Find the First Non-Repeating Character from Stream

GFG

This program finds the first non-repeating character in a given string using a queue-based approach (**firstNonRepeatingCharacter** function).

Approach 1: Function to find the first non-repeating character in a string.

- **Description:** In this approach, we maintain a queue to store characters in the order of their appearance in the string and an array to count the occurrences of each lowercase letter.
- **Steps:**
    1. Initialize an empty string **ans** to store the result.
    2. Initialize an array **count** of size 26 (for lowercase letters) and initialize all elements to 0. This array will be used to count the occurrences of each letter.
    3. Iterate through each character **ch** in the input string **str**:
        - Increment the count for the current character **ch** by accessing the corresponding element in the **count** array (**count[ch - 'a']++**).
        - Push the current character onto the queue.
        - Remove characters from the front of the queue until a non-repeating character is found (i.e., its count is 1).
        - If the queue becomes empty during this process, it means no non-repeating character was found in the substring, so we add **'#'** to the result string **ans**.
        - Otherwise, we add the first non-repeating character (found at the front of the queue) to the result string **ans**.
    4. After processing all characters in the string, the **ans** string will contain the first non-repeating characters for each substring.
- **Time Complexity: O(n), where n is the length of the input string str. We perform a single pass through the string.**
- **Space Complexity:** We create a queue to store characters in the order of their appearance in the string. **In the worst case, when all characters are unique, the queue can potentially store all characters from the input string, leading to a space complexity of O(n).**