import java.util.Arrays;

public class Quicksort {

    public static void quickSort(int[] arr, int low, int high) {

        if (low < high) {

            // Partitioning index

            int pivotIndex = partition(arr, low, high);

            // Recursively sort the two halves

            quickSort(arr, low, pivotIndex - 1);  // Left side of pivot

            quickSort(arr, pivotIndex + 1, high); // Right side of pivot

        }

    }

    private static int partition(int[] arr, int low, int high) {

        int pivot = arr[high];

        int i = low - 1;  // Index of smaller element

        for (int j = low; j < high; j++) {

            if (arr[j] <= pivot) {

                i++;  // Increment the index of smaller element

                int temp = arr[i];

                arr[i] = arr[j];

                arr[j] = temp;

            }

        }

        int temp = arr[i + 1];

        arr[i + 1] = arr[high];

        arr[high] = temp;

        return i + 1;

    }

    public static void main(String[] args) {

        int[] arr = {3, 6, 8, 10, 1, 2, 1};

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

        quickSort(arr, 0, arr.length - 1);

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

    }

}