OBJECTIVE

Design a custom DoublyLinkedList class without using any built-in list or collections, and implement logic to analyze whether the list contains a palindromic sequence based on custom rules.

Problem Statement

Create a custom doubly linked list class in Python that stores a sequence of characters (A-Z, a-z). Implement a method that detects whether the stored sequence forms a custom palindrome.

Custom Palindrome Definition

A sequence is considered a custom palindrome if:

- It reads the same forwards and backwards ignoring the case.
- It ignores all vowels (a, e, i, o, u).
- It ignores repeating characters.

Requirements:-

Define Classes

- Node class
- DoublyLinkedList class

Implement Methods

- \`insert(char)\`: inserts a character at the end.
- \`is_custom_palindrome()\`: returns True if custom palindrome rules are satisfied.
- \`__str__()\`: returns the list as a printable string.

Driver Code

- Accept a string input.
- Insert all characters into the list.
- Print if it’s a custom palindrome or not.

⚠ Constraints:-

- Do not use any built-in list, deque, or similar data structures.
- Do not use \`[::-1]\`, \`.reverse()\` or similar reverse operations.

- Ignore case (treat A and a as same).
- Ignore characters: 'a', 'e', 'i', 'o', 'u' and any repeated characters from left.

Sample Input / Output:-

Input	Output
"Deified"	True
"ddoodd"	True
"banana"	False

Submission Guidelines:-

- Push code to a GitHub or GitLab repo.
- `README.md` should include:
  - Problem summary
  - Approach used
  - Time & space complexity
  - Sample test cases.
- Share the repository URL via reply to the assignment email.