

NAME : **Sujan Biswas**

BATCH : BCSE 2nd Year (Lateral)

ROLL NO: 302010501003

Microprocessor lab in ASSIGNMENT #1:

1.Load the contents of the memory locations 2200H into registers .Add these register and store the results in memory location 2202 H & 2203 H.

SL NO	ADDRESS	OPCODE IN HEX	LABEL	INSTRUCTIONS	COMMENTS
1	2200	3A,00,22		LDA 2200	Load accumulator direct address of 2200H.
2	2203	47		MOV B,A	Move accumulator to b register.
3	2204	3A,01,22		LDA 2201	Load accumulator direct into memory location 2201H.
4	2207	0E,00		MVI C,00	Move immediate value 2200 memory address into c register.
5	2209	80		ADD B	Add register b with accumulator
6	220A	D2,0E,22		JNC L1	When carry flag is 0,its jump.
7	220D	0C		INR C	Increase c ,when carry is 1.
8	220E	32,02,22	L1	STA 2202	Store the accumulator memory address 2202H.
9	2211	79		MOV A,C	Move c register to accumulator
10	2212	32,03,22		STA 2203	Store accumulator memory address 2203H
11	2215	76		HLT	Termination the program

8085 Simulator - C:\Users\USER\Documents\Q4.asm

File Tools Settings Simulation Load Sample Program Help

8085 Assembly Language Editor

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
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	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0
Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	00	0	0	0	0	0	0	0	0

Memory Range: 2500 ---- FFFF
Memory Editor

Memory Address	Value
2500	3A
2502	22
2503	47
2504	3A
2505	01
2506	22
2507	0E
2509	80
250A	D2
250B	0E
250C	25
250D	0C
250E	32
250F	02
2510	22
2511	79
2512	32
2513	02

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

Assemble

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0
SOD SID INTR TRAP R7.5 R6.5 R5.5	0 0 0 0 0 0 0

For SIM instruction For RIM instruction

SOD SDE *	R7.5 MSE M7.5 M6.5 M5.5
0 0 0 1 0	0 0 0 0 0

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

Simulate : Start From → 0000

Run all At a Time Step By Step

No. Converter Tool :

Hexadecimal	Decimal	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0

I/O Port Editor

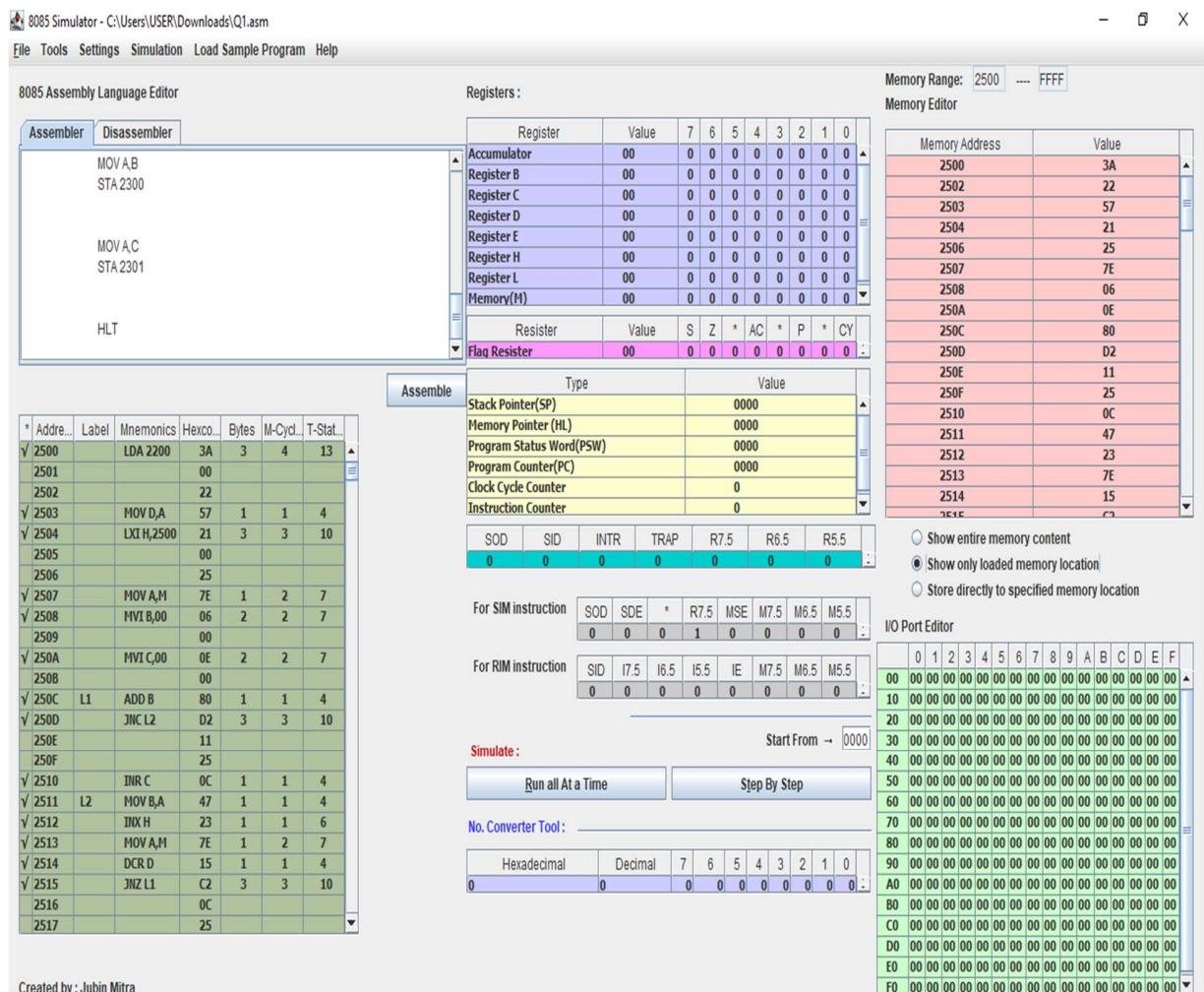
0 1 2 3 4 5 6 7 8 9 A B C D E F
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Created by : Jubin Mitra

2.Find the sum of N numbers stored in consecutive locations starting from 2500 H.The value of N is stored in 2200 H. Store the results in location 2300 H and 2301 H.

SL NO	ADDRESS	OPCODE IN HEX	LABEL	INSTRUCTION	COMMENTS
1	2500	3A,00,22		LDA 2200	Load accumulator direct memory location 2200H .
2	2503	57		MOV D ,A	Move accumulator to d register.
3	2504	21,00,25		LXI H,2500	Load first instruction address 2500H.
4	2507	7E		MOV A,M	Move memory address to accumulator.

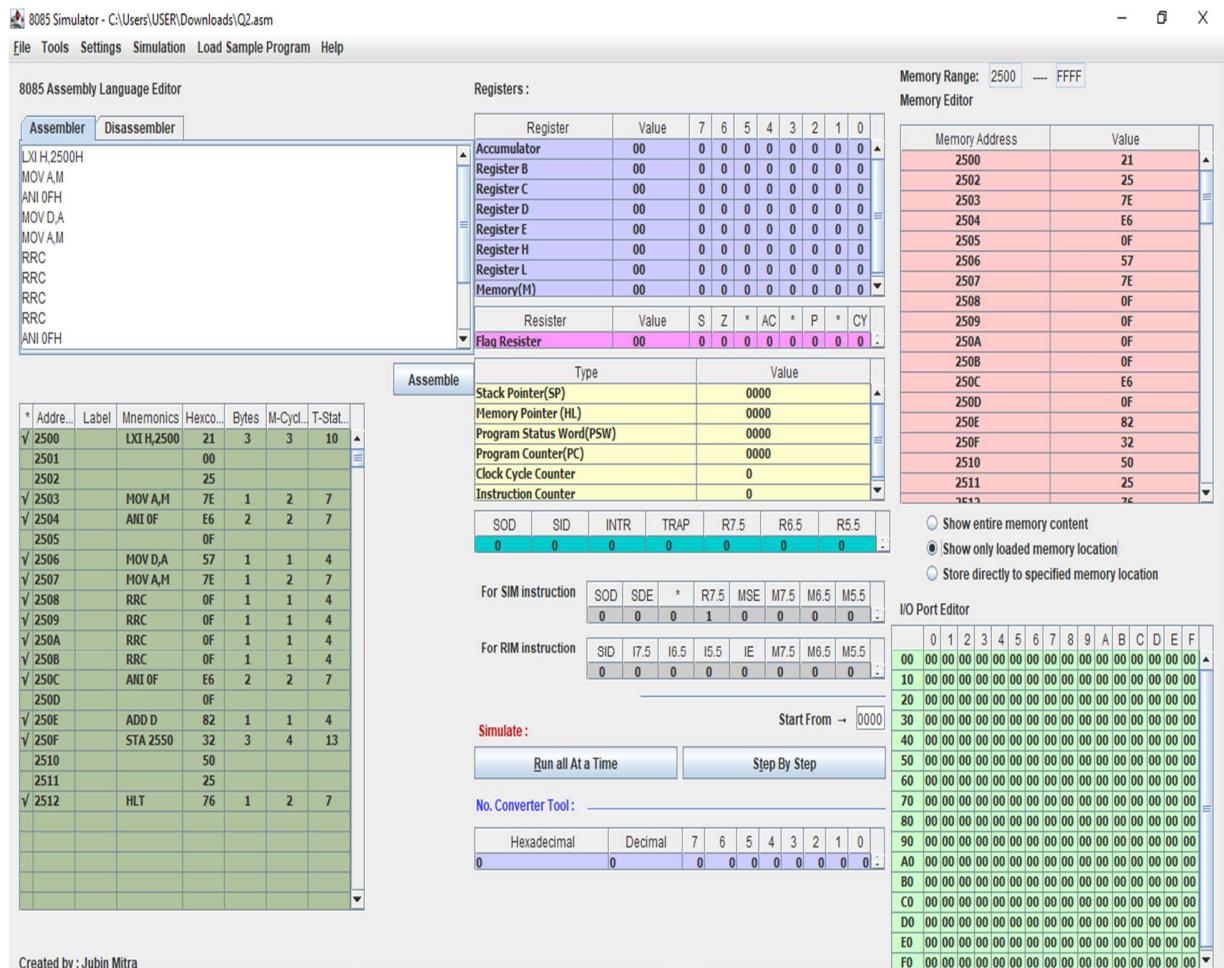
5	2508	06,00		MVI B,00	Move immediate to b register.
6	250A	0E,00		MVI C,00	Move immediate to c register.
7	250C	80	L1	ADD B	Add b register with accumulator.
8	250D	D2,11,25		JNC L2	Jump when carry is O.
9	2510	0C		INRC	Increment c when carry is 1.
10	2511	47	L2	MOV B,A	Move accumulator to b register.
11	2512	23		INX H	Increase HL pair.
12	2513	7E		MOV A,M	Move memory location to accumulator.
13	2514	15		DCR D	Decrement d register.
14	2515	C2,0C,25		JNZ L1	Jump when z flag is 0.
15	2518	78		MOV A,B	Move b register to accumulator
16	2519	32,00,23		STA 2300	Store accumulator memory location 2300
17	251C	79		MOV A,C	Move c register to accumulator
18	251D	32,01,23		STA 2301	Store accumulator memory location 2301
19	2520	76		HLT	Terminate the program.



