**ACADEMIC YEAR: 2024-25**

**Course:** Microprocessor Lab

**Course code:** CSL404

**Year/Sem:** SE-1/IV

| |
|---|
| **Experiment No.:** 6 |
| **Aim:** a. Assembly Language Program to reverse the words in String.<br>b. Assembly language program to find whether the string is palindrome or not. |
| **Name:Aishwarya Shetty** |
| **Roll Number: 33** |
| **Date of Performance:5/3/25** |
| **Date of Submission:12/3/25** |

**Evaluation**

| Performance Indicator | Max. Marks | Marks Obtained |
|---|---|---|
| Performance | 5 | |
| Understanding | 5 | |
| Journal work and timely submission. | 10 | |
| **Total** | **20** | |

| Performance Indicator | Exceed Expectations (EE) | Meet Expectations (ME) | Below Expectations (BE) |
|---|---|---|---|
| Performance | 5 | 3 | 2 |
| Understanding | 5 | 3 | 2 |
| Journal work and timely submission. | 10 | 8 | 4 |

**Checked by**

**Name of Faculty** **:** Ms. Sweety Patil

**Signature** **:**

**Date :**

**Aim:** Assembly Language Program to reverse the words in String.

**Theory:**

This program will read the string entered by the user and then reverse it. Reversing a string is the technique that reverses or changes the order of a given string so that the last character of the string becomes the first character of the string and so on.

**Algorithm:**

1. Start.
2. Initialize the Data Segment.
3. Display the message -1
4. Input the string.
5. Display the message-2
6. Take character count in DI.
7. Point to the end character and read it.
8. Display the character.
9. Decrement the count.
10. Repeat until count is Zero.
11. To terminate the program, using the DOS interrupt:
    a) Initialize AH with 4CH
    b) Call interrupt INT 21H
12. Stop.

