Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

Aim : Learning Techniques on Block Transfer of Data.

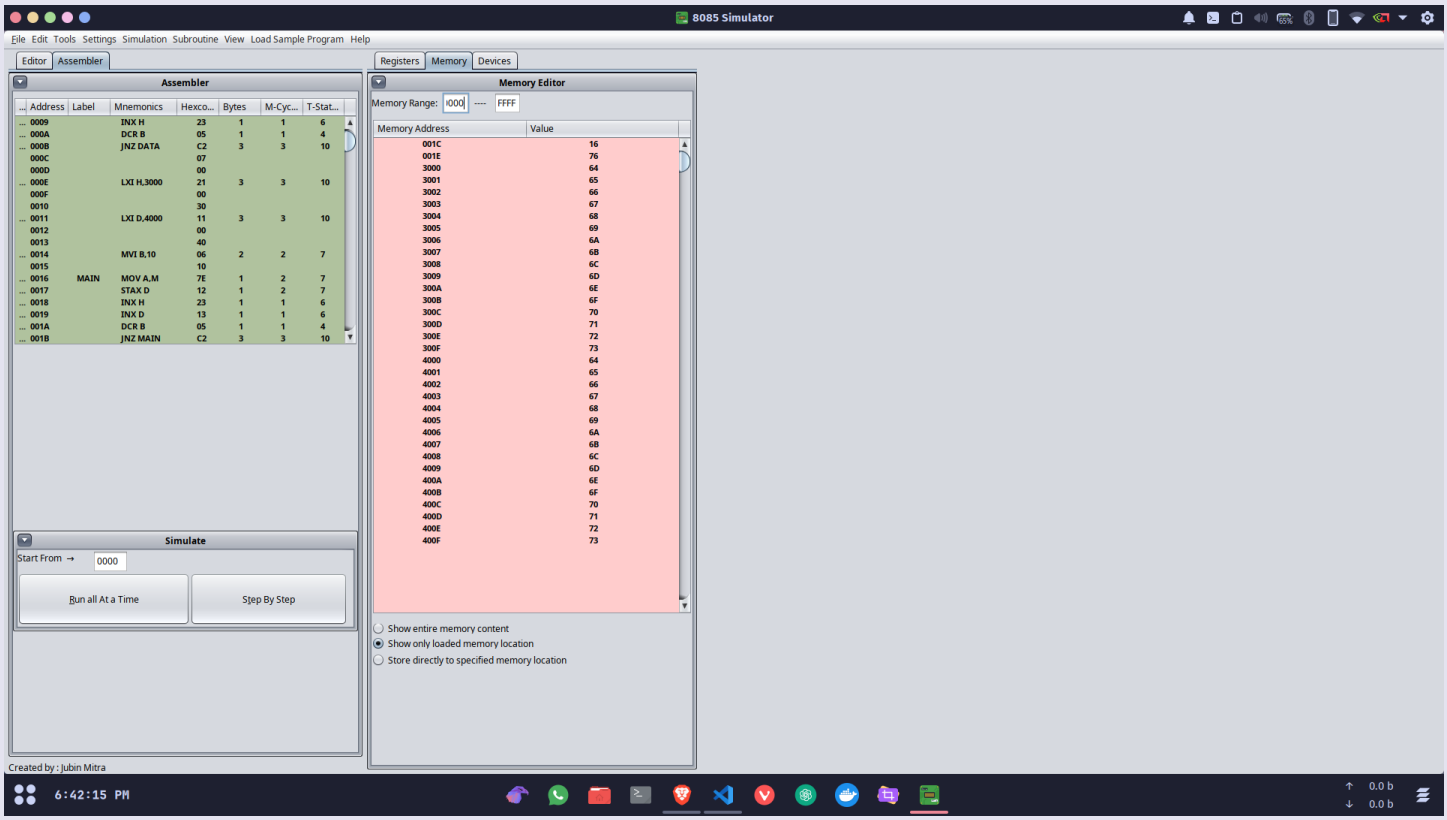
Exercise :

1. **Write a program to transfer a block of data from 3000H to 4000H. The size of block is 16 bytes.**

```
MVI B,10H
MVI C,64H
LXI H,3000H

DATA:
MOV M,C
INR C
INX H
DCR B
JNZ DATA

LXI H,3000H
LXI D,4000H
MVI B,10H

MAIN:
MOV A,M
STAX D

INX H
INX D

DCR B
JNZ MAIN
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

HLT



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

Name: Yash Lakhtariya
En.no: 211621010 12
Branch: CBA Batch - 51
Subject: M&A Practical 4

```
Code: MVI B, 10H    // Load 10H in reg. B (Counter)
      MVI C, 64H    // Load 64H in reg. C (data)
      LXI H, 3000H  // Load 3000H in H,L pair (start addr.)

