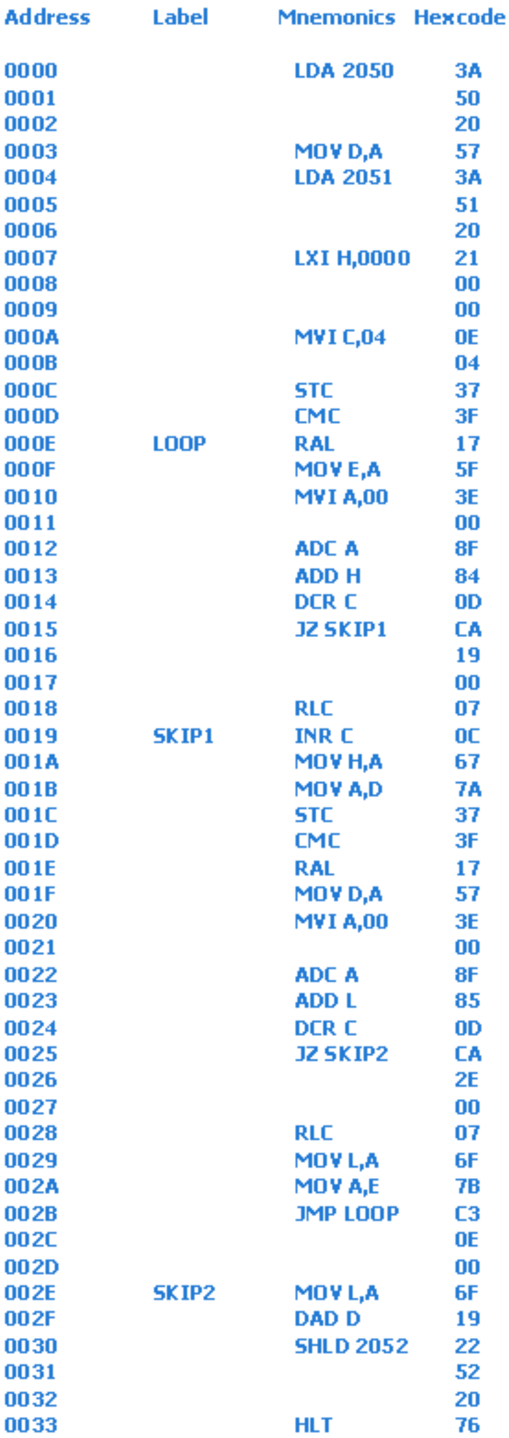
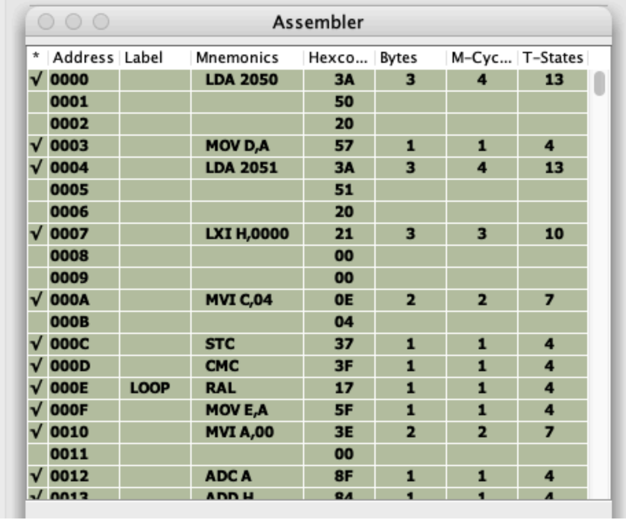
Name : Ritabroto Ganguly
Roll : 001910501090 BCSE-II

