JAVA ARRAY & STRING PRACTICE SHEET (CONSTRAINED)

Rules:

1. No StringBuilder or StringBuffer.
2. Allowed String methods: charAt(), split(), length(), and toCharArray().
3. Language: Java

==
=====================

--- BEGINNER LEVEL ---

1. Sum of Elements
   Calculate the total sum of an integer array.
   - Signature: public int getSum(int[] nums)
   - Input: [1, 2, 3, 4] | Output: 10

2. Find Maximum
   Find the largest integer in an unsorted array.
   - Signature: public int findMax(int[] nums)
   - Input: [5, 2, 9, 1, 6] | Output: 9

3. Count Occurrences
   Count how many times a target integer 'x' appears in the array.
   - Signature: public int countX(int[] nums, int x)
   - Input: nums = [1, 2, 1, 3, 1], x = 1 | Output: 3

4. Reverse an Array
   Reverse the elements of an array in-place.
   - Signature: public void reverse(int[] nums)
   - Input: [10, 20, 30] | Output: [30, 20, 10]

5. Linear Search
   Return the index of the first occurrence of 'target'. Return -1 if not found.
   - Signature: public int search(int[] nums, int target)
   - Input: nums = [4, 8, 2], target = 8 | Output: 1

6. Reverse a String (Manual)
   Reverse a string by creating a char[] and swapping elements manually.

- Signature: public String reverseString(String s)
- Input: "java" | Output: "avaj"

7. Vowel Counter

Count 'a', 'e', 'i', 'o', 'u' using charAt().

- Signature: public int countVowels(String s)
- Input: "Hello World" | Output: 3

8. Palindrome Check

Compare characters from start and end using charAt().

- Signature: public boolean isPalindrome(String s)
- Input: "racecar" | Output: true

9. Remove Whitespace

Use split(" ") and concatenate the resulting parts back together.

- Signature: public String removeSpaces(String s)
- Input: " a b c " | Output: "abc"

10. Toggle Case

Convert lowercase to uppercase (ASCII math: 'a' - 32 = 'A') and vice versa.

- Signature: public String toggleCase(String s)
- Input: "aBc" | Output: "AbC"

--- INTERMEDIATE LEVEL ---

11. Two Sum

Find the indices of two numbers that add up to a target.

- Signature: public int[] twoSum(int[] nums, int target)
- Input: nums = [2, 7, 11], target = 9 | Output: [0, 1]

12. Remove Duplicates (Sorted Array)

Remove duplicates in-place (don't worry about array size, just move values).

- Signature: public int removeDuplicates(int[] nums)
- Input: [1, 1, 2] | Output: 2 (Array starts with [1, 2])

13. Move Zeroes

Move all 0's to the end without changing the order of non-zero elements.

- Signature: public void moveZeroes(int[] nums)
- Input: [0, 1, 0, 3, 12] | Output: [1, 3, 12, 0, 0]

14. Rotate Array
Rotate the array to the right by k steps.
- Signature: public void rotate(int[] nums, int k)
- Input: [1, 2, 3, 4, 5], k = 2 | Output: [4, 5, 1, 2, 3]

15. Valid Anagram
Check if two strings are anagrams using a frequency array of size 26.
- Signature: public boolean isAnagram(String s, String t)
- Input: s = "anagram", t = "nagaram" | Output: true

16. First Unique Character
Find the index of the first non-repeating character using charAt().
- Signature: public int firstUniqChar(String s)
- Input: "leetcode" | Output: 0

17. String Compression (Logic-heavy)
Compress "aaabb" to "a3b2". Since you can't use StringBuilder, use a char[].
- Signature: public String compress(String s)
- Input: "aabbcc" | Output: "a2b2c2"

18. Longest Common Prefix
Find the longest common prefix among an array of strings.
- Signature: public String longestCommonPrefix(String[] strs)
- Input: ["flower","flow","flight"] | Output: "fl"

--- ADVANCED LEVEL ---

19. Sliding Window Maximum
Find the max element in every sliding window of size k.
- Signature: public int[] maxSlidingWindow(int[] nums, int k)
- Input: nums = [1, 3, -1, -3, 5], k = 3 | Output: [3, 3, 5]

20. Longest Substring Without Repeating Characters

Find the length of the longest substring without duplicates using a sliding window.

- Signature: public int lengthOfLongestSubstring(String s)
- Input: "abcabcbb" | Output: 3

21. Trapping Rain Water

Calculate trapped water. Hint: Use two pointers or prefix/suffix arrays.

- Signature: public int trap(int[] height)
- Input: [0, 1, 0, 2, 1, 0, 1, 3] | Output: 6

22. Minimum Window Substring

Smallest substring of S containing all characters of T. Use charAt() extensively.

- Signature: public String minWindow(String s, String t)
- Input: s = "ADOBECODEBANC", t = "ABC" | Output: "BANC"