3. Find the sum of the least significant 4 bits and most significant 4 bits of a byte stored in memory location 2500H. store the result in 2550 H.

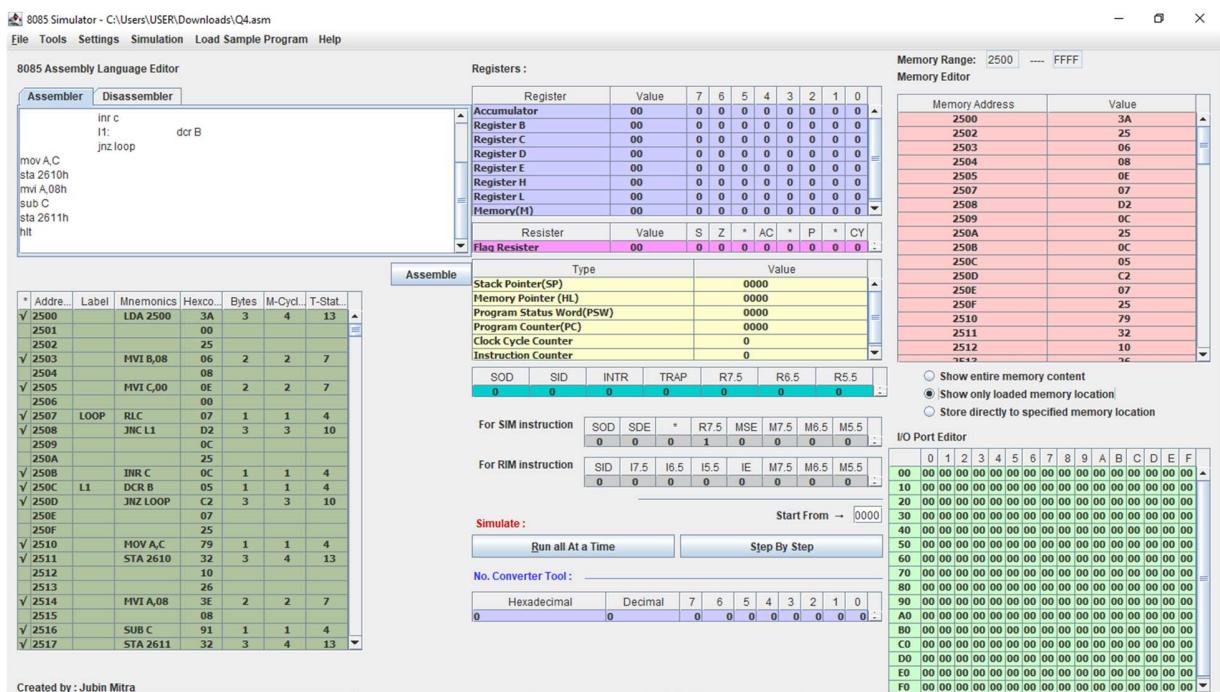
SL NO	ADDRESS	OPCODE IN HEX	LABEL	INSTRUCTION	COMMENTS
1	2500	21,00,25		LXI H,2500H	Contents of memory location 2500H into HL register pair.
2	2503	7E		MOV A,M	Move memory address to accumulator A=M[HL]
3	2504	E6,0F		ANI 0FH	A=A&(0000 1111)
4	2506	57		MOV D,A	D=A
5	2507	7E		MOV A,M	A=M[HL]
6	2508	0F		RRC	Rotate bits of accumulator right without carry bit
7	2509	0F		RRC	Rotate bits of accumulator right without carry bit

8	250A	0F		RRC	Rotate bits of accumulator right without carry bit
9	250B	0F		RRC	Rotate bits of accumulator right without carry bit
10	250C	E6,0F		ANI 0FH	A=A&(0000 1111)
11	250E	82		ADD D	A=A+D
12	250F	32,50,25		STA 2550H	Load the contents of the accumulator in the address location 2550H ,M[2550]
13	2512	76		HLT	Stop the program.



4. Write a program to count the 1's and 0's of a byte stored in 2500 H. Store in 2610 H, and 2511 H, respectively.

SL NO	ADDRESS	OPCODE IN HEX	LABEL	INSTRUCTIONS	COMMENTS
1	2500	21,00,25		LXI H,2500H	Contents of memory location 2500H into HL register pair.
2	2503	7E		MOV A,M	A=M
3	2504	06,08		MVI B,00H	B=08H
4	2506	16,00		MVI D,00H	D=00H
5	2508	07	LOOP	RLC	Rotate accumulator left without carry.
6	2509	D2,0D,00		JNC SKIP	If no carry is generated the jump to label skip
7	250C	14		INR D	D=D+1[To get the one count]
8	250D	05	SKIP	DCR B	B=B-1
9	250E	C2,08,00		JNZ LOOP	If contents of B is not zero then jump to the label LOOP, we need to continue this 8 times to get the count of all set bits
10	2511	7A		MOV A,D	A=D
11	2512	32,10,26		STA 2610H	Load the contents of the accumulator in the address location 2610H, M[2610]=A(store the number of ones)
12	2515	47		MOV B,A	B=A
13	2516	3E,08		MVI A,08H	A=08H
14	2518	90		SUB B	A=A-B(To get a zero count)
15	2519	32,11,25		STA 2511H	Load the contents of the accumulator in the address location 2511H, M[2511]=A(store the number of ones)
16	251C	76		HLT	Stop the program.



5. Write a program to sum two 16 bits binary numbers.

SL NO	ADDRESS	OPCODE IN HEX	LABEL	INSTRUCTIONS	COMMENTS
1	2500	21,00,25		LXI H, 2500H	Contents of memory location 2500H into HL register pair
2	2503	7E		MOV A,M	A=M[HL]]
3	2504	21,02,25		LXI H,2502H	Contents of memory location 2502H into HL register pair
4	2507	46		MOV B,M	B=M[HL]
5	2508	80		ADD B	A=A+B
6	2509	32,10,25		STA 2510	Load the contents of the accumulator in the address location 2510H, M[2510]=A
7	250C	3E,00		MVI A,00H	A=00H
8	250E	8F		ADC A	Add the carry generated
9	250F	32,11,25		STA 2511H	Load the contents of the accumulator in the address location 2511H, M[2511]=A
10	2512	21,01,25		LXI H,2501	Contents of memory location 2501H into HL register pair
11	2515	56		MOV D,M	D=M[HL]
12	2516	82		ADD D	A=A+D
13	2517	21,03,25		LXI H,2503H	Contents of memory location 2503H into HL register pair

14	251A	56		MOV D,M	D=M[HL]
15	251B	82		ADD D	A=A+D
16	251C	32,11,25		STA 2511H	Load the contents of the accumulator in the address location 2511H,M[2511]=A
17	251F	3E,00		MVI A,00H	A=00H
18	2521	8F		ADC A	Add the carry to the accumulator
19	2522	32,12,25		STA 2512H	Load the contents of the accumulator in the address location 2512H,M[2512]=A
20	2525	76		HLT	Stop the program

8085 Simulator - C:\Users\USER\Downloads\Q4.asm

File Tools Settings Simulation Load Sample Program Help

8085 Assembly Language Editor

Assembler **Disassembler**

```
LXI H,2500H
MOV A,M
LXI H,2502
MOV B,M
ADD B
STA 2510
MVI A,00H
ADC A
STA 2511
LXI H,2501
```

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	00	0	0	0	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
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	00	0	0	0	0	0	0	0	0
Register L	00	0	0	0	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	* AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0

Memory Editor

Memory Range: 2500 --- FFFF

Memory Address	Value
2500	21
2502	25
2503	7E
2504	21
2505	02
2506	25
2507	32
2508	80
2509	10
250A	25
250B	3E
250C	8F
250D	32
250E	11
250F	25
2510	21
2511	01
2512	21
2513	01

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

Assemble

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0000
Program Status Word(PSW)	0000
Program Counter(PC)	0000
Clock Cycle Counter	0
Instruction Counter	0

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	1	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

Simulate : Start From → 0000

Run all At a Time **Step By Step**

No. Converter Tool :

Hexadecimal	Decimal	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0

IO Port Editor

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Created by : Jubin Mitra

NAME : **Sujan Biswas**

BATCH: BSCE 2ND year , (LATERAL)

