

MIT (Assignment – 4)

U20CS110
Krishna Pandey

(1). Write an assembly language program in 8085 to search a given number in an array of n numbers. If number is found, then store AA in memory location 4051 otherwise store BB in 4051.

;<Question-1>

JMP start

start: LXI H, 2050H

MOV B, M

INX H

MOV C, M

skip: INX H

MOV A, M

CMP B

JZ loop

DCR C

JNZ skip

MVI A, BBH

STA 4051H

HLT

loop: MVI A, AAH

STA 4051H

HLT

The screenshot displays an 8085 assembly simulator interface. On the left, the **Registers** panel shows the following values: A/PSW (0x AA 56), BC (0x DD 02), DE (0x 00 00), HL (0x 20 55), SP (0x FF FF), and PC (0x 08 1F). Below this, the **Flags** panel shows Z (checked), S (unchecked), P (checked), C (unchecked), and AC (checked). The central area displays the assembly code for `main.asm`:

```

1 ;<Question-5>
2
3 JMP start
4
5 start: LXI H, 2050H
6       MOV B, M
7       INX H
8       MOV C, M
9 skip: INX H
10      MOV A, M
11      CMP B
12      JZ loop
13      DCR C
14      JNZ skip
15      MVI A, BBH
16      STA 4051H
17      HLT
18 loop: MVI A, AAH
19      STA 4051H
20      HLT

```

On the right, the **Memory View** panel shows a memory dump starting at address 200. The address 205 is highlighted, showing the value `DD 05 AA BB CC DD EE`. The **Start Address** is set to `0x 1100`.

Memory View



0x

4051

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
401	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
402	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
403	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
404	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
405	00	AA	00	00	00	00	00	00	00	00	00	00	00	00	00	00
406	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
407	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
408	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
409	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x 3300



(2).Write an assembly language program to sort numbers in ascending order where six numbers are stored in consecutive memory locations starting from 2000H.

;<Question-2>

JMP START

START: LXI H, 2000H

MVI D, 00H

MOV C, M

DCR C

INX H

CHECK: MOV A, M

INX H

CMP M

JC NEXTBYTE

JZ NEXTBYTE

MOV B, M

MOV M, A

DCX H

MOV M, B

INX H

MVI D,01H

NEXTBYTE: DCR C

JNZ CHECK

MOV A, D

CPI 01H

JZ START

HLT

The screenshot shows an 8086 assembly simulator interface. On the left, the **Registers** window displays the status of various registers: A/PSW (0x0002), BC (0x0000), DE (0x0000), HL (0x0000), SP (0x0000), and PC (0x0800). Below it, the **Flags** window shows Z, S, P, C, and AC flags, all of which are currently cleared (indicated by empty checkboxes). The main window displays the assembly code for `main.asm`, with line numbers 1 through 20. The code implements a sorting algorithm, starting with a jump to `START`, loading the address `2000H` into the `HL` register, and initializing the `DE` register to `00H`. It then enters a loop where it compares the current byte with the next byte in memory and swaps them if necessary. The `PC` register is shown at `0x0800`. On the right, the **Memory View** window shows the memory locations from `2000H` to `200FH`. The first six locations (`2000H` to `2005H`) contain the values `06`, `02`, `05`, `13`, `24`, and `35`, which are highlighted in green. The remaining locations contain `00`. At the bottom, the **Start Address** is set to `0x1100`.

(3).Write an assembly language program to take all numbers which are not 00H from array, and store them into different location. Numbers are stored at 3001 onwards, 3000 is holding the size of array, the results will be stored from 4000.

;<Question-3>

JMP START

START: LXI H,3000H

MOV C,M

LXI D,4000H

LOOP: INX H

MOV A,M

ORI 00H

JZ SKIP

STAX D

INX D

SKIP: DCR C

JNZ LOOP

HLT

The screenshot displays an assembly language simulator interface. On the left, the **Registers** section shows the following values: A/PSW (0x0002), BC (0x0000), DE (0x0000), HL (0x0BB8), SP (0xFFFF), and PC (0x0807). Below this, the **Flags** section shows Z, S, P, C, and AC flags, all of which are currently cleared (indicated by empty checkboxes).

In the center, the **main.asm** code editor displays the following assembly code:

```

1 ;<Question-3>
2
3 JMP START
4 START: LXI H,3000
5       MOV C,M
6       LXI D,4000
7
8 LOOP: INX H
9       MOV A,M
10      ORI 00H
11      JZ SKIP
12      STAX D
13      INX D
14
15 SKIP: DCR C
16       JNZ LOOP
17
18 HLT
19

```

On the right, the **Memory View** section shows a hexadecimal memory dump starting at address 300. The first row (300) contains the values 05, 0A, 12, 56, 45, 99, followed by zeros. The address 3000 is highlighted in the address field, and the value 0x3000 is shown in the input field. The **Start Address** is set to 0x2200.

Registers

A/PSW	0x 99 56
BC	0x 00 00
DE	0x 40 05
HL	0x 30 05
SP	0x FF FF
PC	0x 08 18

Flags

Z	<input checked="" type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input checked="" type="checkbox"/>

Load at 0x0800

Memory View

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
400	0A	12	56	45	99	00	00	00	00	00	00	00	00	00	00	00
401	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
402	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
403	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
404	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
405	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
406	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
407	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
408	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
409	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x

```
main.asm
1 ;<Question-3>
2
3 JMP START
4 START: LXI H,3000H
5       MOV C,M
6       LXI D,4000H
7
8 LOOP: INX H
9       MOV A,M
10      ORI 00H
11      JZ SKIP
12      STAX D
13      INX D
14
15 SKIP: DCR C
16      JNZ LOOP
17
18 HLT
19
```

(4).Write an assembly language program to insert a new value in an array of data only if it is not present in the array. Assume suitable data and memory location.

;<Question-4>

JMP start

start: LXI H, 0000H

MOV B, M

INX H

MOV C, M

skip: INX H

MOV A, M

CMP B

JZ loop

DCR C

JNZ skip

MOV A, B

STA 0007H

HLT

loop: HLT

Registers

A/PSW	0x1056
BC	0x1000
DE	0x0000
HL	0x0006
SP	0xFFFF
PC	0x0818

Flags

Z	<input checked="" type="checkbox"/>
S	<input type="checkbox"/>
P	<input checked="" type="checkbox"/>
C	<input type="checkbox"/>
AC	<input checked="" type="checkbox"/>

Memory View

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	10	05	11	23	0A	BB	77	10	00	00	00	00	00	00	00	00
001	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
002	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
003	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
004	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
005	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
006	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
007	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
008	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
009	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Start Address at: 0x0