1. A set of *N* data bytes is stored in m/m locations starting from 2501H. The value of *N* is stored in 2500H. Write a program to store these data bytes from m/m location 2600H if D0 or D7 is 1; otherwise reject the data byte.

2. There are *N* data bytes stored from m/m location 2200H. The value of *N* is stored in 21FFH. Write an 8085 program to find the sum of integers whose LSB and MSB are 1. Store the result in 2500H and 2501H.

3. Write an 8085 program to generate *N*th fibonacci number using function and store it in 2050H. The value of *N* (8-bits) is stored in memory 2060H.

4. Write a program to transfer a block of bytes of size *N* from location1 to location2 (location2 > location1) when the size of overlap between the two locations is defined by *M*. The values of *N* and *M* are stored in 201EH and 201FH, respectively.

5. Write a program to flash “**BCSE II**” in the address and data fields with a flashing rate of 0.5 seconds.

1.

LDA 2050H //[A]🡨[2050H]

MOV B,A //[B]🡨[A]

MVI C,01H //[C]🡨01H

LXI H,2051H //[H-L]🡨2051H

LXI D,2600H //[D-E]🡨2600H

LOOP: MOV A,M //[A]🡨[[H-L]]

ANI 80H //AND ACCUMULATOR WITH 80H

JNZ SKIP1 //JUMP WHEN [A]!=0

MOV A,M //[A]🡨[[H-L]]

ANI 01H //AND ACCUMULATOR WITH 01H

JNZ SKIP1 //JUMP WHEN [A]!=0

JMP SKIP2 //JUMP AT SKIP2

SKIP1:MOV A,M //[A]🡨[[H-L]]

XCHG //[H-L]🡨[D-E] AND [D-E]🡨[H-L]

MOV M,A //[[H-L]]🡨[A]

INX H //[H-L]🡨[H-L]+0001H

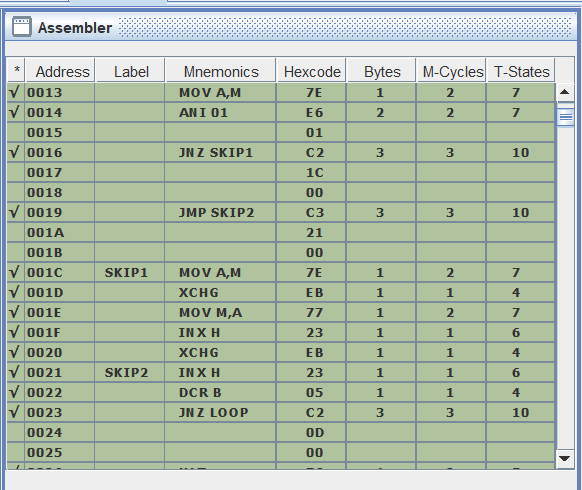
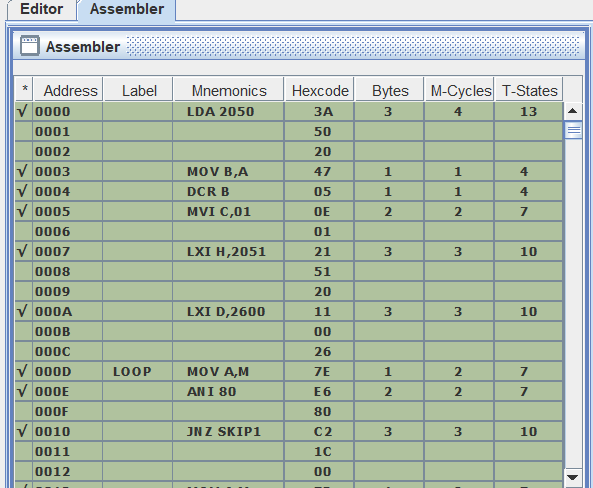
XCHG //[H-L]🡨[D-E] AND [D-E]🡨[H-L]

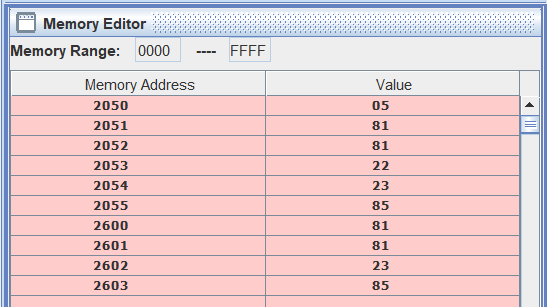
SKIP2: INX H //[H-L]🡨[H-L]+0001H

DCR B //[B]🡨[B]-01H

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

HLT //HALT





2.

