# Beginner - Intermediate - Advanced String Questions

## Level 1 — Warm-up / Basics

1. **Reverse a String**
   Input: `"hello"` → Output: `"olleh"`
   *(Try both iterative swap method and building a new string.)*

2. **Check Palindrome**
   `"madam"` → true
   `"apple"` → false

3. **Count vowels and consonants**
   `"education"` → vowels: 5, consonants: 4

4. **Remove spaces from a String**
   `"a b c"` → `"abc"`

5. **Change case**
   `"Hello"` → `"hELLO"`

---

## Level 2 — Pattern & Logic Building

6. **Count occurrences of each character** (without using Map)
   `"banana"` → `b:1, a:3, n:2`

7. **Remove duplicates**
   `"programming"` → `"progamin"`

8. **Find first non-repeating character**
   `"swiss"` → `'w'`

9. **Check if two strings are anagrams** (same letters in different order)
   `"listen"`, `"silent"` → true

10. **Find all substrings of a String**
    `"abc"` → `"a"`, `"b"`, `"c"`, `"ab"`, `"bc"`, `"abc"`

## Level 3 — Real Interview Style

11. **Rotate a string by k positions**
    `"abcdef", k=2` → `"cdefab"`

12. **Check if string is rotation of another**
    `"waterbottle"`, `"erbottlewat"` → true

13. **Find longest palindrome substring**
    `"babad"` → `"bab"` or `"aba"`

14. **Find all permutations of a string**
    `"abc"` → `"abc"`, `"acb"`, `"bac"`, `"bca"`, `"cab"`, `"cba"`

15. **Remove all occurrences of a substring**
    `"abcabcabc", remove="ab"` → `"ccc"`

---

## Level 4 — Advanced & Tricky

16. **String compression**
    `"aaabbcaaa"` → `"a3b2c1a3"`
    *(Leetcode style)*

17. **Pattern matching without library functions**
    `"abcabcabcd"`, pattern `"abcd"` → index 6

18. **Zigzag conversion** (like Leetcode #6)
    `"PAYPALISHIRING"`, rows=3 → `"PAHNAPLSIIGYIR"`

19. **Minimum window substring** (hard version without Maps)
    Given two strings, find the smallest window in first string containing all chars of second.

20. **Multiply two large numbers stored as strings**
    `"123456789" * "987654321"` → `"121932631112635269"`

### Level 5 — Pure Logic Monsters

21. **Implement `indexOf` without using `indexOf`**
22. **Implement `split` without using `split`**
23. **Implement `replace` without using `replace`**
24. **Convert integer to Roman numeral** (string manipulation)
25. **Convert Roman numeral to integer**

---

If you grind these in order, you'll cover:

- Traversing strings manually
- Comparing and swapping characters
- Working with ASCII values
- Building substrings manually
- Handling edge cases like empty strings and repeated chars
- Thinking about **time complexity** when you can't lean on built-in structures