8085 MPU Problem Sheet #2

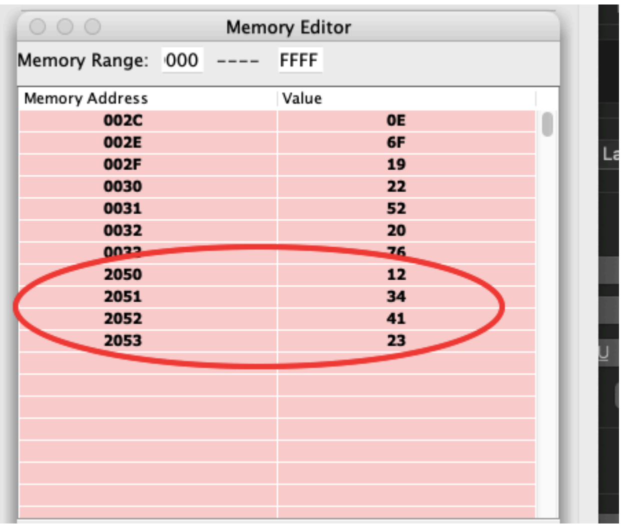
1. Two numbers MN_H and KL_H are stored in 2050_H and 2051_H, respectively. Write a program to assemble them as NK_H and LM_H store them in 2052_H and 2053_H.



Address	Label	Mnemonics	Hexcode	
0000		LDA 2050	3A	A = MN
0001			50	
0002			20	D = A = MN
0003		MOV D,A	57	A = KL
0004		LDA 2051	3A	
0005			51	
0006			20	
0007		LXI H,0000	21	HL = 0000H
0008			00	
0009			00	C = 04H is the counter for loop
000A		MVI C,04	0E	
000B			04	set carry flag 1
000C		STC	37	complement carry flag to make it 0
000D		CMC	3F	Rotate A left, D7 comes to Carry flag and carry flag = 0 comes to D0
000E	LOOP	RAL	17	E = A
000F		MOV E,A	5F	A = 00H
0010		MVI A,00	3E	
0011			00	
0012		ADC A	8F	A = Carry Flag
0013		ADD H	84	H will store NK
0014		DCR C	0D	Decrement C to check if C was 1 or not next
0015		JZ SKIP1	CA	
0016			19	
0017			00	Rotate A left so that next bit from KL can be put in D0 of NK
0018		RLC	07	restore C after testing
0019	SKIP1	INR C	0C	H = A
001A		MOV H,A	67	A = D
001B		MOV A,D	7A	
001C		STC	37	Carry flag = 1
001D		CMC	3F	complement carry flag to make it 0
001E		RAL	17	Rotate A left, D7 comes to Carry flag and carry flag = 0 comes to D0
001F		MOV D,A	57	D = A
0020		MVI A,00	3E	A = 00H
0021			00	
0022		ADC A	8F	A = Carry Flag
0023		ADD L	85	L will store LM
0024		DCR C	0D	Decrement C to check if C was 1 or not next
0025		JZ SKIP2	CA	
0026			2E	
0027			00	Rotate A left so that next bit from MN can be put in D0 of LM
0028		RLC	07	L = A
0029		MOV L,A	6F	E = A
002A		MOV A,E	7B	
002B		JMP LOOP	C3	
002C			0E	
002D			00	L = A
002E	SKIP2	MOV L,A	6F	
002F		DAD D	19	HL = DE + HL
0030		SHLD 2052	22	
0031			52	
0032			20	
0033		HLT	76	



*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LDA 2050	3A	3	4	13
	0001			50			
	0002			20			
√	0003		MOV D,A	57	1	1	4
√	0004		LDA 2051	3A	3	4	13
	0005			51			
	0006			20			
√	0007		LXI H,0000	21	3	3	10
	0008			00			
	0009			00			
√	000A		MVI C,04	0E	2	2	7
	000B			04			
√	000C		STC	37	1	1	4
√	000D		CMC	3F	1	1	4
√	000E	LOOP	RAL	17	1	1	4
√	000F		MOV E,A	5F	1	1	4
√	0010		MVI A,00	3E	2	2	7
	0011			00			
√	0012		ADC A	8F	1	1	4
√	0013		ADD H	84	1	1	4



Memory Range: 000 ---- FFFF

Memory Address	Value
002C	0E
002E	6F
002F	19
0030	22
0031	52
0032	20
0033	76
2050	12
2051	34
2052	41
2053	23

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

Address	Label	Mnemonics	Hexcode
0000		LDA 2050	3A
0001			50
0002			20
0003		LHLD 2051	2A
0004			51
0005			20
0006		MVI H,00	26
0007			00
0008		MVI D,00	16
0009			00
000A		MOV E,L	5D
000B		DCR A	3D
000C	LOOP	DAD D	19
000D		DCR A	3D
000E		JNZ LOOP	C2
000F			0C
0010			00
0011		SHLD 2052	22
0012			52
0013			20
0014		HLT	76