DATA:               // Loading data in mem. from 3000 H
      MOV M, C      // Copy contents of C in memory 3000H
      INR C         // Increment data
      INX H         // Increment address
      DCR B         // Decrement counter
      JNZ DATA      // Repeat loop till counter is zero

      LXI H, 3000H  // Again load address in H,L pair
      LXI D, 4000H  // Load addr. to copy in D,E pair
      MVI B, 10H    // Set counter

MAIN:
      MOV A, M      // Load data from memory to A
      STAX D        // Copy from A to mem. pointed by D,E
      INX H         // Increment first addr.
      INX D         // Increment second addr.
      DCR B         // decrement counter
      JNZ MAIN      // Repeat loop till counter is zero

      HLT           // End execution
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0000		MVI B,10	06	2	2	7
0001			10			
0002		MVI C,64	0E	2	2	7
0003			64			
0004		LXI H,3000	21	3	3	10
0005			00			
0006			30			
0007	DATA	MOV M,C	71	1	2	7
0008		INR C	0C	1	1	4
0009		INX H	23	1	1	6
000A		DCR B	05	1	1	4
000B		JNZ DATA	C2	3	3	10
000C			07			
000D			00			
000E		LXI H,3000	21	3	3	10
000F			00			
0010			30			
0011		LXI D,4000	11	3	3	10
0012			00			
0013			40			
0014		MVI B,10	06	2	2	7
0015			10			
0016	MAIN	MOV A,M	7E	1	2	7
0017		STAX D	12	1	2	7
0018		INX H	23	1	1	6
0019		INX D	13	1	1	6
001A		DCR B	05	1	1	4
001B		JNZ MAIN	C2	3	3	10
001C			16			
001D			00			
001E		HLT	76	1	2	5

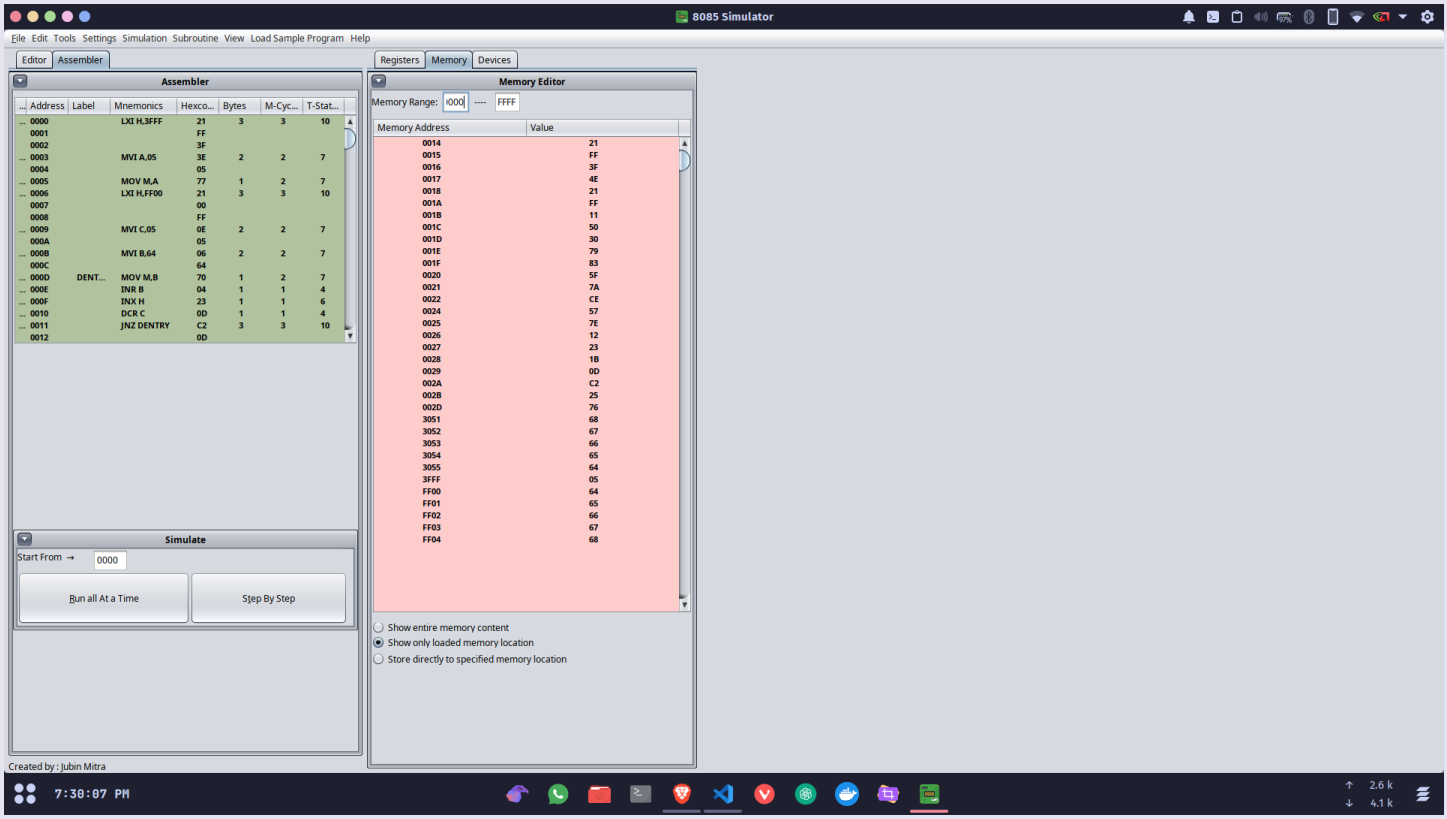
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

