TASK 1

import java.util.Arrays;

public class EvenOddArrangement {

    public static void main(String[] args) {

        int[] arr = {1, 2, 3, 5, 7, 2, 2, 7, 8, 9}; // Input array

        rearrangeEvenOdd(arr);

        System.out.println("Output: " + Arrays.toString(arr));

    }

    public static void rearrangeEvenOdd(int[] arr) {

        int left = 0;

        int right = arr.length - 1;

        while (left < right) {

            // Move the left pointer until an odd number is found

            while (arr[left] % 2 == 0 && left < right) {

                left++;

            }

            // Move the right pointer until an even number is found

            while (arr[right] % 2 != 0 && left < right) {

                right--;

            }

            // Swap the odd number from the left with the even number from the right

            if (left < right) {

                int temp = arr[left];

                arr[left] = arr[right];

                arr[right] = temp;

            }

        }

    }

}

TASK 2

import java.util.HashSet;

public class NoDuplication {

    public static void main(String[] args) {

        int[][] array2D = {

            {1, 2, 3, 4, 5},

            {6, 7, 8, 9, 10},

            {11, 12, 13, 14, 15},

            {16, 17, 18, 19, 20}

        };

        int[] array1D = new int[20];

        noDup(array2D, array1D);

        System.out.print("Output: ");

        for (int i : array1D) {

            System.out.print(i + " ");

        }

    }

    public static void noDup(int[][] array2D, int[] array1D) {

        HashSet<Integer> seen = new HashSet<>();

        int index = 0;

        for (int i = 0; i < array2D.length; i++) {

            for (int j = 0; j < array2D[i].length; j++) {

                int element = array2D[i][j];

                if (!seen.contains(element)) {

                    seen.add(element);

                    array1D[index++] = element;

                }

            }

        }

        while (index < array1D.length) {

            array1D[index++] = 0;

        }

    }

}

TASK 3

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.util.ArrayList;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class NLArray {

    public static void main(String[] args) {

        String fileName = "example.txt";

        String[] words = wordTokenize(fileName);

        System.out.println("Tokenized Words:");

        for (String word : words) {

            System.out.println(word);

        }

        String[] emails = extractEmail(fileName);

        System.out.println("Extracted Emails:");

        for (String email : emails) {

            System.out.println(email);

        }

    }

    public static String[] wordTokenize(String fileName) {

        ArrayList<String> wordList = new ArrayList<>();

        try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {

            String line;

            while ((line = br.readLine()) != null) {

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

                for (String word : words) {

                    if (!word.trim().isEmpty()) {

                        wordList.add(word);

                    }

                }

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

        return wordList.toArray(new String[0]);

    }

    public static String[] extractEmail(String fileName) {

        ArrayList<String> emailList = new ArrayList<>();

        String emailRegex = "[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,6}";

        Pattern emailPattern = Pattern.compile(emailRegex);

        try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {

            String line;

            while ((line = br.readLine()) != null) {

                Matcher matcher = emailPattern.matcher(line);

                while (matcher.find()) {

                    emailList.add(matcher.group());

                }

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

        return emailList.toArray(new String[0]);

    }

}

TASK 4

public class NLArray {

    public static void main(String[] args) {

        int[][] arr = {

            {1, 2, 3, 4, 5},

            {6, 7, 8, 9, 10},

            {11, 12, 13, 14, 15},

            {16, 17, 18, 19, 20},

            {21, 22, 23, 24, 25}

        };

        System.out.println("Extracted Boundaries:");

        extractBoundaries(arr);

        System.out.println("Cropped Center Part:");

        cropCenterPart(arr);

    }

    public static void extractBoundaries(int arr[][]) {

        int rows = arr.length;

        int cols = arr[0].length;

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

            System.out.print(arr[0][j] + " ");

        }

        System.out.println();

        for (int i = 1; i < rows - 1; i++) {

            System.out.print(arr[i][0] + " ");

            for (int j = 1; j < cols - 1; j++)

                System.out.print("  ");

            }

            System.out.print(arr[i][cols - 1] + " ");

            System.out.println();

        }

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

            System.out.print(arr[rows - 1][j] + " ");

        }

        System.out.println();

    }

    public static void cropCenterPart(int arr[][]) {

        int rows = arr.length;

        int cols = arr[0].length;

        for (int i = 1; i < rows - 1; i++) {

            for (int j = 1; j < cols - 1; j++) {

                System.out.print(arr[i][j] + " ");

            }

            System.out.println();

        }

    }

}