1.

public class MergeArrays {

    public static void main(String[] args) {

        int[] X = { 0, 2, 0, 3, 0, 5, 6, 0, 0 };

        int[] Y = { 1, 8, 9, 10, 15 };

        mergeArrays(X, Y);

        for (int num : X) {

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

        }

    }

    public static void mergeArrays(int[] X, int[] Y) {

        int m = X.length;

        int n = Y.length;

        int i = m - 1;

        int j = n - 1;

        while (i >= 0 && j >= 0) {

            if (X[i] > Y[j]) {

                X[i--] = X[i];

            } else {

                X[i--] = Y[j--];

            }

        }

        while (j >= 0) {

            X[i--] = Y[j--];

        }

    }

}

Output:

**0 2 0 1 3 8 9 10 15**

**3.**

import java.util.HashMap;

public class StringCount {

    public static void main(String[] args) {

        String str = "Good Morning Everybody.";

        int wordCount = countWords(str);

        System.out.println("Number of words in the string: " + wordCount);

    }

    public static int countWords(String str) {

        HashMap<String, Integer> wordCounts = new HashMap<> ();

        for (String word: str.split("\\s+")) {

            wordCounts.put(word, wordCounts.getOrDefault(word, 0) + 1);

        }

        return wordCounts.size();

    }

}

**OUTPUT:**

Number of words in the string: 3

4.

import java.util.HashSet;

import java.util.Set;

public class DuplicateChar {

    public static void main(String[] args) {

        String str = "good morning everybody";

        Set<Character> duplicates = findDuplicates(str);

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

        for (char ch : duplicates) {

            System.out.println(ch);

        }

    }

    public static Set<Character> findDuplicates(String str) {

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

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

        for (char ch : str.toCharArray()) {

            if (!seenCharacters.add(ch)) {

                duplicates.add(ch);

            }

        }

        return duplicates;

    }

}

**OUTPUT:**

**Duplicate characters:**

**r**

**d**

**e**

**g**

**y**

**n**

**o**