Microprocessor (8085) Assignment 1

Q1. Load the contents of the memory locations 2200_H and 2201_H into registers. Add these registers and store the result in memory locations 2202_H and 2203_H.

```
        LXI H,2200      ;STORE THE ADDRESS 2200 INTO MEMORY
        MVI C,00        ;SET THE VALUE OF C REG AS 0 TO STORE THE CARRY
        MOV A,M         ;MOVE THE CONTENT OF MEMORY TO ACCUMULATOR
        INX H           ;INCREMENT THE VALUE OF MEMORY LOCATION , [HL]=[HL]+1
        ADD M           ;ADDING THE VALUE OF MEMORY TO THE ACCUMULATOR
        JNC LABEL       ;IF NO CARRY SKIP AND JUMP TO LABEL
        INR C           ;IF CARRY THEN STORE THE CARRY IN C REGISTER

LABEL:  STA 2202        ;STORE THE RESULT INTO LOCATION 2202
        MOV A,C         ;MOV THE CARRY VALUE TO ACC
        STA 2203        ;STORE THE CARRY VALUE INTO LOCATION 2203
        HLT             ;END THE PROGRAM
```

2. Find the sum of N numbers stored in consecutive locations starting from 2500_H. The value of N is stored in 2200_H. Store the result in locations 2300_H and 2301_H.

```
        LDA 2200H ; LOAD THE CONTENT OF 2200 INTO THE ACCUMULATOR(THE VALUE OF N)
        MOV B,A   ;MOV THE CONTENT OF ACCUMULATOR TO REGISTER B
        LXI H,2500H ; LOAD THE HL PAIR WITH THE ADDRESS 2500
        MVI A,00H  ; STORE 0 IN THE ACCUMULATOR
        MVI C,00H  ; STORE 0 IN THE C REGISTER
LABEL1: ADD M    ; ADDING THE VALUE OF MEMORY LOCATION TO ACCUMULATOR CONTENT
        INR L      ; INCREMENT THE MEMORY LOCATION POINTER
        JNC LABEL  ; IF NO CARRY , JUMP TO LABEL
        INR C  ; ELSE INCREMENT THE C REGISTER
LABEL: DCR B ; DECREMENT THE LOOP COUNTER
        JNZ LABEL1  ; IF LOOP COUNTER NOT ZERO THEN REPEAT THE LOOP
```

```
STA 2300H;  STORE THE RESULT INTO 2300
MOV A,C  ; MOVE THE CARRY VALUE TO ACCUMULATOR
STA 2301H  ; STORE THE ACCUMULATOR VALUE TO ADDRESS 2301
HLT        ; END THE PROGRAM
```

3. Find the sum of the least significant 4 bits and most significant 4 bits of a byte stored in memory location 2500_H. Store the result in 2550_H.

```
LXI H,2500   ;LOADING THE ADDRESS INTO MEM
MOV B,M    ;MOV THE VALUE FROM MEM TO B REG
MOV A,B    ;MOV THE CONTENT OF B TO A REG
ANI 0F       ; AND THE ACC CONTENT WITH 0F(HEX)
MOV C,A     ;MOV THE ACC CONTENT TO C REG
MOV A,B    ;MOV THE B REG CONTENT TO ACC
ANI F0       ;AND THE ACC CONTENT WITH
RRC          ;RIGHT ROTATE THE ACC
RRC          ;RIGHT ROTATE THE ACC
RRC          ;RIGHT ROTATE THE ACC
RRC          ;RIGHT ROTATE THE ACC
ADD C      ; ADD THE ACC CONTENT TO C
STA 2550   ;STORE THE RESULT INTO 2550
HLT          ;END THE PROGRAM
```