ROLL NO : 302010501003

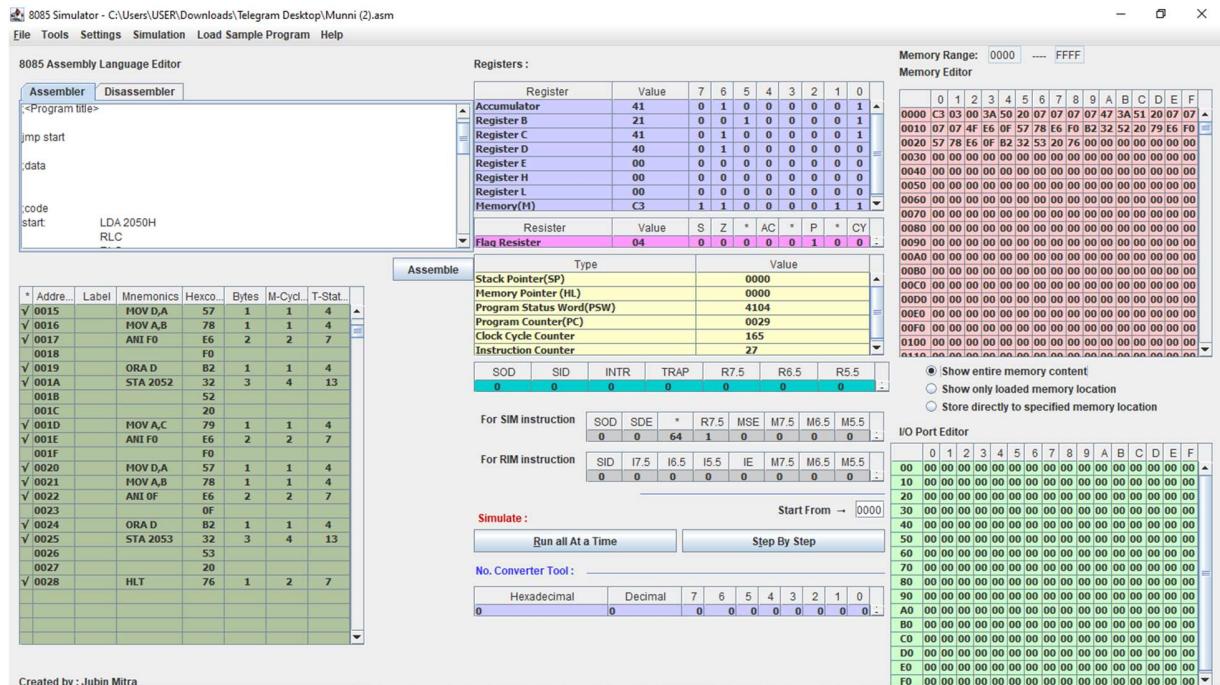
Microprocessor lab Assignment problem sheet #2

- Two numbers MN_H and KL_H are stored in 2050 H and 2051 H , respectively . Write a program to assemble them a NK_H and LM_H store them in 2052 H and 2053 H.

SL NO.	ADDRESS	OPCODE IN HEX CODE	LABEL	INSTRUCTIONS	COMMENTS
1	0000	C3 03 00		JMP START	Start the program
	0001				
	0002				
2	0003	3A	START	LDA2050 H	Load accumulator direct memory location 2050 H [A=MN H]
	0004	50			
	0005	20			
3	0006	07		RLC	Rotate accumulator left without carry
4	0007	07		RLC	Rotate accumulator left without carry
5	0008	07		RLC	Rotate accumulator left without carry
6	0009	07		RLC	Rotate accumulator left without carry [After 4 rotation ,A=NM H]
7	000A	47		MOV B,A	Move accumulator to B register [B=A=NM H]
8	000B	3A		LDA 2051 H	Load accumulator direct memory location 2051 H [A=KL H]
	000C	51			
	000D	20			
9	000E	07		RLC	Rotate accumulator left without carry
10	000F	07		RLC	Rotate accumulator left without carry
11	0010	07		RLC	Rotate accumulator left without carry
12	0011	07		RLC	Rotate accumulator left without carry [After 4 rotation, A= LK H]
13	0012	4F		MOV C,A	Move accumulator to C register [C=A= LK H]
14	0013	E6		ANI OF	A=OK H
	0014	0F			
15	0015	57		MOV D,A	Move accumulator to D register [D=A=OK H]
16	0016	78		MOV A,B	Move B register to accumulator [A=B=NM H]
17	0017	E6		ANI FO	A=NO H
	0018	F0			
18	0019	B2		ORA D	A=NO H OK H=NK H
19	001A	32		STA 2052 H	Load the contents of the accumulator in the address
	001B	52			
	001C	20			

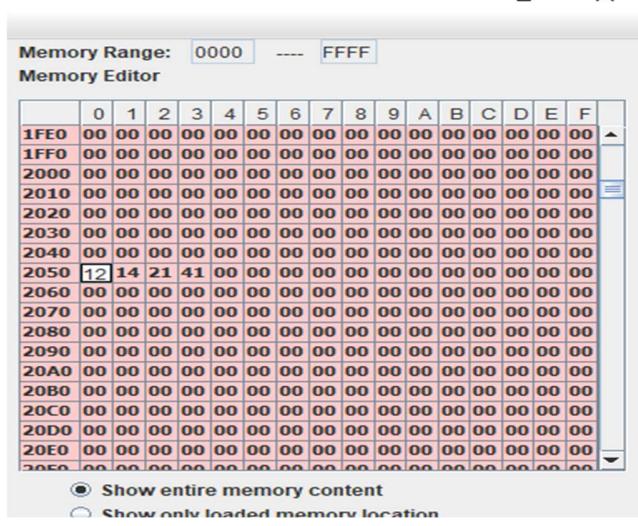
					location 2052 H, M[2052= NK H]
20	001D	79		MOV A,C	Move C register to accumulator [A=C=LK H]
21	001E	E6		ANI F0	A=L0 H
	001F	F0			
22	0020	57		MOV D,A	Move accumulator to D register[A=D=L0 H]
23	0021	78		MOV A,B	Move B register accumulator to[A=B=NM H]
24	0022	E6		ANI OF	A=0M H
	0023	OF			
25	0024	B2		ORA D	A=0M H L0 H=LM H
26	0025	32		STA 2053 H	Load the contents of the accumulator in the address location 2053 H, M[2053=LM H]
	0026	53			
	0027	20			
27	0028	76		HLT	Terminate the program.

Simulator with loaded machine code:



Created by : Jubin Mitra

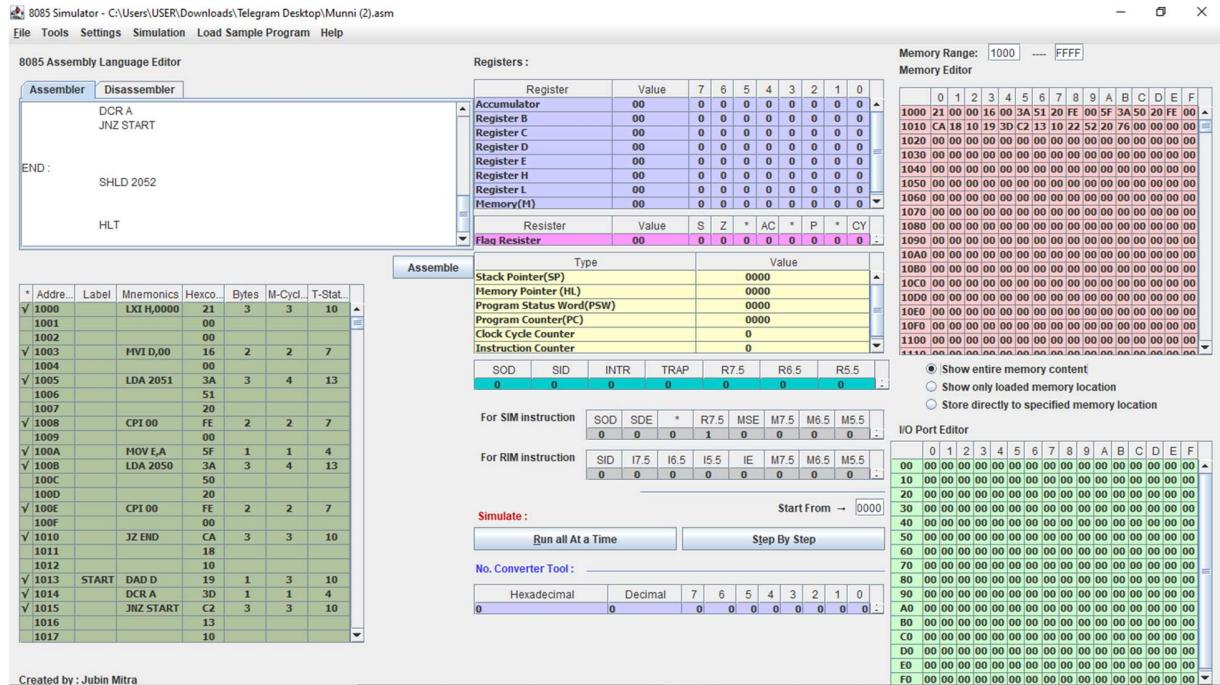
Sample input and output (M[2050 H]=12 H, M[2051 H]=14 H)



2. Two numbers A & B are stored in 2050 H and 2051 H , respectively . Write a program to perform A*B and store the results in 2052 H and 2053 H.

SL NO.	ADDRESS	OPCODE IN HEX CODE	LABEL	INSTRUCTIONS	COMMENTS
1	1000	21		LXI H,0000	Contents of memory location 0000 H into HL register pair
	1001	00			
	1002	00			
2	1003	16		MVI D,00	Move immediate to D register
	1004	00			
3	1005	3A		LDA 2051 H	Load accumulator direct memory location 2051 H Get B
	1006	51			
	1007	20			
4	1008	FE		CPI 00 H	If B=0?
	1009	00			
5	100A	5F		MOV E,A	Move accumulator to E register E=A
6	100B	3A		LDA 2050 H	Load accumulator direct memory location 2050H Get A
	100C	50			
	100D	20			
7	100E	FE		CPI 00 H	Is A=0?
	100F	00			
8	1010	CA		JZ END	If A=0,nothing to do
	1011	18			
	1012	10			
9	1013	19	START	DAD D	HL + DE
10	1014	3D		DCR A	Decrement the A
11	1015	C2		JNZ START	If DE has not been added A times ,add again
	1016	13			
	1017	10			
12	1018	22	END	SHLD 2052 H	Store results as specified
	1019	52			
	101A	20			
13	101B	76		HLT	Stop the program

Simulator with loaded machine code:



Sample input and output (A= M[2050 H]=12 H,B=M[2051 H]=12 H, A*B=144 H, M[2052 H]= 12 H, M[2053 H]=04 H).



