

**LAB Report**

**COURSE TITLE –** Microprocessor Lab

**COURSE CODE –** CSE 360

***Submitted To***

***Md. Ismail***

*Lecturer of UITS*

***Submitted By***

*Md. Shoyaif Rahman (****0432310005101050****)*

***Semester:*** *Autumn-2025 (6th)*

***Department:*** *CSE*

***Batch:*** *53*

***Section:****6A2*

Date: 29/10/25

Assembly Language Lab Report

# Lab report: 06

# Experiment Name: Write an assembly language to reverse a string. String must be declared at data segment of code.

# Process:

1. Start the program and set up the stack and data segment (MOV AX,@DATA and MOV DS,AX).

2. Define the string MSG in the data segment containing 'Hello World'.

3. Create another array REV in the data segment with 11 bytes to store the reversed string.

4. Initialize registers for reversing the string:

* SI = OFFSET MSG → pointer to the start of the original string.
* DI = OFFSET REV → pointer to the start of the reversed string.
* CX = 11 → number of characters in the string.

5. Move SI to point to the last character of MSG using ADD SI,CX and DEC SI.

6. Start the loop REVERSE\_LOOP to reverse the string:

 Load a character from MSG into AL using MOV AL,[SI].

 Store the character into REV using MOV [DI],AL.

 Decrement SI to move backwards through MSG.

 Increment DI to move forward through REV.

 Repeat the loop until all characters are copied in reverse order (LOOP REVERSE\_LOOP).

7. Initialize registers for printing the reversed string:

* SI = OFFSET REV → pointer to the start of the reversed string.
* CX = 11 → number of characters to print.

8. Start the loop PRINT\_LOOP to display the reversed string:

* Load the current character into DL using MOV DL,[SI].
* Print the character using DOS interrupt INT 21H with AH=2.
* Increment SI to point to the next character.
* Repeat the loop until all characters are printed (LOOP PRINT\_LOOP).

9. Exit the program using MOV AH,4CH and INT 21H.

## 2. Implementation (Program Code – ASM)

.MODEL SMALL

.STACK 100H

.DATA

MSG DB 'Hello World'

REV DB 11 DUP(?)

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV SI, OFFSET MSG

MOV DI, OFFSET REV

MOV CX, 11

ADD SI, CX

DEC SI

REVERSE\_LOOP:

MOV AL, [SI]

MOV [DI], AL

DEC SI

INC DI

LOOP REVERSE\_LOOP

MOV SI, OFFSET REV

MOV CX, 11

PRINT\_LOOP:

MOV DL, [SI]

MOV AH, 02H

INT 21H

INC SI

LOOP PRINT\_LOOP

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

**output:**

****

**3.Result**

## The program prints the string 'Hello World' in reverse order, displaying each character one by one using a loop and DOS interrupt 21H. The reversed string 'dlroW olleH' is stored in the array REV before being printed. The program demonstrates the use of loops, string traversal in both forward and backward directions, and basic DOS character output to display text on the screen.

## 4. Conclusion

The program effectively demonstrates how to reverse a string in memory and print it character by character using 8086 assembly language and DOS interrupt 21H. It illustrates sequential memory access using index registers SI and DI, loop control with CX, and proper handling of string traversal for both reading and writing. This example reinforces the concepts of memory addressing, loop operations, and character output, highlighting fundamental techniques for string manipulation in assembly language.