All 27 Java Problems with Libraries

--- CHALLENGE 2: ARRAY MODULE 3 ---

1. Binary Search

```java
import java.util.Arrays;
public class BinarySearch {
    public static void main(String[] args) {
        int[] arr = {5, 12, 18, 24, 33, 45};
        int key = 24;
        int result = Arrays.binarySearch(arr, key);
        System.out.println((result >= 0) ? "Found at index: " + result : "Not Found");
    }
}
```

2. Third Largest and Second Smallest

```java
import java.util.Arrays;
public class FindElements {
    public static void main(String[] args) {
        int[] arr = {10, 5, 20, 8, 15};
        Arrays.sort(arr);
        System.out.println("Third Largest: " + arr[arr.length - 3]);
        System.out.println("Second Smallest: " + arr[1]);
    }
}
```

3. Merge Two Arrays

```java
import java.util.Arrays;
public class MergeArrays {
    public static void main(String[] args) {
        int[] a = {1, 2, 3};
        int[] b = {4, 5, 6};
        int[] result = new int[a.length + b.length];
        System.arraycopy(a, 0, result, 0, a.length);
        System.arraycopy(b, 0, result, a.length, b.length);
        System.out.println(Arrays.toString(result));
    }
}
```

4. Insertion Sort

```java
public class InsertionSort {
    public static void main(String[] args) {
        int[] arr = {9, 5, 1, 4, 3};
        for (int i = 1; i < arr.length; i++) {
            int key = arr[i], j = i - 1;
            while (j >= 0 && arr[j] > key) arr[j + 1] = arr[j--];
            arr[j + 1] = key;
        }
        for (int num : arr) System.out.print(num + " ");
    }
}
```

5. Remove Duplicate Elements

```
import java.util.*;
public class RemoveDuplicates {
    public static void main(String[] args) {
        int[] arr = {2, 2, 4, 4, 5, 6};
        Set<Integer> set = new LinkedHashSet<>();
        for (int i : arr) set.add(i);
        System.out.println(set);
    }
}
```

6. Check Anagram

```
import java.util.Arrays;
public class AnagramCheck {
    public static void main(String[] args) {
        String a = "listen", b = "silent";
        char[] ca = a.toCharArray(), cb = b.toCharArray();
        Arrays.sort(ca); Arrays.sort(cb);
        System.out.println(Arrays.equals(ca, cb) ? "Anagram" : "Not Anagram");
    }
}
```

7. Odd and Even Numbers

```
public class OddEven {
    public static void main(String[] args) {
        int[] arr = {2, 3, 4, 5, 6};
        for (int i : arr) {
            if (i % 2 == 0) System.out.println(i + " is Even");
            else System.out.println(i + " is Odd");
        }
    }
}
```

8. Remove Given Element

```
import java.util.Arrays;
public class RemoveElement {
    public static void main(String[] args) {
        int[] arr = {3, 8, 5, 2, 8};
        int val = 8;
        arr = Arrays.stream(arr).filter(x -> x != val).toArray();
        System.out.println(Arrays.toString(arr));
    }
}
```

9. Insert Element at Index

```
import java.util.Arrays;
public class InsertElement {
    public static void main(String[] args) {
        int[] arr = {10, 20, 30};
        int pos = 1, val = 15;
        int[] newArr = new int[arr.length + 1];
        for (int i = 0, j = 0; i < newArr.length; i++) {
            if (i == pos) newArr[i] = val;
            else newArr[i] = arr[j++];
```

```java
        }
        System.out.println(Arrays.toString(newArr));
    }
}
```

10. Multiply Two Matrices

```java
public class MatrixMultiply {
    public static void main(String[] args) {
        int[][] a = {{1, 2}, {3, 4}};
        int[][] b = {{2, 0}, {1, 2}};
        int[][] c = new int[2][2];
        for (int i = 0; i < 2; i++)
            for (int j = 0; j < 2; j++)
                for (int k = 0; k < 2; k++)
                    c[i][j] += a[i][k] * b[k][j];
        for (int[] row : c)
            for (int val : row)
                System.out.print(val + " ");
    }
}
```