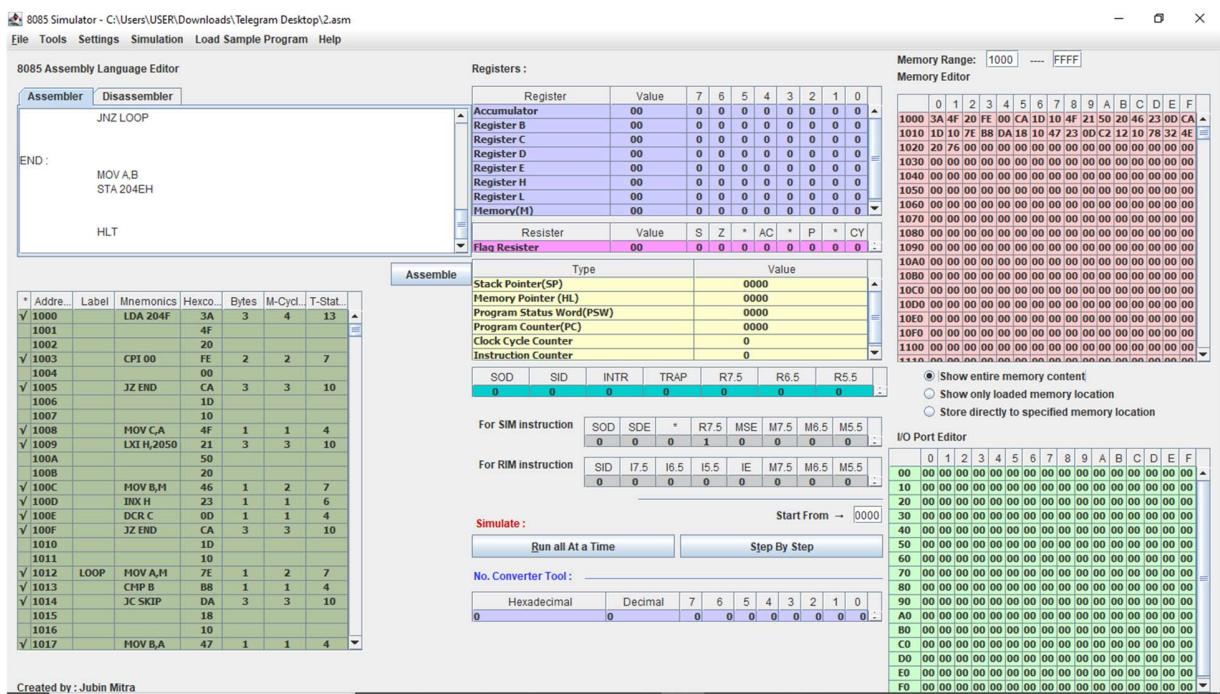
3.N numbers are stored in consecutive m/m location starting the from 2050 H. The value N is stored in 204F H.

I)Find maximum among the N numbers.

SL NO.	ADDRESS	OPCODE IN HEX CODE	LABEL	INSTRUCTIONS	COMMENTS
--------	---------	--------------------	-------	--------------	----------

1	1000	3A		LDA 204F H	Load accumulator direct memory location 204F H [A=N H]
	1001	4F			
	1002	20			
2	1003	FE		CPI 00 H	Is N =0?
	1004	00			
3	1005	CA		JZ END	If N =0,nothing to do
	1006	1D			
	1007	10			
4	1008	4F		MOV C,A	Move accumulator to C register
5	1009	21		LXI 2050 H	Contents of memory location 2050 H into HL register pair
	100A	50			
	100B	20			
6	100C	46		MOV B,M	Move memory address to B register B is current maximum number
7	100D	23		INX H	Increase the HL register ,then go the next number
8	100E	0D		DCR C	Decrement the C register , then check the number
9	100F	CA		JZ END	If end , nothing to do ,store this
	1010	1D			
	1011	7E			
10	1012	7E	LOOP	MOV A,M	Move the memory address to accumulator
11	1013	B8		CMP B	Compare against current maximum
12	1014	DA		JC SKIP	If B>A, do nothing
	1015	18			
	1016	10			
13	1017	47		MOV B,A	Move accumulator to B register load a new maximum
14	1018	23	SKIP	INX H	Increase the HL ,then go the next number
15	1019	0D		DCR C	Decrement the C then check the number
16	101A	C2		JNZ LOOP	If numbers left, continue checking
	101B	12			
	101C	10			
17	101D	78	END	MOV A,B	Move B register to accumulator [A=Maximum]
18	101E	32		STA 204E H	Store the maximum
	101F	4E			
	1020	20			
19	1021	76		HLT	Stop the program

Simulator with loaded machine code:



ii) Find the minimum among the N numbers.

We store the result (Minimum) in 204E H memory location.

SL NO.	ADDRESS	OPCODE IN HEX CODE	LABEL	INSTRUCTIONS	COMMENTS
1	1000	3A		LDA 204F H	Load accumulator direct memory location 204F H [A=N H]
	1001	4F			
	1002	20			
2	1003	FE		CPI 00 H	Is N=0?
	1004	00			
3	1005	CA		JZ END	If N =0,nothing to do
	1006	1D			
	1007	10			
4	1008	4F		MOV C,A	Move accumulator to C register
5	1009	21		LXI H ,2050	Contents of memory location 2050 H into HL register pair
	100A	50			
	100B	20			
6	100C	46		MOV B,M	Move memory address to B register , B is current minimum number
7	100D	23		INX H	Increase the HL register ,then go the next number
8	100E	0D		DCR C	Decrement the C register ,then check the number
9	100F	CA		JZ END	If end ,nothing to do ,store this
	1010	1D			

	1011	10			
10	1012	7E	LOOP	MOV A,M	Move the memory address to accumulator
11	1013	B8		CMP B	Compare against current minimum
12	1014	DA		JC SKIP	If B>A, do nothing
	1015	18			
	1016	10			
13	1017	47		MOV B,A	Move accumulator to B register, load a new minimum
14	1018	23	SKIP	INX H	Increase the HL ,then go to the next number
15	1019	0D		DCR C	Decrement the C register ,then check the number
16	101A	C2		JNZ LOOP	If numbers left, continue checking
	101B	12			
	101C	10			
17	101D	78	END	MOV A, B	Move B register to accumulator [A=Minimum]
18	101E	32		STA 204E H	Store the minimum
	101F	4E			
	1020	20			
19	1021	76		HLT	Stop the program

Simulator with loaded machine code:

The screenshot shows the 8085 Simulator interface with the following details:

- Assembly Language Editor:** Displays the assembly code with labels and addresses.
- Registers:** Shows the state of all 16-bit registers (Accumulator, Register B, Register C, Register D, Register E, Register H, Register L, Memory(M)) and the Flag Register.
- Memory Editor:** Shows the memory dump from address 1000 to FFFF. The memory contains the assembled machine code and initial values for registers and flags.
- I/O Port Editor:** Shows the state of I/O ports from 0 to F.
- Control Buttons:** Includes buttons for "Run all At a Time", "Step By Step", and "No. Converter Tool".

Sample input and output (M[204F H]=06 H, 6 numbers 6A,10,45,7E,B6, and DF,Starting from M[2050 H], M[204E H]=10

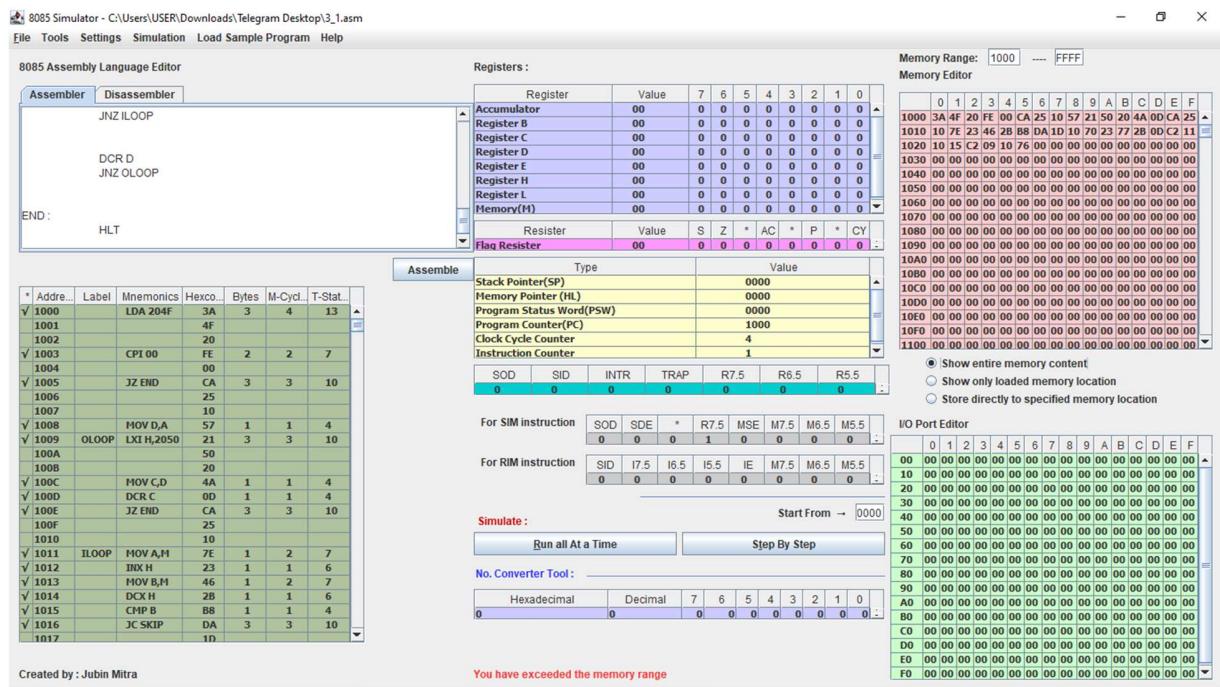
M[204E H]=10

iii) Sort the N numbers in ascending order.

