

1. BLOCK TRANSFER FROM ONE MEMORY ADDRESS TO ANOTHER

MVI H, 0A ; Size of data

LXI B, 2000H

LXI D, 3000H

UP:

LDAX B ; Move data from memory address pointed by B into the accumulator

STAX D ; Move data from accumulator to memory address pointed by D

INX B

INX D

DCR H

JNZ UP

HLT

2. INVERT BLOCK TRANSFER FROM ONE MEMORY ADDRESS TO ANOTHER

MVI H, 0AH

LXI B, 2000H

LXI D, 3009H

UP:

LDAX B

STAX D

INR B

DCR H

DCR D

JNZ UP

HLT

3. ADDITION OF TWO 8-BIT NUMBERS, SUM 8-BITS

LXI H, 2000H

MOV B, M

INX H

MOV A, M
ADD B
INX H
MOV M, A
HLT

4. ADDITION OF TWO 8-BIT NUMBERS, SUM 16 BITS

LXI H, 2000H
MVI C, 00H
MOV A, M
INX H
MOV B, M
ADD B
JNC DOWN
INR C

DOWN:

 INX H
 MOV M, A
 INX H
 MOV M, C
HLT

5. ADDITION OF TWO 16-BIT NUMBERS (WITH CARRY)

MVI C, 00H
LHLD 2000H ; Load the contents of memory at address 2000H into HL
XCHG ; Exchange the contents of HL and DE registers
LHLD 2002H
DAD D ; HL = HL + DE
JNC DOWN

INR C

DOWN:

SHLD 3000H ; Store HL pair to address 3000H

MOV A, C

STA 3002H

HLT

6. SUBTRACTION OF TWO 8-BIT NUMBERS (DISPLAY OF BORROW)

MVI C, 00H

LXI H, 2000H

MOV B, M

INX H

MOV A, M

SUB B

JNC DOWN

INR C

DOWN:

INX H

MOV M, A

INX H

MOV M, C

HLT

7. SUBTRACTION OF TWO 16-BIT NUMBERS (DISPLAY OF BORROW)

MVI B, 00H

LHLD 2000H

XCHG

LHLD 2002H

MOV A, E
SUB L
MOV E, A
MOV A, D
SBB H
JNC DOWN
INR B

DOWN:

MOV D, A
XCHG
SHLD 2004H
MOV A, B
STA 2006H
HLT

8. MULTIPLICATION OF TWO 8-BIT NUMBERS

LXI H, 2000H
MOV B, M
INX H
MOV C, M
MVI D, 00H
XRA A ; A = 0 (clear accumulator)

UP:

ADD B
JNC DOWN
INR D

DOWN:

```
DCR C
JNZ UP
STA 2200H
MOV A, D
STA 2200H
HLT
```

9. DIVISION OF A 8-BIT NUMBER BY ANOTHER 8-BIT NUMBER

```
MVI C, 00H
LXI H, 2000H
MOV A, M
INX H
MOV B, M
```

UP:

```
CMP B
JC DOWN
SUB B
INR C
JMP UP
```

DOWN:

```
INX H
MOV M, C
INX H
MOV M, A
HLT
```

10. TO FIND THE SQUARE OF AN 8-BIT NUMBER USING LOOKUP TABLE

LDA 3000H ; User number will be stored here (00-09)

MOV L, A
MVI H, 20H
MOV A, M
STA 3001H
HLT

LOOK UP TABLE:

Address	Data
2000H	00H
2001H	01H
2002H	04H
2003H	09H
2004H	10H
2005H	19H

11. TO FIND THE SQUARE ROOT OF AN 8-BIT NUMBER

MVI D, 01H
MVI E, 01H
LDA 2000H

UP:

SUB D
JZ DOWN
INR D
INR D
INR E
JMP UP

DOWN:

MOV A, E
STA 2001H

HLT

12. TO PRINT THE MULTIPLICATION TABLE OF INPUT 8-BIT NUMBER IN DECIMAL FORM

LDA 2000H

MOV D, A

MVI C, 00H

LXI H, 2200H

MVI B, 0AH

UP:

MOV A, D

ADD C

MOV M, A

INX H

MOV C, A

DCR B

JNZ UP

HLT

13. 1'S COMPLEMENT OF 8-BIT NUMBER

LDA 2000H

CMA ; Complement the accumulator data

STA 2200H

HLT

14. 1'S COMPLEMENT OF 16-BIT NUMBER

LDA 2000H

CMA

STA 2200H

LDA 2001H
CMA
STA 2201H
HLT

15. 2'S COMPLEMENT OF 8-BIT NUMBER

LDA 2000H
CMA
INR A
STA 2201H
HLT

16. 2'S COMPLEMENT OF 16-BIT NUMBER

MVI B, 00H
LDA 2000H
CMA
INR A
JNZ DOWN
INR B

DOWN:

STA 2200H
LDA 2001H
CMA
ADD B ; First complement accumulator, then add carry
STA 2201H
HLT

17. TO FIND THE LARGEST OF TWO NUMBERS

LXI H, 2000H

MOV B, M
INX H
MOV A, M
CMP B
JNC DOWN
MOV A, B

DOWN:

INX H
MOV M, A
HLT

18. TO FIND THE SMALLEST OF TWO NUMBERS

LXI H, 2000H
MOV B, M
INX H
MOV A, M
CMP B
JC DOWN
MOV A, B

DOWN:

