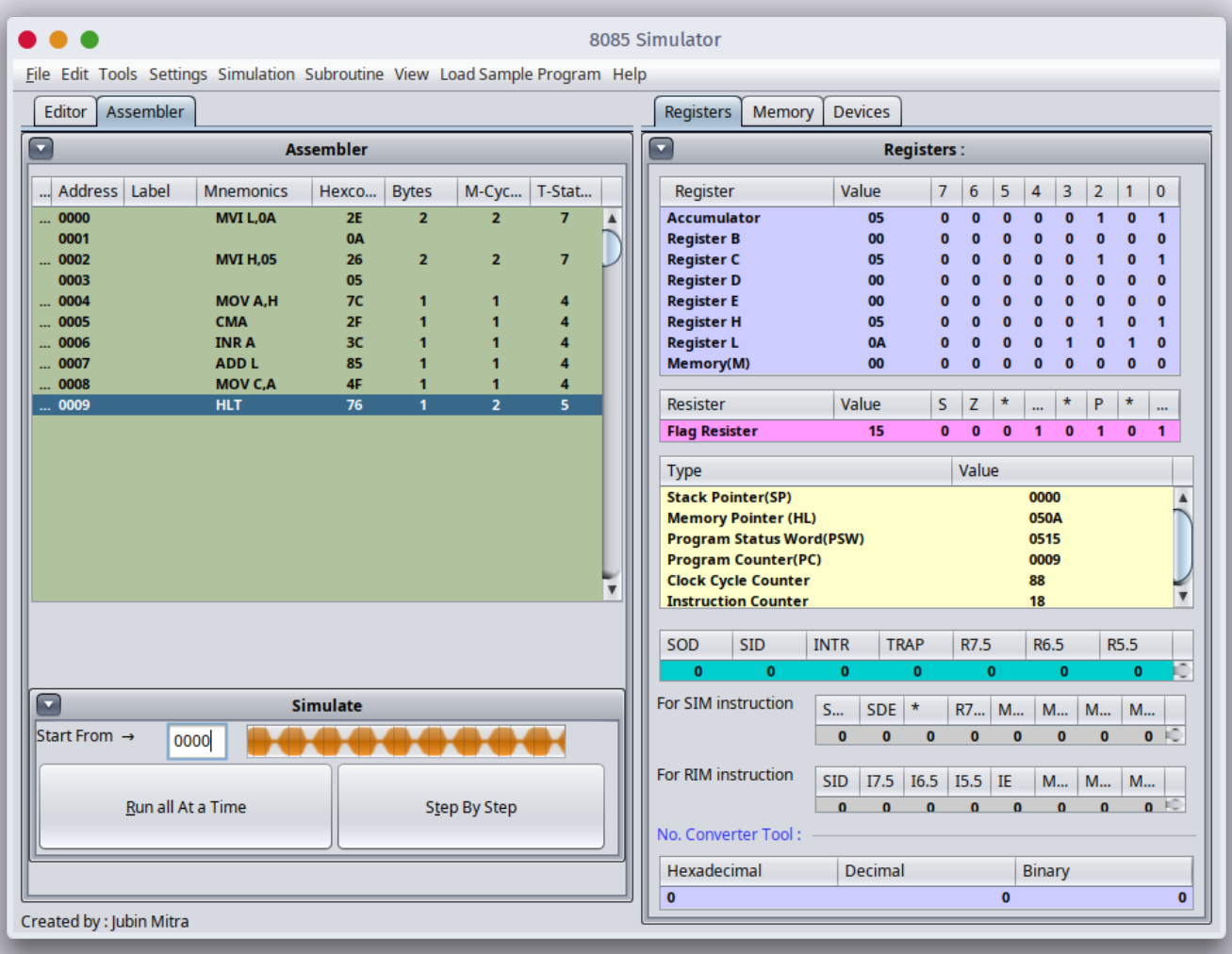
Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
001F		DCR H	25	1	1	4
0020		JNZ SUM	C2	3	3	10
0021			17			
0022			00			
0023		LXI B, 2000	01	3	3	10
0024			00			
0025			20			
0026		MOV A, E	7B	1	1	4
0027		STAX B	02	1	2	7
0028		INX B	03	1	1	6
0029		MOV A, D	7A	1	1	4
002A		STAX B	02	1	2	7
002B		HLT	76	1	2	5

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

4. Write a program to subtract the contents of register H from register L without using any of the subtract instructions.

(a) Apply 2's Complement method.

