LAB-3 (COA)

OBJECTIVE-1:-

Find the largest / smallest number (8-bit number) from a given array of size N.

For largest Number:-

Pre-lab:-

- Assembly code:-

```
.data
Array_length db 04h
value db 09h, 10h, 05h, 03h
result db 00h
.code
MAIN PROC
    mov ax, data
    mov ds, ax
    mov cl, Array_length
    dec cl
    LEA si, value
    mov al, [si]

up: inc si
    cmp al, [si]
    jnl down
    mov al, [si]

down : dec cl
     jnz up
     LEA DI, result
     mov [DI], al

  END MAIN
```

- Input /output of Assembly code:-

Input:-

Sl. No.	Memory location	operand (data)
1	0710 : 0000	04
2	0710 : 0001	09
3	0710 : 0002	10
4	0710 : 0003	05
5	0710 : 0004	03

Output:-

Sl No.	Memory location	Operand (data)
1	0710 : 0005	10

For smallest Number :-

Pre-lab:-

- Assembly code :-

```
.data
Array-length db 04h
value db 09h,10h,05h,03h
result db 00h
.code
MAIN PROC
    mov ax, data
    mov ds, ax
    mov cl, Array-length
    dec cl
    LEA si, value
    mov al, [si]

up: inc si
    cmp al, [si]
    jc down
    mov al, [si]

down: dec cl
    jnz up
    LEA DI, result
    mov [DI], al

  END MAIN
```

- Input/output for Assembly code:-

Input:-

Sl. No.	Memory Location	Operand (data)
1	0710 : 0000	04
2	0710 : 0001	09
3	0710 : 0002	10
4	0710 : 0003	05
5	0710 : 0004	03

Output :-

Sl. No.	Memory Location	Operand (data)
1	0710 : 0005	03

OBJECTIVE 2:-

Arrange the elements (8-bit number) of a given array of size N in ascending / descending order.

For Ascending order:-

Pre-lab:-

- Assembly code:-

```
.DATA
count DB 06
value DB 09H, 0FH, 14H, 45H, 24H, 3FH

.CODE
MAIN PROC
    MOV AX, DATA
    MOV DS, AX
    MOV CH, count
    dec CH
UP2 : MOV CL, CH
    LEA SI, value
UP1: MOV AL, [SI]
    CMP AL, [SI+1]
    JC DOWN
    MOV DL, [SI+1]
    XCHG [SI], DL
    MOV [SI+1], DL
DOWN : INC SI
    DEC CL
    JNZ UP1
    DEC CH
    JNZ UP2
```

END MAIN

- Input/output for Assembly code:-

Input:-

Sl No.	Memory Location	Operand (data)
1	0710 : 0000	06
2	0710 : 0001	09
3	0710 : 0002	0F
4	0710 : 0003	14
5	0710 : 0004	45
6	0710 : 0005	24
7	0710 : 0006	3F

Output:-

Sl No.	Memory Location	Operand (data)
1	0710 : 0000	06
2	0710 : 0001	09
3	0710 : 0002	0F
4	0710 : 0003	14
5	0710 : 0004	24
6	0710 : 0005	3F
7	0710 : 0006	45

For Descending Order :-

Pre-Lab:-

- Assembly code :-

```
.DATA
COUNT DB 06
VALUE DB 09H, 0FH, 14H, 45H, 24H, 3FH

.CODE
MAIN PROC
   MOV AX, DATA
   MOV DS, AX
   MOV CH, COUNT
   DEC CH

   UP2:
   MOV CL, CH
   LEA SI, VALUE

   UP1:
   MOV AL, [SI]
   CMP AL, [SI+1]
   JNC DOWN
   MOV DL, [SI+1]
   XCHG [SI], DL
   MOV [SI+1], DL

   DOWN:
   INC SI
   DEC CL
   JNZ UP1
   DEC CH
   JNZ UP2
END MAIN
```

- Input/output for Assembly code:-

Input:-

Sl No.	Memory Location	Operand (data)
1	0710:0000	06
2	0710:0001	09
3	0710:0002	0F
4	0710:0003	14
5	0710:0004	45
6	0710:0005	24
7	0710:0006	3F

Sl. No.	Memory Location	Operand (data)
1	0710:0000	06
2	0710:0001	45
3	0710:0002	3F
4	0710:0003	24
5	0710:0004	14
6	0710:0005	0F
7	0710:0006	09

Conclusion:-

The lab exercise on 8086 assembly language programming focused on array manipulation, particularly finding the smallest number in an array, largest number in an array, sorting the dements of array in ascending order and sorting the elements of array in descending order. Through practical implementation, we depend our understanding of low-level programming concepts and learned essential skills in algorithmic thinking and precise coding practices for microprocessor-based systems.

Post LAB

Q1 What are the directions available for data declaration in 8086 microprocessor?

Ans⇒ Directives available for data declaration in 8086 microprocessor are:-

- DB (Define Byte) - Defines one or more bytes of data.
- DW (Define Word) - Defines one ormore words (16-bit data).
- DD (Define Doubleword) - Defines one or more doublewords. (32-bit data).
- DQ (Define Quadword) - Defines one ormore quadwords (64-bit data).
- DT - Defines a variable that is 10 bytes.
- CRLF = Defines a carriage return, or terminator byte.

Q2 State the difference between END, ENDP, and ENDS directions.

Ans⇒

END: Marks the end of the entire program.

ENDP: Marks the end of a procedure (similar to a function)

ENDS: Marks the end of a segment (a logical division of the Program).

Q3 Find the sum and average of a given array of size N.

Code:-

```
.DATA
    ARRAY_LENGTH DB 04H
    ARRAY DB 09H, 10H, 05H, 03H
    SUM DB 00H
    AVERAGE DB 00H

.CODE
MAIN
    MOV AX, DATA
    MOV DS, AX
    MOV CX, ARRAY_LENGTH
    LEA SI, ARRAY
    MOV AL, [SI]

    ; CALCULATE THE SUM
    XOR BX, BX ; CLEAR BX
    SUM_LOOP: ADD BL, AL
              INC SI
              MOV AL, [SI]
              DEC CX
              JNZ SUM_LOOP
    ; CALCULATE THE AVERAGE
    MOV AL, BL
    MOV BL, ARRAY_LENGTH
    DIV BL
    MOV AVERAGE, AL

    MAIN ENDP
END MAIN
```