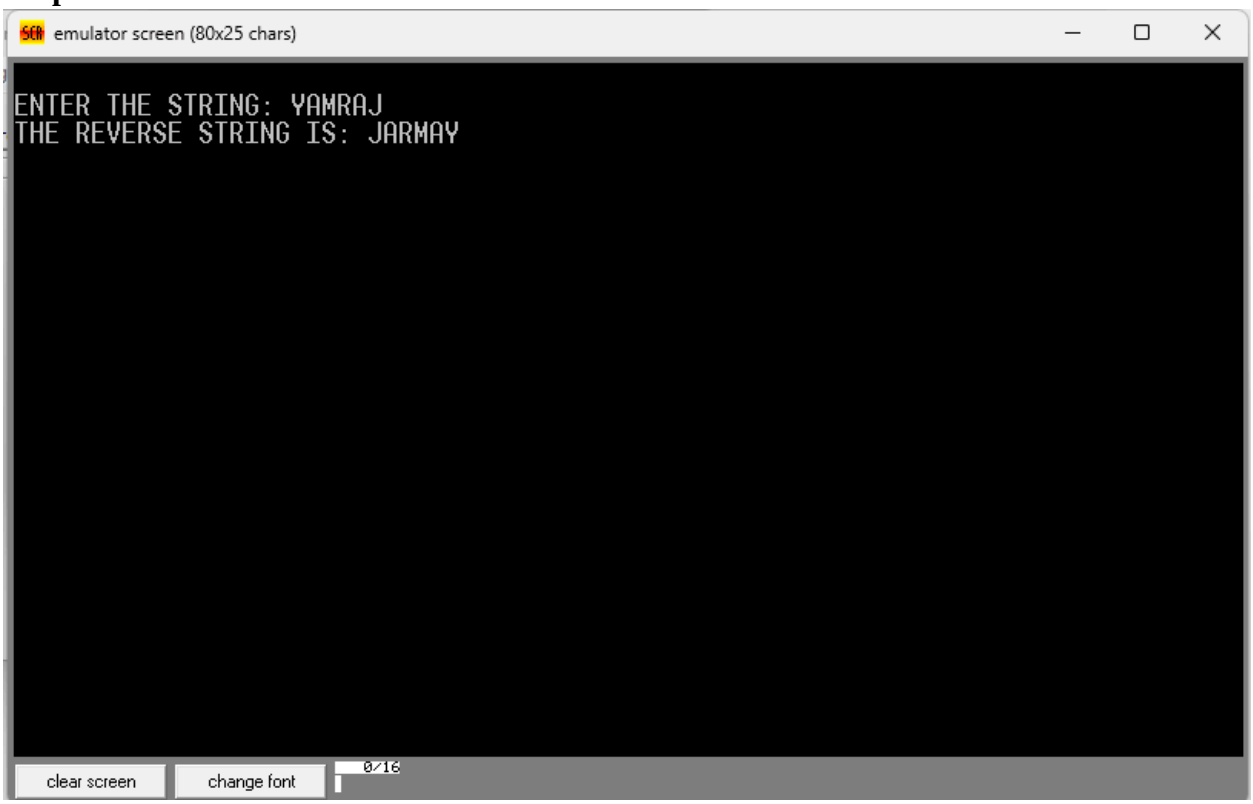
**Program:**

**a)**

```
org 100h
.DATA
M1 DB 10,13,'ENTER THE STRING: $'
M2 DB 10,13,'THE REVERSE STRING IS: $'
BUFF DB 80H
.CODE
MOV AH,09H
LEA DX,M1
INT 21H
MOV AH,0AH
LEA DX,BUFF
INT 21H
MOV AH,09H
LEA DX,M2
INT 21H
```

```
MOV CH,00H
MOV CL,BUFF + 1
LEA BX,BUFF + 1
MOV SI,CX
L1:MOV DL,[BX+SI]
MOV AH,02H
INT 21H
DEC SI
JNZ L1
ret
```

**Output:**



**Conclusion:**

1. What is MOV AH, 09H and INT 21H used for?
   **MOV AH, 09H**: This instruction loads the **AH** register with the value `09H`, which is the DOS interrupt service for printing a string to the screen. It prepares the system for the string output function.

   **INT 21H**: This triggers **interrupt 21H**, which is a software interrupt used to invoke various DOS services. When **AH = 09H**, it tells DOS to display the string pointed to by the **DX** register. The string must be terminated with a `$` symbol.

2. What is MOV AH, 02H and INT 21H used for?

   **MOV AH, 02H**: This instruction sets the **AH** register to `02H`, which is the DOS service for outputting a single character to the screen.

   **INT 21H**: This triggers the **interrupt 21H**, which invokes DOS services. When **AH = 02H**, it tells DOS to print the character specified in the **DL** register to the screen.

**Aim:** Assembly Language Program to find whether the string is palindrome or not.

**Theory:**

A Palindrome String is a string when read in a forward or backward direction remains the same. One of the approach to check this is iterate through the string till middle of string and compare a character from back and forth.

**Algorithm:**

1. Initialize the data segment.
2. Display message M1.
3. Input the string.
4. Get the starting address of the string.
5. Get the right most character.
6. Get the left most character.
7. Check for palindrome.
8. If not, Go to step 14.
9. Decrement the end pointer.
10. Increment the starting pointer.
11. Decrement the counter.
12. If count not equal to Zero, Go to step 5.
13. Display message M2.
14. Display message M3.
15. Terminate and exit to DOS.
    a) Initialize AH with 4CH
    b) Call interrupt INT 21H
16. Stop

