Name pranjal bajaj

Cms 023-21-0022

Lab 1

Task 1

import java.util.Scanner;

public class ReverseArrayCopy {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Prompt the user for the array size

        System.out.print("Enter the number of elements in the array: ");

        int arraySize = scanner.nextInt();

        // Create the first array

        int[] originalArray = new int[arraySize];

        // Get the values for the first array from the user

        System.out.println("Enter " + arraySize + " values for the array:");

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

            originalArray[i] = scanner.nextInt();

        }

        // Display the original array

        System.out.print("Original Array: ");

        displayArray(originalArray);

        // Create the reversed array

        int[] reversedArray = new int[arraySize];

        // Copy the values from the original array to the reversed array in reverse order

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

            reversedArray[i] = originalArray[arraySize - 1 - i];

        }

        // Display the reversed array

        System.out.print("Reversed Array: ");

        displayArray(reversedArray);

        // Close the scanner

        scanner.close();

    }

    // Helper method to display an array

    public static void displayArray(int[] array) {

        for (int value : array) {

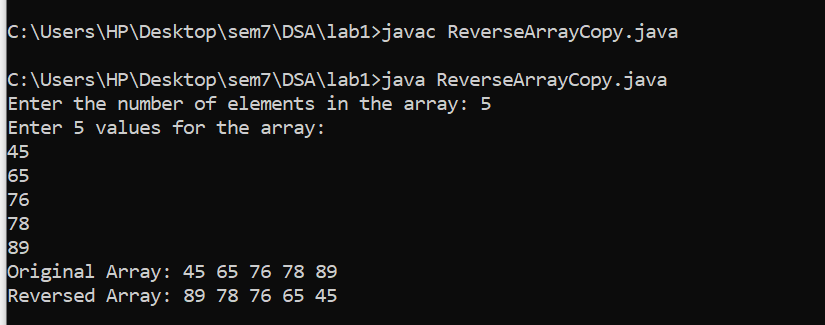
            System.out.print(value + " ");

        }

        System.out.println();

    }

}



Task 2

import java.util.Scanner;

public class Task {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Step 1: Get the size of the first array

        System.out.print("Enter the size of the first array: ");

        int size1 = scanner.nextInt();

        // Step 2: Create the first array

        int[] firstArray = new int[size1];

        // Step 3: Get the values for the first array from the user

        System.out.println("Enter " + size1 + " values for the first array:");

        for (int i = 0; i < size1; i++) {

            firstArray[i] = scanner.nextInt();

        }

        // Step 4: Get the size of the second array

        System.out.print("Enter the size of the second array: ");

        int size2 = scanner.nextInt();

        // Step 5: Create the second array

        int[] secondArray = new int[size2];

        // Step 6: Get the values for the second array from the user

        System.out.println("Enter " + size2 + " values for the second array:");

        for (int i = 0; i < size2; i++) {

            secondArray[i] = scanner.nextInt();

        }

        int[] mergedArray = new int[size1 + size2];

        int i = 0, j = 0, k = 0;

        while (i < size1 && j < size2) {

            if (firstArray[i] < secondArray[j]) {

                mergedArray[k++] = firstArray[i++];

            } else {

                mergedArray[k++] = secondArray[j++];

            }

        }

        // Step 7: Display the values of the first array

        System.out.println("Merged Array:");

        for (int value : mergedArray) {

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

        }

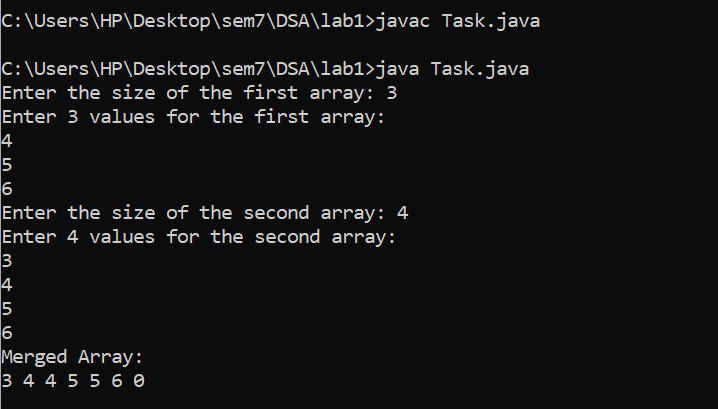
        System.out.println();

