.1. Indirect Addressing -To add two 8-bit numbers.

```
LXI H, 8100
MOV A, M          //8100=02
INX H             //8101==03
ADD M
INX H             //8102
MOV M, A
HLT
```

OR LXI H,8100

```
LDAX H
MOV B, A
INX H
LDAX H
ADD B
INX H
STAX H
HLT
```

2.To add 16 bit numbers using direct and indirect addressing mode

(i) Direct Addressing mode-using DAD

```
LHLD 8100                  //8100=L=02, 8101=H=03
XCHG                              //E=02,    D=03
LHLD 8102                  //8102=L=01, 8103=H=04
DAD D
SHLD 8104
HLT
```

(ii) Indirect addressing mode

```
LXI B, 8100
LDAX B
MOV D, A
INX B
LDAX B
ADD D
STA 8104
INX B
LDAX B
MOV D, A
INX B
LDAX B
ADC D
STA 8105
HLT
```

1. Write a program to add 8-bit numbers using carry (using JNC instruction).

```
MVI C, 00
LXI H, 8100
MOV A, M
INX H
ADD M
JNC 820A
INR C
820A: INX H
MOV M, A
INX H------8103
MOV M, C

HLT
```

Input: 8100=02

8101=08

2. Write a program for the sum of series of numbers.

```
MVI C, 0A                                                    M
LXI H, 8100                          8100=02,8101=03,8104=04…..
MVI A,00
Back: ADD M
INX H
DCR C
JNZ Back
STA 810A

HLT
```

3. Write a program to add ten 8-bit numbers

```
MVI C, 00
MVI B, 09
LXI H, 8100
MOV A, M
Back: INX H
ADD M
JNC Next
INR C
Next: DCR B
JNZ Back
INX H
MOV M, A
```

INX H

MOV M, C
HLT

4. Write a program to find negative numbers in a block of data.

```
MVI C, 05
MVI B, 00                    A-1000 0001=81
LXI H, 8101                  1000 0000
Back: MOV A, M
ANI 80          // If MSB =1 then number is negative
JZ Skip         // jump if MSB=0 then number is positive
INR B
Skip: INX H
DCR C
JNZ Back
MOV A, B
STA 8109
HLT
```

5. Write a program for data transfer from memory block B1 to memory block B2.

```
MVI C, 0A
LXI H, 8100
LXI D, 8200
Back: MOV A, M
STAX D
INX H
INX D
DCR C
JNZ Back
HLT
```

1. Write a program to count the number of 1's in a number

```
LDA 8100                    A=01010001
MVI B, 08
MVI D, 00
Loop1: RLC          //Rotate left circular// put D7 in D0 and carry flag
JNC Loop2
INR D
Loop2: DCR B
JNZ Loop1
MOV A, D
STA 810A
HLT
```

Input 8100:25

2. Write a program to find the largest no. in array of 10 elements

```
MVI B, 05
LXI H, 8100
MOV A, M
INX H
Back: CMP M                 A=02 M=04  A<M,A<-M04
JNC Next
MOV A, M
Next: INX H
DCR B
JNZ Back
STA 810A
HLT
```

3. Write a program to calculate the sum of series of even numbers

```
LDA 8100
MOV C, A
MVI B,00
LXI H,8101
Back: MOV A, M
ANI 01
JNZ Skip   8008  800E
MOV A, B
ADD M                          // B=0, M=02,
MOV B, A              //B=02
Skip: INX H
DCR C
JNZ Back
MOV A, B
STA 810B
HLT
```

4. Write a program to arrange the numbers in ascending order.

```
MVI C, 08          //C=08
Repeat: MOV D, C                                    8101=02()H-L-8102=05(A) 8103=06(M)
LXI H, 8101
Loop: MOV A, M
INX H
CMP M
JC Skip
MOV B, M                       B=02
MOV M, A
DCX H
MOV M, B
INX H
Skip: DCR D
JNZ Loop
DCR C
```

```
JNZ Repeat
HLT
```

5. Write a program to find the even parity in the consecutive memory locations in 8085

```
MVI B, 0A
MVI C,00
LXI H,8100
Back: MOV A, M          02=00000010,03=00000011
ANI FF                    11111111==
JPO Next                 //Jump if odd parity
INR C
Next: INX H
DCR B
JNZ Back
MOV A, C
STA 810A
HLT
```