

MIT ASSIGNMENT – 1

Q1: Store the data byte 25H into memory location 3000H

;Program 1

MVI a,25h

STA 3000h

hlt

The screenshot shows an 8085 assembler simulator interface. On the left, a text editor displays the assembly code for Program 1:

```
1  
2 ;Program 1  
3 MVI a,25h  
4 STA 3000h  
5 hlt  
6
```

On the right, the 'Memory' tab is active, showing a memory dump starting at address 3000h. The dump table is as follows:

Address (Hex)	Address	Data
3000	12288	37
3001	12289	0
3002	12290	0
3003	12291	0
3004	12292	0
3005	12293	0
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0

Below the memory dump, the 'Assembler Message' window shows:

Line No	Assembler Message
0	Program assembled successfully

Q2: Write a program to add two 8-bit numbers. Store the result at one memory location.

;Program 2

MVI a,25h

STA 3000h

MVI a,32h

STA 3001h

MVI a,30h

STA 3005h

;-----ADDITION-----

LDA 3000h

MOV b,a

LDA 3001h

ADD b

STA 3002h

hlt

Load me at

```

1 ;Program 2
2 MVI a,25h
3 STA 3000h
4 MVI a,32h
5 STA 3001h
6 MVI a,30h
7 STA 3005h
8 ;-----ADDITION-----
9 LDA 3000h
10 MOV b,a
11 LDA 3001h
12 ADD b
13 STA 3002h
14 hlt
15

```

Data Stack KeyPad **Memory** I/O Ports

Start 3000h OK

Address (Hex)	Address	Data
3000	12288	37
3001	12289	50
3002	12290	87
3003	12291	0
3004	12292	0
3005	12293	48
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0

Line No Assembler Message

0 Program assembled successfully

Q3: Write a program to subtract two 8-bit numbers. Store the result at one memory location.

Program 3

MVI a,37h

STA 3000h

MVI a,25h

STA 3001h

MVI a,30h

STA 3005h

;-----SUBTRACTION-----

LDA 3001h

MOV b,a

LDA 3000h

SUB b

STA 3002h

hlt

Load me at

```

1 ;Program 3
2 MVI a,37h
3 STA 3000h
4 MVI a,25h
5 STA 3001h
6 MVI a,30h
7 STA 3005h
8 ;-----SUBTRACTION-----
9 LDA 3001h
10 MOV b,a
11 LDA 3000h
12 SUB b
13 STA 3002h
14 hlt
15

```

Data Stack KeyPad **Memory** I/O Ports

Start 3000h OK

Address (Hex)	Address	Data
3000	12288	55
3001	12289	37
3002	12290	18
3003	12291	0
3004	12292	0
3005	12293	48
3006	12294	0
3007	12295	0
3008	12296	0
3009	12297	0
300A	12298	0
300B	12299	0

Line No Assembler Message

0 Program assembled successfully

Q4: Exchange the contents of memory locations 2000H and 4000H

;Program 4

MVI a,25h

STA 4000h

MVI a,32h

STA 2000h

;-----

LDA 4000h

MOV b,a

LDA 2000h

STA 4000h

MOV a,b

STA 2000h

hlt

Start	2000h	OK
Address (Hex)	Address	Data
2000	8192	37
2001	8193	0
2002	8194	0
2003	8195	0
2004	8196	0
2005	8197	0
2006	8198	0
2007	8199	0
2008	8200	0
2009	8201	0
200A	8202	0
200B	8203	0

Line No	Assembler Message
0	Program assembled successfully

Start	4000h	OK
Address (Hex)	Address	Data
4000	16384	50
4001	16385	0
4002	16386	0
4003	16387	0
4004	16388	0
4005	16389	0
4006	16390	0
4007	16391	0
4008	16392	0
4009	16393	0
400A	16394	0
400B	16395	0

Line No	Assembler Message
0	Program assembled successfully

Q5: Write a program to add two 16 bit numbers. Numbers are stored in four consecutive memory location as 8-bit numbers. (Use instruction ADC)

;Program 5

LDA 00h

MOV B, A

LDA 02h

ADD B

STA 04h

LDA 01h

MOV C, A

LDA 03h

ADC C

STA 05h

hlt

The screenshot shows an 8085 assembler simulator interface. On the left, the 'Registers' panel displays the state of the 8085 registers: A (2B), BC (63), DE (00), HL (00), PSW (00), PC (42), SP (FF), and Int-Reg (00). The 'Flag' panel shows S=1, Z=0, AC=0, P=0, and C=0. Below these are sections for 'Decimal - Hex Conversion', 'I/O Ports', and 'Memory'. The main assembly window shows a program with 14 lines of code, including comments and instructions like LDA, MOV, ADD, STA, ADC, and hlt. On the right, the 'Memory' panel shows a dump of memory locations from 0000 to 000B, with addresses and data values. At the bottom, a message box indicates 'Program assembled successfully'.

Address (Hex)	Address	Data
0000	0	43
0001	1	99
0002	2	21
0003	3	89
0004	4	64
0005	5	188
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0
000B	11	0

Q6: Write a program to subtract two 16 bit numbers. Numbers are stored in four consecutive memory location as 8-bit numbers. (Use instruction SBB)

;Program 6

LDA 02h

MOV B, A

LDA 00h

SUB B

STA 04h

LDA 03h

MOV C, A

LDA 01h

SBB C

STA 05h

hlt

Registers			Flag
<i>A</i>	0A		<i>S</i> 0
<i>BC</i>	15	59	
<i>DE</i>	00	00	<i>Z</i> 0
<i>HL</i>	00	00	
<i>PSW</i>	00	00	<i>AC</i> 0
<i>PC</i>	42	17	
<i>SP</i>	FF	FF	<i>P</i> 1
<i>Int-Reg</i>	00		<i>C</i> 0

Decimal - Hex Conversion

Decimal

Hex

0

0

→ To Hex

← To Dec

I/O Ports

0

-

+

00

Update Port Value

Memory

0

-

+

2B

Update Memory

Load me at

1

2

3

4

5

6

7

8

9

10

11

12

13

14

;

P

r

o

g

r

a

m

6

L

D

A

0

2

h

M

O

V

B

,

A

L

D

A

0

0

h

S

U

B

B

S

T

A

0

4

h

L

D

A

0

3

h

M

O

V

C

,

A

L

D

A

0

1

h

S

S

B

C

S

T

A

0

5

h

h

l

t

Start

OK

Address (Hex)	Address	Data
0000	0	43
0001	1	99
0002	2	21
0003	3	89
0004	4	22
0005	5	10
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0
000B	11	0

Line No	Assembler Message
0	Program assembled successfully