        // Close the scanner

        scanner.close();

    }

}



Task 3

import java.util.HashMap;

public class Task {

    public static void main(String[] args) {

        // Step 1: Initialize the array with hard-coded values

        int[] array = {1, 2, 3, 4, 5, 2, 3, 6, 1, 7, 8, 1};

        // Step 2: Print the array values

        System.out.println("Array values:");

        for (int value : array) {

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

        }

        System.out.println();

        // Step 3: Count duplicate values using a HashMap

        HashMap<Integer, Integer> countMap = new HashMap<>();

        for (int value : array) {

            countMap.put(value, countMap.getOrDefault(value, 0) + 1);

        }

        // Step 4: Print the duplicate values and their counts

        System.out.println("Duplicate values:");

        int duplicateCount = 0;

        for (HashMap.Entry<Integer, Integer> entry : countMap.entrySet()) {

            if (entry.getValue() > 1) {

                System.out.println("Value: " + entry.getKey() + ", Count: " + entry.getValue());

                duplicateCount++;

            }

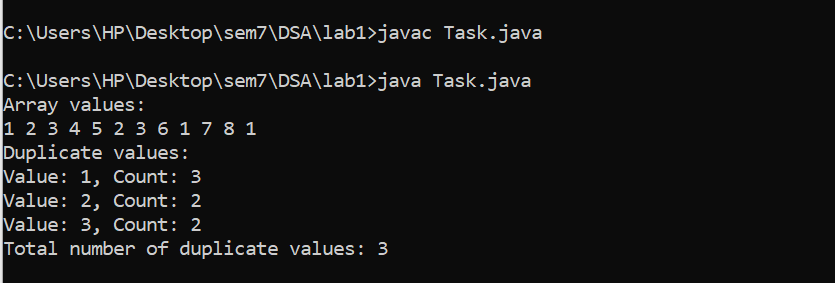
        }

        // Step 5: Print the total number of duplicate values

        System.out.println("Total number of duplicate values: " + duplicateCount);

    }

}



Task 4

import java.util.Arrays;

import java.util.HashSet;

public class Task {

    public static void main(String[] args) {

        // Step 1: Initialize the array with hard-coded values

        int[] originalArray = {1, 2, 3, 4, 5, 2, 3, 6, 1, 7, 8, 1};

        // Step 2: Print the original array

        System.out.println("Original Array: " + Arrays.toString(originalArray));

        // Step 3: Remove duplicate values using a HashSet

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

        for (int value : originalArray) {

            uniqueValues.add(value);

        }

        // Step 4: Convert the HashSet back to an array

        int[] newArray = new int[uniqueValues.size()];

        int index = 0;

        for (int value : uniqueValues) {

            newArray[index++] = value;

        }

        // Step 5: Print the new array

        System.out.println("New Array (Without Duplicates): " + Arrays.toString(newArray));

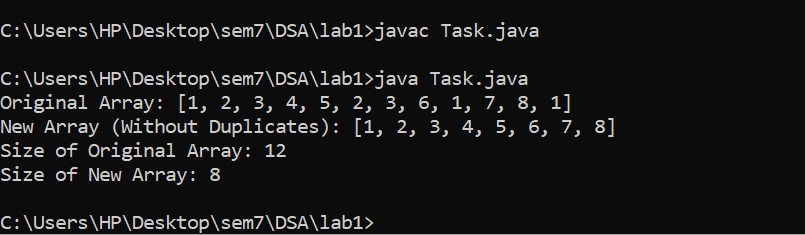
        // Step 6: Print the sizes of both arrays

        System.out.println("Size of Original Array: " + originalArray.length);

        System.out.println("Size of New Array: " + newArray.length);

    }

}



Task 5

import java.util.HashMap;

public class Task {

    public static void main(String[] args) {

        // Step 1: Initialize the array with hard-coded values

        int[] array = {1, 2, 3, 4, 5, 2, 3, 6, 1, 7, 8, 1};

        // Step 2: Print the array values

        System.out.println("Array values:");

        for (int value : array) {

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

        }

        System.out.println();

        // Step 3: Count the frequency of each element using a HashMap

        HashMap<Integer, Integer> frequencyMap = new HashMap<>();

        for (int value : array) {

            frequencyMap.put(value, frequencyMap.getOrDefault(value, 0) + 1);

