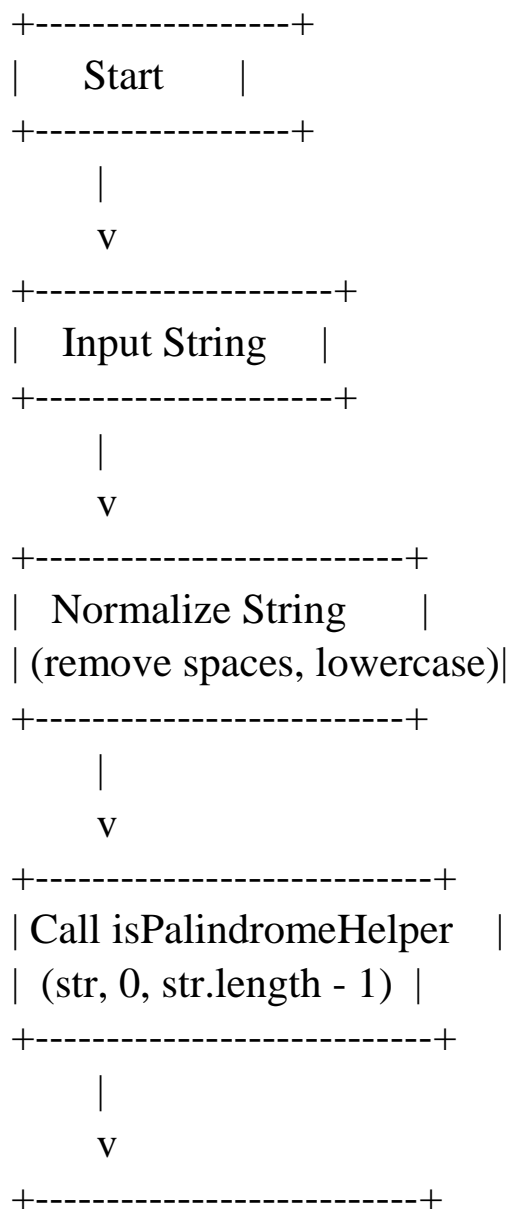# ❖Palindrome:

Explanation: -

The problem is to determine if a given string is a palindrome, meaning it reads the same forwards and backwards, ignoring spaces and case sensitivity. The solution involves using a recursive approach to compare characters from both ends of the string, moving inward. If all corresponding characters are the same, the string is identified as a palindrome; otherwise, it is not.

Time Complexity: - O(n)

Space Complexity: - O(n)

```
+-----------------+
|     Start       |
+-----------------+
        |
        v
+-------------------+
|   Input String    |
+-------------------+
        |
        v
+------------------------+
|   Normalize String     |
| (remove spaces, lowercase)|
+------------------------+
        |
        v
+---------------------------+
| Call isPalindromeHelper   |
|  (str, 0, str.length - 1) |
+---------------------------+
        |
        v
+---------------------------+
```

```
|   isPalindromeHelper      |
+--------------------------+
      |
      v
+---------------------------------+
| If left >= right, return true   |
+---------------------------------+
      |
      v
+-------------------------------------------+
| If str.charAt(left) != str.charAt(right),|
| return false (not a palindrome)          |
+-------------------------------------------+
      |
      v
+------------------------------------+
|  Call isPalindromeHelper(str, left+1,|
|         right-1)              |
+------------------------------------+
      |
      v
+----------------------+
|  Output Result       |
| (Is palindrome or not)|
+----------------------+
      |
      v
+------------------+
|     End      |
+------------------+
```