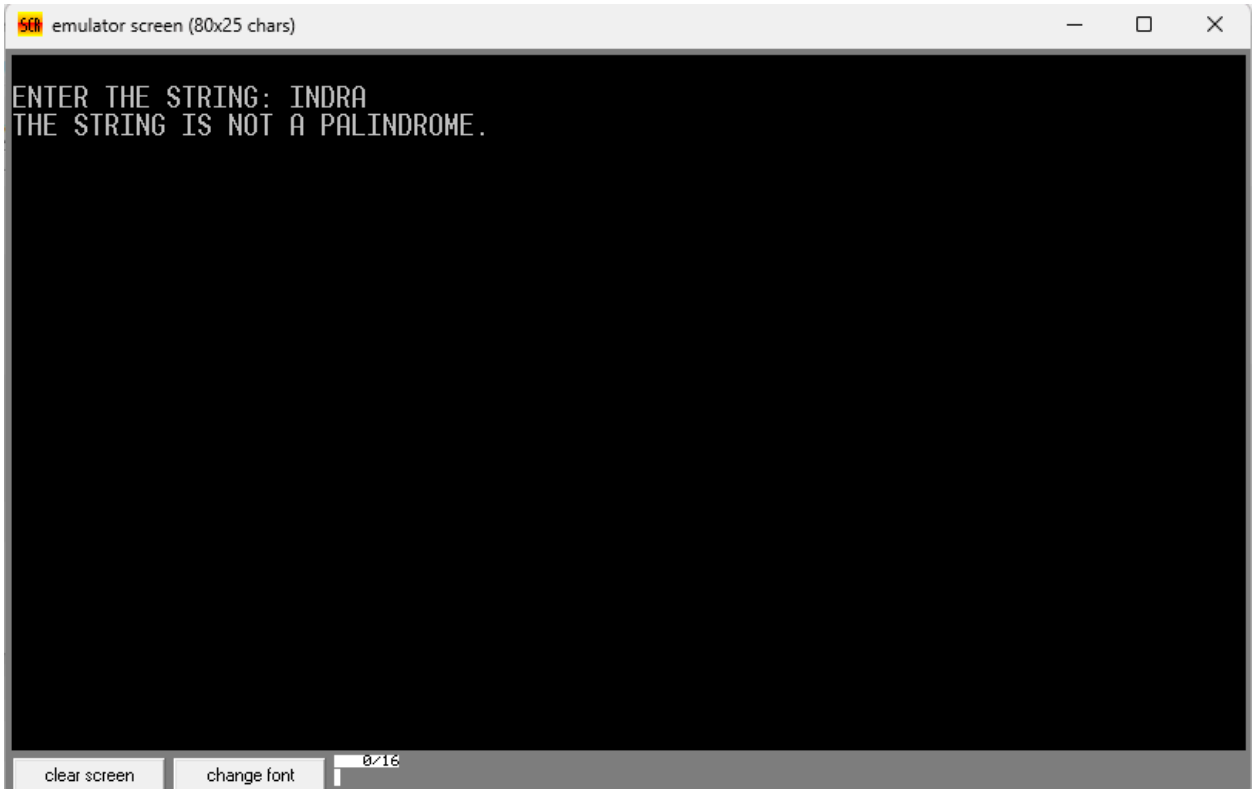
**Code:**

```
org 100h
.DATA
M1 DB 10,13, "ENTER THE STRING: $"
M2 DB 10,13, "THE STRING IS A PALINDROME. $"
M3 DB 10,13, "THE STRING IS NOT A PALINDROME. $"
BUFF DB 80h
.CODE
mov ah,09H
lea dx,M1
int 21h
mov ah,0Ah
lea dx,BUFF
```

```
int 21h
mov CH,00h
mov cl,BUFF+1
lea bx,BUFF+2
mov di,cx
dec di
sar cl,1
mov si,00h
back:mov al,[bx+di]
mov ah,[bx+si]
cmp al,ah
jnz last
dec di
inc si
dec cl
jnz back
mov ah,09h
lea dx,M2
int 21h
jmp l2

last:mov ah,09h
lea dx,m3
int 21h
jmp l2
l2:ret
```

**Output:**




**Conclusion:**

1. What is the difference between SUB and CMP?
   **SUB** (Subtract) performs subtraction between two operands and stores the result in the destination operand.
   **CMP** (Compare) subtracts the second operand from the first but **does not store** the result. Instead, it updates the **flags** (like Zero, Carry, etc.) based on the result to be used in conditional jumps or decisions.

2. What is the use of JMP instruction.
   The **JMP** (Jump) instruction is used to **unconditionally transfer control** to a different part of the program. It changes the instruction pointer (IP) to the target address, effectively causing the program to jump to that location and continue execution from there.