NAME-ARPAN MANDAL DEPT-CSE YEAR-2nd

ROLL-001910501061 GROUP-A3

SUBJECT-MICROPROCESSOR AND ASSEMBLY LANGUAGE LAB

ASSIGNMENT-2

-------------------------------------------------------------------------------------------------------------------------------------

1. Two numbers MNH and KLH are stored in 2050H and 2051H, respectively. Write a program to assemble them as NKH and LMH store them in 2052H and 2053H.

2. Two numbers A & B are stored in 2050H and 2051H, respectively. Write a program to perform A×B and store the result in 2052H and 2053H.

3. *N* numbers are stored in consecutive m/m location starting from 2050H. The value *N* is stored in 204FH.

i) Find the maximum among the *N* numbers.

ii) Find the minimum among the *N* numbers.

iii) Sort the *N* numbers in ascending order.

iv) Sort the *N* numbers in descending order.

4. *N* numbers are stored in consecutive m/m location starting from 2050H. The value *N* is stored in 204FH. Write a program to copy the even and odd numbers starting from 2100H and 2200H, respectively. Store the total no. of even and odd numbers in 2300H and 2201H, respectively.

5. *N* numbers are stored in consecutive m/m location starting from 2050H. The value *N* is stored in 204FH. Write a program to test whether a number stored in 204EH is present in the list. If present, store its position in the list at 204DH; otherwise store FFH

1.

LDA 2050H // [A]🡨[2050H]

ANI 0FH // [A]🡨[A]^[0FH]

RRC // ROTATE ACCUMULATOR RIGHT

RRC // “

RRC // “

RRC // “

MOV B,A // [B]🡨[A]

LDA 2050H // [A]🡨[2050H]

ANI F0H // [A]🡨[A]^[F0H]

RRC //ROTATE ACCUMULATOR RIGHT

RRC // “

RRC // ‘’

RRC // ‘’

MOV C,A // [C]🡨[A]

LDA 2051H //[A]🡨[2051H]

ANI 0FH //[A]🡨[A]^[0FH]

RRC //ROTATE ACCUMULATOR RIGH

RRC // ‘’

RRC // ‘’

RRC // ‘’

MOV D,A // [D]🡨[A]

LDA 2051H //[A]🡨[2051H]

ANI F0H // [A]🡨[A]^[F0H]

RRC //ROTATE ACCUMULATOR RIGHT

RRC // ‘’

RRC // ‘’

RRC // ‘’

MOV E,A //[E]🡨[A]

MOV A,B //[A]🡨[B]

ADD E //[A]🡨[A]+[E]

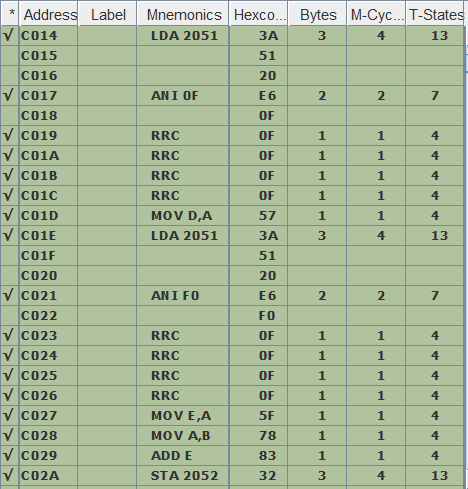
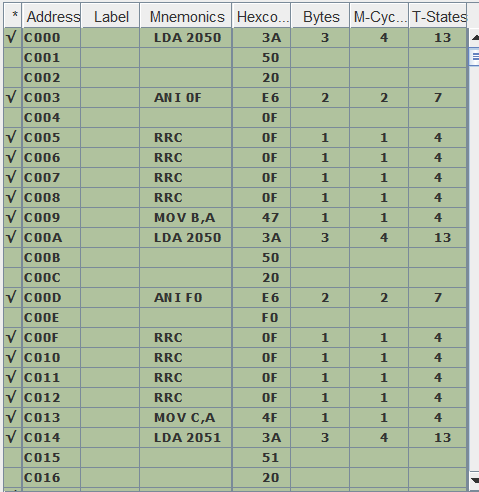
STA 2052H //[2052H]🡨[A]

