**2)**import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class SymmetricDifference {

public static List<Integer> symmetricDifference(int[] A, int[] B) {

List<Integer> diff\_A = new ArrayList<>();

List<Integer> diff\_B = new ArrayList<>();

// Знаходимо елементи, які є в A, але не в B

for (int x : A) {

if (!contains(B, x)) {

diff\_A.add(x);

}

}

// Знаходимо елементи, які є в B, але не в A

for (int x : B) {

if (!contains(A, x)) {

diff\_B.add(x);

}

}

// Об'єднуємо обидві різниці, щоб отримати симетричну різницю

List<Integer> symmetricDiff = new ArrayList<>(diff\_A);

symmetricDiff.addAll(diff\_B);

return symmetricDiff;

}

// Функція, що перевіряє, чи містить масив заданий елемент

private static boolean contains(int[] array, int element) {

for (int x : array) {

if (x == element) {

return true;

}

}

return false;

}

public static void main(String[] args) {

// Задання масивів A та B (можна ввести їх з клавіатури або вказати сталі значення)

int[] A = {1, 2, 3, 4, 5};

int[] B = {3, 4, 5, 6, 7};

// Виклик функції для знаходження симетричної різниці

List<Integer> C = symmetricDifference(A, B);

// Вивід результату

System.out.println("Масив A: " + Arrays.toString(A));

System.out.println("Масив B: " + Arrays.toString(B));

System.out.println("Симетрична різниця A \\ B ∪ B \\ A: " + C);

}

}

**3)** import java.util.Scanner;

public class SymmetricRows {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введіть розмірність матриці (n): ");

int n = scanner.nextInt();

// Ініціалізація матриці

int[][] A = new int[n][n];

System.out.println("Введіть елементи матриці:");

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

A[i][j] = scanner.nextInt();

}

}

// Перевірка симетричних послідовностей в рядках

for (int i = 0; i < n; i++) {

if (isSymmetricSequence(A[i])) {

System.out.println("Рядок " + (i + 1) + " має симетричну послідовність.");

}

}

}

// Функція для перевірки чи є послідовність симетричною

private static boolean isSymmetricSequence(int[] row) {

int length = row.length;

for (int i = 0; i < length / 2; i++) {

if (row[i] != row[length - 1 - i]) {

return false;

}

}

return true;

}

}

**4)** import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

public class RemoveWordsWithRepeatingLetters {

public static void main(String[] args) {

String text = "Your input text here"; // Замініть це речення на власний текст

// Розділити текст на слова

String[] words = text.split("[\\s.,;!?]+");

// Вилучення слів з повторенням літер

List<String> result = removeWordsWithRepeatingLetters(words);

// Виведення результату

System.out.println("Вихідний текст: " + text);

System.out.println("Текст без слів з повторенням літер: " + String.join(" ", result));

}

// Функція для вилучення слів з повторенням літер

private static List<String> removeWordsWithRepeatingLetters(String[] words) {

List<String> result = new ArrayList<>();

Set<Character> seenLetters = new HashSet<>();

for (String word : words) {

boolean hasRepeatingLetters = false;

for (char letter : word.toCharArray()) {

if (!seenLetters.add(Character.toLowerCase(letter))) {

hasRepeatingLetters = true;

break;

}

}

if (!hasRepeatingLetters) {

result.add(word);

}

seenLetters.clear();

}

return result;

}

}