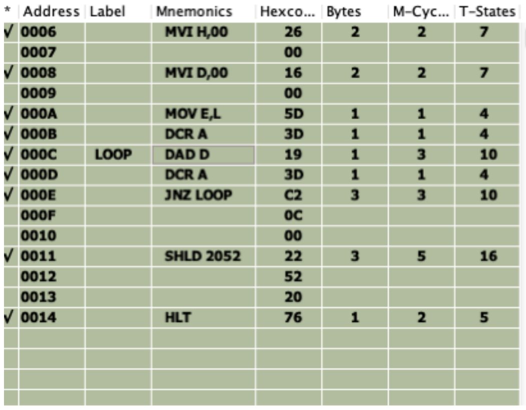
A = 'A'

HL = 'B'

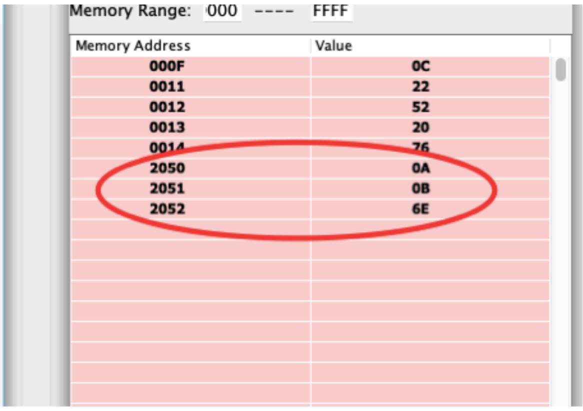
OTHERWISE H BECOMES M[2502] FROM SECOND RUN ONWARDS

D = 00H

E = L
SUM VALUE STARTS WITH A , SO WE LOOP ONE TIME LESS
HL = HL + DE



*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0006		MVI H,00	26	2	2	7
	0007			00			
√	0008		MVI D,00	16	2	2	7
	0009			00			
√	000A		MOV E,L	5D	1	1	4
√	000B		DCR A	3D	1	1	4
√	000C	LOOP	DAD D	19	1	3	10
√	000D		DCR A	3D	1	1	4
√	000E		JNZ LOOP	C2	3	3	10
	000F			0C			
	0010			00			
√	0011		SHLD 2052	22	3	5	16
	0012			52			
	0013			20			
√	0014		HLT	76	1	2	5

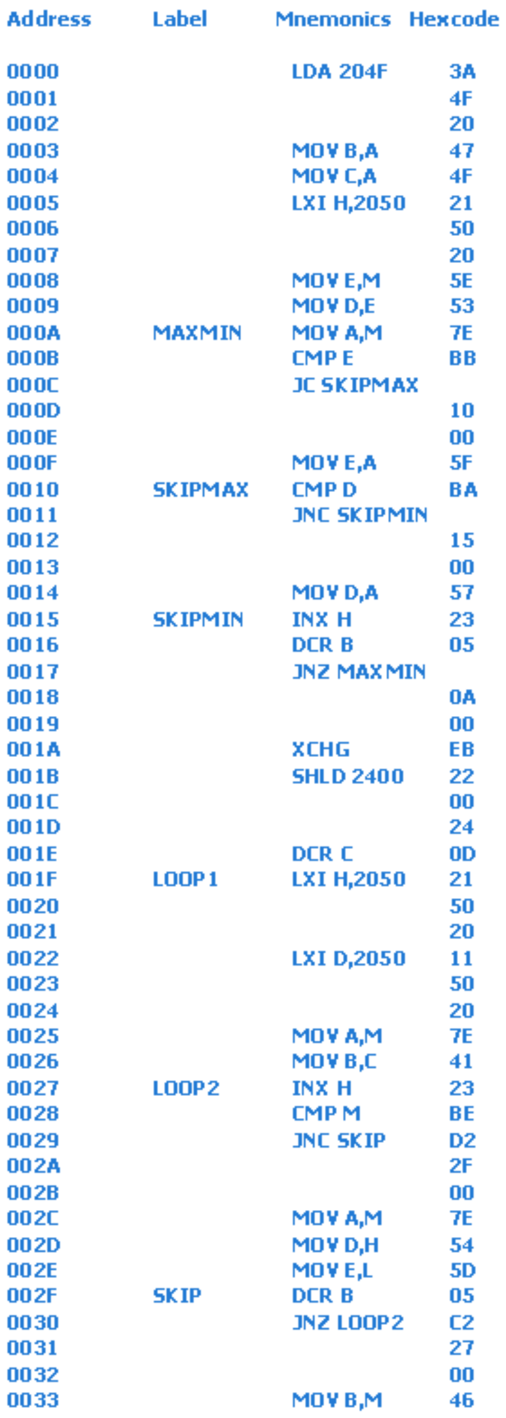


Memory Range: 000 ---- FFFF

Memory Address	Value
000F	0C
0011	22
0012	52
0013	20
0014	76
2050	0A
2051	0B
2052	6E