We will be using bubble sort algorithm.

SL NO.	ADDRESS	OPCODE IN HEX CODE	LABEL	INSTRUCTIONS	COMMENTS
1	1000	3A		LDA 204F H	Get N
	1001	4F			
	1002	20			
2	1003	FE		CPI 00 H	Is N =?
	1004	00			
3	1005	CA		JZ END	Is N =0, nothing to do
	1006	25			
	1007	10			
4	1008	57		MOV D,A	D=outer loop counter
5	1009	21	OLOOP	LXI H ,2050	Contents of memory location 2050 H into HL register pair
	100A	50			
	100B	20			
6	100C	4A		MOV C,D	C=inner loop counter
7	100D	0D		DCR C	Numbers of comparisons is 1 less than the length
8	100E	CA		JZ END	If no comparisons are to be made ,do nothing
	100F	25			
	1010	10			
9	1011	7E	ILOOP	MOV A,M	Get first number
10	1012	23		INX H	Go to next number
11	1013	46		MOV B,M	Get second number
12	1014	2B		DCX H	Go back to current position
13	1015	B8		CMP B	Compare 2 nd number against 1 st number
14	1016	DA		JC SKIP	If 2 nd number>1 st number, do nothing
	1017	1D			
	1018	10			
15	1019	70		MOV M,B	Put 2 nd number first
16	101A	13		INX H	Go to next location
17	101B	77		MOV M,A	Put 1 st number second
18	101C	2B		DCX H	Go back to previous location
19	101D	0D	SKIP	DCR C	Comparison done
20	101E	C2		JNZ ILOOP	Start from the next location
	101F	11			
	1020	10			
21	1021	15		DCR D	One pass finished
22	1022	C2		JNZ OLOOP	Go to the next pass
	1023	09			
	1024	10			
23	1025	76		HLT	Stop the program

Simulator with loaded machine code:



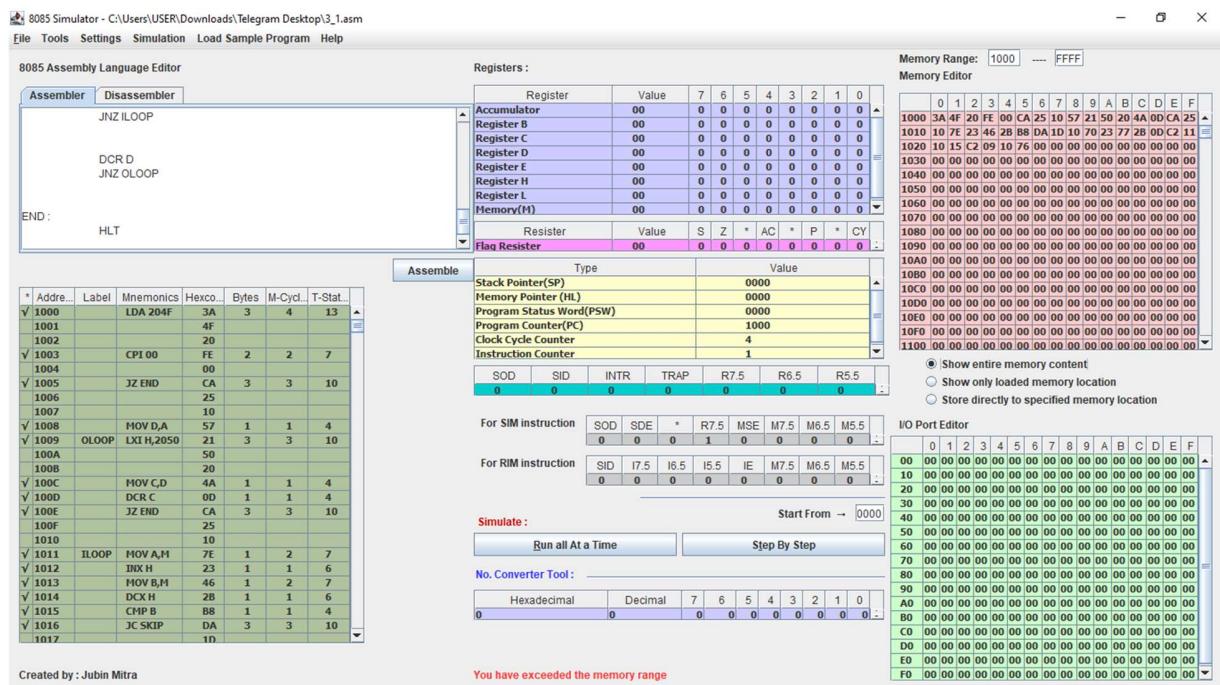
iv) Sort the N numbers in descending order.

We will be using bubble sort algorithm.

SL NO.	ADDRESS	OPCODE IN HEX CODE	LABEL	INTRUCTIONS	COMMENTS
1	1000	3A		LDA 204F H	Get N

	1001	4F			
	1002	20			
2	1003	FE		CPI 00 H	Is N =?
	1004	00			
3	1005	CA		JZ END	Is N =0,nothing to do
	1006	25			
	1007	10			
4	1008	57		MOV D,A	D=outer loop counter
5	1009	21	OLOOP	LXI H ,2050	Contents of memory location 2050 H into HL register pair
	100A	50			
	100B	20			
6	100C	4A		MOV C,D	C=inner loop counter
7	100D	0D		DCR C	Numbers of comparisons is 1 less than the length(D)
8	100E	CA		JZ END	If no comparisons are to be made ,do nothing
	100F	25			
	1010	10			
9	1011	7E	ILOOP	MOV A,M	Get first number
10	1012	23		INX H	Go to next number
11	1013	46		MOV B,M	Get second number
12	1014	2B		DCX H	Go back to current position
13	1015	B8		CMP B	Compare 2 nd number against 1 st number
14	1016	DA		JC SKIP	If 2 nd number<1 st number, do nothing
	1017	1D			
	1018	10			
15	1019	70		MOV M,B	Put 2 nd number first
16	101A	13		INX H	Go to next location
17	101B	77		MOV M,A	Put 1 st number second
18	101C	2B		DCX H	Go back to previous location
19	101D	0D	SKIP	DCR C	Comparison done
20	101E	C2		JNZ ILOOP	Start from the next location
	101F	11			
	1020	10			
21	1021	15		DCR D	One pass finished
22	1022	C2		JNZ OLOOP	Go to the next pass
	1023	09			
	1024	10			
23	1025	76		HLT	Stop the program

Simulator with loaded machine code:

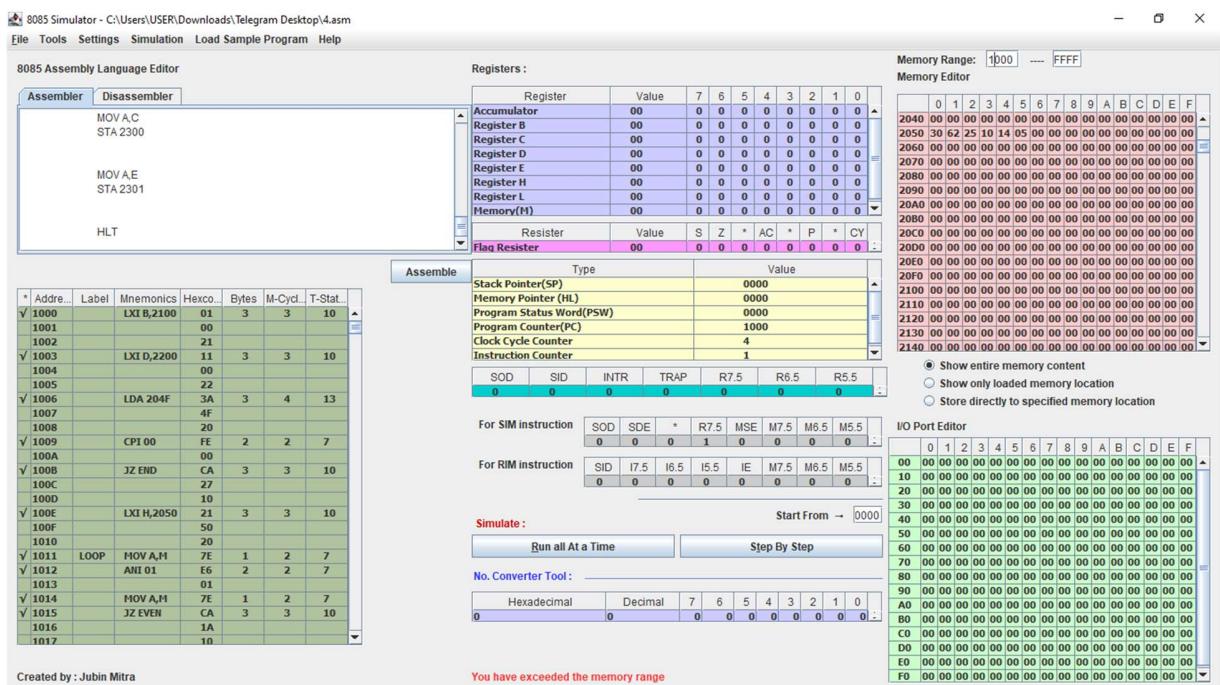


4.N numbers are stored in consecutive m/m location starting from 2050 H. The value N is stored in 204F H. Write a program to copy the even and odd numbers starting from 2100 H and 2200 H, respectively. Store the total number of even and odd numbers in 2300 H and 2301 H, respectively

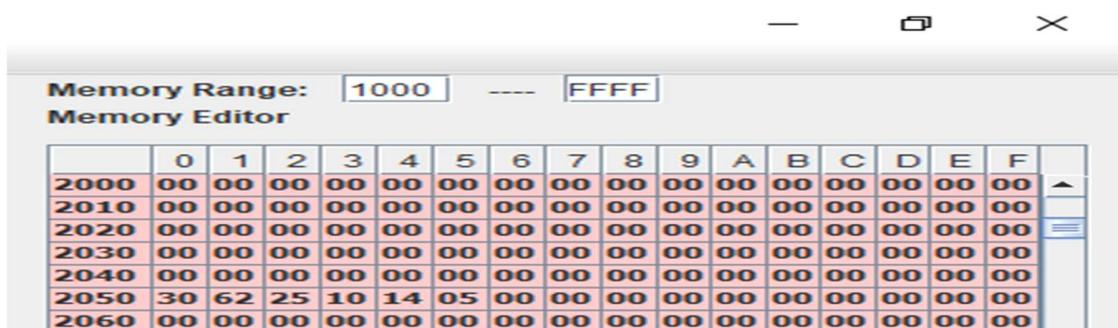
SL NO.	ADDRESS	OPCODE IN HEX CODE	LABEL	INSTRUCTIONS	COMMENTS
1	1000	01		LXI B , 2100	Starting address of even numbers
	1001	00			
	1002	21			
2	1003	11		LXI D,2200	Starting address of odd numbers
	1004	00			
	1005	22			
3	1006	3A		LDA 204F	Get N
	1007	4F			
	1008	20			
4	1009	FE		CPI 00 H	Is N =0?