4. Write a program to count the '1's and '0's of a byte stored in 2500_H. Store the result in 2610_H and 2511_H, respectively.

```
//CALCULATING THE NUMBER OF ZEROES AND ONES IN THE GIVEN
//8 BIT NUMBER
    LXI H,2500   ;LOAD THE ADDRESS INTO MEMORY
    MVI C,08      ;STORING THE COUNTER OF LOOP INTO C REG
    MOV A,M      ;MOVING THE MEM CONTENT TO ACC
    MVI B,00       ;SETTING THE VALUE OF B REG TO 0,FOR COUNT OF 0
LABEL1:RLC          ;LOOP ROTATE LEFT THROUGH CARRY 8 TIMES
    JC LABEL        ;IF CARRY SKIP
```

```
        INR B           ; ELSE INCREMENT VALUE OF B REG
LABEL:DCR C             ;DECREMENT THE COUNTER VALUE
        JNZ LABEL1      ;REPEAT IF COUNTER NOT ZERO
        MOV A,B         ;STORE COUNT OF ZERO IN ACC
        STA 2610        ;STORE THE COUNT OF ZERO FROM ACC TO ADDRESS 2610
        MVI A,08        ;STORE 08H IN ACC
        SUB B           ; SUB THE NUM OF ZEROES TO GET NUMBER OF 1'S
        STA 2611        ;STORE THE COUNT OF 1 INTO ADDRESS 2611
        HLT             ;END THE PROGRAM
```

//HERE WE CALCULATE THE NUMBER OF 1'S IN THE NUMBER AND SUBTRACT IT FROM 8 TO GET THE NUMBER OF ZEROES

5. Write a program to sum two 16-bits binary numbers.

```
// ADDITION OF TWO 16BIT NUMBERS SUM 16 BITS OR MORE
// MANUALLY STRORE 1ST 16 BIT NO IN THE MEMORY LOCATION 2500 AND 2501
//HIGHER BYTE INTO 2501 ,LOWER INTO 2500
// MANUALLY STRORE 2ND 16 BIT NO IN THE MEMORY LOCATION 2502 AND 2503
//HIGHER BYTE INTO 2503 AND LOWER BYTE INTO 2502
//ADD THEM USING THE INSTRUCTIONS
//STORE THE RESULT IN MEMORY LOCATION 2300
        LHLD 2500       ;L=[2500],H=[2501]
        XCHG            ;EXCHANGE VALUE HL WITH DE REGISTER PAIR
        LHLD 2502       ; L=[2502], H=[2503]
        MVI C,00        ;SET THE C REG VALUE TO ZERO FOR STORING THE CARRY
        DAD D           ;[HL]=[HL]+[DE]
        JNC LABEL       ;IF NOT CARRY JUMP TO LABEL
        INR C           ;ELSE STORE THE CARRY BY INCREMENTING THE C   REGISTER
LABEL:  SHLD 3300       ;STORE THE RESULT INTO LOCATIONS 3300 AND 330
        MOV A,C         ;MOV THE CARRY VALUE TO ACC
```

```
STA 3302        ;STORE THE CARRY VALUE TO ADDRESS 3302
HLT             ;END THE PROGRAM
```