3. N numbers are stored in consecutive m/m location starting from 2050_H. The value N is stored in $204F_H$.

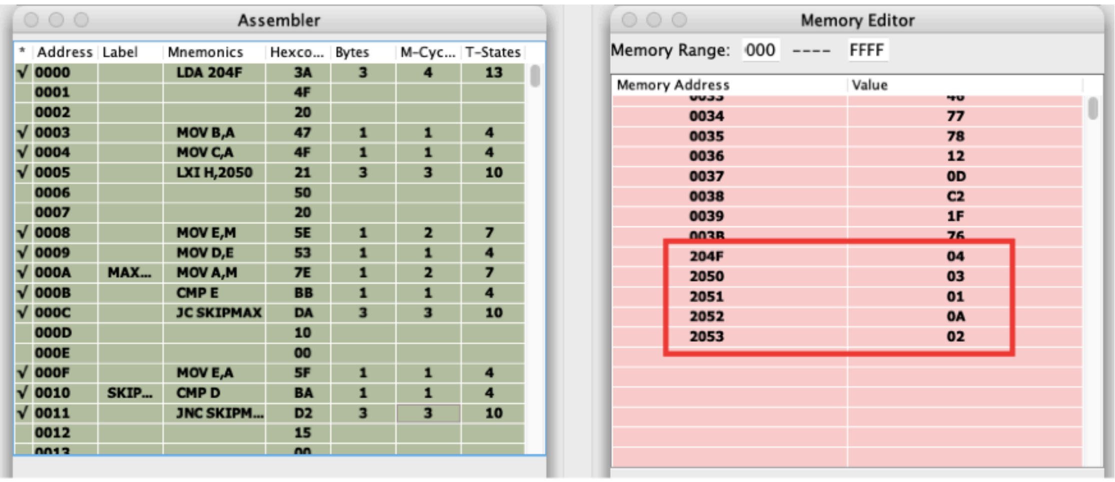
- i) Find the maximum among the N numbers.
- ii) Find the minimum among the N numbers.
- iii) Sort the N numbers in ascending order.
- iv) Sort the N numbers in descending order.



Address	Label	Mnemonics	Hexcode	
0000		LDA 204F	3A	
0001			4F	
0002			20	
0003		MOV B,A	47	FOR MAXMIN LOOP VARIABLE
0004		MOV C,A	4F	FOR SORT LOOP VARIABLE
0005		LXI H,2050	21	
0006			50	
0007			20	***FOR FINDING BOTH MIN AND MAX***
0008		MOV E,M	5E	MAX VALUE WILL BE IN E
0009		MOV D,E	53	MIN VALUE WILL BE IN D
000A	MAXMIN	MOV A,M	7E	
000B		CMP E	BB	
000C		JC SKIPMAX		
000D			10	A IS LESSER
000E			00	
000F		MOV E,A	5F	
0010	SKIPMAX	CMP D	BA	
0011		JNC SKIPMIN		A IS GREATER OR EQUAL TO
0012			15	
0013			00	
0014		MOV D,A	57	
0015	SKIPMIN	INX H	23	
0016		DCR B	05	
0017		JNZ MAXMIN		
0018			0A	
0019			00	
001A		XCHG	EB	
001B		SHLD 2400	22	2400H STORES MAX AND 2401 STORES MIN
001C			00	
001D			24	***FOR USING SELECTION SORT***
001E		DCR C	0D	
001F	LOOP1	LXI H,2050	21	
0020			50	
0021			20	
0022		LXI D,2050	11	
0023			50	
0024			20	
0025		MOV A,M	7E	
0026		MOV B,C	41	
0027	LOOP2	INX H	23	
0028		CMP M	BE	
0029		JNC SKIP	D2	***JNC FOR ASCENDING, JC FOR DESCENDING***
002A			2F	
002B			00	
002C		MOV A,M	7E	
002D		MOV D,H	54	
002E		MOV E,L	5D	
002F	SKIP	DCR B	05	
0030		JNZ LOOP2	C2	
0031			27	
0032			00	
0033		MOV B,M	46	SWAP BEGIN FOR LAST POSITION AND POSITION OF CURRENT HIGHEST VALUE

```
0034        MOV M,A     77
0035        MOV A,B     78
0036        STAX D      12      SWAP END
0037        DCR C       0D
0038        JNZ LOOP1   C2
0039                    1F
003A                    00
003B        HLT         76
```