2. **Write a program to reverse a string stored in memory starting from FF00H. The length of the string is given in memory location 3FFF H. Store the reversed string at location 3050H.**

```
// LOAD DATA FOR TASK
        LXI H,3FFF
        MVI A,05
        MOV M,A
        LXI H,FF00
        MVI C,05
        MVI B,64

DENTRY:     MOV M,B
        INR B
        INX H
        DCR C
        JNZ DENTRY
// ----------------------------------------
// REVERSE
        LXI H,3FFF
        MOV C,M
        LXI H,FF00
        LXI D,3050
        MOV A,C
        ADD E
        MOV E,A
        MOV A,D
        ACI 00
        MOV D,A

STORE:      MOV A,M
        STAX D
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

```
        INX H
        DCX D
        DCR C
        JNZ STORE
        HLT
```



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

PAGE NO.
DATE

Code:

```
// Load Data Block
LXI H, 3FFFH  // addr. to store length of data
MVI A, 05H    // setting some length in A
MOV M, A      // copy length from A to M(3FFF)
LXI H, FF00H  // load start addr. in H,L pair
MVI C, 05H    // set counter / length
MVI B, 64H    // load data in B reg.

DENTRY:
MOV M, B      // copy from reg. B to M(FF00H) initially
INR B         // increment data
INX H         // increment addr.
DCR C         // decrement counter
JNZ DENTRY    // repeat till counter is zero

// Reverse and store
LXI H, 3FFFH  // load length addr. in H,L pair
MOV C, M      // load length from M to C
LXI H, FF00H  // load start addr. in H,L pair
LXI D, 3050H  // load second addr. in D,E pair
MOV A, C      // load length from C to A
ADD E         // add lower byte of D,E pair
MOV E, A      // update lower byte of D,E
MOV A, D      // load upper byte of D,E in A
ACI 00H       // increment if carry
MOV D, A      // update upper byte of D,E
STORE:
MOV A, M      // load from H,L pointer to A
STAX D        // store in D,E pair
INX H         // increment addr. - 1
DCX D         // decrement addr. - 2
DCR C         // decrement counter
JNZ STORE     // repeat till counter is zero
              // end
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0000		LXI H,3FFF	21	3	3	10
0001			FF			
0002			3F			
0003		MVI A,05	3E	2	2	7
0004			05			
0005		MOV M,A	77	1	2	7
0006		LXI H,FF00	21	3	3	10
0007			00			
0008			FF			
0009		MVI C,05	0E	2	2	7
000A			05			
000B		MVI B,64	06	2	2	7
000C			64			
000D	DENTRY	MOV M,B	70	1	2	7
000E		INR B	04	1	1	4
000F		INX H	23	1	1	6
0010		DCR C	0D	1	1	4
0011		JNZ DENTRY	C2	3	3	10
0012			0D			
0013			00			
0014		LXI H,3FFF	21	3	3	10
0015			FF			
0016			3F			
0017		MOV C,M	4E	1	2	7
0018		LXI H,FF00	21	3	3	10
0019			00			
001A			FF			
001B		LXI D,3050	11	3	3	10
001C			50			
001D			30			
001E		MOV A,C	79	1	1	4
001F		ADD E	83	1	1	4