	100A	00			
5	100B	CA		JZ END	If N=0, nothing to do
	100C	27			
	100D	10			
6	100E	21		LXI H ,2050	Contents of memory location 2050 H into HL register pair/input
	100F	50			
	1010	20			
7	1011	7E	LOOP	MOV A, M	Get current numbers
8	1012	E6		ANI 01	Check odd or not
	1013	01			
9	1014	7E		MOV A, M	Restore the number
10	1015	CA		JZ EVEN	If 0, then even
	1016	1A			
	1017	10			
11	1018	12		STAX D	This is an odd number
12	1019	13		INX D	One odd number added
13	101A	02	EVEN	STAX B	This is an even number
14	101B	03		INX B	One even number added
15	101C	23	LEND	INX H	Go to the next address
16	101D	3A		LDA 204F	Get n
	101E	4F			
	101F	20			
17	1020	3D		DCR A	One number checked
18	1021	32		STA 204F	Store N for later use
	1022	4F			
	1023	20			
19	1024	C2		JNZ LOOP	If numbers left ,continue
	1025	11			
	1026	10			
20	1027	79	END	MOV A,C	A=C=number of even number because BC started from 2100 H
21	1028	32		STA 2300 H	Store numbers of even number
	1029	00			
	102A	23			
22	102B	7B		MOV A,E	A=E=number of odd numbers because DE started from 2200 H
23	102C	32		STA 2301 H	Store number of odd numbers
	102D	01			
	102E	23			
24	102F	76		HLT	Stop the program

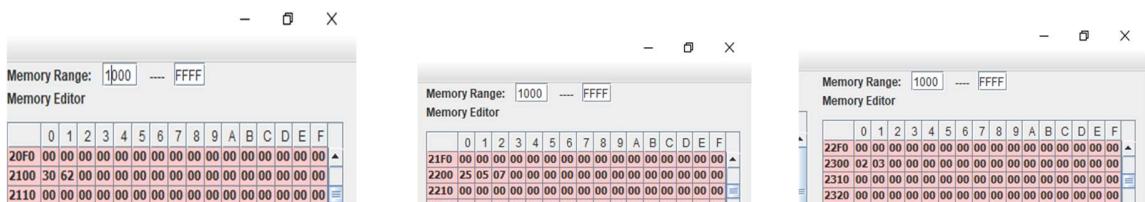
Simulator with loaded machine code:



Sample input (M[204F H]=06 H, 6 numbers 30 H,62 H,25 H,10 H,24 H and 05 H starting from M[2050 H]):



Sample output (2 even numbers 30 H, 62 H starting from M[2100 H], 3 odd numbers 25 H,05 H, 07 H starting from M[2200 H], M[2300 H]=02 H,M[2301]=03 H):

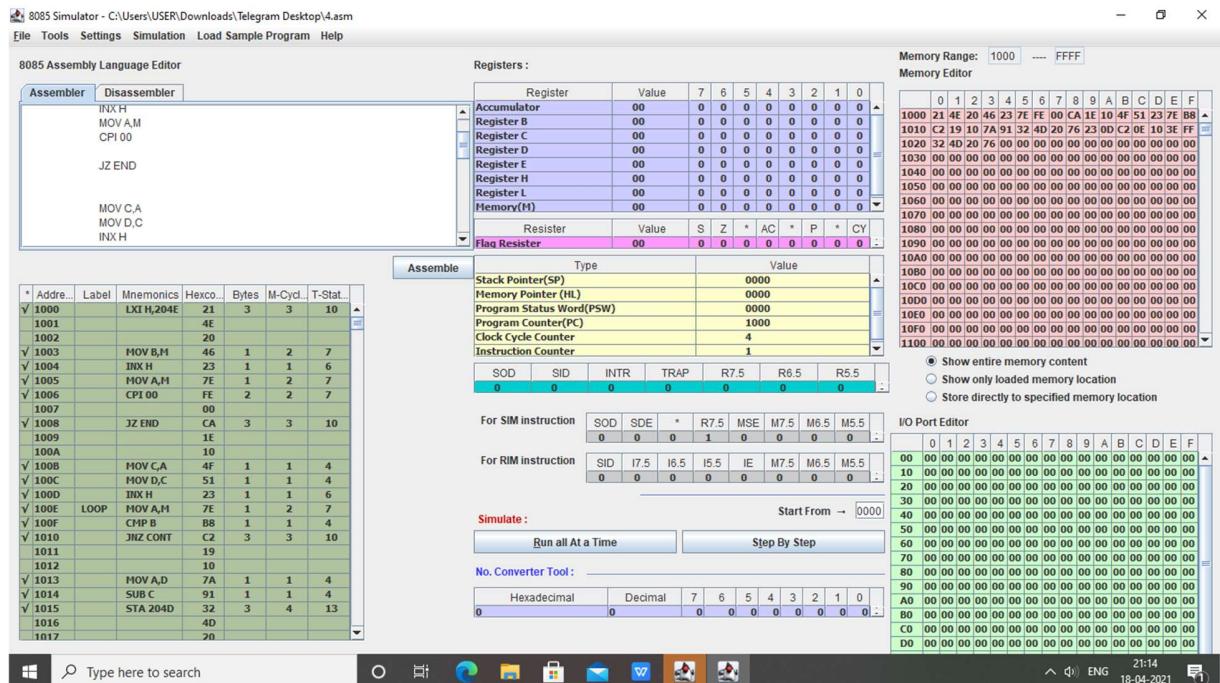


5. N numbers are stored in consecutive m/m location starting from 2050 H. The value N is stored in 204E H. Write a program to test whether a number stored in 204E H is present in the list. If present , store its position in the list at 204D H ;otherwise store FF H.

We will give position by a zero – based index.

SL NO.	ADDRESS	OPCODE IN HEX CODE	LABEL	INSTRUCTIONS	COMMENTS
1	1000	21		LXI H, 204E	B=Numbers to search (key)
	1001	4E			
	1002	20			
2	1003	46		MOV B,M	Move memory location to B register
3	1004	23		INX H	HL=204F H
4	1005	7E		MOV A,M	Get N
5	1006	FE		CPI 00 H	Is N=0?
	1007	00			
6	1008	CA		JZ END	If N=0,do not search
	1009	1E			
	100A	10			
7	100B	4F		MOV C,A	Move accumulator to C register [C=A=Counter]
8	100C	51		MOV D,C	Move C register to D register[D=total no of items]
9	100D	23		INX H	HL=2050 H=Starting address of input
10	100E	7E	LOOP	MOV A,M	Move memory address to accumulator, load current number
11	100F	B8		CMP B	Compare against key
12	1010	C2		JNZ CONT	If unequal ,continue loop
	1011	19			
	1012	10			
13	1013	7A		MOV A,D	A=D=Totals number of items
14	1014	91		SUB C	A=position of key in list
15	1015	32		STA 204D	Store found position
	1016	4D			
	1017	20			
16	1018	76		HLT	We have nothing to do anymore
17	1019	23		INX H	Go to next number
18	101A	0D		DCR C	Checked one number
19	101B	C2		JNZ LOOP	If number left,continue
	101C	0E			
	101D	10			
20	101E	3E	END	MVI A,FF	We haven't found key
	101F	FF			
21	1020	32		STA 204D	Store the number
	1021	4D			
	1022	20			
22	1023	76		HLT	Stop the program.

Simulator with loaded machine code:



NAME : **Sujan Biswas**

BATCH: BSCE 2ND year , (LATERAL)

ROLL NO : 302010501003

Microprocessor lab Assignment problem sheet #3

Problem 1: A set of N data bytes is stored in m/m locations starting from 2501H. The value of N is stored in 2500H. Write a program to store these data bytes from m/m location 2600H if D0 or D7 is 1; otherwise reject the data byte.

