■ Python String Coding Questions

Basic Level

1 Find the length of a string without using len().
2 Reverse a string (different methods).
3 Check if a string is a palindrome.
4 Count the frequency of characters in a string.
5 Count vowels and consonants in a string.
6 Count the number of words in a string.
7 Remove all whitespace from a string.
8 Convert uppercase to lowercase and vice versa (without using .swapcase()).
9 Remove all duplicate characters from a string.
10 Check if a string contains only digits.

Intermediate Level

1 Find the first non-repeating character in a string.
2 Find the first repeating character in a string.
3 Check if two strings are anagrams.
4 Reverse each word in a string.
5 Remove all special characters from a string.
6 Find the longest word in a sentence.
7 Replace all spaces with %20 (like URL encoding).
8 Print all substrings of a string.
9 Find the longest common prefix of given strings.
10 Implement substring search (without using .find() / regex).

Advanced Level

1 Find the longest palindrome substring.
2 Find the longest repeating substring.
3 Generate all permutations of a string.
4 Remove all adjacent duplicate characters.
5 Implement string compression (e.g., 'aaabbc' → 'a3b2c1').
6 Check if one string is a rotation of another.
7 Find the edit distance (Levenshtein distance) between two strings.
8 Find the smallest window substring containing all characters of another string.
9 Implement pattern matching using KMP algorithm.
10 Encode and decode a string using Run Length Encoding (RLE).

Extra Important Questions

1 Write a program to check if a string is a pangram (contains all alphabets).
2 Find the most frequent character in a string.
3 Find the least frequent character in a string.
4 Check if a string is an isogram (no repeating letters).
5 Remove all duplicate words from a string.
6 Find the second most frequent character in a string.

7 Sort characters of a string in alphabetical order.
8 Sort words of a sentence by length.
9 Capitalize the first letter of each word (without using .title()).
10 Find the longest word that is also a palindrome in a sentence.
11 Check if one string is a subsequence of another.
12 Count the number of substrings that are palindromes.
13 Find the minimum number of deletions to make two strings anagrams.
14 Implement a basic spell checker: given a dictionary of words, check if an input word exists or suggest closest matches.
15 Convert a string from camelCase to snake_case and vice versa.
16 Print all possible combinations of characters from a string.
17 Find the longest word formed from given letters.
18 Implement string multiplication (e.g., 'abc' * 3 → 'abcabcabc') without using *.
19 Replace all consecutive spaces with a single space.
20 Write a program to check if a string follows a given pattern (like regex but manually).