Microprocessor (8085) Assignment 2

1. Two numbers MN_H and KL_H are stored in 2050_H and 2051_H, respectively. Write a program to assemble them as NK_H and LM_H store them in 2052_H and 2053_H.

```
// ASSEMBLE THE NUMS STORED IN 2050 AND 2051 WHICH ARE
// MN AND KL AS NK AND LM RESPECTIVELY AND STORE IN 2052
// AND 2053
// THE FIRST PART OF ASSEMBLING
        LXI H,2050 ;  STORE THE ADDRESS 2050 INTO THE MEMORY
        MOV A,M  ; MOV THE VALUE OF MEMORY TO ACCUMULATOR
        ANI 0F ; AND THE MEMORY CONTENT WITH 0F
        RLC ; LEFT ROTATE ACCUMULATOR
        RLC  ; LEFT ROTATE ACCUMULATOR
        RLC ; LEFT ROTATE ACCUMULATOR
        RLC  ; LEFT ROTATE ACCUMULATOR
        MOV B,A ; MOV THE VALUE OF ACCUMULATOR TO B REG
        INX H; INCREMENT THE MEMORY POINTER TO NEXT LOC
        MOV A,M  ; MOV THE MEMORY CONTENT TO ACCUMULATOR
        ANI F0  ; AND WITH F0H
        RRC; RIGHT ROTATE ACCUMULATOR
        RRC  ; RIGHT ROTATE ACCUMULATOR
        RRC   ; RIGHT ROTATE ACCUMULATOR
        RRC  ; RIGHT ROTATE ACCUMULATOR
        ADD B ; ADD THE ACCUMULATOR CONTENT WITH B REG CONTENT
```

```
        STA 2052 ; STORE THE RESULT INTO ADDRESS 2052
// THIS IS FOR THE NEXT PART OF ASSEMBLING
        LXI H,2050  ;  STORE THE ADDRESS 2050 INTO THE MEMORY
        MOV A,M ; MOV THE VALUE OF MEMORY TO ACCUMULATOR
        ANI F0  ; AND THE MEMORY CONTENT WITH 0F
        RRC ;RIGHT ROTATE ACCUMULATOR
        RRC ;RIGHT ROTATE ACCUMULATOR
        RRC ; RIGHT ROTATE ACCUMULATOR
        RRC  ;RIGHT ROTATE ACCUMULATOR
        MOV B,A ; MOVE THE CONTENT FROM ACCUMULATOR TO REG B
        INX H ; INCREMENT THE MEMORY POINTER TO POINT NXT LOC
        MOV A,M ; MOVE THE MEMORY CONTENT TO ACCUMULATOR
        ANI 0F ; AND THE ACCUMULATOR CONTENT WITH 0FH
        RLC; ROTATE LEFT THE ACCUMULATOR
        RLC ;ROTATE LEFT THE ACCUMULATOR
        RLC ; ROTATE LEFT THE ACCUMULATOR
        RLC ;ROTATE LEFT THE ACCUMULATOR
        ADD B  ; ADD THE ACCUMULATOR CONTENT WITH B REG VALUE
        STA 2053  ; STORE THE ACCUMULATOR CONTENT AT ADDRESS 2053
        HLT ; END THE PROGRAM
```

We are extracting the lowermost four bits and the uppermost four bits and then shifting the positions of them to get them at proper positions and then adding them to obtain the final numbers .

Example: To assemble MN and KL as NK we AND MN with 0F to abstract 0N and KL with F0 to obtain K0 then left rotate 0N four times to get N0 and right rotate K0 four times to get 0K and then add them to obtain NK .. Similarly the next part is also done .

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

```
        LXI H,2050  ; STORE THE ADDRESS INTO MEMORY
```

```
        MOV B,M ; MOV THE VALUE OF CONTENT IN MEMORY TO B REG
        MVI A,00 ; SET THE A REG VALUE TO ZERO
        MVI C,00 ; SET THE C REG VALUE TO ZERO
        INX H ; INCREMENT THE MEMORY POINTER TO POINT TO NXT LOC
LOOP:   ADD M ; ADD THE ACCUMULATOR CONTENT WITH MEMORY
        JNC SKIP ; IF NO CARRY GOTO SKIP
        INR C ; IF CARRY INCREMENT THE C REGISTER VALUE

 SKIP:  DCR B  ; DECREMENT THE VALUE OF B REG
        JNZ LOOP  ; IF NOT ZERO REPEAT THE LOOP
        STA 2052 ; STORE THE RESULT INTO 2052
        MOV A,C ; STORE THE CARRY PART INTO ACCUMULATOR
        STA 2053 ; TRANSFER ACCUMULATOR CONTENT TO ADDRESS 2053
        HLT; END THE PROGRAM
```

The multiplication has been performed using the concept of repeated addition . To d AxB we can add B a times or add A b times both will give the same result as AXB . Finally the result has been stored in the desired location .

