Class Work

1. 10 data bytes are stored in memory location 2051H onwards. Write an 8085 assembly language program to count number of 1's in each byte and store this count in corresponding memory locations D001H onwards.

```
        LXI H, 2051    ; load HL Pair with 2051
        LXI D, D001    ; load DE Pair with D001
        MVI A, 0A      ; Initialize Accumulator to 0A
LOOP:   STA 4050       ; store A to memory 4050
        MOV A, M       ; copy memory content to A
        INX H          ; Increment HL Pair
        MVI C, 08      ; Initialize c to 08
        MVI B, 00      ; Initialize B to 00
        CALL START     ; call subroutine start
        LDA 4050       ; load memory 4050 to A
        DCR A          ; Decrement Accumulator
        JNZ LOOP       ; Jump to LOOP if z=0
        HLT

START:  RRC            ; Rotate Accumulator right
        JNC END        ; Jump to END if CY=0
        INR B          ; Increment Reg B
END:    DCR C          ; Decrement Reg C
        JNZ START      ; Jump to START if z≠0
        MOV A, B       ; copy Reg B to Reg A
        STAX D         ; store Accumulator to memory(DE)
        INX D          ; Increment Reg Pair DE
        RET            ; Return

# ORG  2051H
# DB   05H, 033H, 04H, 08H, 09H, 05H, 033H, 04H, 08H, 09H
```

2. Write an 8085 ALP to count positive and negative data bytes from 10 bytes stored on location 3001 onwards. Store count of Positive data bytes in location 4001H and count of negative data bytes on location 5001H.

```
        LXI H, 3001     ; load HL Pair with 3001
        MVI D, 00       ; Initialize Reg D to 00
        MVI E, 00       ; Initialize Reg E to 00
        MVI B, 0A       ; Initialize Reg B to 0A

START:  MOV A, M        ; Copy memory content to Reg A
        INX H           ; Increment HL Pair
        ADI 00          ; Add 00 to Accumulator
        JM MINUS        ; jump to MINUS if S=1
        JP PLUS         ; jump to PLUS if S=0

MINUS:  INR D           ; Increment Reg D
        JMP END         ; Jump to END

PLUS:   INR E           ; Increment Reg E

END:    DCR B           ; Decrement Reg B
        JNZ START       ; Jump to START if Z=0
        MOV A, E        ; Copy Reg E to Reg A
        STA 4001        ; store Accumulator to 4001
        MOV A, D        ; Copy Reg D to Reg A
        STA 5001        ; store Accumulator to 5001
        HLT             ; stop

#  ORG  3001
#  DB   05H, 83H, 04H, 88H, 09H, 85H, 33H, 84H, 08H, 09H
```
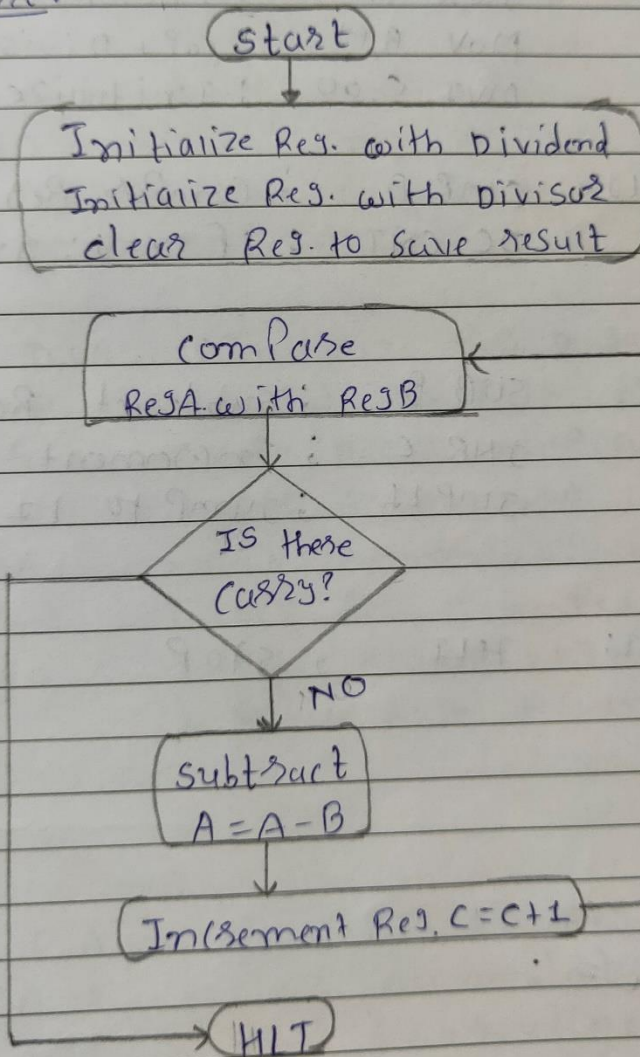
DATE

3. write assembly language Program to do division of two numbers. specify the memory location of each and every instruction and also draw Flowchart.

