***LAB-04***

***DATA STRUCTURES & ALGORITHM***

**TASKS:**

1. package lab4;

public class ArraySwapper {

public static void main(String[] args) {

int[] qasim1 = {5, 4, 3, 1};

int[] qasim2 = {5, 6, 7, 8};

// Swapping elements

for (int i = 0; i < qasim1.length; i++) {

int temp = qasim1[i];

qasim1[i] = qasim2[i];

qasim2[i] = temp;

}

// Displaying swapped arrays

System.out.println("Qasim 1 after swap: ");

for (int num : qasim1) {

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

}

System.out.println("\nQasim 2 after swap: ");

for (int num : qasim2) {

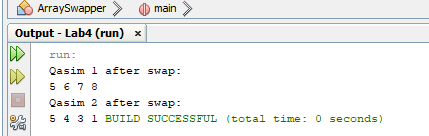
System.out.print(num + " ");

}

}

}

**Output:**



2. package lab4;

import java.util.Arrays;

public class ArrayMerge {

// Existing array

private int[] existingArray;

// Constructor to initialize the existing array

public ArrayMerge(int[] existingArray) {

this.existingArray = existingArray;

}

// Method to merge a new array with the existing array

public int[] mergeArrays(int[] newArray) {

// Calculate the size of the merged array

int[] mergedArray = new int[existingArray.length + newArray.length];

// Copy existing array elements to merged array

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

mergedArray[i] = existingArray[i]; // Copy each element

}

// Copy new array elements to merged array

for (int j = 0; j < newArray.length; j++) {

mergedArray[existingArray.length + j] = newArray[j]; // Append new elements

}

return mergedArray; // Return the merged array

}

// Method to print an array with a label

public void printArray(String label, int[] array) {

System.out.print(label + ": "); // Print the label

for (int value : array) {

System.out.print(value + " "); // Print each value

}

System.out.println(); // New line after printing the array

}

// Main method to demonstrate merging of arrays

public static void main(String[] args) {

// Initialize the existing array

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

// Create an instance of ArrayMerge

ArrayMerge arrayMerger = new ArrayMerge(existingArray);

// Initialize a new array to merge

int[] newArray = {6, 7, 8};

// Print both arrays

arrayMerger.printArray("Existing Array", existingArray);

arrayMerger.printArray("New Array", newArray);

// Merge the arrays and get the result

int[] mergedArray = arrayMerger.mergeArrays(newArray);

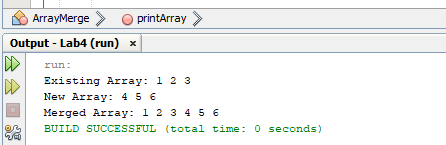
// Print the merged array

arrayMerger.printArray("Merged Array", mergedArray);

}

}

**Output:**



3. package lab4;

public class PalindromeChecker {

public static void main(String[] args) {

String[] qasim = {"madam", "hello", "racecar", "world"};

for (String str : qasim) {

if (isPalindrome(str)) {

System.out.println(str + " is a palindrome.");

} else {

System.out.println(str + " is not a palindrome.");

}

}

}

// Method to check if a string is a palindrome

public static boolean isPalindrome(String str) {

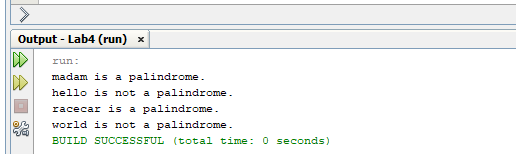
String reversed = new StringBuilder(str).reverse().toString();

return str.equals(reversed); // Check if the string is equal to its reverse

}

}

**Output:**



4. package lab4;

public class EvenOddCounter {

public static void main(String[] args) {

int[] qasim = {1, 3, 4, 5, 6, 7, 8, 9, 10}; // Sample array of integers

int evenCount = 0; // Counter for even numbers

int oddCount = 0; // Counter for odd numbers

// Loop through the array to count even and odd numbers

for (int number : qasim) {

if (number % 2 == 0) {

evenCount++; // Increment even counter

} else {

oddCount++; // Increment odd counter

}

}

// Output the results

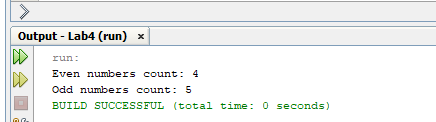
System.out.println("Even numbers count: " + evenCount);

System.out.println("Odd numbers count: " + oddCount);

}

}

**Output:**



5. package lab4;

import java.util.Arrays;

import java.util.HashSet;

public class ArraysMerger {

public static void main(String[] args) {

// Define two integer arrays

int[] qasim1 = {1, 2, 3, 4, 5};

int[] qasim2 = {4, 5, 6, 7};

// Merge the arrays

int[] mergedArray = mergeAndRemoveDuplicates(qasim1, qasim2);

// Print the resulting array

System.out.println(Arrays.toString(mergedArray));

}

// Method to merge two arrays and remove duplicates

public static int[] mergeAndRemoveDuplicates(int[] array1, int[] array2) {

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

// Add elements from the first array

for (int num : array1) {

set.add(num);

}

// Add elements from the second array

for (int num : array2) {

set.add(num);

}

// Convert the set back to an array

int[] result = new int[set.size()];

int index = 0;

for (int num : set) {

result[index++] = num;

}

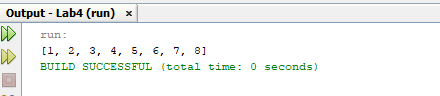
// Return the merged array without duplicates

return result;

}

}

**Output:**



**HOME-TASKS:**

1.