3. *N* numbers are stored in consecutive m/m location starting from 2050_H. The value *N* is stored in $204F_H$.

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

```
// N NUMBERS STARTING FROM 2050 ARE THERE THE VALUE OF N
// STORED IN THE LOCATION 204F FIND THE MAXIMUM AMONG THEM

        LXI H,204F ; STORE THE ADDRESS 204F INTO MEMORY
        MOV B,M ; MOVE THE MEMORY CONTENT INTO B (N VALUE)
        DCR B ; DECREMENT THE LOOP COUNTER
        INX H ; INCREMENT THE MEMORY LOCATION POINTER
        MOV A,M  ; MOVE THE MEMORY CONTENT TO ACCUMULATOR
```

```
SKIP:   INX H ; INCREMENT THE MEMORY POINTER
        CMP M ; COMPARE WITH M
        JNC LOOP ;JUMP IF ACC CONTENT IS LESS OR EQUAL TO MEM CONTENT
        MOV A,M ; STORE MEMORY CONTENT TO ACCUMULATOR
LOOP:   DCR B ; DECREMENT THE LOOP COUNTER
        JNZ SKIP ; IF NOT ZERO REPEAT THE LOOP
        STA 3050 ; STORE THE MAX VALUE IN ADDRESS 3050
        HLT ; END THE PROGRAM
```

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

```
        LXI H,204F ; STORE THE ADDRESS INTO MEMORY
        MOV B,M ; MOVE THE MEMORY CONTENT TO B REGISTER
        INX H ; INCREMENT THE MEMORY POINTER
        MOV A,M ; STORE THE MEMORY POINTER TO ACCUMULATOR
        DCR B ;DECREMENT THE LOOP COUNTER

LOOP:   INX H ; INREMENT THE MEMORY POINTER
        CMP M ; COMPARE THE VALUES IN MEMORY AND ACCUMULATOR
        JC SKIP ; IF ACC CONTENT >= MEMORY CONTENT , SKIP
        MOV A,M ; ELSE STORE THE MEMORY CONTENT IN ACCUMULATOR

SKIP:   DCR B ; DECREMENT THE LOOP COUNTER
        JNZ LOOP ; IF NOT ZERO REPEAT THE LOOP
        STA 3050 ; STORE THE LOWEST VALUE IN ADDRESS 3050
        HLT; END THE PROGRAM
```

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

```
START:  LXI H,204F    ; STORE THE ADDRESS IN MEMORY
        MVI D,00      ; SET THE VALUE OF D REGISTER AS ZERO
        MOV C,M       ; MOV THE VALUE OF MEMORY CONTENT TO C REG (VALUE OF N)
        DCR C         ; DECREMENT C
        INX H         ; INCREMENT THE MEMORY POINTER

CHECK:  MOV A,M       ; MOVE THE CONTENT TO MEMORY TO ACCUMULATOR
        INX H         ; INCREMENT THE MEMORY POINTER
        CMP M         ; COMPARE THE VALUE OF ACCUMULATOR WITH MEMORY
        JC NEXTVAL    ; IF ACCUMULATOR CONTENT IS LESSER, JUMP TO NEXTVAL
        JZ NEXTVAL    ; IF ACCUMULATOR CONTENT IS EQUAL , JUMP TO NEXTVAL
      // SWAPPING THE CONTENTS BCZ OF VIOLATION OF ASCENDING ORDER
        MOV B,M   ; MOV MEMORY CONTENT TO B
        MOV M,A   ; MOV ACCUMULATOR CONTENT TO MEMORY
        DCX H     ; DECREMENT THE MEMORY POINTER
        MOV M,B   ; MOV THE B REG CONTENT TO MEMORY
        INX H     ; INCREMENT THE VALUE OF MEMORY
        MVI D,01  ; IF COMPARISON OCCURS SET THE D REG VALUE TO 1

NEXTVAL:DCR C   ; DECREMENT VALUE OF C
        JNZ CHECK   ; IF THE INNER LOOP NOT FINISHED , REPEAT
        MOV A,D   ; MOV THE D CONTENT TO ACCUMULATOR
        CPI 01   ; COMPARE IMMEDIATE VALUE OF ACCUMULATOR TO 01H
        JZ START   ; IF SWAPS HAVE OCCURRED NEXT ITERATION OF OUTER LOOP
        HLT   ; END THE PROGRAM
```