STA 2200H
HLT

19. TO FIND THE NUMBER OF ONE'S IN AN 8-BIT NUMBER

LDA 2000H
MVI B, 08H
MVI C, 00H

UP:

RAR

JNC DOWN

INR C

DOWN:

DCR B

JNZ UP

MOV A, C

STA 2200H

HLT

20. TO FIND THE NUMBER OF ODD AND EVEN NUMBERS IN A DATA ARRAY

LXI H, 2000H

MOV C, M ; Size of array

MVI B, 00H ; For even

MVI D, 00H ; For odd

UP:

INX H

MOV A, M

RAR

JC DOWN1

INR B

JMP DOWN2

DOWN1:

INR D

DOWN2:

DCR C

JNZ UP

MOV A, B

STA 2200H ; No. of even numbers

MOV A, D

STA 2201H ; No. of odd numbers

HLT

21. TO FIND THE SMALLEST NUMBER IN A DATA ARRAY

LXI 2000H

MOV C, M ; Size of array

INX H

DCR C

MOV A, M

UP:

INX H

CMP M

JC DOWN

MOV A, M

DOWN:

DCR C

JNZ UP

STA 2200H

HLT

22. TO FIND THE LARGEST NUMBER IN A DATA ARRAY

LXI H, 2000H

MOV C, M ; Size of array

DCR C

INX H

MOV A, M

UP:

INX H

CMP M

JNC DOWN

MOV A, M

DOWN:

DCR C

JNZ UP

STA 2200H

HLT

23. TO ARRANGE A SERIES IN ASCENDING ORDER

LXI H, 2000H

MOV B, M

UP2:

MOV C, B

INX H

UP1:

MOV A, M

INX H

MOV D, M

CMP D
JC DOWN
MOV M, A
DCX H
MOV M, D
INX H

DOWN:

DCR C
JNZ UP1
DCR B
JNZ UP2

HLT

24. TO ARRANGE A SERIES IN DESCENDING ORDER

LXI H, 2000H
MOV B, M

UP2:

MOV C, M
INX H

UP1:

MOV A, M
INX H
MOV D, M
CMP D
JNC DOWN
MOV M, A
DCX H

MOV M, D

INX H

DOWN:

DCR C

JNZ UP1

DCR B

JNZ UP2

HLT

25. TO FIND THE SUM OF A SERIES OF 8-BIT NUMBERS IN HEX AND BCD FORM

MVI C, 05H ; Size of array

MVI B, 00H ; For carry

MVI D, 00H

XRA A

LXI H, 2000H

UP:

MOV D, M

ADD D

JNC DOWN

INR B

DOWN:

INX H

DCR C

JNZ UP

STA 2200H ; Addition result

MOV A, B

STA 2201H ; Carry

HLT

26. TO FIND THE NUMBER OF EVEN AND ODD NUMBERS IN A DATA ARRAY

MVI B, 00H ; For even
MVI D, 00H ; For odd
MVI C, 09H ; Size of array
LXI H, 00H

UP:

MOV A, M
RAR
JNC DOWN1
INR B
JMP DOWN2

DOWN1:

INR D

DOWN2:

INX H
DCR C
JNZ UP

MOV A, B
STA 2200H ; Even numbers
MOV A, D
STA 2201H ; Odd numbers
HLT

27. TO FIND THE SUM OF EVEN NUMBERS AND ODD NUMBERS IN A DATA ARRAY

MVI B, 00H ; For even

MVI D, 00H ; For odd

MVI C, 06H

XRA A

LXI 2000H

UP:

MOV A, M

RAR

JC DOWN

ADD B

MOV B, A

XRA A

JMP DOWN2

DOWN:

ADD D

MOV D, A

XRA A

DOWN2:

INX H

DCR C

JNZ UP

MOV A, B

STA 2200H

MOV A, D

STA 2201H

HLT

28. TO FIND THE FACTORIAL OF A NUMBER

LXI H, 2000H

MOV B, M

MVI D, 01H

UP:

CALL MUL ; Call subroutine for multiplication

DCR B

JNZ UP

STA 2200H

HLT

MOV C, B

XRA A

UP:

ADD D

DCR C

JNC UP

MOV D, A

RET

29. TO FIND THE FIRST TEN ELEMENTS OF A FIBONACCI SERIES

LXI H, 2000H

MVI B, 00H

MVI D, 0AH ; For counter

MVI C, 01H

XRA A

UP:

MOV M, A

ADD C
MOV B, C
MOV C, A
MOV A, B
INX H
DCR D
JNZ UP
HLT

30. TO FIND THE SUM OF UPPER AND LOWER NIBBLE OF AN 8-BIT NUMBER

LXI H, 2000H
MOV A, M
ANI F0H
RRC
RRC
RRC
RRC
MOV B, A
MOV A, M
ANI 0FH
ADD B
STA 2200H
HLT

31. ADDITION OF TWO 8-BIT NUMBERS USING A SUBROUTINE

assembly
CopyEdit
MVI D, 00H
LDA 2500H
MOV B, A

```
LDA 2501H  
CALL ADDITION  
STA 2502H  
MOV A, D  
STA 2503H  
HLT
```

ADDITION:

```
ADD B  
JNC DOWN  
INX D
```

DOWN:

```
RET
```