        }

        // Step 4: Print the frequency of each element

        System.out.println("Frequency of each element:");

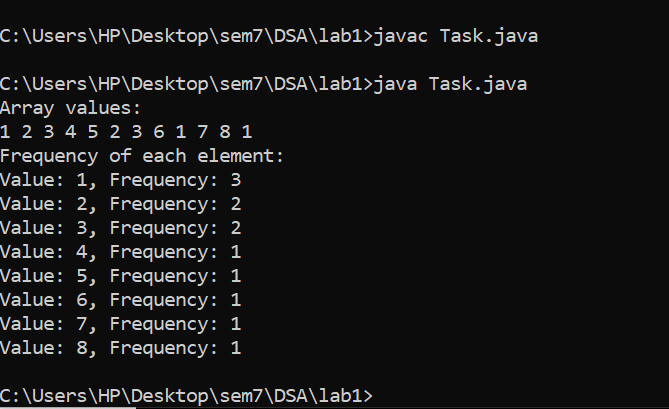
        for (HashMap.Entry<Integer, Integer> entry : frequencyMap.entrySet()) {

            System.out.println("Value: " + entry.getKey() + ", Frequency: " + entry.getValue());

        }

    }

}



Task 6

public class Task {

    public static void main(String[] args) {

        // Step 1: Initialize the 2D array with hard-coded values

        int[][] array = {

            {5, 5, 5, 5},

            {5, 5, 5, 5},

            {5, 5, 6, 10},

            {5, 5, 5, 5}

        };

        // Step 2: Print the original array

        System.out.println("Original Array:");

        for (int[] row : array) {

            for (int value : row) {

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

            }

            System.out.println();

        }

        // Step 3: Check if the array is row-magic

        int rowSum = calculateRowSum(array[0]);

        boolean isRowMagic = true;

        for (int[] row : array) {

            if (calculateRowSum(row) != rowSum) {

                isRowMagic = false;

                break;

            }

        }

        // Step 4: Print the result

        if (isRowMagic) {

            System.out.println("Row Magic");

        } else {

            System.out.println("Array is not Row-Magic");

        }

    }

    // Helper method to calculate the sum of a row

    private static int calculateRowSum(int[] row) {

        int sum = 0;

        for (int value : row) {

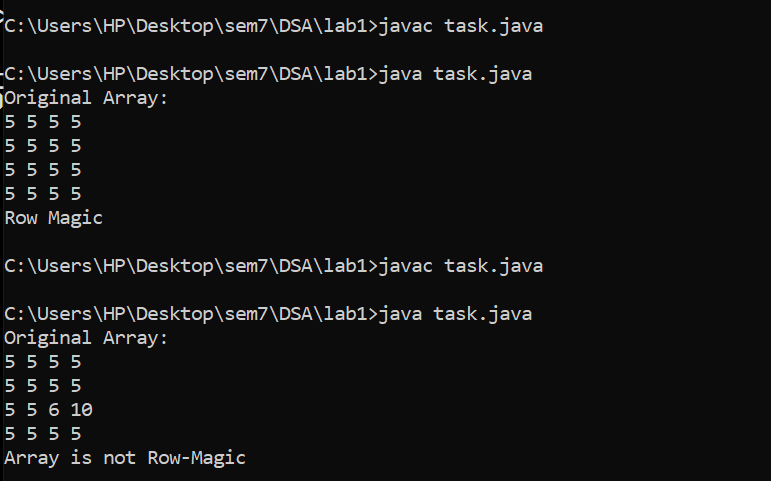
            sum += value;

        }

        return sum;

    }

}



Task 7

public class Task {

    public static void main(String[] args) {

        // Step 1: Initialize the 2D array with hard-coded values

        int[][] array = {

            {5, 5, 5, 5},

            {5, 5, 5, 5},

            {5, 5, 5, 5},

            {5, 5, 5, 5}

        };

        // Step 2: Print the original array

        System.out.println("Original Array:");

        for (int[] column : array) {

            for (int value : column) {

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

            }

            System.out.println();

        }

        // Step 3: Check if the array is row-magic

        int colSum = calculateRowSum(array[0]);

        boolean isColumnMagic = true;

        for (int[] column : array) {

            if (calculateRowSum(column) != colSum) {

                isColumnMagic = false;

                break;

            }

        }

        // Step 4: Print the result

        if (isColumnMagic) {

            System.out.println("Cloumn Magic");