Algorithm :

Here we have two loops for sorting the numbers .The Bubble sort algorithm have been used in

program. Whenever a number is greater then its next number then swapping occurs .There is a variable which is used to indicate whether any swap occurred within the inner loop , the D register is doing that thing.

This swapping is done by the inner loop CHECK and the outer loop START continues the iterations until there are no more swaps which means that all the numbers have been sorted successfully.

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

```
// SORT IN DESCENDING ORDER

START: MVI B,00 ; STORE 0 IN B REGISTER
       LXI H,204F ; STORE THE ADDRESS 204F INTO MEMORY
       MOV C,M ; MOVE THE CONTENT OF MEMORY TO C REGISTER
       DCR C ; DECREMENT THE VALUE OF C REGISTER CONTENT
       INX H ; INCREMENT THE MEMORY LOCATION POINTER

LOOP : MOV A,M ;STORE THE MEMORY CONTENT IN THE ACCUMULATOR
       INX H ; INCREMENT THE MEMORY LOCATION POINTER
       CMP M ; COMPARE ACCUMULATOR CONTENT WITH MEMORY
       JNC NEXTLOOP ; IF ACC CONTENT >MEMORY CONTENT JUMP
// SWAPPING THE CONTENTS BCZ OF VIOLATION OF DESCENDING ORDER
       MOV B,M ; MOVE MEMORY CONTENT TO B REG
       MOV M,A ; MOV THE ACCUMULATOR CONTENT TO A REG
       DCX H ; DECREMENT THE MEMORY LOCATION POINTER
       MOV M,B ; MOV THE B REGISTER CONTENT TO MEMORY
       INX H ; INCREMENT THE MEMORY LOCATION POINTER
       MVI B,01 ; SET THE SWAP FLAG TO 1

NEXTLOOP: DCR C  ; DECREMENT THE C REGISTER VALUE (INNER LOOP COUNTER)
       JNZ LOOP ; IF NOT ZERO REPEAT THE LOOP
```

DCR B ; DECREMENT THE VALUE OF B REGISTER

JZ START ; IF B IS 0 , THIS MEANS SWAP OCCURRED , REPEAT OUTERLOOP

HLT ; END THE PROGRAM

Algorithm :

Here we have two loops for sorting the numbers .The Bubble sort algorithm have been used in program. Whenever a number is smaller then its next number then swapping occurs .There is a variable which is used to indicate whether any swap occurred within the inner loop , the B register is doing that thing.

This swapping is done by the inner loop CHECK and the outer loop START continues the iterations until there are no more swaps which means that all the numbers have been sorted successfully.

