

Addition of two 8-bit numbers

```
MVI C,00 ;Initialize C register to 00
LDA 4150h ;Load the value to Accumulator.
MOV B,A ;Move the content of Accumulator to B register.
LDA 4151h ;Load the value to Accumulator.
ADD B ;Add the value of register B to Accumulator.
JNC LOOP ;Jump on no carry.
INR C ;Increment value of register C.
LOOP: STA 4152h ;Store the value of Accumulator.
MOV A,C ;Move content of Register C to Accumulator
STA 4153h ;Store the value of Accumulator
HLT
```

Addition of two 16-bit numbers

```
LHLD 2000H ;Get first 16-bit number
XCHG ;Save first 16-bit number in DE
LHLD 2002H ;Get second 16-bit number in HL
DAD D ;Add DE and HE
SHLD 2004H ;Store 16-bit result in memory locations 2004H and 2005H.
HLT ;Terminate program execution
```

Subtraction of two 8-bit numbers

```
MVI C,00h ;Initialize C register to 00
LDA 4150h ;Load the value to Accumulator.
MOV B,A ;Move the content of Accumulator to B register.
LDA 4151h ;Load the value to Accumulator.
SUB B ;Add the value of register B to Accumulator.
JNC LOOP ;Jump on no carry.
CMA ;Complement Accumulator Content
INR A ;Increment value of register C.
INR C ;Increment value of register C.
LOOP: STA 4152h ;Store the value of Accumulator.
MOV A,C ;Move content of Register C to Accumulator
STA 4153h ;Store the value of Accumulator
HLT
```

Subtraction of two 16-bit numbers

```

LXI H,9876H ;load HL with 9876H data
LXI B,5432H ;load BC with 9876H data
MOV A,L ;move the content of L reg. into Accumulator
SUB C ;subtract the content of L reg. into Accumulator
STA 2000H ;Store the LSB into 2000H
MOV A,H ;move the content of H reg. into Accumulator
SUB B ;subtract the content of H reg. into Accumulator
STA 2001H ;Store the LSB into 2000H
HLT ;Stop

```

Multiplication of two 8- bit nos. using repeated Addition

```

MVI D,00h ;Initialize D register to 00
MVI A,00h ;Initialize A register to 00
LXI H,4150h
MOV B,M ;Get the second number in B register
INX H
MOV C,M ;Get the second number in C register
LOOP: ADD B ;Add content of A to reg. B
JNC NEXT ;Jump on no carry to NEXT
INR C ;Increment the content of reg. C
NEXT: DCR C ;decrement the content of reg. C
JNZ LOOP ;Jump on no zero to address
STA 4152h ;Store the result in memory
MOV A,D
STA 4153h ;Store the MSB of result in memory
HLT

```

Division of two 8- bit nos. using repeated Subtraction

```

LXI H,4150h ;Load HL with 4150H
MOV B,M ;Get the dividend in b register.
MVI C,00h ;Clear C reg. for quotient.
INX H
MOV A,M ;Get the divisor in A register
NEXT: CMP B ;Compare A reg.with reg B
JC LOOP ;Jump on no carry to NEXT
SUB B ;Increment the content of reg. C
INR C ;decrement the content of reg. C
JMP NEXT ;Jump on no zero to address
LOOP: STA 4152h ;Store the result in memory
MOV A,C
STA 4153h ;Store the MSB of result in memory
HLT

```

1's complement of 8 bit number

```
LDA 2200H ;Get the number
CMA ;Complement number
STA 2300H ;Store the result
HLT ;Terminate program execution
```

2's complement of a no

```
LDA 2200H ;Get the number
CMA ;Complement the number
ADI 01H ;Add one in the number
STA 2300H ;Store the result
HLT ;Terminate program execution
```

Find smallest Number From an array

```
LXI H,3000h ;Load H-L pair with address 3000H
MOV C,M ;Move counter from memory to reg. C.
INX H ;Increment H-L pair.
MOV A,M ;Move the 1 st number from memory to reg. A
DCR C ;Decrement counter.
LOOP: INX H ;Increment H-L pair.
MOV B,M ;Move the next number from memory to reg. B
CMP B ;Compare B with A.
JC AHEAD ;Jump to AHEAD if there is a carry.
MOV A,B ;Move largest from reg. B to reg. A.
AHEAD: DCR C ;Decrement counter.
JNZ LOOP ;Jump to LOOP if counter is not zero.
INX H ;Increment H-L pair.
MOV M,A ;Move the result from reg. A to memory.
HLT
```

Find smallest Number From an array

```
LXI H,3000h ;Load H-L pair with address 3000H
MOV C,M ;Move counter from memory to reg. C.
INX H ;Increment H-L pair.
MOV A,M ;Move the 1 st number from memory to reg. A
DCR C ;Decrement counter.
LOOP: INX H ;Increment H-L pair.
MOV B,M ;Move the next number from memory to reg. B
CMP B ;Compare B with A.
JC AHEAD ;Jump to AHEAD if there is a carry.
MOV A,B ;Move largest from reg. B to reg. A.
AHEAD: DCR C ;Decrement counter.
JNZ LOOP ;Jump to LOOP if counter is not zero.
INX H ;Increment H-L pair.
MOV M,A ;Move the result from reg. A to memory.
HLT
```

Transfer Block of data bytes from one memory location to another

```

MVI C,0AH ;Initialize counter
LXI H,2200H ;Initialize source memory pointer
LXI D,2300H ;Initialize destination memory pointer
BACK: MOV A,M ;Get byte from source memory block
STAX D ;Store byte in the destination memory block
INX H ;Increment source memory pointer
INX D ;Increment destination memory pointer
DCR C ;Decrement counter
JNZ BACK ;If counter 0 repeat
HLT ;Terminate program execution

```

Arrange data bytes in ascending order

```

LXI H,4200H
MOV C,M
DCR C
REPEAT: MOV D,C
LXI H,4201H
LOOP: MOV A,M
INX H
CMP M
JC SKIP
MOV B,M
MOV M,A
DCX H
MOV M,B
INX H
SKIP: DCR D
JNZ LOOP
DCR C
JNZ REPEAT
HLT

```

Arrange data bytes in ascending order

```

LXI H,4200H
MOV C,M
DCR C
REPEAT: MOV D,C
LXI H,4201H
LOOP: MOV A,M
INX H
CMP M
JNC SKIP
MOV B,M

```

```
MOV M,A
DCX H
MOV M,B
INX H
SKIP: DCR D
JNZ LOOP
DCR C
JNZ REPEAT
HLT
```