BEFORE:



Assembler

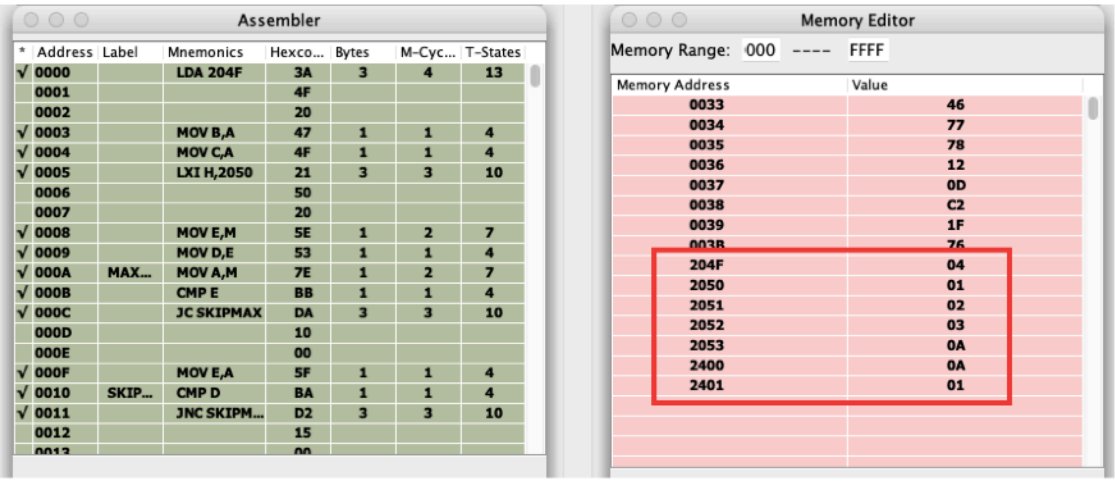
*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LDA 204F	3A	3	4	13
	0001			4F			
	0002			20			
√	0003		MOV B,A	47	1	1	4
√	0004		MOV C,A	4F	1	1	4
√	0005		LXI H,2050	21	3	3	10
	0006			50			
	0007			20			
√	0008		MOV E,M	5E	1	2	7
√	0009		MOV D,E	53	1	1	4
√	000A	MAX...	MOV A,M	7E	1	2	7
√	000B		CMP E	BB	1	1	4
√	000C		JC SKIPMAX	DA	3	3	10
	000D			10			
	000E			00			
√	000F		MOV E,A	5F	1	1	4
√	0010	SKIP...	CMP D	BA	1	1	4
√	0011		JNC SKIPM...	D2	3	3	10
	0012			15			
	0013			00			

Memory Editor

Memory Range: 000 ---- FFFF

Memory Address	Value
0033	46
0034	77
0035	78
0036	12
0037	0D
0038	C2
0039	1F
003B	76
204F	04
2050	03
2051	01
2052	0A
2053	02

AFTER:



