📦 **Array (10 Questions)**

1. Declare an array of integers with size 5 and initialize it with values from 1 to 5.
2. Write a program to find the maximum value in an integer array.
3. Rotate an array to the right by 2 positions. (e.g., [1,2,3,4] ➡ [3,4,1,2])
4. Find the second largest element in an array.
5. Check if a given array is a palindrome.
6. Remove duplicates from a sorted array.
7. Merge two sorted arrays into one sorted array.
8. Count the number of even and odd elements in an array.
9. Rearrange the array so that negative and positive numbers alternate.
10. Find the element that appears only once when every other element appears twice.

🥞 **Stack (10 Questions)**

1. Implement a stack using an array or list.
2. Push 5 elements to a stack and then pop 3 elements.
3. Check for balanced parentheses in a string.
4. Reverse a string using a stack.
5. Implement a stack with an additional getMin() that returns the minimum in O(1).
6. Convert infix expression to postfix.
7. Evaluate a postfix expression.
8. Sort a stack using another temporary stack.
9. Design a browser back button using stack.
10. Remove adjacent duplicates from a string using stack (e.g., "abbaca" → "ca").

🚶‍♂️ **Queue (10 Questions)**

1. Implement a basic queue using a list or array.
2. Enqueue 4 elements and dequeue 2. Print the remaining queue.
3. Implement a circular queue.
4. Design a queue using two stacks.
5. Generate binary numbers from 1 to N using a queue.
6. Check if a given queue is a palindrome (without using extra space).
7. Implement a priority queue manually (without built-in libraries).
8. Reverse the first K elements of a queue.
9. Implement a queue to simulate task scheduling (e.g., round-robin).
10. Simulate a printer job queue with priorities.

📋 **List (10 Questions)**

1. Create a list and insert elements at the beginning, end, and a specific index.
2. Find the middle element in a linked list.
3. Reverse a linked list.
4. Detect a loop in a linked list.
5. Merge two sorted linked lists.
6. Remove duplicates from an unsorted list.
7. Delete the N-th node from the end of the list.
8. Partition a list around a value X (e.g., values less than X come before).
9. Rotate a linked list to the right by K places.
10. Clone a linked list with a random pointer.

📚 **Dictionary / HashMap (10 Questions)**

1. Create a dictionary with student names as keys and marks as values.
2. Count the frequency of each character in a string.
3. Find the first non-repeating character in a string.
4. Implement a simple phonebook lookup system.
5. Find all pairs with a given sum in an array using a dictionary.
6. Group words that are anagrams of each other.
7. Check if two strings are isomorphic.
8. Find the most frequent word in a paragraph.
9. Use a dictionary to track inventory in a shop.
10. Implement a LRU (Least Recently Used) cache using dictionary and doubly-linked list.