Practical – 7

Aim: Write an Assembly Language Program to find 1's & 2's complement of an 8-bit number.

Source Code :

```
;<Program title>

jmp start

;data

;code

start: nop

LDA 0001H

CMA

STA 0002H

ADI 1

STA 0003H

Hlt
```

Output:

Registers

A		85
BC	00	00
DE	00	00
HL	00	00
PSW	00	00
PC	42	11
SP	FF	FF
Int-Reg		00

Flag

S	1
Z	0
AC	0
P	0
C	0

Address (Hex)	Address	Data
0000	0	0
0001	1	123
0002	2	132
0003	3	133
0004	4	0
0005	5	0
0006	6	0
0007	7	0
0008	8	0
0009	9	0
000A	10	0
000B	11	0
000C	12	0
000D	13	0

Practical – 8

Aim: Write an Assembly Language Program to find the sum of 5 numbers using loop.

Source Code :

```
;<Program title>

jmp start

;data

;code

start: nop

MVI B, 05H

LXI H, 0001H

LOOP: MOV C,M

ADD C

INX H

DCR B

JNZ LOOP

STA 0006H

Hlt
```

Output:

Registers

A	0F	
BC	00	03
DE	00	00
HL	00	06
PSW	00	00
PC	42	14
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	**1**
AC	0
P	**1**
C	0

Address (Hex)	Address	Data
0000	0	0
0001	1	1
0002	2	2
0003	3	4
0004	4	5
0005	5	3
0006	6	15
0007	7	0
0008	8	0
0009	9	0
000A	10	0
000B	11	0
000C	12	0
000D	13	0

Practical – 9

Aim: Write an Assembly Language Program to find smallest Number From an array.

Source Code :

```
;<Program title>

jmp start

;data

;code

start: nop

LXI H, 0001H

MOV C,M

INX H

MOV B,M

DCR C

LOOP: INX H

MOV A,M

CMP B

JNC SKIP

MOV B,A

SKIP: DCR C

JNZ LOOP

LXI H, 0007H

MOV M,B

Hlt
```

Output:

Registers

A	09	
BC	06	00
DE	00	00
HL	00	07
PSW	00	00
PC	42	1B
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	**1**
AC	0
P	**1**
C	0

Address (Hex)	Address	Data
0000	0	0
0001	1	5
0002	2	6
0003	3	11
0004	4	8
0005	5	12
0006	6	9
0007	7	6
0008	8	0
0009	9	0
000A	10	0
000B	11	0
000C	12	0
000D	13	0

Practical – 10

Aim: Write an Assembly Language Program to arrange given numbers in ascending order.

Source Code :

```
;<Program title>

jmp start

;data

;code

start: nop

START: LXI H,8040H

MVI D, 00H

MOV C, M

DCR C

INX H

CHECK: MOV A,M

INX H

CMP M

JC NEXTBYT

MOV B,M

MOV M,A

DCX H

MOV M,B

INX H

MVI D,01H

NEXTBYT: DCR C

JNZ CHECK
```

MOV A,D

RRC

JC START

Hlt

Output:

Registers			Flag	
A	00		S	0
BC	0A	00	Z	1
DE	00	00	AC	0
HL	80	45	P	1
PSW	00	00	C	0
PC	42	23		
SP	FF	FF		
Int-Reg	00			

Address (Hex)	Address	Data
8040	32832	5
8041	32833	12
8042	32834	8
8043	32835	15
8044	32836	13
8045	32837	10
8046	32838	0
8047	32839	0
8048	32840	0
8049	32841	0
804A	32842	0
804B	32843	0
804C	32844	0
804D	32845	0

Before Sorting In Ascending Order

Address (Hex)	Address	Data
8040	32832	5
8041	32833	8
8042	32834	10
8043	32835	12
8044	32836	13
8045	32837	15
8046	32838	0
8047	32839	0
8048	32840	0
8049	32841	0
804A	32842	0
804B	32843	0
804C	32844	0
804D	32845	0

After Sorting In Ascending Order

Practical – 11

Aim: Write an Assembly Language Program to find the factorial of a given number.

Source Code :

```
;<Program title>

jmp start

;data

;code

start: nop

LXI H, 8000H

MOV B,M

MVI D,01H

LOOP1: CALL FACT

DCR B

JNZ LOOP1

INX H

MOV M,D

Hlt

FACT: MOV C,B

XRA A

ML: ADD D

DCR C

JNZ ML

MOV D,A

RET

Hlt
```

Output:

Registers

A	78	
BC	00	00
DE	78	00
HL	80	01
PSW	00	00
PC	42	14
SP	FF	FF
Int-Reg	00	

Flag

S	0
Z	**1**
AC	0
P	**1**
C	0

Address (Hex)	Address	Data
8000	32768	5
8001	32769	120
8002	32770	0
8003	32771	0
8004	32772	0
8005	32773	0
8006	32774	0
8007	32775	0
8008	32776	0
8009	32777	0
800A	32778	0
800B	32779	0
800C	32780	0
800D	32781	0