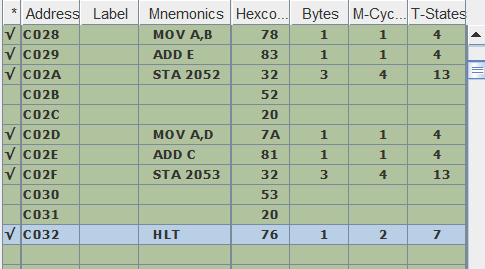
MOV A,D //[A]🡨[D]

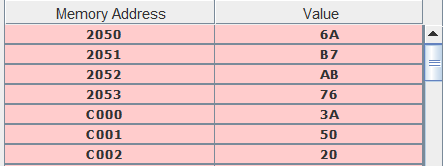
ADD C //[A]🡨[A]+[C]

STA 2053H //[2053H]🡨[A]

HLT //HALT







2.

LDA 2050H //[A]🡨[2050H]

MOV B,A //[B]🡨[A]

LDA 2051H //[A]🡨[2051H]

MOV D,A //[D]🡨[A]

MVI C,00H //[C]🡨 00H

MVI A,00H //[A]🡨 00H

LOOP: ADD D //[A]🡨[A]+[D]

JNC JUMP //JUMP IF CS=0

INR C //[C]🡨[C]+1

JUMP:DCR B //[B]🡨[B]-1

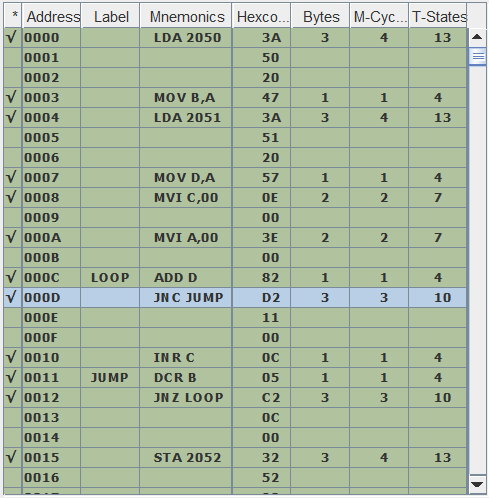
JNZ LOOP //JUMP IF [B]!=0

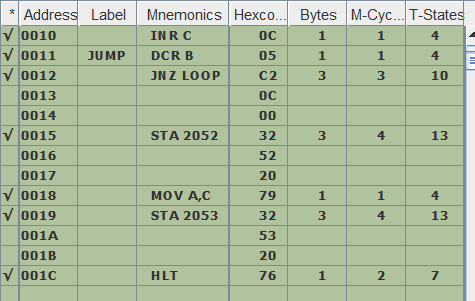
STA 2052H //[2052H]🡨[A]

MOV A,C //[A]🡨[C]

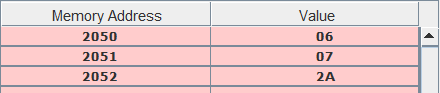
STA 2053H //[2053H]🡨[A]

HLT //HALT









3.

i)

LXI H,2050H //[H-L]🡨2050H

LDA 204FH //[A]🡨204FH

MOV B,A //[B]🡨[A]

MOV A,M //[A]🡨[[H-L]]

LOOP:CMP M //COMPARE [[H-L]] WITH ACCUMULATOR

JNC SKIP //JUMP IF CS =0

MOV A,M //[A]🡨[[H-L]]