Assembler

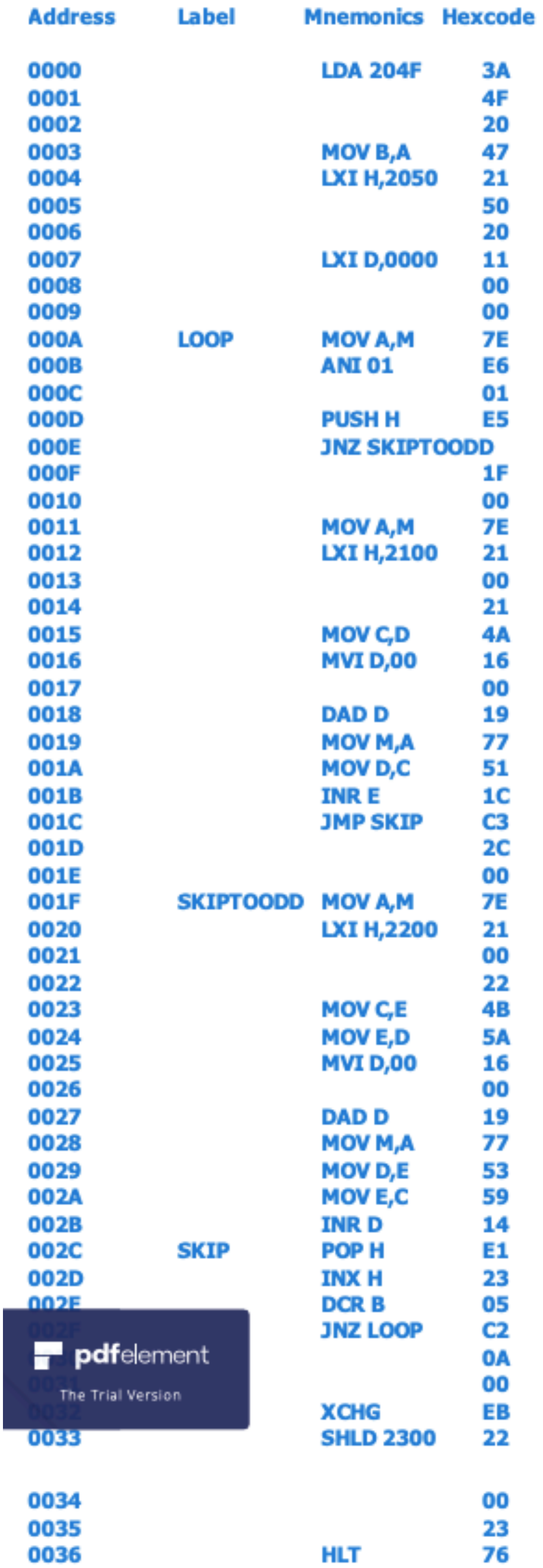
*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LDA 204F	3A	3	4	13
	0001			4F			
	0002			20			
√	0003		MOV B,A	47	1	1	4
√	0004		MOV C,A	4F	1	1	4
√	0005		LXI H,2050	21	3	3	10
	0006			50			
	0007			20			
√	0008		MOV E,M	5E	1	2	7
√	0009		MOV D,E	53	1	1	4
√	000A	MAX...	MOV A,M	7E	1	2	7
√	000B		CMP E	BB	1	1	4
√	000C		JC SKIPMAX	DA	3	3	10
	000D			10			
	000E			00			
√	000F		MOV E,A	5F	1	1	4
√	0010	SKIP...	CMP D	BA	1	1	4
√	0011		JNC SKIPM...	D2	3	3	10
	0012			15			
	0013			00			

Memory Editor

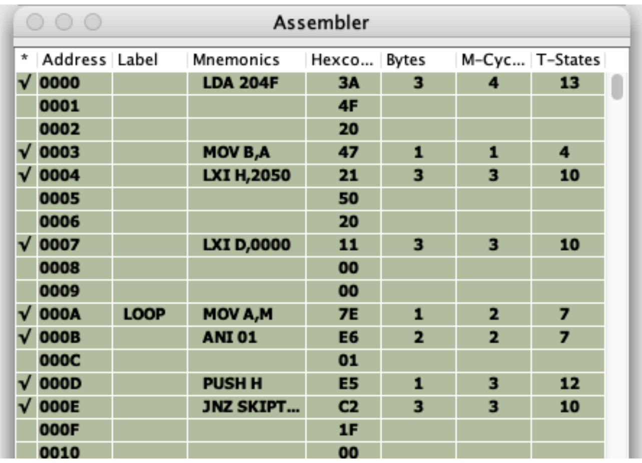
Memory Range: 000 ---- FFFF

Memory Address	Value
0033	46
0034	77
0035	78
0036	12
0037	0D
0038	C2
0039	1F
003B	76
204F	04
2050	01
2051	02
2052	03
2053	0A
2400	0A
2401	01

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 no. of even and odd numbers in 2300_H and 2301_H, respectively.