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

LXI H,204F ; STORE THE ADDRESS 204F INTO MEMORY

MOV C,M ; MOVE THE MEMORY CONTENT TO C REG

INX H ; INCREMENT THE MEMORY LOCATION POINTER

MVI B,00 ; SET THE B REG VALUE AS 0

LXI D,2200 ; STORE THE ADDRESS 2200 INTO DE REG PAIR

LOOP: MOV A,M ; MOV THE MEMORY CONTENT TO ACCUMULATOR

ANI 01 ; AND THE ACCUMULATOR CONTENT WITH 01H

JZ SKIP ; IF THE NUMBER IF EVEN SKIP

MOV A,M ; IF ODD MOVE THE MEMORY CONTENT TO ACCUMULATOR

STAX D ; STORE THE ACCUMULATOR CONTENT INTO MEMORY LOC POINTED BY DE REGISTER PAIR

INR B ;INCREMENT THE ODD COUNT

INX D ; INCREMENT THE VALUE OF DE POINTER

SKIP: INX H ; INCREMENT THE MEMORY LOCATION POINTER

DCR C ; DECREMENT THE VALUE OF C REGISTER(LOOP COUNTER)

```
        JNZ LOOP; IF NOT ZERO REPEAT THE LOOP
        MOV A,B ; MOVE THE VALUE OF B REGISTER TO ACCUMULATOR
        STA 2301 ; STORE THE COUNT OF ODD NUMBERS IN 2301 LOCATION

// FOR THE EVEN NUMBERS REPEAT THE SAME PROGRAM
        LXI H,204F ; STORE THE LOCATION 204F INTO MEMORY
        MOV C,M ; MOVE THE MEMORY CONTENT TO C REG
        INX H ; INCREMENT THE VALUE OF MEMORY LOCATION
        MVI B,00 ; SET THE VALUE OF B REGISTER AS 0
        LXI D,2100 ; STORE THE LOCATION 2100 INTO DE REGISTER PAIR

LOOP2:  MOV A,M  ; MOVE THE CONTENTS OF MEMORY TO ACCUMULATOR
        ANI 01 ; AND THE ACCUMULATOR CONTENT WITH 01H
        JNZ SKIP2 ;IF NUMBER IS ODD SKIP AND GOTO SKIP2
        MOV A,M ; MOVE THE MEMORY CONTENT TO ACCUMULATOR
        STAX D ; STORE THE VALUE OF ACCUMULATOR TO MEMORY LOCATION
POINTED BY DE REGISTER PAIR
        INR B;  INCREMENT THE COUNT OF EVEN
        INX D ; INCREMENT THE LOCATION POINTED BY DE REGISTER PAIR

SKIP2:  INX H ; INCREMENT THE MEMORY LOCATION
        DCR C ; DECREMENT THE LOOP COUNTER
        JNZ LOOP2 ; IF NOT ZERO REPEAT THE LOOP
        MOV A,B ; MOVE THE EVEN COUNT TO ACCUMULATOR
        STA 2300 ; STORE THE EVEN COUNT FROM ACCUMULATOR TO 2300
        HLT ; END THE PROGRAM
```

5. *N* numbers are stored in consecutive m/m location starting from 2050_H. The value *N* is stored in $204F_H$. Write a program to test whether a number stored in $204E_H$ is present in the list. If present, store its position in the list at $204D_H$; otherwise store FF_H.

```
        LXI H,204F ; STORE THE ADDRESS INTO MEMORY
        LDA 204E ; LOAD ACCUMULATOR WITH THE CONTENT OF LOCN 204E
        MOV C,M  ; THE VAL OF N IS STORED IN C REG NOW
        MVI B,00 ; SET THE B REGISTER VALUE AS 0

LOOP:   INX H  ; INCREMENT THE MEMORY LOCATION POINTER
        MOV E,M ; MOV THE CONTENT OF MEMORY LOCATION TO E REGISTER
        CMP E ; COMPARE THE CONTENT OF ACCUMULATOR TO E REG
        JNZ SKIP ; IF CONTENT NOT EQUAL SKIP
        MOV A,B ; STORE THE B REG CONTENT TO ACCUMULATOR
        STA 204D ; STORE THE VALUE OF ACCUMULATOR IN ADDRESS 204D
        HLT ; END THE PROGRAM

SKIP:   INR B ; INCREMENT THE B REGISTER VALUE TO GET NEXT INDEX
        LDA 204E ; LOAD NUMBER TO BE CHECKED FOR EQUALITY INTO ACCUMULATOR
        DCR C ;DECREMENT THE LOOP COUNTER
        JNZ LOOP ; IF NOT ZERO REPEAT THE LOOP
        MVI B,FF ; MOVE FF TO B REGISTER
        MOV A,B ; MOVE THE B REGISTER CONTENT TO ACCUMULATOR
        STA 204D ; STORE THE ACC CONTENT INTO 204D
        HLT ; END THE PROGRAM
```

