Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (fall, Year:2024), B.Sc. in CSE (Day)

Lab Report #06

Course title: Microprocessor & Microcontroller Lab
Course Code: CSE 304 **Section:** 222_D13

Lab Experiment Name: Implement Procedure in Assembly Language Programming

Student Details

Name	ID
Md. Moshiur Rahman	221902324

Submission Date : 08/12/2024
Course Teacher's Name : Md. Jahid Tanvir

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …………………………… **Signature:.........................**

Comments:.............................. **Date:.............................**

1. TITLE OF THE LAB REPORT EXPERIMENT

Implement Procedure in Assembly Language Programming.

2. OBJECTIVES

To understand 8086 instructions related to Procedure using Assembly Language Program.

• Write an Assembly Language code that takes any 5 of decimal digits (0 9) as input and calculates the average, largest and smallest of them in three different procedures and show the output like the following:
Input:
Enter the elements of array: 2 4 1 3 5
Output:
AVERAGE = 3
LARGEST = 5
SMALLEST = 1

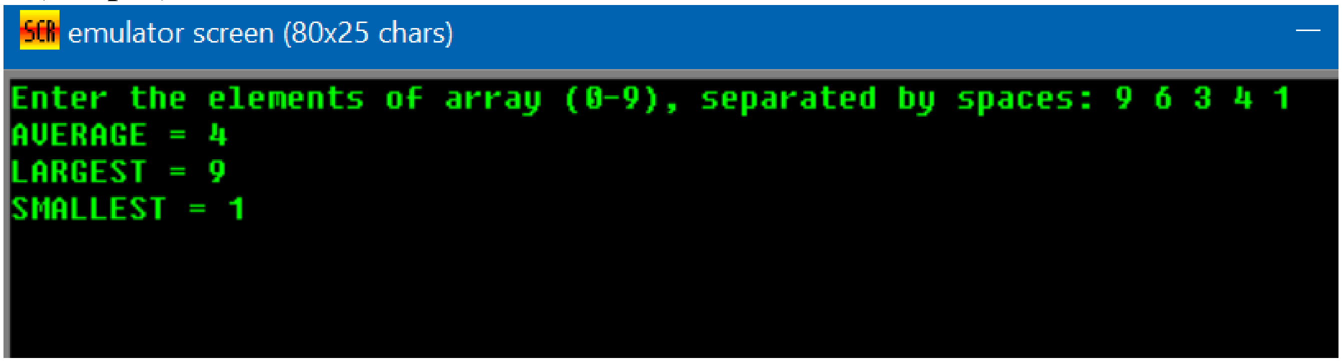
3. IMPLEMENTATION

Src code:

```
.MODEL SMALL
.STACK 100H
.DATA
    array DB 5 DUP(?)
    msgInput DB 'Enter the elements of array (0-9), separated by spaces: $'
    msgAvg DB 13, 10, 'AVERAGE = $'
    msgLargest DB 13, 10, 'LARGEST = $'
    msgSmallest DB 13, 10, 'SMALLEST = $'
    avg DB ?
    largest DB ?
    smallest DB ?

.CODE
MAIN PROC
    ; Initialize data segment
    MOV AX, @DATA
    MOV DS, AX

    ; Display input prompt
    LEA DX, msgInput
    MOV AH, 09H
    INT 21H

    ; Read 5 decimal numbers
    MOV CX, 5               ; Expect 5 numbers
    LEA DI, array           ; Point to the array
READ_LOOP:
    MOV AH, 01H             ; Read character
    INT 21H
    CMP AL, ' '             ; Skip spaces
    JE READ_LOOP
    SUB AL, '0'             ; Convert ASCII to integer
    MOV [DI], AL            ; Store number in array
    INC DI          ; Move to the next position
    LOOP READ_LOOP

    ; Call procedures
    CALL CalculateAverage
    CALL FindLargest
    CALL FindSmallest

    ; Display Average
    LEA DX, msgAvg
    MOV AH, 09H
    INT 21H
    MOV AL, avg
    ADD AL, '0'         ; Convert to ASCII
    MOV DL, AL
    MOV AH, 02H
    INT 21H

    ; Display Largest
    LEA DX, msgLargest
    MOV AH, 09H
    INT 21H
    MOV AL, largest
    ADD AL, '0'         ; Convert to ASCII
    MOV DL, AL
    MOV AH, 02H
    INT 21H

    ; Display Smallest
    LEA DX, msgSmallest
    MOV AH, 09H
    INT 21H
    MOV AL, smallest
    ADD AL, '0'         ; Convert to ASCII
    MOV DL, AL
    MOV AH, 02H
    INT 21H

    ; Exit program
    MOV AH, 4CH
    INT 21H
MAIN ENDP

; Procedure to calculate average
CalculateAverage PROC
    XOR AX, AX          ; Clear AX for sum
    MOV CX, 5
    LEA DI, array
SUM_LOOP:
    ADD AL, [DI]        ; Add each element to AL
    INC DI       ; Move to next element
    LOOP SUM_LOOP
    MOV BL, 5
    DIV BL              ; Divide sum by 5
    MOV avg, AL         ; Store result in avg
    RET
CalculateAverage ENDP

; Procedure to find largest number
FindLargest PROC
    MOV AL, [array]     ; Initialize largest with the first element
    MOV CX, 4
    LEA DI, array + 1
```

```
FIND_LARGEST_LOOP:
    CMP AL, [DI]
    JGE SKIP_LARGEST            ; Skip if AL is larger or equal
    MOV AL, [DI]        ; Update largest
SKIP_LARGEST:
    INC DI
    LOOP FIND_LARGEST_LOOP
    MOV largest, AL     ; Store result in largest
    RET
FindLargest ENDP

; Procedure to find smallest number
FindSmallest PROC
    MOV AL, [array]     ; Initialize smallest with the first element
    MOV CX, 4
    LEA DI, array + 1
FIND_SMALLEST_LOOP:
    CMP AL, [DI]
    JLE SKIP_SMALLEST           ; Skip if AL is smaller or equal
    MOV AL, [DI]        ; Update smallest
SKIP_SMALLEST:
    INC DI
    LOOP FIND_SMALLEST_LOOP
    MOV smallest, AL    ; Store result in smallest
    RET
FindSmallest ENDP

END MAIN
```