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

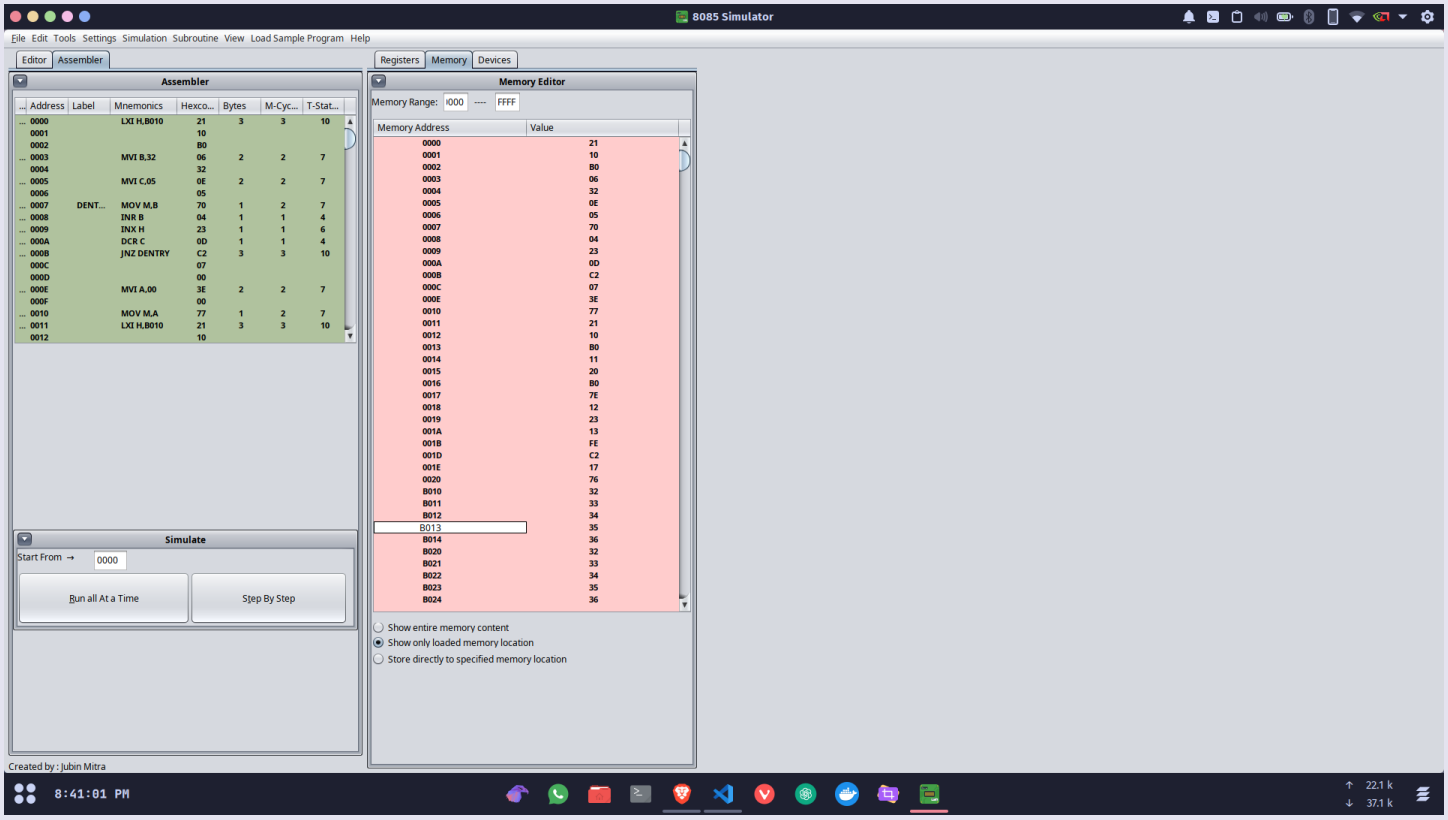
Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0020						
0021		MOV E,A	5F	1	1	4
0022		MOV A,D	7A	1	1	4
0023		ACI 00	CE	2	2	7
0024			00			
0025		MOV D,A	57	1	1	4
0026	STORE	MOV A,M	7E	1	2	7
0027		STAX D	12	1	2	7
0028		INX H	23	1	1	6
0029		DCX D	1B	1	1	6
002A		DCR C	0D	1	1	4
002B		JNZ STORE	C2	3	3	10
002C			26			
002D			00			
002E		HLT	76	1	2	5
002F						
30						

