STRING

1. **Write a basic program to take input (String) from User and just print it.**

2. **Write a program to count the number of occurrences of each character in the string and print it.**
   [Solution: https://www.geeksforgeeks.org/java-program-count-occurrences-character/]

3. **Write a program to remove all whitespaces in a given string.**
   [Solution: https://www.geeksforgeeks.org/how-to-remove-all-white-spaces-from-a-string-in-java/]

4. **Find Duplicate characters in a string.**
   [Solution: https://www.w3schools.in/java-program/java-program-find-duplicate-characters-string/]

5. **Write a program to reverse the string in place.**
   [Solution: https://www.java67.com/2016/06/how-to-reverse-string-in-place-in-java.html]

6. **Write a program to check whether given two strings are anagram or not.**
   [Practice here: https://practice.geeksforgeeks.org/problems/anagram/0]

7. **Why strings are immutable in Java?**
   [Solution: https://www.geeksforgeeks.org/java-string-is-immutable-what-exactly-is-the-meaning/]

8. **How do you convert string to integer and integer to string in java?**
   [Solution: https://javaconceptoftheday.com/string-to-integer-integer-to-string-conversion-in-java/]

9. **Write a program to reverse each word in the given string.**
   [Solution: https://www.geeksforgeeks.org/reverse-individual-words/]

10. **Check whether the String is a palindrome or not.**
    [Solution: https://practice.geeksforgeeks.org/problems/palindrome-string/0]

String

Intermediate Level Questions:

1. **Write a Code to check whether one string is a rotation of another**
   [Practice here: https://www.geeksforgeeks.org/a-program-to-check-if-strings-are-rotations-of-each-other/]

2. **Write a program to remove Duplicate characters from the String.**
   [Follow here: https://www.geeksforgeeks.org/remove-duplicates-from-a-given-string/]

3. **Write a Program to check whether a string is a valid shuffle of two strings or not.**
   [Follow here: https://www.geeksforgeeks.org/check-whether-a-given-string-is-an-interleaving-of-two-other-given-strings/]

4. **Write a program to find the longest Palindrome in a string.[Lonest palindromic Substring]**
   [Practice here: https://practice.geeksforgeeks.org/problems/longest-palindrome-in-a-string/0]

5. **Find Longest Recurring Subsequence in String.**
   [Practice here: https://practice.geeksforgeeks.org/problems/longest-repeating-subsequence/0]

6. **Print all Subsequences of a string.**
   [Follow here: https://www.geeksforgeeks.org/print-subsequences-string/]

7. **Print all the permutations of the given string**
   [Practice here: https://practice.geeksforgeeks.org/problems/permutations-of-a-given-string/0]

8. **Split the Binary string into ~~two~~ substring with equal 0's and 1's.**
   [Follow here: https://www.geeksforgeeks.org/split-the-binary-string-into-substrings-with-equal-number-of-0s-and-1s/]

9. **Rearrange characters in a string such that no two adjacent are same**
   [Practice here: https://practice.geeksforgeeks.org/problems/rearrange-_characters/0]

10. **Write a program to find the smallest window that contains all characters of string itself.**
[Practice here: https://practice.geeksforgeeks.org/problems/smallest-distant-window/0]

11. **Number of Substrings with count of each character as "K".**

12. **Find the longest common subsequence between two strings.**
[Practice here: https://practice.geeksforgeeks.org/problems/longest-common-subsequence/0]

13. **Word Wrap Problem [VERY IMP].**
[Practice here: https://practice.geeksforgeeks.org/problems/word-wrap/0]

14. **Program to generate all possible valid IP addresses from given string.**
[Follow here: https://www.geeksforgeeks.org/program-generate-possible-valid-ip-addresses-given-string/]

15. **EDIT Distance [Very Imp]**
[Practice here: https://practice.geeksforgeeks.org/problems/edit-distance/0]

16. **Find next greater number with same set of digits. [Very Very IMP]**
[Practice here: https://practice.geeksforgeeks.org/problems/next-permutation/0]

17. **Try your hands on all these conversions:**
    - ➔ Prefix to Infix
    - ➔ Prefix to Postfix
    - ➔ Postfix to prefix
    - ➔ Postfix to infix

[Follow link: https://www.geeksforgeeks.org/prefix-infix-conversion/]

18. **Convert a Sentence into its equivalent mobile numeric keypad sequence.**
[Follow here: https://www.geeksforgeeks.org/convert-sentence-equivalent-mobile-numeric-keypad-sequence/]

19. **Balanced Parenthesis problem.[Imp]**
[Practice here: https://practice.geeksforgeeks.org/problems/parenthesis-checker/0]

20. **Minimum number of swaps for bracket balancing.**
[Practice here: https://practice.geeksforgeeks.org/problems/minimum-swaps-for-bracket-balancing/0]

21. **Minimum number of bracket reversals needed to make an expression balanced.**

[Practice here: https://practice.geeksforgeeks.org/problems/count-the-reversals/0]

22. **Word break Problem[Very Imp]**

[Practice here: https://practice.geeksforgeeks.org/problems/word-break/0]

23. **Minimum rotations required to get the same string.**

[Follow here: https://www.geeksforgeeks.org/minimum-rotations-required-get-string/]

24. **Find the first repeated word in string.**

[Practice here: https://practice.geeksforgeeks.org/problems/second-most-repeated-string-in-a-sequence/0]

25. **Efficiently find first repeated character in a string without using any additional data structure in one traversal**

[Practice here : https://practice.geeksforgeeks.org/problems/find-first-repeated-character/0]

26. **Count All Palindromic Subsequence in a given String.**

[Practice here: https://practice.geeksforgeeks.org/problems/count-palindromic-subsequences/1]

27. **Number of flips to make binary string alternate**

[Practice here: https://practice.geeksforgeeks.org/problems/min-number-of-flips/0]

28. **Count of number of given string in 2D character array**

[Follow here: https://www.geeksforgeeks.org/find-count-number-given-string-present-2d-character-array/]

29. Search a Word in a 2D Grid of characters.

[Practice here: https://practice.geeksforgeeks.org/problems/find-the-string-in-grid/0]

30. Boyer Moore Algorithm for Pattern Searching.

[Follow here: https://www.geeksforgeeks.org/boyer-moore-algorithm-for-pattern-searching/]

Anshika G

1. Atoi (InterviewBit)
2. Longest common prefix (LeetCode)
3. Longest Palindromic substring (LeetCode)
4. Next greater no. with same set of digits (GFG)
5. Implements strstr (InterviewBit)
6. Regular Expression matching (LeetCode)
7. Wildcard matching (LeetCode)
8. Reverse string wordwise (GFG)
9. Letter combination of phone number (LeetCode)
10. Word break (GFG)