

## Lab-2

**Objective 1: Find the sum and average of N numbers (integer numbers).**

**Program:**

**; Find sum and average of N 16-bit number**

```
        MOV SI,2000H
        MOV CL,[SI] ;DECLARE N IN THE MEMORY LOCATION 2000
        MOV CH,00H
        MOV BX,CX
        MOV AX,0000H
L2:     INC SI      ;16-bit data to be given from 2002 Memory location onwards
        INC SI
        ADD AX,[SI]
        JNC L1
        INC CH
L1:     DEC CL
        JNZ L2
        INC SI
        INC SI
        MOV [SI],AX
        INC SI
        INC SI
        MOV [SI],CH
        DIV BX
        INC SI
        INC SI
        MOV [SI],AX
        INC SI
        INC SI
        MOV [SI],DX
        HLT
```

**Objective 2: Count no. of 0s in an 8-bit number.**

**Program:**

**; counting no of 0s using indirect addressing mode**

```
        MOV BX,2000H ; SI CAN BE USED INSTEAD OF BX
```

```

        MOV AL,[BX]      ; DATA TO BE ENTERED IN 2000 MEMORY LOCATION
        MOV CL,08H
        MOV CH,00H
L2:     SHR AL,1H
        JC L1
        INC CH
L1:     DEC CL
        JNZ L2
        INC BX
        MOV [BX],CH      ; RESULT TO BE CHECKED IN 2001 MEMORY LOCATION
        HLT

```

**Objective 3: Move a block of 16-bit data from one location to another.**

**Program:**

; Move a block of 16-bit data from one location to another (Intra Segment Transfer).

```
MOV AX,2000H
```

```

        MOV SI, 3000H
        MOV DI, 5000H
        MOV CL,05H
        MOV DS,AX
L1: MOV BX,[SI]
        MOV [DI],BX
        INC SI
        INC SI
        INC DI
        INC DI
        DEC CL
        JNZ L1
        HLT

```

OR

**Program:**

; Move a block of 16-bit data from one location to another (Inter Segment Transfer).

```

        MOV SI, 3000H
        MOV DI, 5000H
        MOV CL,05H
L1:     MOV AX,2000H

```

```
MOV DS,AX
MOV BX,[SI]
INC SI
INC SI
MOV AX,4000H
MOV DS,AX
MOV [DI],BX
INC DI
INC DI
DEC CL
JNZ L1
HLT
```

**Obj-4 Multiplication of two 16-bit numbers without using MUL instruction using direct addressing mode.**

**Program:**

```
MOV BX,[1000H] ; BX = MULTIPLICAND
MOV CX,[1002H] ; CX = MULTIPLIER
MOV DX,0000H   ; DX= STORES HIGHER 16-BIT RESULTS
MOV AX, 0000H
L2: ADD AX,BX
    JNC L1
    INC DX
L1: DEC CX
    JNZ L2
    MOV [1004H], AX
    MOV [1006H],DX
    HLT
```