Name: **Muhammad Maiz Nadeem**
Reg. ID: **SP21-BCS-052**

Question 1:

Find out all valid / invalid combinations of MOV instruction using data register and immediate values or both operands as data register.

Answer:

```
1    ORG 100h
2
3    .DATA
4
5        X DW 11
6        Y DW 0x1AB2
7
8    .CODE
9
10       ; VALID:
11       MOV AX, BX          ; Register to Register
12       MOV AX, X           ; Reference to Register
13       MOV AX, 9           ; Immediate to Register
14       MOV Y, BX           ; Register to Reference
15       MOV Y, 5            ; Immediate to Reference
16
17       MOV 38h, BX         ; Register to memory address
18       MOV 5, 6            ; Immediate to Immediate
19
20       ; NOT VALID
21       MOV X, Y            ; Reference to Reference
22
```

Question 2:

Now execute the following instructions (single-step) in EMU8086 and observe the changes in contents of destination operand and explain those changes. If any instruction gives error, correct that error.

- **MOV AL, 256**
- **MOV AX, F1ABh**
- **MOV AX, -123**
- **MOV BX, 123**
- **MOV AH, 010010001b**
- **MOV 1234h, BX**
- **MOV DX, 33h**
- **MOV CX, 'AB'**
- **MOV CH, AL**
- **MOV DL, BL**
- **MOV AH, BL**
- **MOV AX, CL**

Answer:

```
ORG 100h

.DATA

.CODE

    MOV AL, 255             ; 256 exceeds the 0-255 lower byte limit.
    MOV AX, 0xF1ABh         ; Not a hex value until preceded by 0x because F is in front.
    MOV AX, -123            ; Negative decimal in 16-bit register.
    MOV BX, 123             ; Positive decimal in 16-bit register.
    MOV AH, 010010001b      ; Binary value in 8-bit register.
    MOV 1234h, BX           ; 16-bit register in a hex memory location.
    MOV DX, 33h             ; Hex value in a register.
    MOV CX, 'AB'            ; 16-bit ASCII value of AB in register.
    MOV CH, AL              ; Lower byte of Accumulator to Higher Byte of Count register.
    MOV DL, BL              ; 8-bit to 9-bit.
    MOV AH, BL              ; Higher to Lower (8-bit to 8-bit).
    MOV AX, CX              ; Move operation on 2 different registers.

RET
```

Question 3:

In this question we will learn a little about the format in which data is stored in memory(or registers).

Our data will be composed of Bytes. Let's assume that our data is composed of following 2 Bytes:

Data = 0x7A5B (0X prefix means that the following number is interpreted as a hexadecimal number)

If data is stored such that Most significant Byte is placed in the higher address place and Least Significant Byte is stored in the lower address place then we say that data is stored in Little Endian format e.g; if we run MOV [200H],0x7A5B then data in Little Endian format is stored as follow

Address	Data
200H	5B
201H	7A

Similarly , If data is stored such that Most significant Byte is placed in the lower address and Least Significant Byte is stored in the higher address place then we say that data is stored in Big Endian format e.g; if we run MOV [200H],0x7A5B then data in Big Endian format is stored as follow

Address	Data
200H	7A
201H	5B

Now based on the commands that you ran in question 2, which storage format does your emulator machine (EMU8086) follows?

Emu8086 follows little endial storage format.