Level 2

The Frequency Frontier: Explore, Detect, Repeat

1. First Repeating Element

🎯 **Task:** Return the first element that repeats (based on first appearance).
📌 **Description:** Focuses on identifying the earliest recurring element in the array—ideal for showcasing frequency tracking and first-occurrence logic.

🧪 **Example:**
Input: arr = [10, 5, 3, 4, 3, 5, 6]
Output: 3

2. First Non-Repeating Element

🎯 **Task:** Return the first element that appears only once.
📌 **Description:** Highlights array traversal combined with frequency counting—clean demonstration of hash maps in action.

🧪 **Example:**
Input: arr = [9, 4, 9, 6, 5, 4]
Output: 5

3. Unique Intersection of Two Arrays

🎯 **Task:** Return all common elements (no duplicates), any order.
📌 **Description:** Emphasizes set-based logic and deduplication strategies across data sources—efficient comparisons in unsorted arrays.

🧪 **Example:**
Input: arr1 = [4, 3, 1, 2, 2], arr2 = [5, 2, 3, 4, 4, 5]
Output: [2, 3, 4] *(any order)*

4. Find Me If You Can

🎯 **Task:** Print index of key x or -1 (linear search).
📌 **Description:** Classic search problem—demonstrates index tracking and basic iteration.

🧪 **Example:**
Input: arr = [6, 34, 3, 2, 1, 5, 3, 2], x = 5
Output: 4 *(1-based indexing)*

5. Count My Presence

🎯 **Task:** Count occurrences of key x.
📌 **Description:** Solid frequency analysis—perfect for showcasing use of hash tables or counters.

🧪 **Example:**
Input: arr = [7, 5, 4, 5, 6, 5, 5, 7, 5], x = 5
Output: 4

6. Odd One Out

🎯 **Task:** Exactly one element appears odd # times, others even → output that element (XOR).
📌 **Description:** Clever use of XOR for space-efficient solving—great for surprising interview-style logic.

🧪 **Example:**
Input: arr = [4, 2, 4, 5, 2, 5, 3, 3, 4]
Output: 4

7. First Repeating (Index Version)

🎯 **Task:** Return first index (1-based) of element that appears ≥ 2. If none, print -1.
📌 **Description:** Combines value tracking with indexing—highlights lookup optimization.

🧪 **Example:**
Input: arr = [1, 5, 3, 4, 3, 5, 6]
Output: 2 *(element 5 repeats first)*

8. First Non-Repeating (Value Version)

🎯 **Task:** Return first non-repeating element or -1.
📌 **Description:** Stresses value output and edge-case handling—clean logic for visibility into array states.

🧪 **Example:**
Input: arr = [9, 4, 9, 6, 5, 4]
Output: 5

9. Unique Intersection (Revisited)

🎯 **Task:** Given two unsorted arrays, print all common elements without duplicates, in any order.
📌 **Description:** Demonstrates intersection of unordered datasets—normalization before comparison.

🧪 **Example:**
Input:
arr1 = [4, 3, 1, 2, 2]
arr2 = [5, 2, 3, 4, 4, 5]
Output: [2, 3, 4]