Test(Output)



```
Enter the elements of array (0-9), separated by spaces: 9 6 3 4 1
AVERAGE = 4
LARGEST = 9
SMALLEST = 1
```

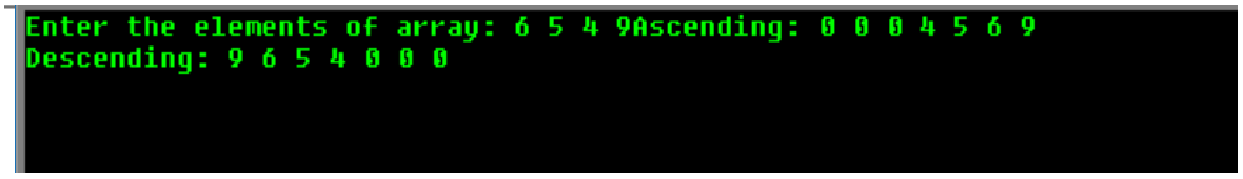
2.
Write an Assembly Language code that takes any 7 of decimal digits (0 9) in any order as input and rearrange them in ascending and descending order. Use two different procedures for arranging the digits in ascending and descending order

Src code:

```
.MODEL SMALL
.STACK 100H
.DATA
ARRAY   DB  7 DUP(?)
MSG1      DB  'Enter the elements of array (7 digits): $'
MSG2      DB  'Ascending: $'
MSG3      DB  'Descending: $'
```

```
SPACE   DB  ' $'

.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    ; Display input prompt
    MOV AH, 09H
    LEA DX, MSG1
    INT 21H

    ; Input 7 digits
    LEA SI, ARRAY
    MOV CX, 7
INPUT_LOOP:
    MOV AH, 01H
    INT 21H          ; Read a character
    CMP AL, ' '                 ; Skip space characters
    JE SKIP_DIGIT
    SUB AL, '0'                 ; Convert ASCII to number
    MOV [SI], AL
    INC SI

SKIP_DIGIT:
    LOOP INPUT_LOOP

    ; Ascending Sort
    CALL SORT_ASC

    ; Display Ascending
    MOV AH, 09H
    LEA DX, MSG2
    INT 21H
    CALL DISPLAY_ARRAY

    ; Descending Sort
    CALL SORT_DESC

    ; Display Descending
    MOV AH, 09H
    LEA DX, MSG3
    INT 21H
    CALL DISPLAY_ARRAY

    ; Exit
    MOV AH, 4CH
    INT 21H
MAIN ENDP

; Ascending Sort Procedure
SORT_ASC PROC
    MOV CX, 6                   ; 6 passes for 7 elements
ASC_OUTER:
    PUSH CX
    LEA SI, ARRAY
ASC_INNER:
    MOV AL, [SI]
    CMP AL, [SI+1]
    JLE NEXT_ASC          ; No swap needed
    ; Swap
    MOV BL, [SI+1]
    MOV [SI+1], AL
    MOV [SI], BL
NEXT_ASC:
    INC SI
    LOOP ASC_INNER
    POP CX
    LOOP ASC_OUTER
    RET
SORT_ASC ENDP

; Descending Sort Procedure
SORT_DESC PROC
    MOV CX, 6             ; 6 passes for 7 elements
DESC_OUTER:
    PUSH CX
    LEA SI, ARRAY
DESC_INNER:
    MOV AL, [SI]
    CMP AL, [SI+1]
    JGE NEXT_DESC         ; No swap needed
    ; Swap
    MOV BL, [SI+1]
    MOV [SI+1], AL
    MOV [SI], BL
NEXT_DESC:
    INC SI
    LOOP DESC_INNER
    POP CX
    LOOP DESC_OUTER
    RET
SORT_DESC ENDP

; Display Array Procedure
DISPLAY_ARRAY PROC
    LEA SI, ARRAY
    MOV CX, 7
DISPLAY_LOOP:
    MOV AL, [SI]
    ADD AL, '0'           ; Convert number back to ASCII
    MOV DL, AL
    MOV AH, 02H
    INT 21H        ; Display digit

    ; Print space
    MOV DL, ' '
    MOV AH, 02H
    INT 21H

    INC SI
    LOOP DISPLAY_LOOP

    ; New line
    MOV DL, 0DH
    INT 21H
    MOV DL, 0AH
    INT 21H
    RET
DISPLAY_ARRAY ENDP

END MAIN
```

Test(Output)



```
Enter the elements of array: 6 5 4 9Ascending: 0 0 0 4 5 6 9
Descending: 9 6 5 4 0 0 0
```

4. ANALYSIS AND DISCUSSION [3 marks]

The provided assembly language code demonstrates a basic program flow for processing user input, performing calculations, and displaying output. The key components include:

Input Handling: The code prompts the user to enter 5 decimal digits separated by spaces, and then reads each digit one by one, converting the ASCII input to numeric values and storing them in the array data structure

5. SUMMARY:

The code reads 5 decimal digits as input, calculates their average, and finds the largest and smallest numbers. It then displays the results in a formatted output.