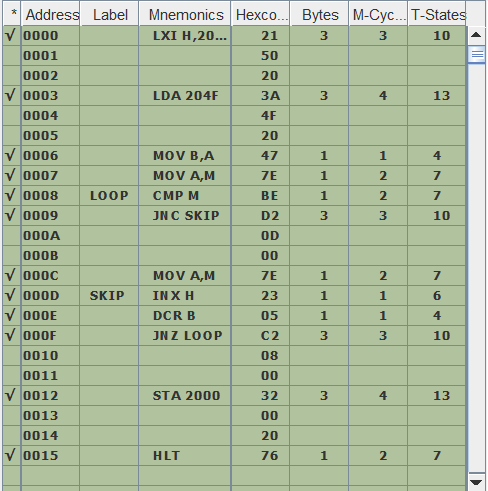
SKIP:INX H // [H-L]🡨[H-L]+1

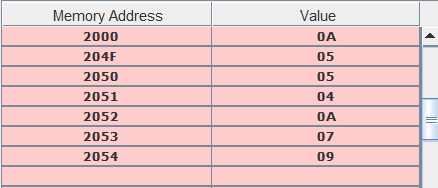
DCR B //[B]🡨[B]-1

JNZ LOOP //JUMP IF [B]!=0

STA 2000H //[2000H]🡨[A]

HLT //HALT





\*LARGEST VALUE STORE IN 2000H MEMORY LOCATION

ii)

LXI H,2050H //[H-L]🡨2050H

LDA 204FH //[A]🡨204FH

MOV B,A //[B]🡨[A]

MOV A,M //[A]🡨[[H-L]]

LOOP:CMP M //COMPARE [[H-L]] WITH ACCUMULATOR

JC SKIP //JUMP IF CS !=0

MOV A,M //[A]🡨[[H-L]]

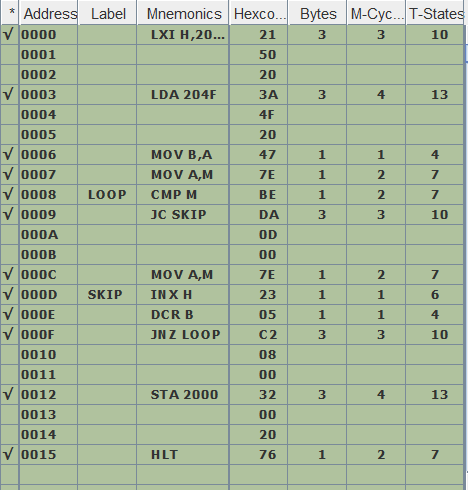
SKIP:INX H // [H-L]🡨[H-L]+1

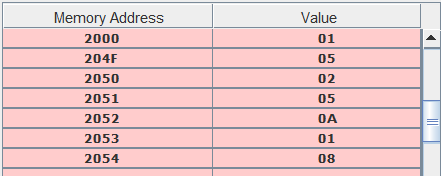
DCR B //[B]🡨[B]-1

JNZ LOOP //JUMP IF [B]!=0

STA 2000H //[2000H]🡨[A]

HLT // HALT





\*SMALLEST VALUE STORE AT 2000H MEMORY LOCATION

iii)

LDA 204FH //[A]🡨[204FH]

MOV D,A //[D]🡨[A]

LOOP1: LXI H,2050H //[H-L]🡨2050H

SHLD 2100H //[2100H]🡨[L]&[2101H]🡨[H]

MOV B,D //[B]🡨[D]

MOV A,M //[A]🡨[[H-L]]

LOOP: CMP M //COMPARE ACCUMULATOR WITH [[H-L]]

JNC SKIP //JUMP IF CY=0

MOV A,M //[A]🡨[[H-L]]

SHLD 2100H //[2100H]🡨[L]&[2101H]🡨[H]

SKIP: INX H //[H-L]🡨[H-L]+1

DCR B //[B]🡨[B]-1

JNZ LOOP //JUMP IF ZERO=0

DCX H //[H-L]🡨[H-L]-1

DCR D //[D]🡨[D]-1

MOV E,M //[E]🡨[[H-L]]

