LAB 04

ARRAYS IN JAVA

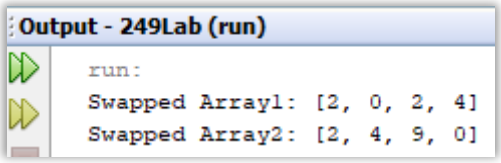
Lab Tasks

1. Write a program that takes two arrays of size 4 and swap the elements of those arrays e a program which takes an integer value (k) as input and prints the sequence of numbers from k to 0 in descending order.

Source Code :

```
public class Main {
   public static void main(String[] args){
       int[] array1={2,4,9,0};
       int[] array2={2,0,2,4};
    swapArrays(array1,array2);
        System.out.println("Swapped Array1: " + Arrays.toString(array1));
        System.out.println("Swapped Array2: " + Arrays.toString(array2));
   }public static void swapArrays(int[] arr1, int[] arr2) {
        for (int i = 0; i < arr1.length; i++) {
            int temp = arr1[i];
            arr1[i] = arr2[i];
            arr2[i] = temp;
        }
   }
}
```

Output :



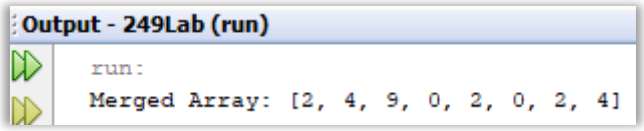
2. Add a method in the class that takes array and merge it with the existing one.

Source Code :

```
public class Main {
   public static void main(String[] args){
       int[] array1={2,4,9,0};
       int[] array2={2,0,2,4};
 int[] mergedArray = mergeArrays(array1, array2);
        System.out.println("Merged Array: " +
Arrays.toString(mergedArray));
   }
 public static int[] mergeArrays(int[] arr1, int[] arr2) {
        int[] merged = new int[arr1.length + arr2.length];
        System.arraycopy(arr1, 0, merged, 0, arr1.length);
```

```
        System.arraycopy(arr2, 0, merged, arr1.length, arr2.length);
        return merged;
    }
}
```

Output :

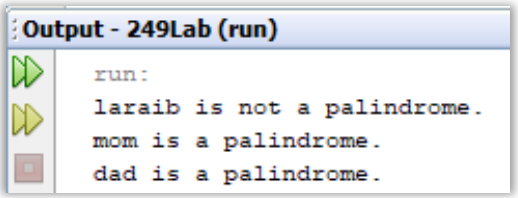


3. In a JAVA program, take an array of type string and then check whether the strings are palindrome or not.

Source Code :

```
public class Main {
   public static void main(String[] args){
        String[] strArray = {"laraib", "mom", "dad"};
        checkPalindromes(strArray);
   }
 public static void checkPalindromes(String[] arr) {
        for (String str : arr) {
            String reversed = new StringBuilder(str).reverse().toString();
            if (str.equals(reversed)) {
                System.out.println(str + " is a palindrome.");
            } else {
                System.out.println(str + " is not a palindrome.");
            }
        }
    }
}
```

Output :



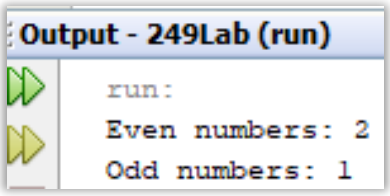
4. Given an array of integers, count how many numbers are even and how many are odd.