Microprocessor (8085) Assignment 3

1. A set of *N* data bytes is stored in m/m locations starting from 2501_H. The value of *N* is stored in 2500_H. Write a program to store these data bytes from m/m location 2600_H if D_0 or D_7 is 1; otherwise reject the data byte.

```
        LXI H,2500 ; STORE THE ADDRESS INTO MEMORY
        MOV C,M ; MOVE THE MEMORY CONTENT TO C REG
        LXI D,2600 ;STORE ADDRESS 2600 INTO DE REG PAIR

LOOP:   INX H ; INCREMENT THE MEMORY LOCATION
        MOV A,M ; MOVE THE MEMORY CONTENT TO ACCUMULATOR
        ANI 81 ; AND THE ACCUMULATOR CONTENT WITH 81H
        JZ SKIP ; IF D0 OR D7 IS NOT ONE , SKIP
        MOV A,M ; MOVE THE MEMORY CONTENT TO ACCUMULATOR
        STAX D ; IF D0 OR D7 IS ZERO , STORE THE CONTENT OF ACCUMULATOR INTO MEMORY LOCATION POINTED BY DE REGISTER PAIR
        INX D ; INCREMENT THE DE REG PAIR CONTENT

SKIP:   DCR C ; DECREMENT THE LOOP COUNTER
        JNZ LOOP ; IF NOT ZERO REPEAT THE LOOP
        HLT ; END THE PROGRAM
```

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

```
//ADD NUMBERS HAVING THE LSB AND MSB AS 1
//IN ARRAY AND STORING RESULT IN 2500H AND CARRY IN 2501H
```

```
        LXI H,21FF ; STORE THE CONTENT OF 21FF INTO MEMORY LOCATION
        MOV B,M ; MOVE THE VALUE OF MEMORY TO B REGISTER
        MVI D,00 ; SET THE D REGISTER CONTENT TO 0
        MVI C,00 ; SET THE C REGISTER CONTENT TO 0
        INX H ; INCREMENT THE MEMORY LOCATION POINTER

LOOP:   MOV A,M ; MOVE THE MEMORY CONTENT TO ACCUMULATOR
        CMA ; COMPLEMENT THE ACCUMULATOR
        ANI 81 ; AND WITH 81H
        JNZ SKIP2 ; IF THE NUMBER HAS LSB AND MSB NOT BOTH ZERO
        MOV A,D ; MOVE THE D REG CONTENT TO A REG
        ADD M ; ADD THE ACCUMULATOR CONTENT
        JNC SKIP1 ; IF NO CARRY GOTO SKIP1
        INR C ; INCREMENT THE C REG VALUE TO STORE THE CARRY

SKIP1:  MOV D,A ; MOVE THE ACCUMULATOR CONTENT TO D REGISTER TO SAVE CURRENT
ADD RESULT

SKIP2:  INX H ; INCREMENT THE MEMORY LOCATION POINTER
        DCR B ; DECREMENT THE LOOP COUNTER
        JNZ LOOP ; IF NOT ZERO REPEAT THE LOOP
        MOV A,D ; MOVE THE RESULT TO ACCUMULATOR
        STA 2500 ; STORE THE RESULT FROM ACC TO 2500H
        MOV A,C ; MOVE THE CARRY VALUE IN THE ACCUMULATOR
        STA 2501 ; STORE THE ACCUMULATOR CONTENT INTO 2501
        HLT ; END THE PROGRAM
```