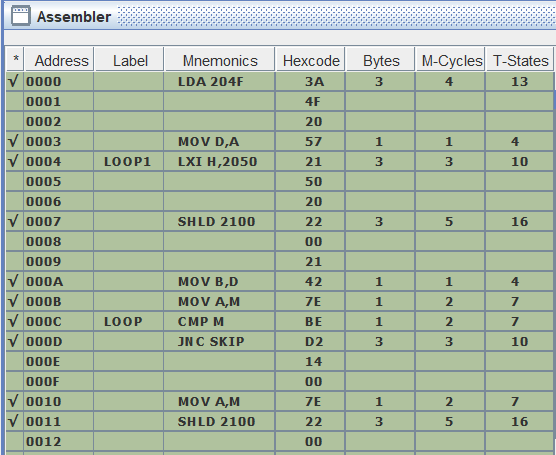
MOV M,A //[[H-L]]🡨[A]

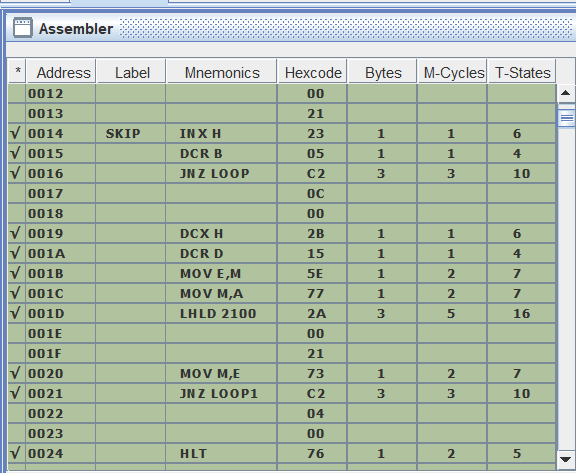
LHLD 2100H //[L]🡨[2100H] [H]🡨[2101H]

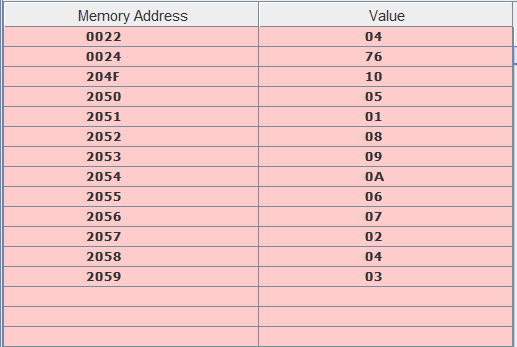
MOV M,E //[[H-L]]🡨[E]

JNZ LOOP1 //JUMP IF ZERO=0

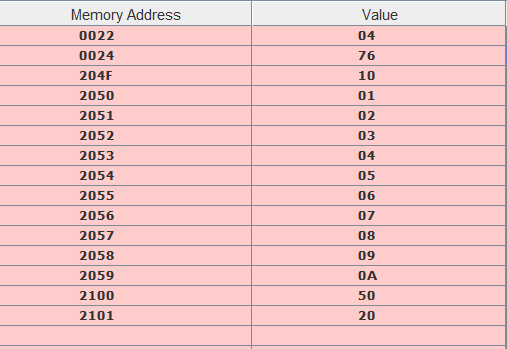
HLT //HALT







MEMORY ELEMENT BEFORE SORTING



MEMORY ELEMENT AFTER SORTING

Iv)

LDA 204FH //[A]🡨[204FH]

MOV D,A //[D]🡨[A]

LOOP1: LXI H,2050H //[H-L]🡨2050H

SHLD 2100H //[2100H]🡨[L]&[2101H]🡨[H] MOV B,D //[B]🡨[D]

MOV A,M //[B]🡨[[H-L]]

LOOP: CMP M //COMPARE ACCUMULATOR WITH [[H-L]]

JC SKIP //JUMP IF CY!=0

MOV A,M //[A]🡨[[H-L]]

SHLD 2100H //[L]🡨[2100H]&[H]🡨[2101H]

SKIP: INX H //[H-L]🡨[H-L]+1

DCR B //[B]🡨[B]-1

JNZ LOOP //JUMP IF ZERO=0

DCX H //[H-L]🡨[H-L]-1

DCR D //[D]🡨[D]-1

MOV E,M //[E]🡨[H-L]