Solution:

Line #	Address in Hex	Label	Instruction (Mnemonics)	Opcode in Hex	Remarks
1.	0000		LXI H, 2600	21	HL = 2600
2.	0001			00	
3.	0002			26	
4.	0003		XCHG	EB	swap(DE, HL)
5.	0004		LXI H, 2500	21	Now HL = 2500
6.	0005			00	
7.	0006			25	
8.	0007		MOV B, M	46	
9.	0008	LOOP	INX H	23	HL = HL + 1
10.	0009		MOV A, M	7E	contents of the memory stored into the accumulator
11.	000A		RRC	OF	rotate the content of the accumulator by the affecting the conditions of the Carry Bit
12.	000B		JNC COND2	D2	If no carry is generated then jump to condition 2
13.	000C			16	
14.	000D			00	
15.	000E		RLC	07	rotate the contents of the accumulator to the left to restore the number back
16.	000F		XCHG	EB	SWAP(HL, DE)

17.	0010		MOV M, A	77	contents of the HL register pair in A
18.	0011		INX H	23	HL = HL + 1
19.	0012		XCHG	EB	SWAP(DE, HL)
20.	0013		JMP SKIP	C3	if condition 1 satisfied no need to check condition 2
21.	0014			20	
22.	0015			00	
23.	0016	COND2	RLC	07	rotate the contents of the accumulator to restore the number
24.	0017		RLC	07	check the D7 bit
25.	0018		JNC SKIP	D2	if the second condition also fails jump to skip
26.	0019			20	
27.	001A			00	
28.	001B		RRC	0F	rotate the accumulator right to retrieve the number back
29.	001C		XCHG	EB	SWAP(DE, HL)
30.	001D		MOV M, A	77	M = A;
31.	001E		INX H	23	HL = HL + 1
32.	001F		XCHG	EB	SWAP(DE, HL)
33.	0020	SKIP	DCR B	05	decrement the loop counter
34.	0021		JNZ LOOP	C2	continue till no numbers are left
35.	0022			08	
36.	0023			00	
36.	0024		HLT	76	stop all operations

Sample input and output(M[2500 H] = N = 05 H , 5 numbers 76 H , 73 H , 02 H , DF H and EE H starting from M[2501 H], 3 numbers 73 H , DF H and EE H starting from M[2600 H]):

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Assembler

Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
0014			20			
0015			00			
✓ 0016	COND2	RLC	07	1	1	4
✓ 0017		RLC	07	1	1	4
✓ 0018		JNC SKIP	D2	3	3	10
0019			20			
001A			00			
✓ 001B		RRC	0F	1	1	4
✓ 001C		XCHG	EB	1	1	4
✓ 001D		MOV M,A	77	1	2	7
✓ 001E		INX H	23	1	1	6
✓ 001F		XCHG	EB	1	1	4
✓ 0020	SKIP	DCR B	05	1	1	4
✓ 0021		JNZ LOOP	C2	3	3	10
0022			08			
0023			00			
✓ 0024		HLT	76	1	2	5

Registers Memory Devices

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
0010	77
0011	23
0012	EB
0013	C3
0014	20
0016	07
0017	07
0018	D2
0019	20
001B	0F
001C	EB
001D	77
001E	23
001F	EB
0020	05
0021	C2
0022	08
0024	76
2500	05
2501	76
2502	73
2503	02
2504	DF
2505	EE
2600	73
2601	DF
2602	EE

Simulate

Start From → 0000

Run all At a Time Step By Step

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

Created by : Jubin Mitra

Problem 2: There are N data bytes stored from m/m location 2200H. The value of N is stored in 21FFH. Write an 8085 program to find the sum of integers whose LSB and MSB are 1. Store the result in 2500H and 2501H.

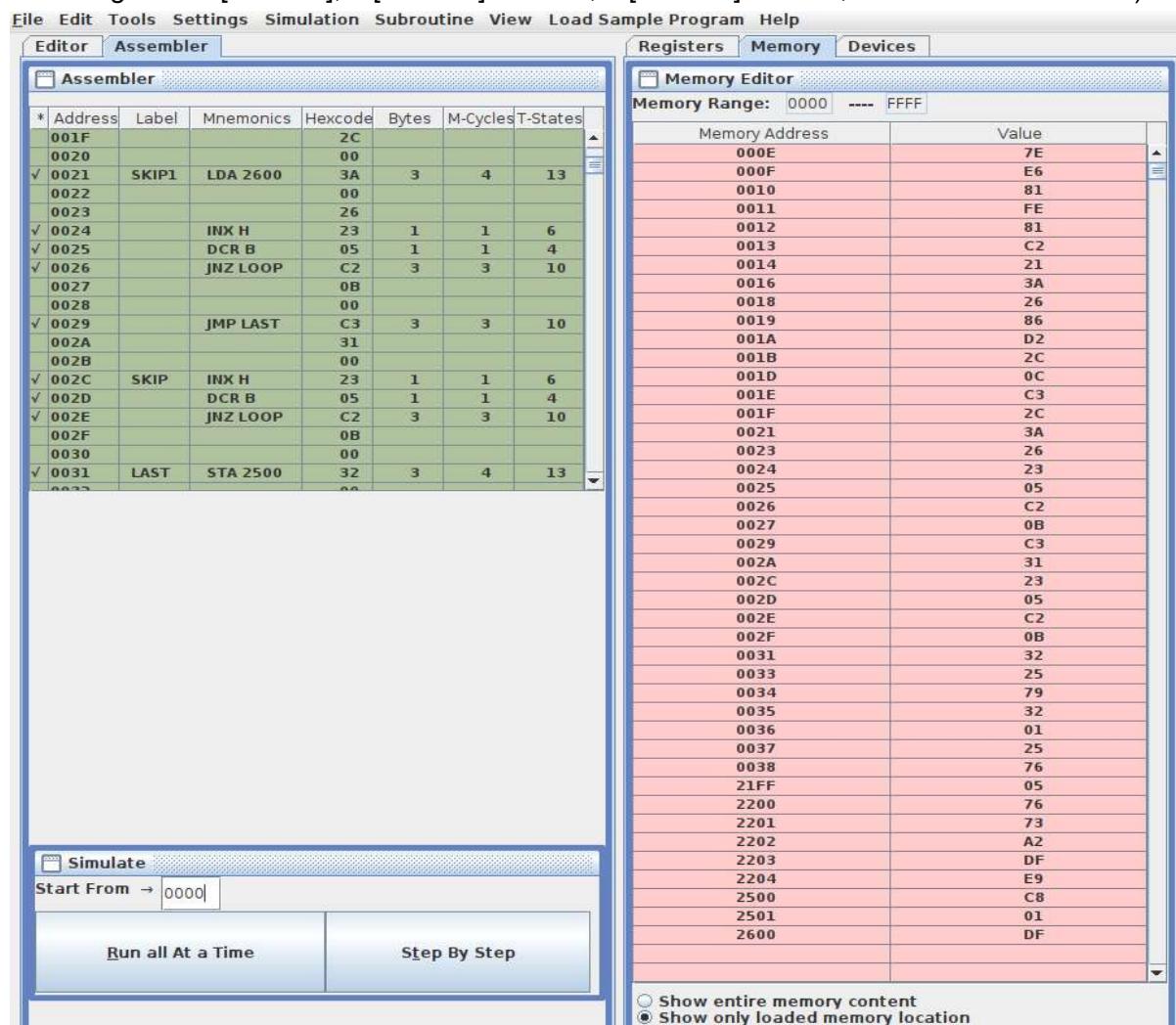
Solution:

Line #	Address in Hex	Label	Instruction (Mnemonics)	Opcode in Hex	Remarks
1.	0000		LXI H, 21FF	21	HL = 21FF
2.	0001			FF	
3.	0002			21	
4.	0003		MOV B, M	46	B = M
5.	0004		LXI H, 2200	21	HL = 2200
6.	0005			00	
7.	0006			22	
8.	0007		MVI C, 00	0E	C = 0
9.	0008			00	
10.	0009		MVI A, 00	3E	A = 0
11.	000A			00	
12.	000B	LOOP	STA 2600	32	A = M[2600]
13.	000C			00	
14.	000D			26	
15.	000E		MOV A, M	7E	A = contents of memory specified by HL register pair
16.	000F		ANI 81	E6	A = A & (1000 0001) to check if lsb and msb is one or not
17.	0010			81	
18.	0011		CPI 81	FE	compare to check D7 and D0 is one or not
19.	0012			81	
20.	0013		JNZ SKIP1	C2	if not satisfied move to skip
21.	0014			21	
22.	0015			00	
23.	0016		LDA 2600	3A	A = contents of memory location 2600 used as dummy m/m
24.	0017			00	

25.	0018			26	
26.	0019		ADD M	86	if condition is satisfied perform addition
27.	001A		JNC SKIP	D2	Ig no carry is present move to skip
28.	001B			2C	
29.	001C			00	
30.	001D		INR C	0C	increment the carry
31.	001E		JMP SKIP	C3	move to skip
32.	001F			2C	
33.	0020			00	
34.	0021	SKIP1	LDA 2600	3A	A = contents of memory location 2600 used as dummy m/m
35.	0022			00	
36.	0023			26	
37.	0024		INX H	23	HL = HL + 1
38.	0025		DCR B	05	B = B - 1
39.	0026		JNZ LOOP	C2	continue till last number
40.	0027			0B	
41.	0028			00	
42.	0029		JMP LAST	C3	no number present the jump to last
43.	002A			31	
44.	002B			00	
45.	002C	SKIP	INX H	23	HL = HL + 1
46.	002D		DCR B	05	B = B - 1
47.	002E		JNZ LOOP	C2	continue till last number
48.	002F			0B	
49.	0030			00	
50.	0031	LAST	STA 2500	32	store the sum in m/m 2500

51.	0032			00	
52.	0033			25	
53.	0034		MOV A, C	79	move the carry
54.	0035		STA 2501	32	store the MSB in m/m 2501
55.	0036			01	
56.	0037			25	
57.	0038		HLT	76	stop all operations

Sample input and output(M[21FF H] = N = 05 H , 5 numbers 76 H , 73 H , A2 H , DF H and E9 H starting from M[2501 H] , M[2500 H] = C8 H , M[2501 H] = 01 H , DF H + E9 H = 1C8 H):