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

```
        LXI H,2060 ; STORE THE ADDRESS INTO MEMORY
        MOV C,M ; MOVE THE MEMORY LOCATION CONTENT TO C REG
        MVI B,00 ; SET THE VALUE OF  B REG AS 0
        MVI D,01 ;SET THE VALUE OF D REG AS 1
        DCR C ; DECREMENT THE C REG VALUE
        JZ SKIP ; IF NOT FIRST FIB THEN SKIP
        CALL FIB ; CALL THE SUBROUTINE FIB
        HLT ; END THE PROGRAM

SKIP:   MVI A,01 ; STORE THE FIRST FIB VALUE IN ACCUMULATOR
        STA 2050 ; STORE THE CONTENT OF ACCUMULATOR INTO 2050
        HLT ; END THE PROGRAM

FIB:    XRA A ; CLEAR THE ACCUMULATOR
        ADD D ; ADD THE ACCUMULATOR CONTENT WITH D REG VALUE
        ADD B ; ADD THE B REG CONTENT TO ACCUMULATOR
        MOV B,D ; MOVE THE D REG VALUE TO B REGISTER
        MOV D,A ; MOVE THE CONTENT OF ACCUMULATOR TO D
        DCR C ; DECREMENT THE LOOP COUNTER
        JNZ FIB ; IF NOT ZERO CALL FIB AGAIN
        STA 2050 ; STORE THE FINAL RESULT INTO 2050H ADDRESS
        RET; RETURN FROM THE FUNCTION ONCE THE NTH FIB HAS BEEN SAVED INTO DEST
```

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 $201E_H$ and $201F_H$, respectively.

```
        LDA 201E ; LOAD ACCUMULATOR WITH BLOCK SIZE N
        LXI H,201F ; LOAD MEMORY WITH ADDRESS 201FH
```

```
        MOV C,M ; MOVE THE CONTENTS OF MEMORY TO C REG(OVERLAP SIZE)
        SUB C ; A=A-M , FINDING THE DIFF BETWEEN N AND M
// NOW IN THE ACC WE HAVE THE INCREMENT SIZE
        MOV C,A ; STORING THE DIFF INTO C REG
        MVI B,00 ; SET B REG VALUE AS ZERO
        LXI H,3000 ; STORE ADDRESSS 3000 INTO MEMORY
        DAD B ; HL= HL+BC
        LXI D,3000 ; STORE THE ADDRESS 3000 INTO THE DE REG PAIR
        XCHG ; EXCHANGE THE CONTENTS OF DE AND HL
// TILL HERE HL WILL HAVE THE ADDRESS 3000 AND DE WILL HAVE THE ADDRESS 3000+(N-M)
        LDA 201E ; LOAD THE ACCUMULATOR WITH THE BLOCK SIZE N
        MOV C,A ; MOVE THE BLOCK SIZE TO C REG
        DAD B ;HL=HL+BC
//TILL HERE HL WILL POINT TO THE LOCATION 3000+N,LOCATION AFTER THE LAST VALUE OF THE BLOCK
        DCX H ; DECREMENT THE MEMORY POINTER
        XCHG ; EXCHANGE THE CONTENTS OF HL AND DE
        DAD B ; HL=HL+BC
        DCX H ; DECREMENT THE VALUE OF MEMORY LOCATION POINTER
        XCHG;EXCHANGE THE VALUE OF HL AND DE AGAIN

LOOP:   MOV A,M ; STORE THE MEMORY CONTENT INTO ACCUMULATOR
        STAX D ; STORE THE ACCUMULATOR CONTENT TO LOCATION POINTED BY DE REGISTER PAIR
        DCX D ; DECREMENT THE LOCATION POINTER OF DE
        DCX H ; DECREMENT THE MEMORY LOCATION POINTER HL
        DCR C ; DECREMENT THE LOOP COUNTER
        JNZ LOOP ; IF NOT ZERO REPEAT THE LOOP
        HLT ; END  THE PROGRAM
```

ALGORITHM:

Firstly we find the difference between the block size and the overlap size . Then we find the location where the last location of the current block will be placed by doing the arithmetic : base address+(2*n-m)-1 .

Then we point to the location where the last location is currently placed by the arithmetic:

Base address+n-1 .Then we keep on storing the elements in these memory locations by register operations n

Times and finally we get the desired outcome .