Terna Engineering College
Computer Engineering Department

Program: Sem V

Course: Microprocessor Lab

Faculty: ARATHI BOYANAPALLI

LAB Manual

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Experiment No. 4

A.1 Aim:
Write assembly language programs to sort numbers in ascending / descending order. [Use BIOS/DOS interrupts to read input and display results.]

A.2 Prerequisite:
Basic knowledge of sorting the numbers and interrupts of 8086

A.3 Outcome:
After successful completion of this experiment students will be able to

1. Use appropriate instructions to program microprocessors to perform various tasks.
2. Develop the program in assembly/ mixed language for Intel 8086 processor
3. Demonstrate the execution and debugging of assembly/ mixed language program

A.4 Theory:

A Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of elements in the respective memory locations. In the given diagram the numbers are sorted in ascending order from an array of n numbers, where size n is stored in memory location 2000:500 and numbers are sorted from memory location 2000:501

Input Data ⇨	04	F9	F2	39	05
Memory Address(offset) ⇨	500	501	502	503	504

Output Data ⇨	05	39	F2	F9
Memory Address(offset) ⇨	501	502	503	504

A.5 Algorithm:

1. Defines the memory model
2. Initialize the data segment with variables. First variables will be the one which allow the user to give a list of numbers and second will be an array with size and it holds numbers . Third variable will be holding Length of the Array.
3. Initialize the code segment
4. Apply DOS interrupt functions to allow the user to give the inputs.
5. Set the counter
6. Travel from starting memory location and compare the first two two numbers if first number is greater than second number then swap them(for ascending order) and vice versa for descending order
7. Decrease the counter and increment SI
8. Continue until the counter become zero

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no ERP access available)

Roll No. : 50	Name: Amey Thakur
Class: TE-Comps B	Batch: B3
Date of Experiment: 13/08/2020	Date of Submission: 13/08/2020
Grade:	

B.1 Observations and learning:

(Software Code written by a student and output of the program)

- **Input to sort number in Ascending order -**

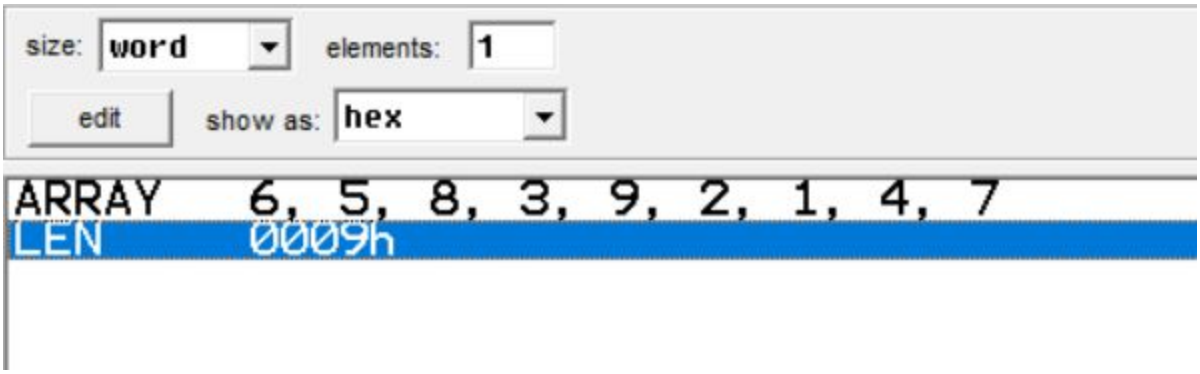
```
DATA SEGMENT
    ARRAY DB 6, 5, 8, 3, 9, 2, 1, 4, 7
    LEN DW $-ARRAY
DATA ENDS

CODE SEGMENT
ASSUME DS:DATA CS:CODE
    FILLY:
    MOV AX,DATA
    MOV DS,AX
    MOV CX,LEN-1
    SAAKSHI:
    LEA SI,ARRAY
    MOV BX,0
    ARCHIT:
    INC BX
    MOV AL,ARRAY[SI]
    INC SI
    CMP AL,ARRAY[SI]
    JB MEGA
    XCHG AL,ARRAY[SI]
    MOV ARRAY[SI-1],AL
    MEGA:
    CMP BX,CX
```

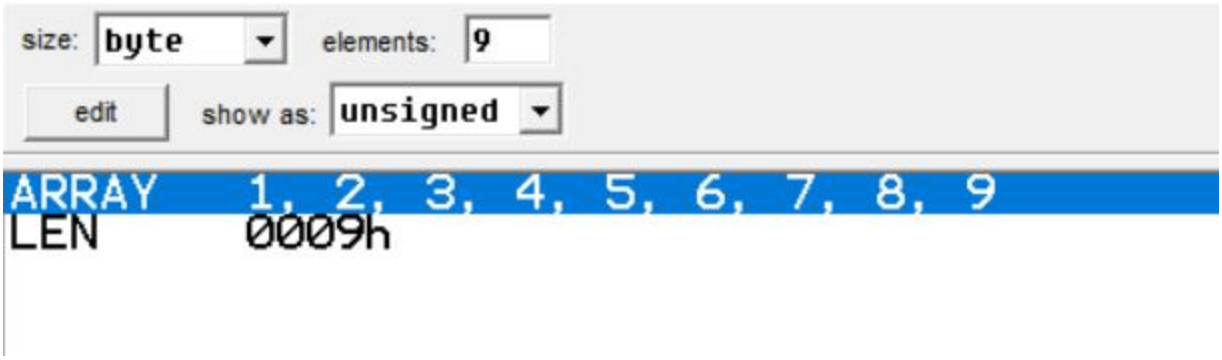
```
        JL  ARCHIT
        LOOP SAAKSHI
        MOV AH,4CH
        INT 21H
        CODE ENDS
END FILLY
```

