Java DSA/Logic Questions for QA - Full Set with Solutions

1. Find the duplicate elements in an array

```
public class DuplicateElements {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 2, 4, 1};
        for (int i = 0; i < arr.length; i++) {
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[i] == arr[j]) {
                    System.out.println("Duplicate: " + arr[i]);
                }
            }
        }
    }
}
```

2. Check if a string is a palindrome

```
public class PalindromeCheck {
    public static void main(String[] args) {
        String str = "madam";
        String reversed = new StringBuilder(str).reverse().toString();
        if (str.equals(reversed)) {
            System.out.println("Palindrome");
        } else {
            System.out.println("Not a palindrome");
        }
    }
}
```

3. Reverse a string without using reverse()

```
public class ReverseString {
    public static void main(String[] args) {
        String str = "hello";
        String reversed = "";
        for (int i = str.length() - 1; i >= 0; i--) {
            reversed += str.charAt(i);
        }
        System.out.println("Reversed: " + reversed);
    }
}
```

4. Count vowels in a string

```
public class VowelCount {
    public static void main(String[] args) {
        String str = "automation";
        int count = 0;
        for (char c : str.toCharArray()) {
            if ("aeiouAEIOU".indexOf(c) != -1) {
                count++;
            }
        }
        System.out.println("Vowels: " + count);
    }
}
```

5. Find the maximum and minimum in an array

```
public class MaxMinArray {
    public static void main(String[] args) {
        int[] arr = {3, 5, 1, 8, 2};
        int max = arr[0], min = arr[0];
        for (int num : arr) {
            if (num > max) max = num;
            if (num < min) min = num;
        }
        System.out.println("Max: " + max + ", Min: " + min);
    }
}
```

6. Check if two strings are anagrams

```
import java.util.Arrays;

public class AnagramCheck {
    public static void main(String[] args) {
        String str1 = "listen", str2 = "silent";
        char[] arr1 = str1.toCharArray();
        char[] arr2 = str2.toCharArray();
        Arrays.sort(arr1);
        Arrays.sort(arr2);
        System.out.println(Arrays.equals(arr1, arr2) ? "Anagrams" : "Not Anagrams");
    }
}
```

7. Remove duplicate characters from a string

```
import java.util.LinkedHashSet;
import java.util.Set;

public class RemoveDuplicates {
    public static void main(String[] args) {
        String str = "automation";
        Set<Character> set = new LinkedHashSet<>();
        for (char c : str.toCharArray()) {
            set.add(c);
        }
```

```java
        StringBuilder result = new StringBuilder();
        for (char c : set) {
            result.append(c);
        }
        System.out.println("After removing duplicates: " + result);
    }
}
```

8. Check if a number is prime

```java
public class PrimeCheck {
    public static void main(String[] args) {
        int num = 17;
        boolean isPrime = num > 1;
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                isPrime = false;
                break;
            }
        }
        System.out.println(isPrime ? "Prime" : "Not Prime");
    }
}
```

9. Find factorial of a number using recursion

```java
public class Factorial {
    public static int factorial(int n) {
        if (n <= 1) return 1;
        return n * factorial(n - 1);
    }

    public static void main(String[] args) {
        int num = 5;
        System.out.println("Factorial: " + factorial(num));
    }
}
```

10. Check if a number is a palindrome

```java
public class PalindromeNumber {
    public static void main(String[] args) {
        int num = 121, reversed = 0, original = num;
        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
        System.out.println(original == reversed ? "Palindrome" : "Not Palindrome");
    }
}
```

11. Fibonacci series up to N terms

```java
public class FibonacciSeries {
    public static void main(String[] args) {
        int n = 10, a = 0, b = 1;
        System.out.print("Fibonacci: ");
        for (int i = 0; i < n; i++) {
            System.out.print(a + " ");
            int next = a + b;
            a = b;
            b = next;
        }
    }
}
```

12. Sum of digits in a number

```java
public class SumOfDigits {
    public static void main(String[] args) {
        int num = 1234, sum = 0;
        while (num != 0) {
            sum += num % 10;
            num /= 10;
        }
        System.out.println("Sum of digits: " + sum);
    }
}
```

13. Find missing number in an array from 1 to N

```java
public class MissingNumber {
    public static void main(String[] args) {
        int[] arr = {1, 2, 4, 5};
        int n = 5, sum = n * (n + 1) / 2;
        for (int num : arr) {
            sum -= num;
        }
        System.out.println("Missing number: " + sum);
    }
}
```

14. Count occurrences of each word in a string

```java
import java.util.HashMap;

public class WordCount {
    public static void main(String[] args) {
        String str = "java code java test code java";
        String[] words = str.split(" ");
        HashMap<String, Integer> map = new HashMap<>();
        for (String word : words) {
            map.put(word, map.getOrDefault(word, 0) + 1);
        }
        System.out.println(map);
    }
}
```

15. Sort an array without using sort()

```
import java.util.Arrays;

public class SortArray {
    public static void main(String[] args) {
        int[] arr = {5, 3, 8, 1, 2};
        for (int i = 0; i < arr.length; i++) {
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[i] > arr[j]) {
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
        System.out.println(Arrays.toString(arr));
    }
}
```

16. Use HashMap to count character frequency

```
import java.util.HashMap;

public class CharFrequency {
    public static void main(String[] args) {
        String str = "tester";
        HashMap<Character, Integer> map = new HashMap<>();
        for (char c : str.toCharArray()) {
            map.put(c, map.getOrDefault(c, 0) + 1);
        }
        System.out.println(map);
    }
}
```

17. Use HashSet to find duplicates in a list

```
import java.util.*;

public class FindDuplicates {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 2, 4, 1};
        Set<Integer> set = new HashSet<>();
        Set<Integer> duplicates = new HashSet<>();
        for (int num : arr) {
            if (!set.add(num)) {
                duplicates.add(num);
            }
        }
        System.out.println("Duplicates: " + duplicates);
    }
}
```

18. Find first non-repeating character in a string

```
import java.util.LinkedHashMap;

public class FirstNonRepeating {
    public static void main(String[] args) {
        String str = "aabbcdeff";
        LinkedHashMap<Character, Integer> map = new LinkedHashMap<>();
        for (char c : str.toCharArray()) {
            map.put(c, map.getOrDefault(c, 0) + 1);
        }
        for (char c : map.keySet()) {
            if (map.get(c) == 1) {
                System.out.println("First non-repeating: " + c);
                break;
            }
        }
    }
}
```