LDA 21FFH

MOV B,A

LXI H,2200H

MVI C,00H

LOOP: MOV A,M

ANI 80H

JZ SKIP

MOV A,M

ANI 01H

JZ SKIP

XRA A

MOV A,C

ADD M

JNC SKIP2

INR D

SKIP2:MOV C,A

SKIP: INX H

DCR B

JNZ LOOP

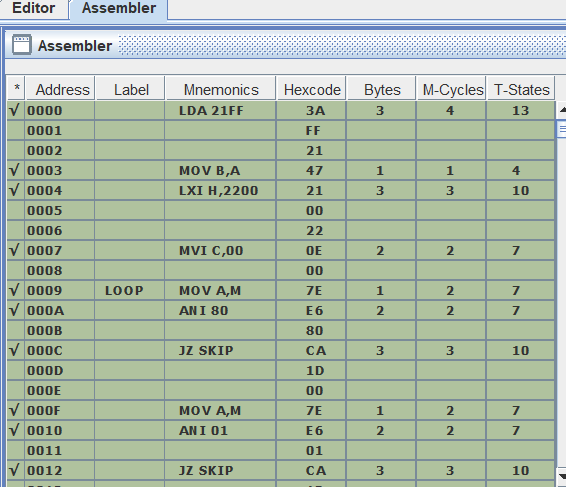
MOV A,C

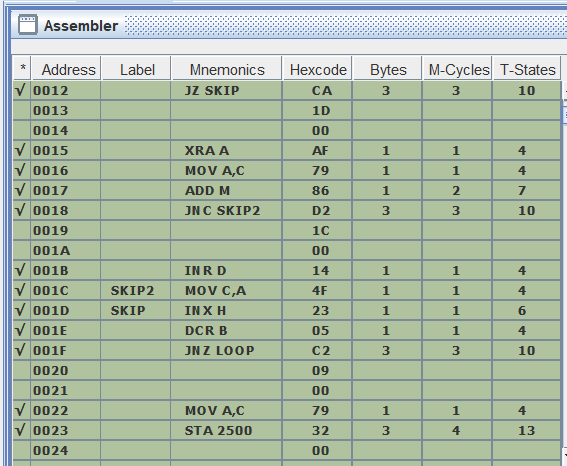
STA 2500H

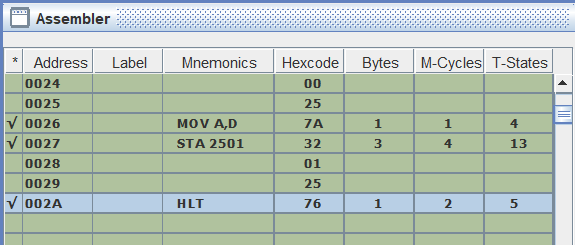
MOV A,D

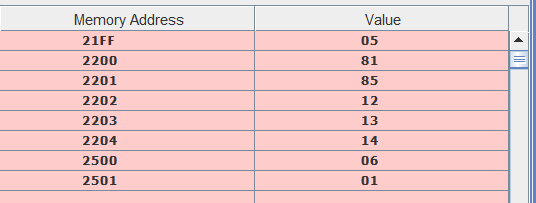
STA 2501H

HLT









3.

LDA 2060H //[A]🡨[2060H]

MOV B,A //[B]🡨[A]

DCR B //[B]🡨[B]-01H

DCR B //[B]🡨[B]-01H

MVI C,00H //[C]🡨00H

MVI D,01H //[D]🡨01H

LOOP: MOV A,C //[A]🡨[C]

ADD D //[A]🡨[A]+[D]

MOV C,D //[C]🡨[D]

MOV D,A //[D]🡨[A]

DCR B //[B]🡨[B]-01H

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

MOV A,D //[A]🡨[D]

STA 2050H //[2050H]🡨[A]

HLT //HALT

