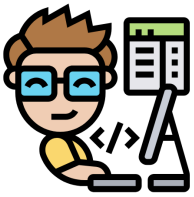

DAILY PROGRAMMING
CHALLENGE

WISDOM SPROUTS
IT Training Hub

Permutations of a String

You are given a string s. Your task is to generate and return all possible permutations of the characters in the string. A permutation is a rearrangement of the characters in the string, and each character must appear exactly once in every permutation. If there are duplicate characters in the string, the resulting permutations should also be unique (i.e., no repeated permutations).

Input:
A string s consisting of lowercase English letters. The length of the string n satisfies $1 \leq n \leq 9$.

Output:

- An array of strings containing all unique permutations of the input string. The order of permutations in the output does not matter.

Examples:

- Example 1
  Input: "abc"
  Output: ["abc", "acb", "bac", "bca", "cab", "cba"]
  Explanation: All possible arrangements of "abc" are listed, and there are no duplicate permutations.

Constraints:

- 1 ≤ s.length ≤ 9
- The string s consists of lowercase English letters.
- If s contains duplicate letters, the permutations must be unique.

Test Cases:

1. Input: "abc"
   Output: ["abc", "acb", "bac", "bca", "cab", "cba"]
2. Input: "aab"
   Output: ["aab", "aba", "baa"]
3. Input: "aaa"
   Output: ["aaa"]

4. Input: "a"
   Output: ["a"]
5. Input: "abcd"
   Output: ["abcd", "abdc", "acbd", "acdb", "adbc", "adcb", "bacd", ..., "dcba"]

Edge Cases:

1. Single character string: If the input string has only one character, the output will be the string itself.
2. All characters the same: If all characters in the string are identical, there will be only one unique permutation.
3. String contains duplicate characters: If the input string contains repeated characters, the output must not contain duplicate permutations.