Flowchart:

```
                    ( start )
                        |
                        v
        +-------------------------------+
        | Initialize Reg. with Dividend |
        | Initialize Reg. with Divisor  |
        |   clear Reg. to save result   |
        +-------------------------------+
                        |
                        v
           +------------------------+
           |       comPare          |<--------+
           |    RegA with RegB      |         |
           +------------------------+         |
                        |                     |
                        v                     |
                    /---------\               |
                   / IS there  \              |
                  <   Carry?    >             |
                   \           /              |
                    \---------/               |
                        | NO                  |
                        v                     |
                  +-----------+               |
                  | subtract  |               |
                  |  A = A-B  |               |
                  +-----------+               |
                        |                     |
                        v                     |
           ( Increment Reg. C = C+1 )---------+
                        |
                        v
                    ( HLT )
```

Address			
0000		LXI H, 2000	; load HL Pair with 2000
0001			
0002			
0003		Mov A,M	; copy Dividend to Reg A.
0004		INX H	; Increment HL Pair
0005		Mov B,M	; copy Divisor to Reg B
0006		MVI C,00	; Initialize Reg C to 00
0007			
0008	L1:	CMP B	; Compare Reg B with Reg A
0009		JC CNT	; if CY=1; Jump to CNT
000A			
000B			
000C		SUB B	; subtract Reg B from Reg A
000D		INR C	; Increment Reg C
000E		JMP L1	; Jump to L1
000F			
0010			
0011	CNT:	HLT	; stop

A BCD numbes between 0 and 99 is stored in a memory location named IN BUF. write a main Program and subroutine to convert the BCD numbes into its equivalent binary numbes. stose the result in a memory location called OUT BUF.

```
    MVI  D,0A     ; Initialize Reg D to 0A
    IN  08        ; Input the data from Port 08
    MOV B,A       ; COPY Reg B to Accumulator
    ANI  OF       ; AND Accumulator content with OF
    MOV C,A       ; COPY Accumulator to Reg C
    MOV A,B       ; COPY Reg B to Reg A
    ANI FO        ; AND Accumulator with FO
    RRC           ; Rotate Right Accumulator
    RRC
    RRC
    RRC
    MOV E,A  ; COPY Reg A to Reg E
    MVI A,00 ; clear Accumulator
    CALL MUL ; Call subroutine MUL
    ADD  C   ; Add content of Reg C to Reg A
    OUT  09  ; output the data to Port 09
    HLT      ; stop
MUL: ADD E   ; Add Reg E to Accumulator
    DCR D    ; Decrement Reg D
    JNZ MUL  ; Jump to MUL if Z=0
    RET      ; Return
```

5. A Railway crossing signal has two flushing lights run by a microcomputer. One light is connected to data bit D7 and the second light is connected to data bit D6. Write a program to turn each signal light alternately ON and OFF at interval of 2 second

$$\text{Let's Assume, } f = 3 MHz$$

$$T = \frac{1}{f} = 0.33 \mu S$$

```
        LXI H, 2050
        MVI D, AA

START:  MOV A, D
        RLC
        MOV D, A
        LXI B, 1776
        LXI B, D548
DELAY:  DCX B      6T
        XTHL       16T
        XTHL
        XTHL
        XTHL
        XTHL
        MVI A, FF  7T
        ANA B      4T
        ORA C      4T
        JNZ DELAY  10T
        JMP START
```

$$2 = T_0 \times T$$

$$T_L = T \times \text{loop } T\text{-stat} \times COUNT$$

$$2 S = 0.33 \times 10^{-6} \times 111 \times COUNT$$

$$COUNT = \frac{2}{0.33 \times 10^{-6} \times 111}$$

$$COUNT \approx 54600$$

$$COUNT = D548H$$

$$\text{Total Delay} = T_0 + T_L$$
$$= 32 \times 0.33 + 2S$$
$$= 10.56 \mu S + 2S$$
$$\approx 2 \text{ second.}$$