--- CHALLENGE 3: STRING QUESTIONS ---

11. Display Only Digits

```java
public class DigitsOnly {
    public static void main(String[] args) {
        String str = "75#41*";
        str.chars().filter(Character::isDigit).forEach(c -> System.out.print((char) c));
    }
}
```

12. Convert Case without String Methods

```java
public class CaseConvert {
    public static void main(String[] args) {
        String s = "HeLLo";
        for (char c : s.toCharArray()) {
            if (c >= 'A' && c <= 'Z') System.out.print((char)(c + 32));
            else if (c >= 'a' && c <= 'z') System.out.print((char)(c - 32));
            else System.out.print(c);
        }
    }
}
```

13. Reverse Words

```java
public class ReverseWords {
    public static void main(String[] args) {
        String s = "I am a developer";
        String[] words = s.split(" ");
        for (int i = words.length - 1; i >= 0; i--)
            System.out.print(words[i] + " ");
    }
}
```

14. Repeated Characters

```
public class RepeatedChars {
    public static void main(String[] args) {
        String s = "programming";
        int[] freq = new int[256];
        for (char c : s.toCharArray()) freq[c]++;
        for (char c : s.toCharArray()) {
            if (freq[c] > 1) {
                System.out.print(c + " ");
                freq[c] = 0;
            }
        }
    }
}
```

15. Count Vowels

```
public class CountVowels {
    public static void main(String[] args) {
        String s = "Hello World";
        int count = 0;
        for (char c : s.toLowerCase().toCharArray())
            if ("aeiou".indexOf(c) != -1) count++;
        System.out.println("Vowels: " + count);
    }
}
```

16. Remove Word from String

```
public class RemoveWord {
    public static void main(String[] args) {
        String str = "Java is a powerful language";
        String word = "powerful";
        str = str.replace(word, "").replaceAll(" +", " ").trim();
        System.out.println(str);
    }
}
```

17. Remove First and Last Character

```
public class RemoveEnds {
    public static void main(String[] args) {
        String str = "Java";
        if (str.length() > 2)
            System.out.println(str.substring(1, str.length() - 1));
        else
            System.out.println("Too short to trim");
    }
}
```

--- CHALLENGE 4: COLLECTIONS ---

18. Insert in LinkedList

```
import java.util.*;
public class LinkedListInsert {
    public static void main(String[] args) {
        LinkedList<String> list = new LinkedList<>();
        list.add("One");
        list.add("Three");
        list.add(1, "Two");
        System.out.println(list);
    }
}
```

19. Check ArrayList is Empty

```
import java.util.*;
public class ArrayListEmpty {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        System.out.println(list.isEmpty() ? "Empty" : "Not Empty");
    }
}
```

20. HashSet to ArrayList

```
import java.util.*;
public class SetToList {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<>(Arrays.asList("A", "B", "C"));
        ArrayList<String> list = new ArrayList<>(set);
        System.out.println(list);
    }
}
```

21. Sort HashMap by Keys

```
import java.util.*;
public class SortHashMap {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("C", 3); map.put("A", 1); map.put("B", 2);
        TreeMap<String, Integer> sorted = new TreeMap<>(map);
        System.out.println(sorted);
    }
}
```

22. Iterate TreeMap

```
import java.util.*;
public class TreeMapIterate {
    public static void main(String[] args) {
        TreeMap<String, Integer> map = new TreeMap<>();
        map.put("X", 24); map.put("Y", 25);
        for (Map.Entry<String, Integer> entry : map.entrySet())
            System.out.println(entry.getKey() + " = " + entry.getValue());
    }
}
```

23. Sort ArrayList using Comparable & Comparator

```
import java.util.*;
class Student implements Comparable<Student> {
    int id; String name;
    Student(int id, String name) { this.id = id; this.name = name; }
    public int compareTo(Student s) { return this.id - s.id; }
}
public class SortStudent {
    public static void main(String[] args) {
        ArrayList<Student> list = new ArrayList<>();
        list.add(new Student(3, "Zara"));
        list.add(new Student(1, "Alex"));
        list.sort(Comparator.comparing(s -> s.name));
        for (Student s : list)
            System.out.println(s.id + " " + s.name);
    }
}
```