Anjuman — I — Islam's

M.H SABOO SIDDIK COLLEGE OF ENGINEERING

8, Saboo Siddik Polytechnic Rd., Byculla, Mumbai — 400 008.

Department of Basic Science and Humanity

🧠 Algorithm: Palindrome Finder

Step 1: Start the program.
Step 2: Declare a character array str[100] to store the input string and an integer variable i for iteration.
Step 3: Display a message:

    *** Palindrome Finder ***

Step 4: Prompt the user to enter a string.

    Enter Your String:

Step 5: Read the string from the user using gets(str).
Step 6: Find the length of the string using strlen(str).
Step 7: Use a for loop to iterate from i = 0 to i < strlen(str) - 1.
Step 8:

In each iteration, compare:
str[i] and str[strlen(str) - 1 - i]

- If both characters are **not equal**, then:
  - Print: "The String is NOT a Palindrome"
  - Exit the program.

Step 9: If the loop completes without mismatches, print:

    The String is a Palindrome

Step 10: End the program.

🧠 Algorithm: String Length Finder

Step 1: Start the program.
Step 2: Declare a **character array** str[100] to store the input string and an **integer variable** length to count the number of characters.
Initialize length = 0.
Step 3: Display a message:

    *** String Length Finder ***

Step 4: Prompt the user to enter a string.

    Enter Your String:

Step 5: Read the string from the user using gets(str).
Step 6: Using a **while loop**, traverse the string until the null character '\0' is reached.
For each character that is **not '\0'**, increment length by 1.

```
while(str[length] != '\0')
        length++;
```

Step 7: After the loop ends, print the value of length.

    String Length: <length> Characters

Step 8: End the program.