        } else {

            System.out.println("Array is not Column-Magic");

        }

    }

    // Helper method to calculate the sum of a row

    private static int calculateRowSum(int[] column) {

        int sum = 0;

        for (int value : column) {

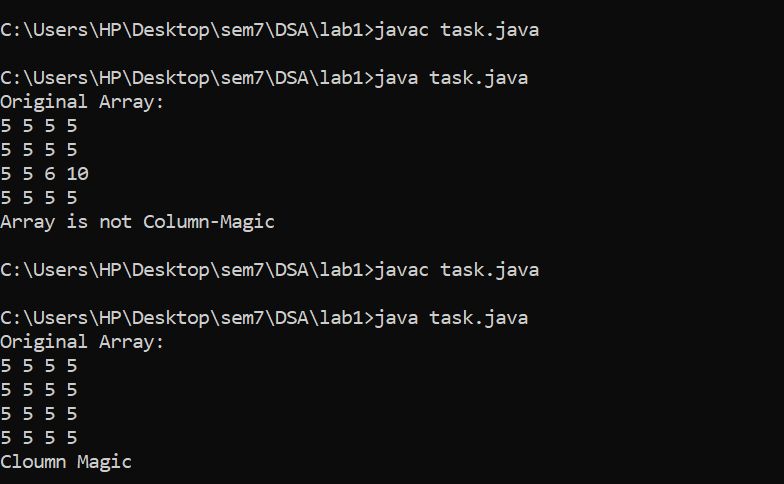
            sum += value;

        }

        return sum;

    }

}



Task 8

public class Task {

    public static void main(String[] args) {

        // Step 1: Initialize the 2D array with hard-coded values

        int[][] array = {

            {3, 7, 2},

            {8, 4, 1},

            {5, 9, 6}

        };

        // Step 2: Print the original array

        System.out.println("Original array values are:");

        printArray(array);

        // Step 3: Flip the array horizontally

        flipArrayHorizontally(array);

        // Step 4: Print the flipped array

        System.out.println("Horizontally flipped array is:");

        printArray(array);

    }

    // Helper method to print a 2D array

    private static void printArray(int[][] array) {

        for (int[] row : array) {

            for (int value : row) {

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

            }

            System.out.println();

        }

    }

    // Helper method to flip a 2D array horizontally

    private static void flipArrayHorizontally(int[][] array) {

        for (int[] row : array) {

            int left = 0;

            int right = row.length - 1;

            while (left < right) {

                int temp = row[left];

                row[left] = row[right];

                row[right] = temp;

                left++;

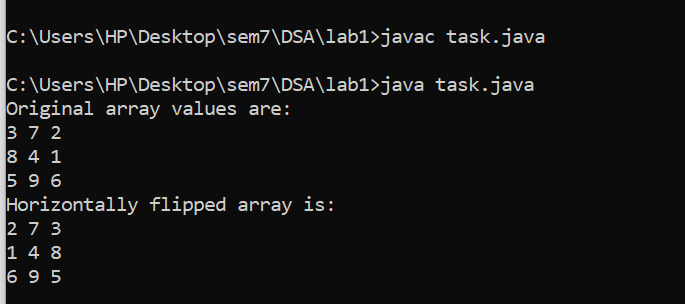
                right--;

            }

        }

    }

}



Task 9

public class Task {

    public static void main(String[] args) {

        // Step 1: Initialize the 2D array with hard-coded values

        int[][] array = {

            {3, 7, 2},

            {8, 4, 1},

            {5, 9, 6}

        };

        // Step 2: Print the original array

        System.out.println("Original array values are:");

        printArray(array);

        // Step 3: Flip the array vertically

        flipArrayVertically(array);

        // Step 4: Print the flipped array

        System.out.println("Vertically flipped array is:");

        printArray(array);

    }

    // Helper method to print a 2D array

    private static void printArray(int[][] array) {

        for (int[] row : array) {

            for (int value : row) {

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

            }

            System.out.println();

        }

    }

    // Helper method to flip a 2D array vertically

    private static void flipArrayVertically(int[][] array) {

        int top = 0;

        int bottom = array.length - 1;

        while (top < bottom) {

            int[] temp = array[top];

            array[top] = array[bottom];

            array[bottom] = temp;

            top++;

            bottom--;

        }

    }

}