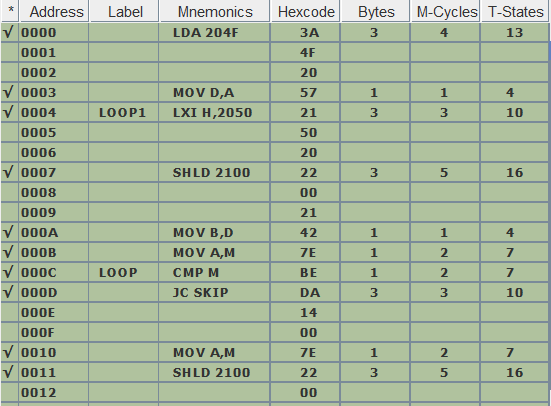
MOV M,A //[H-L]🡨[A]

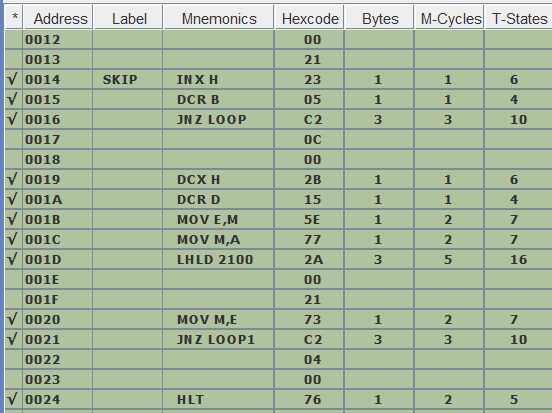
LHLD 2100H //[L]🡨[2100H] [H]🡨[2101H]

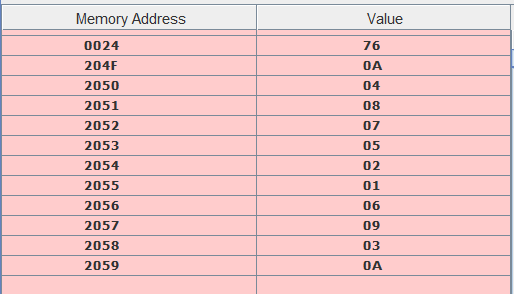
MOV M,E //[[H-L]]🡨[E]

JNZ LOOP1 //JUMP IF ZERO=0

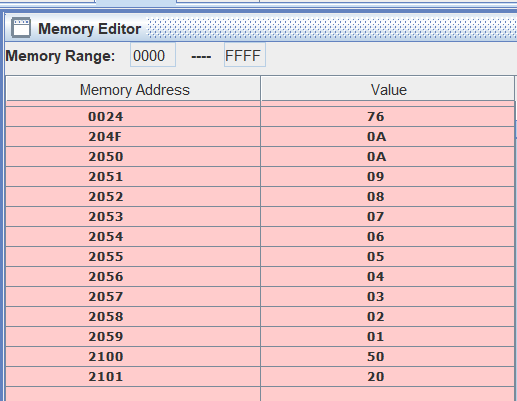
HLT //HALT







MEMORY LOCATIONS BEFORE SORTING



MEMORY LOCATIONS AFTER SORTING

4.

LXI H,2050 H //[H-L]🡨2050H

LDA 204FH //[A]🡨[204FH]

MOV B,A //[B]🡨[A]

MOV C,A //[C]🡨[A]

MVI D,00H //[D]🡨00H

LOOP: MOV A,M //[A]🡨[[H-L]]

ANI 01H //AND ACCUMULATOR WITH 01H

JNZ SKIP //JUMP IF ZERO!=0

INR D //[D]🡨[D]+1

SKIP: INX H //[H-L]🡨[H-L]+1

DCR B //[B]🡨[B]-1

JNZ LOOP //JUMP IF ZERO!=0

MOV A,D //[A]🡨[D]

STA 2300H //[2300H]🡨[A]

MOV A,C //[A]🡨[C]

SUB D //[A]🡨[A]-[D]

STA 2301H //[2301H]🡨[2301H]