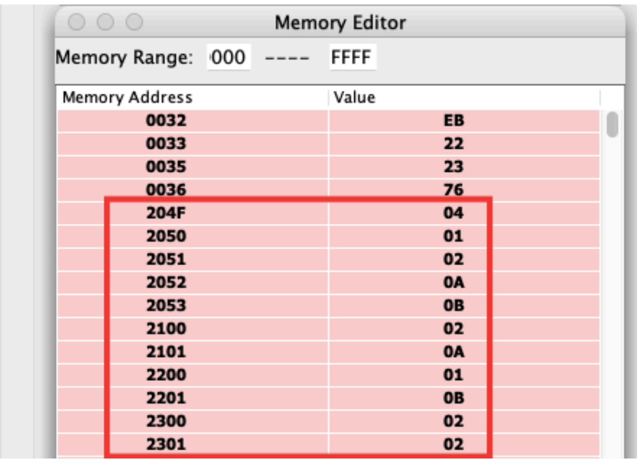


Address	Label	Mnemonics	Hexcode	
0000		LDA 204F	3A	
0001			4F	
0002			20	
0003		MOV B,A	47	
0004		LXI H,2050	21	
0005			50	
0006			20	
0007		LXI D,0000	11	
0008			00	
0009			00	
000A	LOOP	MOV A,M	7E	
000B		ANI 01	E6	
000C			01	
000D		PUSH H	E5	TO TEMP STORE HL VALUE
000E		JNZ SKIPTOODD		NUMBER IS ODD
000F			1F	
0010			00	
0011		MOV A,M	7E	RESTORE A AFTER ANI
0012		LXI H,2100	21	
0013			00	
0014			21	
0015		MOV C,D	4A	TO TEMP STORE D
0016		MVI D,00	16	
0017			00	
0018		DAD D	19	
0019		MOV M,A	77	
001A		MOV D,C	51	RESTORE D
001B		INR E	1C	E STORES EVEN COUNT
001C		JMP SKIP	C3	
001D			2C	
001E			00	
001F	SKIPTOODD	MOV A,M	7E	RESTORE A AFTER ANI
0020		LXI H,2200	21	
0021			00	
0022			22	
0023		MOV C,E	4B	TEMP STORE E
0024		MOV E,D	5A	PUT D IN E AND MAKE E 00 TO PERFOM HL+D
0025		MVI D,00	16	
0026			00	
0027		DAD D	19	HL = HL+DE
0028		MOV M,A	77	
0029		MOV D,E	53	RESTORE D
002A		MOV E,C	59	RESTORE E
002B		INR D	14	D STORES ODD COUNT
002C	SKIP	POP H	E1	RESTORE HL FROM STACK
002D		INX H	23	
002E		DCR B	05	
002F		JNZ LOOP	C2	
0030			0A	
0031			00	
0032		XCHG	EB	
0033		SHLD 2300	22	
0034			00	
0035			23	
0036		HLT	76	



Assembler

*	Address	Label	Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LDA 204F	3A	3	4	13
	0001			4F			
	0002			20			
√	0003		MOV B,A	47	1	1	4
√	0004		LXI H,2050	21	3	3	10
	0005			50			
	0006			20			
√	0007		LXI D,0000	11	3	3	10
	0008			00			
	0009			00			
√	000A	LOOP	MOV A,M	7E	1	2	7
√	000B		ANI 01	E6	2	2	7
	000C			01			
√	000D		PUSH H	E5	1	3	12
√	000E		JNZ SKIPT...	C2	3	3	10
	000F			1F			
	0010			00			