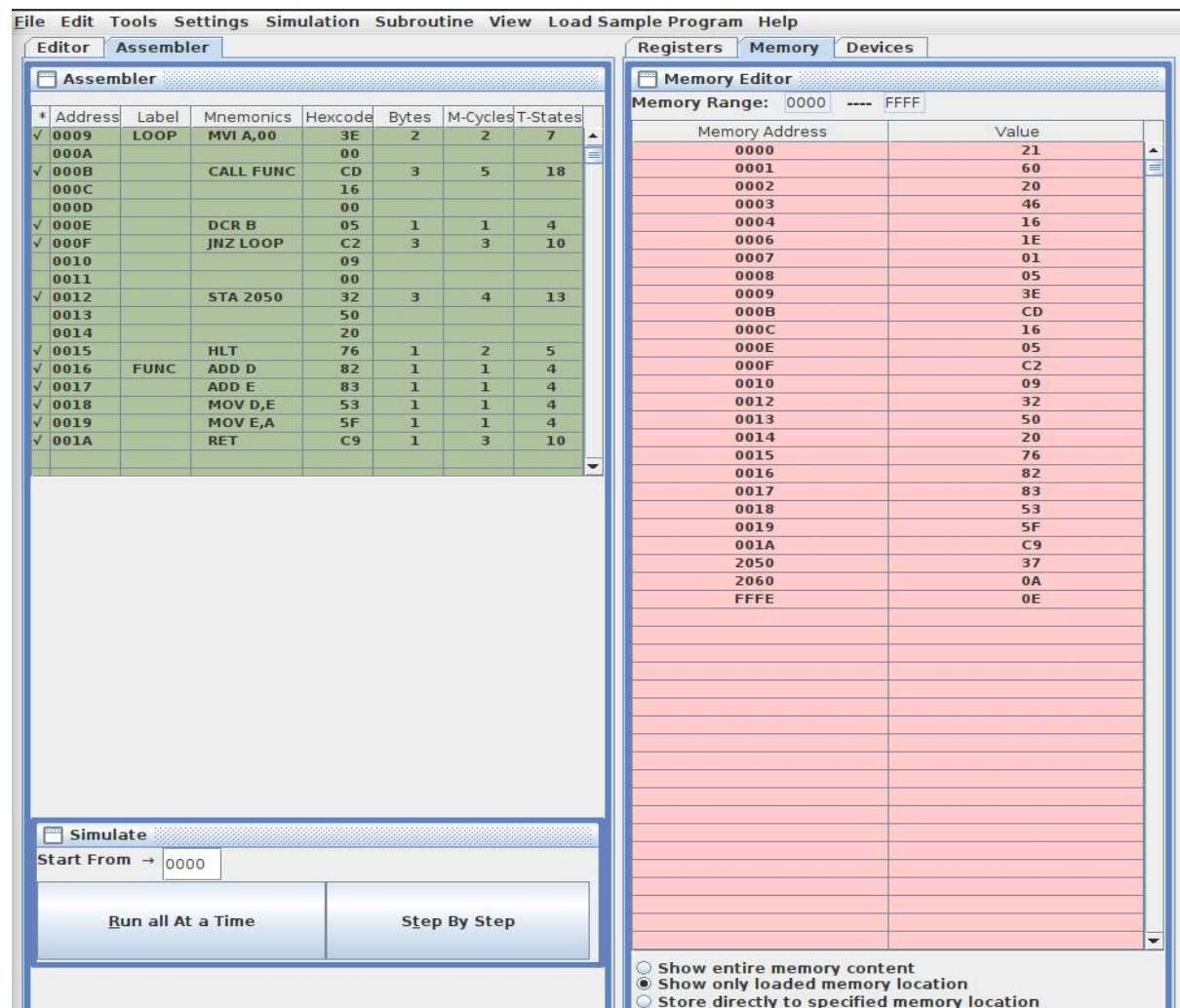
Problem 3: Write an 8085 program to generate Nth fibonacci number using function and store it in 2050H. The value of N (8-bits) is stored in memory 2060H.

Solution :-

Line #	Address in Hex	Label	Instruction (Mnemonics)	Opcode in Hex	Remarks
1.	0000		LXI H, 2060	21	Load the value of N in HL register pair
2.	0001			60	
3.	0002			20	
4.	0003		MOV B, M	46	B = N
5.	0004		MVI D, 00	16	initialise the value of D with zero
6.	0005			00	
7.	0006		MVI E, 01	1E	initialise the value of the register E with 1
8.	0007			01	
9.	0008		DCR B	05	B = B - 1
10.	0009	LOOP	MVI A, 00	3E	initialise the accumulator with zero value
11.	000A			00	
12.	000B		CALL FUNC	CD	make a call to the function to evaluate the fibonacci number
13.	000C			16	
14.	000D			00	
15.	000E		DCR B	05	B = B - 1
16.	000F		JNZ LOOP	C2	continue till nth fibonacci is not evaluated
17.	0010			09	
18.	0011			00	
19.	0012		STA 2050	32	store the contents of the accumulator in the m/m location 2050
20.	0013			50	
21.	0014			20	

22.	0015		HLT	76	Stop all operations
23.	0016	FUNC	ADD D	82	A = A +D
24.	0017		ADD E	83	A = A +E
25.	0018		MOV D, E	53	D = E
26.	0019		MOV E, A	5F	E = A
27.	001A		RET	C9	return back to the statement where function call was made

Sample input and output(M[2060 H] = N = 0A H = 10 10 , M[2050 H] = FIB(0A H) = 37 H = 55 10):



Problem 4: Write a program to transfer a block of bytes of size N from location1 to location2 (location2 > location1) when the size of overlap between the two locations is defined by M. The values of N and M are stored in 201EH and 201FH, respectively.

Solution :-

Line #	Address in Hex	Label	Instruction (Mnemonics)	Opcode in Hex	Remarks
1.	0000		LDA 201E	3A	A = N
2.	0001			1E	
3.	0002			20	
4.	0003		CPI 00	FE	compare the contents of the accumulator with zero
5.	0004			00	
6.	0005		JZ END	CA	If N == 0 no need to perform any operation
7.	0006			33	
8.	0007			00	
9.	0008		LDA 201E	3A	A = N
10.	0009			1E	
11.	000A			20	
12.	000B		MOV C, A	4F	C = A
13.	000C		LXI H, 201F	21	store the m/m location where the overlapping size is present
14.	000D			1F	
15.	000E			20	
16.	000F		MOV B, M	46	B = M
17.	0010		SUB B	90	A = A - B
18.	0011		JZ END	CA	if N - M == 0 we need to perform no operation
19.	0012			33	
20.	0013			00	

21.	0014		JC END	DA	if $N - M < 0$, it's an invalid input
22.	0015			33	
23.	0016			00	
24.	0017		MOV E, A	5F	E = N - M
25.	0018		LHLD 2020	21	HL = location1
26.	0019			20	
27.	001A			20	
28.	001B		MVI D, 00	16	DE = N - M
29.	001C			00	
30.	001D		DAD D	19	HL = location1 + (N - M)
31.	001E		XCHG	EB	DE = location1 + (N - M)
32.	001F		LHLD 2020	2A	HL = location1
33.	0020			20	
34.	0021			20	
35.	0022		DCR C	0D	C = N - 1
36.	0023		MVI B, 00	06	BC = N - 1
37.	0024			00	
38.	0025		DAD B	09	HL = end of location1
39.	0026		XCHG	EB	DE = end of location1
40.	0027		DAD B	09	HL = end of location2
41.	0028		INR C	0C	C = N
42.	0029	LOOP	XCHG	EB	SWAP(DE, HL)
43.	002A		MOV A, M	7E	contents moved into accumulator
44.	002B		XCHG	EB	SWAP(DE, HL)
45.	002C		MOV M, A	7E	store the contents into desired memory location from backwards

46.	002D		DCX H	2B	HL = HL - 1
47.	002E		DCX D	1B	DE = DE - 1
48.	002F		DCR C	0D	decrement the value of loop counter
49.	0030		JNZ LOOP	C2	continue till a single element is left to transfer
50.	0031			29	
51.	0032			00	
52.	0033	END	HLT	76	stop all operations

Sample input(M[201E H] = 05 H , M[201F H] = 03 H , M[2020 H] = 30 H , M[2021 H] = 20 H , location1 = 2030 H , location2 = 2032 H , 5 numbers 0A H , 0B H , 0C H , 0D H and 0E H starting from M[2030 H]):

The screenshot shows a Z80 assembly debugger interface with three main tabs: Assembler, Registers, and Memory.

Assembler Tab: Displays the assembly code with columns for Address, Label, Mnemonics, Hexcode, Bytes, M-Cycles, and T-States. The code includes instructions like DCR C, MVI B,00, DAD B, XCHG, INR C, MOV A,M, MOV M,A, DCX H, DCX D, DCR C, JNZ LOOP, END, and HLT.

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓	0022		DCR C	0D	1	1	4
✓	0023		MVI B,00	06	2	2	7
	0024			00			
✓	0025		DAD B	09	1	3	10
✓	0026		XCHG	EB	1	1	4
✓	0027		DAD B	09	1	3	10
✓	0028		INR C	0C	1	1	4
✓	0029	LOOP	XCHG	EB	1	1	4
✓	002A		MOV A,M	7E	1	2	7
✓	002B		XCHG	EB	1	1	4
✓	002C		MOV M,A	77	1	2	7
✓	002D		DCX H	2B	1	1	6
✓	002E		DCX D	1B	1	1	6
✓	002F		DCR C	0D	1	1	4
✓	0030		JNZ LOOP	C2	3	3	10
	0031			29			
	0032			00			
✓	0033	END	HLT	76	1	2	5

Registers Tab: Shows the CPU registers with their current values: R1=20, R2=46, R3=90, R4=CA, R5=33, R6=DA, R7=33, R8=5F, R9=2A, R10=20, R11=16, R12=19, R13=EB, R14=2A, R15=20, R16=20, R17=0D, R18=06, R19=09, R20=EB, R21=09, R22=0C, R23=7E, R24=EB, R25=77, R26=2B, R27=1B, R28=0D, R29=C2, R30=29, R31=05, R32=03, R33=30, R34=20, R35=01, R36=02.

Memory Tab: Shows the memory content from address 0000 to FFFF. The memory starts with 000E=20, 000F=46, followed by a series of zeros (0010-001F), then values 2A, 20, 16, 19, EB, 2A, 20, 20, 0D, 06, 09, EB, 09, 0C, EB, 7E, EB, 77, 2B, 1B, 0D, C2, 29, 05, 03, 30, 20, 01, 02, 01, 01, 02, 01, 03, 04, 05.

Simulate Tab: Contains buttons for "Run all At a Time" and "Step By Step".

Bottom Options:

- Show entire memory content
- Show only loaded memory location
- Store directly to specified memory location