Source Code :

```
public class Main {
   public static void main(String[] args){
        int[] numArray = {2, 4, 9};
        countEvenOdd(numArray);
}
```

```
    public static void countEvenOdd(int[] arr) {
        int evenCount = 0, oddCount = 0;
        for (int num : arr) {
            if (num % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }
        }System.out.println("Even numbers: " + evenCount);
        System.out.println("Odd numbers: " + oddCount);
    }
}
```

Output :

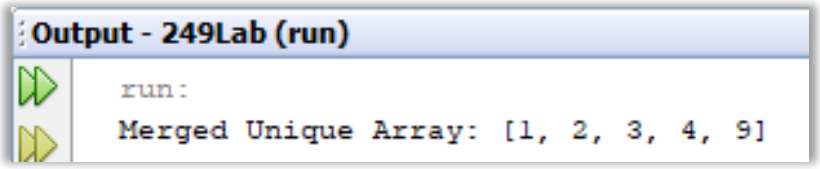


5. Given two integer arrays, merge them and remove any duplicate values from the resulting array.

Source Code :

```
import java.util.HashSet;
public class Main {
   public static void main(String[] args){
       int[] array1 = {2, 4, 9};
        int[] array2 = {1, 2, 3};
        int[] mergedUnique = mergeAndRemoveDuplicates(array1, array2);
        System.out.println("Merged Unique Array: " +
Arrays.toString(mergedUnique));
    }
      public static int[] mergeAndRemoveDuplicates(int[] arr1, int[]
arr2) {
        HashSet<Integer> set = new HashSet<>();
        for (int num : arr1) set.add(num);
        for (int num : arr2) set.add(num);
        int[] result = new int[set.size()];
        int i = 0;
        for (int num : set) result[i++] = num;
        return result;
    }
  }
```

Output :



Home Tasks

1. Write a program that takes an array of Real numbers having size 7 and calculate the sum and mean of all the elements. Also depict the memory management of this task.

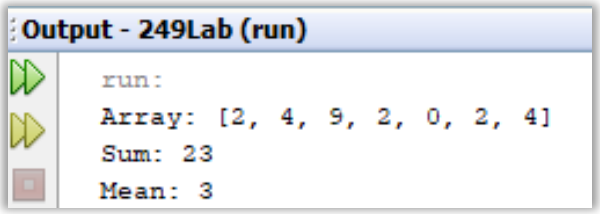
Source Code :

```
public class Main {
    public static void main(String[] args) {
        int[] numbers = {2,4,9,2,0,2,4};

        int sum = calculateSum(numbers);
        int mean = calculateMean(sum, numbers.length);

        System.out.println("Array: " +
java.util.Arrays.toString(numbers));
        System.out.println("Sum: " + sum);
        System.out.println("Mean: " + mean);
    }
    public static int calculateSum(int[] numbers) {
        int sum = 0;
        for (int number : numbers) {
            sum += number;
        }
        return sum;
    }
    public static int calculateMean(int sum, int n) {
        return sum / n;
    }
}
```

Output :



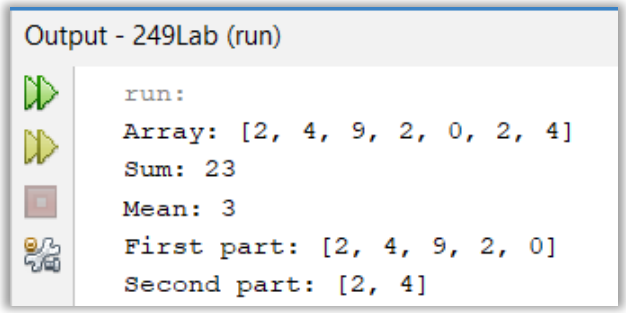
2. Add a method in the same class that splits the existing array into two. The method should search a key in array and if found splits the array from that index of the key.

Source Code :

```
public class lab249 {
    public static void main(String[] args) {
        int[] numbers = {2, 4, 9, 2, 0, 2, 4};
        int sum = calculateSum(numbers);
        int mean = sum / numbers.length;
        System.out.println("Array: " + Arrays.toString(numbers));
        System.out.println("Sum: " + sum);
        System.out.println("Mean: " + mean);
        int key = 0;
```

```
        int[][] splitArrays = splitArray(numbers, key);
        System.out.println("First part: " +
Arrays.toString(splitArrays[0]));
        System.out.println("Second part: " +
Arrays.toString(splitArrays[1]));
    }
    public static int calculateSum(int[] numbers) {
        int sum = 0;
        for (int num : numbers) sum += num;
        return sum;
    }
    public static int[][] splitArray(int[] numbers, int key) {
        int index = -1;
        for (int i = 0; i < numbers.length; i++) {
            if (numbers[i] == key) {
                index = i;
                break;
            }
        }
        if (index == -1) return new int[][] {numbers, new int[0]};
        return new int[][] {Arrays.copyOfRange(numbers, 0, index + 1),
Arrays.copyOfRange(numbers, index + 1, numbers.length)};
    }
}
```