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

3. **Write a program to transfer a block of data from B010H to B020H. The end of block is indicated by 00H.**

```
        LXI H,B010
        MVI B,32
        MVI C,05

DENTRY:     MOV M,B
        INR B
        INX H
        DCR C
        JNZ DENTRY
        MVI A,00
        MOV M,A
// ------------------
        LXI H,B010
        LXI D,B020

CHECK:      MOV A,M
        STAX D
        INX H
        INX D
        CPI 00
        JNZ CHECK
        HLT
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

Code :

```
LXI H,B010H   // load start addr. in H,L pair
MVI B,32H     // load data in reg. B
MVI C,05H     // load length of data in C

DENTRY :
MOV M,B       // Copy from B to H,L pointer
INR B         // increment data
INX H         // increment address
DCR C         // decrement counter
JNZ DENTRY    // repeat till counter is zero
MVI A,00H     // load & clear accumulator
MOV M,A       // store 00H in last addr.

LXI H,B010H   // load addr-1 in H,L pair
LXI D,B020H   // load addr-2 in D,E pair

CHECK :
MOV A,M       // load data from H,L pointer to A
STAX D        // Copy from A to DE pointer
INX H         // increment addr-1
INX D         // increment addr-2
CPI 00        // compare A's content with 00
JNZ CHECK     // repeat till A equals 00
HLT           // end execution
```

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0000		LXI H,B010	21	3	3	10
0001			00			
0002			B0			
0003		MVI B,32	06	2	2	7

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

Address	Label	Mnemonics	Hex	Bytes	M-cycles	T-states
0004			32			
0005		MVI C,05	0E	2	2	7
0006			05			
0007	DENTRY	MOV M,B	70	1	2	7
0008		INR B	04	1	1	4
0009		INX H	23	1	1	6
000A		DCR C	0D	1	1	4
000B		JNZ DENTRY	C2	3	3	10
000C			07			
000D			00			
000E		MVI A,00	3E	2	2	7
000F			00			
0010		MOV M,A	77	1	2	7
0011		LXI H,8010	21	3	3	10
0012			10			
0013			80			
0014		LXI D,8020	11	3	3	10
0015			20			
0016			80			
0017	CHECK	MOV A,M	7E	1	2	7
0018		STAX D	12	1	2	7
0019		INX H	23	1	1	6
001A		INX D	13	1	1	6
001B		CPI 00	FE	2	2	7
001C			00			
001D		JNZ CHECK	C2	3	3	10
001E			17			
001F			00			
0020		HLT	76	1	2	5

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

Exercise : (Answer the following questions)

1) The register always pointing to the location of next instruction to be fetched is __
2) If starting address of 8KB RAM is 3000H, than ending address is ___
3) Define Bus and give its type.
4) A memory chip in a microcomputer system has eight address lines and eight data lines. How many bytes can be stored in it?
5) If the starting address of the chip is 9000H, What will be the ending address?
6) How many address lines are necessary to access 16 Kbytes EPROM & 8 Kbytes RAM?

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
M&A Practical 4

Exercise answers :

1] PC (Program Counter)

2] Starting addr. = 3000 H
size of 8 KB RAM = 2^{13} bytes
In 8085, 64 KB max mem. = 2^{16} bytes
For, 8 KB, $16 - 13 = 3$, $\frac{16}{2^3} = \frac{16}{8} = 2$ blocks or units
1st block of 16 : 3000 to 3FFF
2nd block of 16 : 4000 to 4FFF

→ ending address = 4FFF H

3] Bus is a set of electrical lines used to transfer data signals among different parts of a computer or its parts.
Types of bus : ① Address Bus
② Data Bus
③ Control Bus

4] 2^8 = 256 bytes

5] Depends on size of chip, but max it can go till FFFF H

6] EPROM : 16 KB = 2^{14} => 14 address lines
RAM : 8 KB = 2^{13} => 13 address lines