Screenshots :



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

Contents and comment :

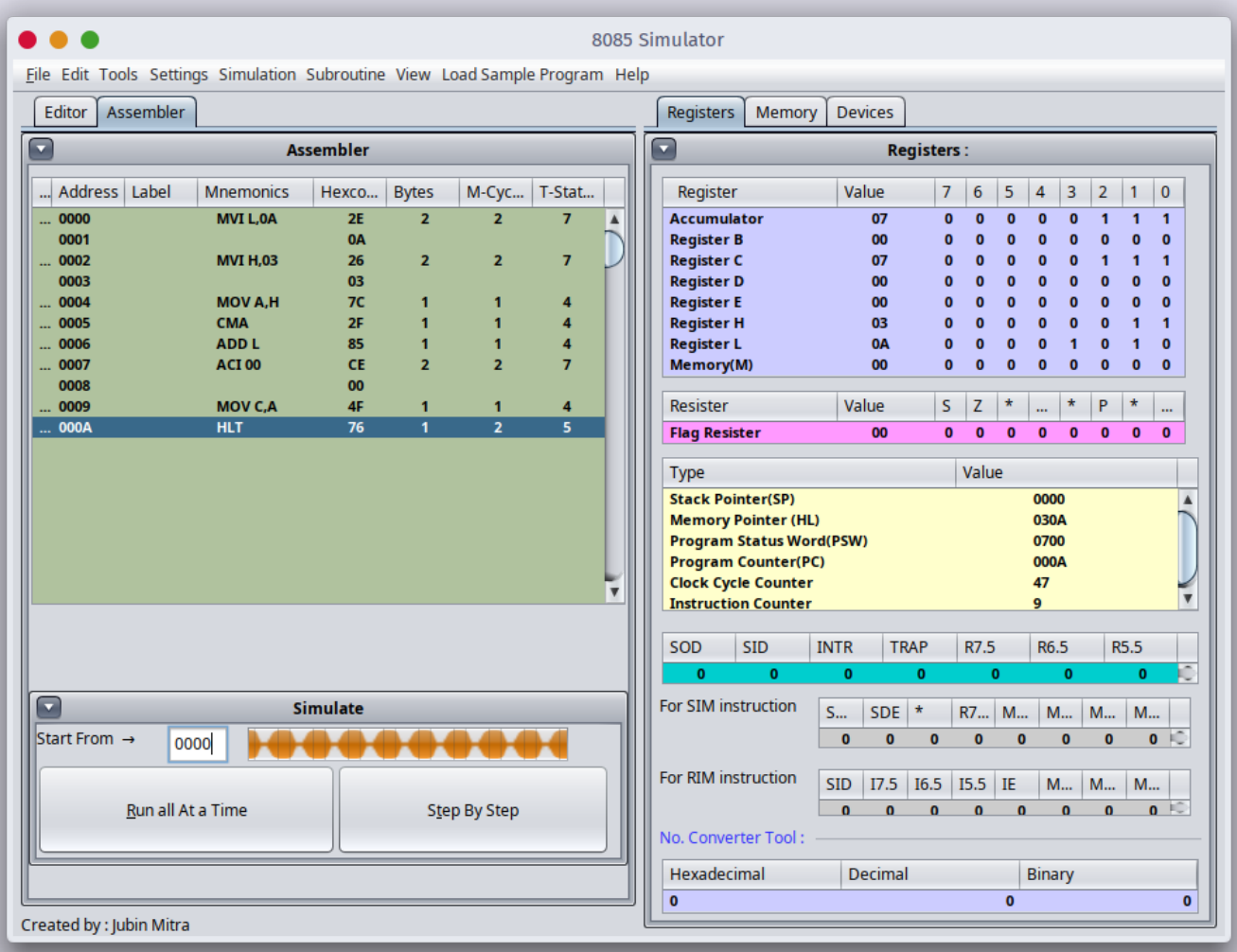
```
Code : MVI  L,04H   // load 04 in L
       MVI  H,09H   // load 09 in H

       MOV  A,H     // copy from H to A
       CMA          // 1's compl of H
       INR A        // 2's compl. of H
       ADD  L       // add L to it
       MOV  C,A     // store in C
       HLT          // end
```

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0000		MVI L,04	2E	2	2	7
0001			04			
0002		MVI H,09	26	2	2	7
0003			09			
0004		MOV A,H	7C	1	1	4
0005		CMA	2F	1	1	4
0006		INR A	3C	1	1	4
0007		ADD L	85	1	1	4
0008		MOV C,A	4F	1	1	4
0009		HLT	76	1	2	5

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

(b) Apply 1's Complement method.



Contents and comment :

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 3

```
Code : MVI  L,04H   // load 04 in L
       MVI  H,09H   // load 09 in H

       MOV  A,H     // copy from H to A
       CMA          // 1's compl of H
       INR A        // 2's compl. of H
       ADD  L       // add L to it
       MOV  C,A     // store in C
       HLT          // end
```

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0000		MVI L,04	2E	2	2	7
0001			04			
0002		MVI H,09	26	2	2	7
0003			09			
0004		MOV A,H	7C	1	1	4
0005		CMA	2F	1	1	4
0006		INR A	3C	1	1	4
0007		ADD L	85	1	1	4
0008		MOV C,A	4F	1	1	4
0009		HLT	76	1	2	5