Output :



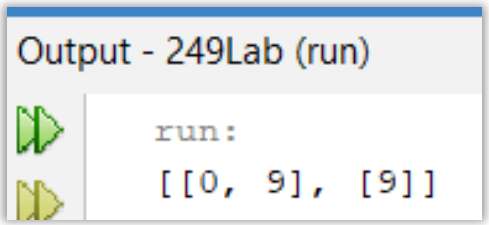
3. Given an array of distinct integers and a target integer, return all unique combinations of numbers that add up to the target. Each number can be used only once in the combination.

Source Code :

```
public class lab249 {
    public static void main(String[] args) {
        int[] candidates = {0,2,4,9};
        int target = 9;
        List<List<Integer>> result = combinationSum(candidates, target);
        System.out.println(result);
    }
  public static List<List<Integer>> combinationSum(int[] candidates, int target){
        List<List<Integer>> result = new ArrayList<>();
        backtrack(candidates, target, 0, new ArrayList<>(), result);
        return result;
    }
    private static void backtrack(int[] candidates, int target, int start,
```

```
List<Integer> current, List<List<Integer>> result) {
        if (target == 0) {
            result.add(new ArrayList<>(current));
            return;
        }
        for (int i = start; i < candidates.length; i++) {
            if (candidates[i] > target) continue;
            current.add(candidates[i]);
            backtrack(candidates, target- candidates[i], i + 1,current, result);
            current.remove(current.size() - 1);
        }
    }
}
```

Output :

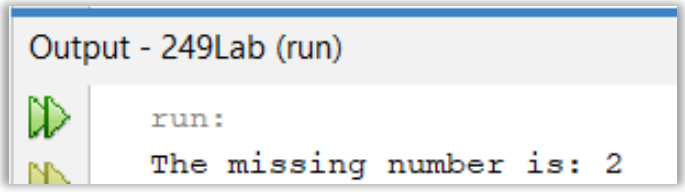


4. You are given an array containing n distinct numbers taken from 0, 1, 2, ..., n. Write a program to find the one number that is missing from the array.

Source Code :

```
public class lab249 {
    public static void main(String[] args) {
        int[] nums = {0, 1, 3, 4, 5};
        System.out.println("The missing number is: " +
findMissingNumber(nums));
    }
    public static int findMissingNumber(int[] nums) {
        int n = nums.length;
        int totalSum = n * (n + 1) / 2;
        int arraySum = 0;
        for (int num : nums) {
            arraySum += num;
        }
        return totalSum - arraySum;
    }
}
```

Output :



5. You are given an array of integers. Write a program to sort the array such that it follows a zigzag pattern: the first element is less than the second, the second is greater than the third, and so on.

Source Code :

```
public class lab249 {
    public static void lab249(int[] arr) {
        Arrays.sort(arr);
        for (int i = 1; i < arr.length; i += 2) {
            if (i + 1 < arr.length && arr[i] < arr[i + 1]) {
                int temp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = temp;
            }
        }
    }
    public static void printArray(int[] arr) {
        for (int num : arr) {
            System.out.print(num + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        int[] arr = {4, 3, 7, 8, 6, 2, 1};
        System.out.println("Original array:");
        printArray(arr);
        lab249(arr);
        System.out.println("Zigzag sorted array:");
        printArray(arr);
    }
}
```

Output :