- **Output (Ascending order) -**

    A. Before Execution



    B. After Execution



- **Input to sort number in Descending order -**

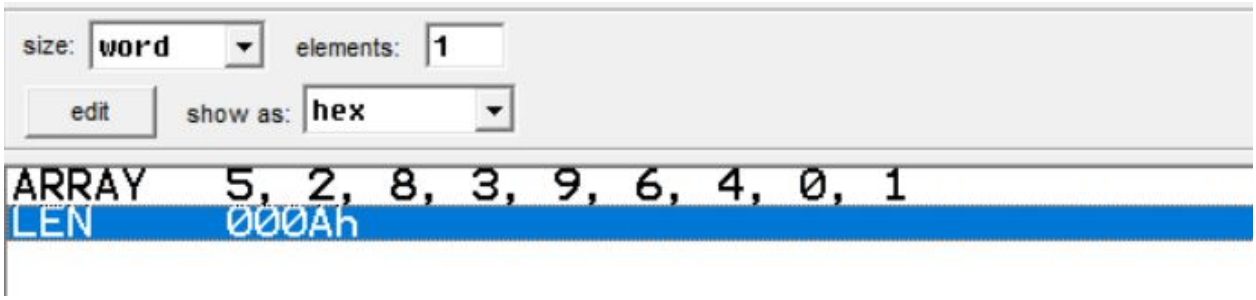
```
DATA SEGMENT
        ARRAY DB 5,2,8,3,9,6,4,0,1,7
        LEN DW $-ARRAY
DATA ENDS

CODE SEGMENT
ASSUME DS:DATA CS:CODE
        FILLY:
        MOV AX,DATA
```

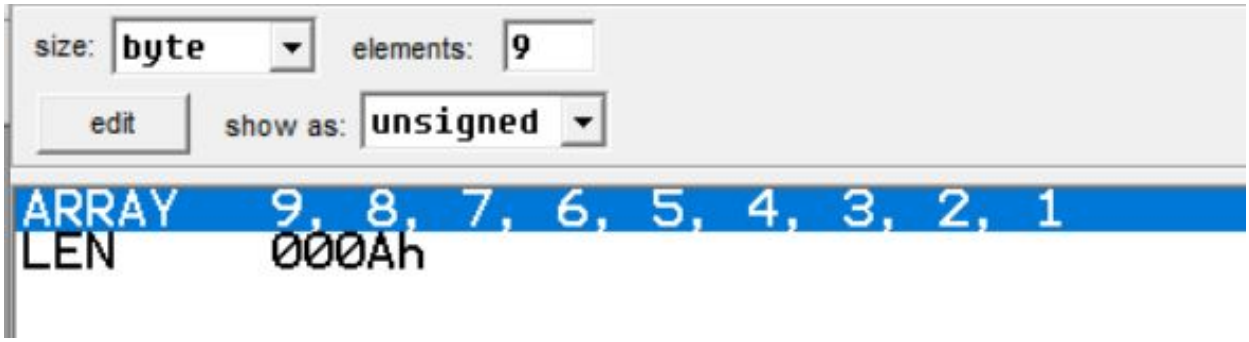
```
        MOV DS,AX
        MOV CX,LEN-1
        SAAKSHI:
        LEA SI,ARRAY
        MOV BX,0
        ARCHIT:
        INC BX
        MOV AL,ARRAY[SI]
        INC SI
        CMP ARRAY[SI],AL
        JB MEGA
        XCHG AL,ARRAY[SI]
        MOV ARRAY[SI-1],AL
        MEGA:
        CMP BX,CX
        JL  ARCHIT
        LOOP SAAKSHI
        MOV AH,4CH
        INT 21H
        CODE ENDS
END FILLY
```

- **Output (Descending order) -**

    A. Before Execution



    B. After Execution



B.2 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.1)

We successfully learned assembly language programs to sort numbers in ascending/descending order.

B.5 Question of Curiosity

Q1. List out and explain the instructions used in above program

Ans:

1. **Move Instruction (MOV) -** Inside the processor, the data resides in the registers. MOV is the basic instruction that moves the constant data in the register or moves that data from one register to another.
2. **The DEC Instruction -** The DEC instruction is used for decrementing an operand by one. It works on a single operand that can be either in a register or in memory.
3. **The ADD and SUB Instructions -** The ADD and SUB instructions are used for performing simple addition/subtraction of binary data in byte, word and doubleword size, i.e., for adding or subtracting 8-bit, 16-bit or 32-bit operands, respectively.
4. **CMP Instruction -** The CMP instruction compares two operands. It is generally used in conditional execution. This instruction basically subtracts one operand from the other for comparing whether the operands are equal or not.

Q2. Write an assembly language program to do bubble sorting.

Ans:

- **Input to do bubble sorting -**

```
.MODEL SMALL

DATA SEGMENT
    ARRAY DB 6, 5, 8, 3, 9, 2, 1, 4, 7
    LEN1 EQU $-ARRAY

CODE SEGMENT
    MOV AX, @DATA
    MOV DS, AX
    MOV CH, LEN1-1

    FILLY:
    MOV CL, CH
    LEA SI, ARRAY
```

```
ARCHIT:
MOV AL, [SI]
INC SI
CMP AL, [SI]
JBE MEGA
XCHG AL, [SI]
MOV [SI-1], AL


MEGA:
DEC CL
JNZ ARCHIT
DEC CH
JNZ FILLY
MOV AH, 4CH
INT 21H

END
```

- **Output (Bubble Sort) -**

  A. Before Execution

  B. After Execution