Assignment Title: Java Programming Assignment

Deadline: 27 November, Evening
Submission Method: GitHub Pull Request

Guidelines and Mark Distribution

1. **Task Completion**: +1 mark
2. **GitHub Submission + Code Ethics** (Proper Indentation, Comments, etc.): +1 mark
3. **Optimization**: +1 mark
4. **Review Comments**: +1 mark
5. **Outstanding Performance**: +1 mark

Note: No submission will be accepted after the deadline.

Programming Language: Java
IDE Recommended: Any

Submission Instructions

- Each student must create a repository on their GitHub account.
- For each task, create a branch with the task name (e.g., `AssignemntTitle_Task1`.).
- Submit the task by creating a pull request (PR) to the main branch.
- Share the pull request link with the instructor for review and grading.

Task 1: Expand Characters in a String

Given a string containing characters followed by digits, expand each character by repeating it according to the digit that follows.

Example 1:

- **Input:** "a1b4c3"
- **Output:** "abbbbccc"
- **Explanation:**
  The character `'a'` is followed by 1, so it appears once.
  The character `'b'` is followed by 4, so it appears four times: `"bbbb"`.
  The character `'c'` is followed by 3, so it appears three times: `"ccc"`.
  The final output is `"abbbbccc"`.

Example 2:

- **Input:** "a1c2"
- **Output:** "acc"
- **Explanation:**
  The character `'a'` is followed by 1, so it appears once.
  The character `'c'` is followed by 2, so it appears twice: `"cc"`.
  The final output is `"acc"`.

Constraints:

- The string will only contain lowercase letters followed by digits.
- The input string length will be at most 100.

Task 2: Character Frequency in a String

Write a Java program that takes a string input and outputs the frequency of each character in a compressed form.

Example 1:

- **Input:** "aabcccdeee"
- **Output:** "a2b1c3d1e3"
- **Explanation:**
  The character `'a'` appears `2` times.
  The character `'b'` appears `1` time.
  The character `'c'` appears `3` times.
  The character `'d'` appears `1` time.
  The character `'e'` appears `3` times.
  Concatenating these frequencies results in `"a2b1c3d1e3"`.

Example 2:

- **Input:** "aaaaaaaaaaaa"
- **Output:** "a12"
- **Explanation:**
  The character `'a'` appears `12` times.
  The final output is `"a12"`.

Constraints:

- The input string will only contain lowercase English letters.
- The input string length will be at most 1000.

Task 3: Prime Number Checker

Write a Java program to determine if a given integer is a prime number.

Example 1:

- **Input:** 21
- **Output:** "The given number is NOT prime"
- **Explanation:**
  The number 21 is divisible by 3 and 7, hence it is not a prime number.

Example 2:

- **Input:** 7
- **Output:** "The given number is PRIME"
- **Explanation:**
  The number 7 is only divisible by 1 and 7, hence it is a prime number.

Constraints:

- The input will be a positive integer between 1 and 10^6.

Task 4: Number to Words Converter

Write a Java program that converts a given integer into its written English form.

Example 1:

- **Input:** 51
- **Output:** "fifty one"
- **Explanation:**
  The number `51` is represented as "fifty" for `50` and "one" for `1`.

Example 2:

- **Input:** 123
- **Output:** "one hundred twenty three"
- **Explanation:**
  The number `123` is represented as "one hundred" for `100`, "twenty" for `20`, and "three" for `3`.

Example 3:

- **Input:** 848
- **Output:** "eight hundred forty eight"
- **Explanation:**
  The number `848` is represented as "eight hundred" for `800`, "forty" for `40`, and "eight" for `8`.

Constraints:

- The input will be a positive integer between 1 and 1000.

Task 5: Longest Substring Without Repeating Characters

Given a string `s`, find the length of the longest substring without repeating characters.

Example 1:

- **Input:** `s = "abcabcbb"`
- **Output:** 3
- **Explanation:** The answer is `"abc"`, with the length of 3.

Example 2:

- **Input:** `s = "bbbbb"`
- **Output:** 1
- **Explanation:** The answer is `"b"`, with the length of 1.

Example 3:

- **Input:** `s = "pwwkew"`
- **Output:** 3
- **Explanation:** The answer is `"wke"`, with the length of 3. Notice that the answer must be a substring, `"pwke"` is a subsequence and not a substring.

Constraints:

- 0 <= `s.length` <= 50,000
- `s` consists of English letters, digits, symbols, and spaces.

Submission Notes:

- All submissions must adhere to the code ethics mentioned above.
- Submit each task via GitHub as mentioned in the guidelines.
- Provide proper comments in your code to enhance readability.
- Ensure your code is optimized for performance.