LXI H,2050 //[H-L]🡨2050H

LXI D,2100 H //[D-E]🡨2100H

LOOP1: MOV A,M //[A]🡨[[H-L]]

MOV B,A /[B]🡨[A]

ANI 01 //AND ACCUMULATOR WITH 01H

JNZ SKIP1 //JUMP IF ZERO!=0

XCHG //[H-L]🡨[D-E],[D-E]🡨[H-L]

MOV M,B //[[H-L]]🡨[B]

INX H //[H-L]🡨[H-L]+1

XCHG //[H-L]🡨[D-E],[D-E]🡨[H-L]

SKIP1: INX H //[H-L]🡨[H-L]+1

DCR C //[C]🡨[C]-1

JNZ LOOP1 //JUMP IF ZERO!=0

LDA 204FH //[A]🡨[204FH]

MOV C,A //[C]🡨[A]

LXI H,2050H //[H-L]🡨2050H

LXI D,2200 //[D-E]🡨2200H

LOOP2: MOV A,M //[A]🡨[[H-L]]

MOV B,A //[B]🡨[A]

ANI 01H //AND ACCUMULATOR WITH 01H

JZ SKIP2 //JUMP IF ZERO=0

XCHG //[H-L]🡨[D-E],[D-E]🡨[H-L]

MOV M,B //[[H-L]]🡨[B]

INX H //[H-L]🡨[H-L]+1

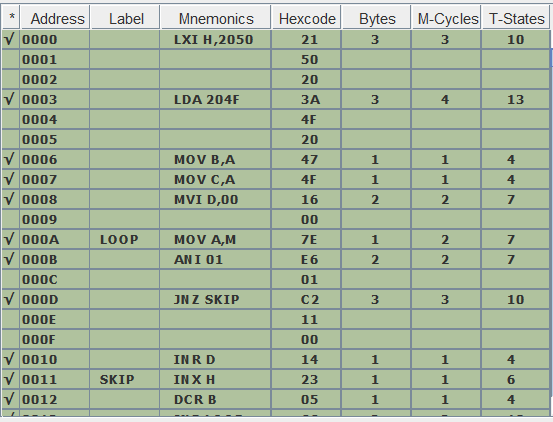
XCHG //[H-L]🡨[D-E],[D-E]🡨[H-L]