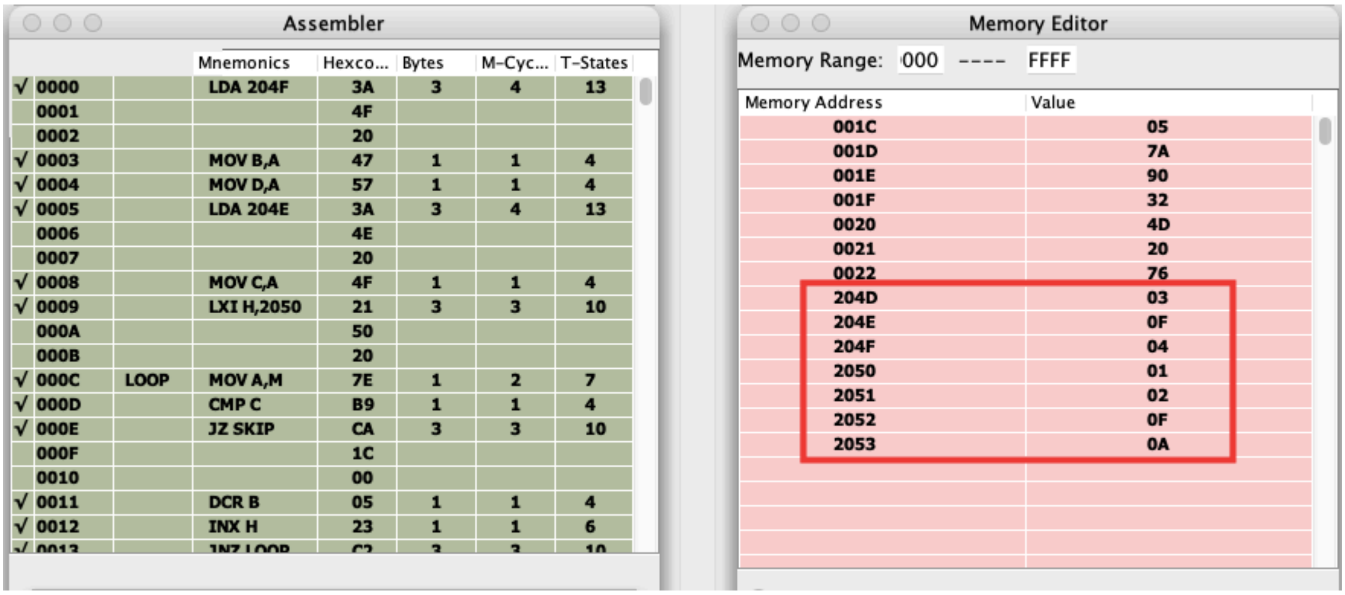
Memory Editor

Memory Range: 000 ---- FFFF

Memory Address	Value
0032	EB
0033	22
0035	23
0036	76
204F	04
2050	01
2051	02
2052	0A
2053	0B
2100	02
2101	0A
2200	01
2201	0B
2300	02
2301	02

5. 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 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.

Address	Label	Mnemonics	Hexcode	
0000		LDA 204F	3A	A = N
0001			4F	
0002			20	
0003		MOV B,A	47	B = A = N
0004		MOV D,A	57	another copy of N
0005		LDA 204E	3A	A = X (X IS THE NUMBER TO BE COMPARED)
0006			4E	
0007			20	
0008		MOV C,A	4F	C = A = X
0009		LXI H,2050	21	m/m ADDRESS VALUE
000A			50	
000B			20	
000C	LOOP	MOV A,M	7E	
000D		CMP C	B9	
000E		JZ SKIP	CA	
000F			1C	
0010			00	
0011		DCR B	05	
0012		INX H	23	
0013		JNZ LOOP	C2	
0014			0C	
0015			00	
0016		MVI B,FF	06	NOT FOUND
0017			FF	
0018		MOV A,B	78	
0019		JMP NOF	C3	
001A			1F	
001B			00	
001C	SKIP	DCR B	05	
001D		MOV A,D	7A	
001E		SUB B	90	POSITION FROM START (2050H)
001F	NOF	STA 204D	32	
0020			4D	
0021			20	
0022		HLT	76	



			Mnemonics	Hexco...	Bytes	M-Cyc...	T-States
√	0000		LDA 204F	3A	3	4	13
	0001			4F			
	0002			20			
√	0003		MOV B,A	47	1	1	4
√	0004		MOV D,A	57	1	1	4
√	0005		LDA 204E	3A	3	4	13
	0006			4E			
	0007			20			
√	0008		MOV C,A	4F	1	1	4
√	0009		LXI H,2050	21	3	3	10
	000A			50			
	000B			20			
√	000C	LOOP	MOV A,M	7E	1	2	7
√	000D		CMP C	B9	1	1	4
√	000E		JZ SKIP	CA	3	3	10
	000F			1C			
	0010			00			
√	0011		DCR B	05	1	1	4
√	0012		INX H	23	1	1	6
√	0013		JNZ LOOP	C2	3	3	10

Memory Range: 000 ---- FFFF

Memory Address	Value
001C	05
001D	7A
001E	90
001F	32
0020	4D
0021	20
0022	76
204D	03
204E	0F
204F	04
2050	01
2051	02
2052	0F
2053	0A