UITS

UNIVERSITY OF INFORMATION TECHNOLOGY AND SCIENCES

Assignment on

Lab Report- 06

Course Title

Microprocessor and MicroControllers

Course Code

CSE 360

Submitted by

KM Jakaria

Section: 6A

Batch: 53

ID: 043231000510103 7

Submitted to

Md. Ismail

Lecturer

Department of CSE

UITS

Date of Submission

10 Nov, 2025

Problem No: 06

Experiment No: 06

Experiment Name: Write an Assembly Language Program to Reverse a String.

Process:

Initialize registers to point to the start of the array. For the character array, use a loop (or direct offset) to access each character and load it into the appropriate register. For the string array, use a loop (or direct offsets) to load the address of each string into the register. Use int 21h function 09h to display each character or string on the screen.

Print a new line after each string (for the string array). Repeat the process until all characters or all strings in the array have been displayed.

Implementation:

```
DATA SEGMENT
   msg1 DB 'String: $'
   msg2 DB 0DH,0AH,'Reversed String: $'
   str1 DB 'JAKARIA$', 0
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:
   MOV AX, DATA
   MOV DS, AX

   LEA DX, msg1
   MOV AH, 9
   INT 21H

   LEA DX, str1
   MOV AH, 9
```

```
    INT 21H

    LEA SI, str1
    MOV CX, 0

FIND_LEN:
    MOV AL, [SI]
    CMP AL, '$'
    JE GOT_LEN
    INC SI
    INC CX
    JMP FIND_LEN

GOT_LEN:
    DEC SI
    LEA DI, str1

    MOV BX, CX
    SHR BX, 1

REVERSE_LOOP:
    MOV AL, [DI]
    MOV DL, [SI]
    MOV [DI], DL
    MOV [SI], AL

    INC DI
    DEC SI
    DEC BX
    JNZ REVERSE_LOOP

    LEA DX, msg2
```

```
    MOV AH, 9
    INT 21H

    LEA DX, str1
    MOV AH, 9
    INT 21H

    MOV AH, 4CH
    INT 21H
CODE ENDS
END START
```

Result:

The application displays every character array element on the screen. Every string in the string array is printed on a separate line by the application.

Conclusion:

The assembly language program to reverse a string was successfully implemented and executed. The program demonstrated pointer manipulation and character swapping in 8086 assembly.