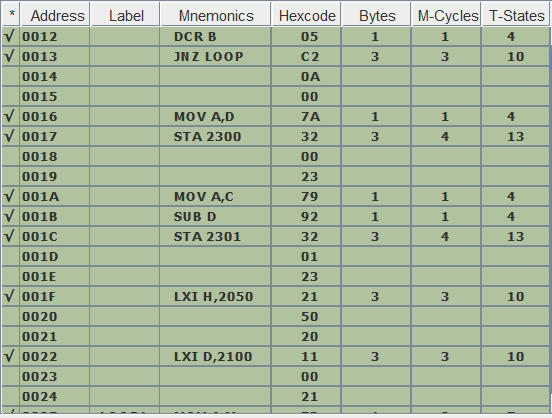
SKIP2: INX H //[H-L]🡨[H-L]+1

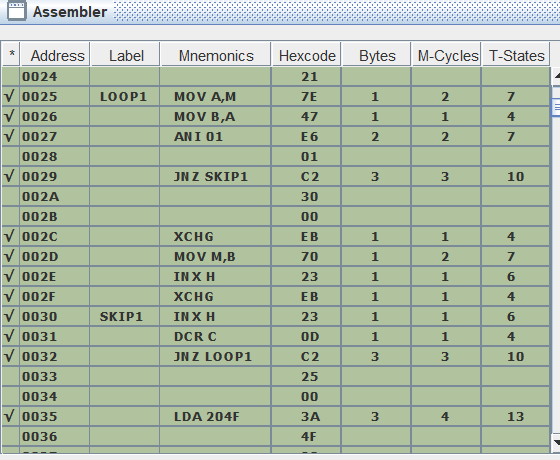
DCR C //[C]🡨[C]-1

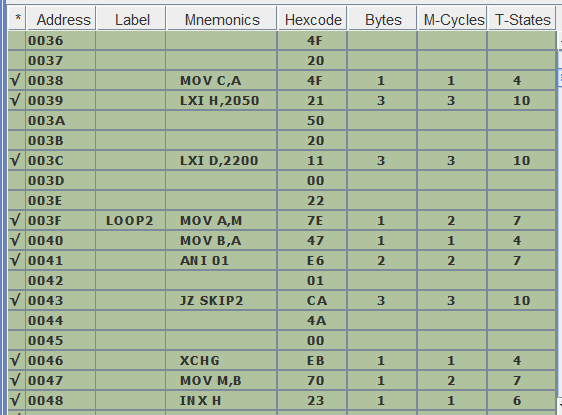
JNZ LOOP2 //JUMP IF ZERO!=0

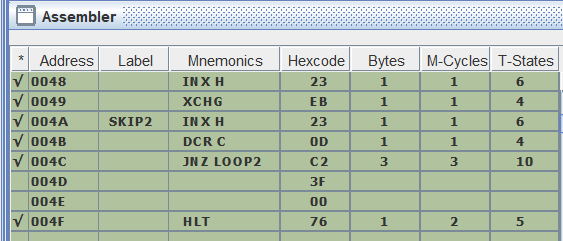
HLT //HALT

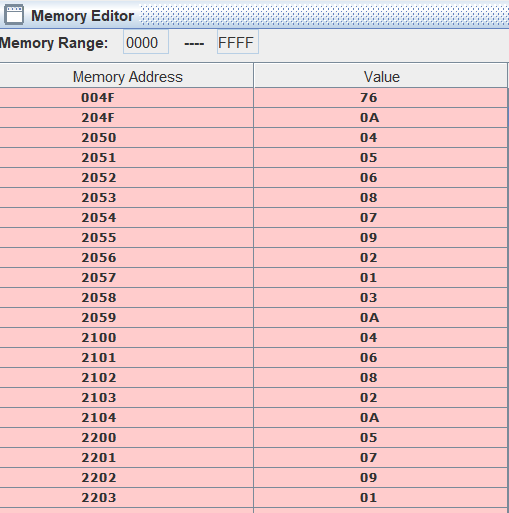


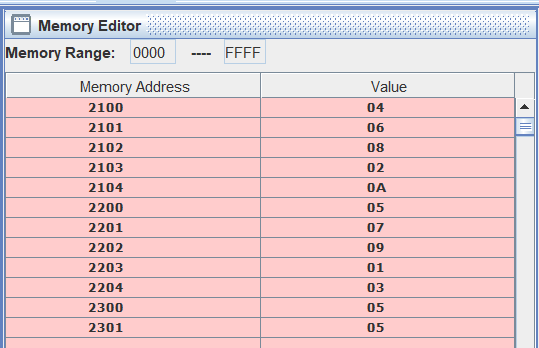












5.

LXI H,2050 //[H-L]🡨2050H

LDA 204F //[A]🡨[204FH]

MOV B,A //[B]🡨[A]

MVI D,00H //[D]🡨00H

MVI C,00 //[C]🡨00H

LDA 204E //A]🡨[204EH]

LOOP: CMP M //COMPARE ACCUMULATOR WITH [[H-L]]

JNZ SKIP //JUMP IF ZREO!=0

INR C //[C]🡨[C]+1

MOV A,D //[A]🡨[D]

STA 204D //[204DH]🡨[A]

JMP END //JUMP AT END

SKIP: INR D //[D]🡨[D]+1

INX H //[H-L]🡨[H-L]+1

DCR B //[B]🡨[B]-1

JNZ LOOP //JUMP IF ZERO!=0

MOV A,C //[A]🡨[C]

ADD C //[A]🡨[A]+[C]

JNZ SKIP1 //JUMP IF ZERO!=0

MVI A,FFH //[A]🡨FFH

STA 204DH //[204DH]🡨[A]

SKIP1: HLT //HALT

END: HLT

