arrShiftSortMerge.java

```java
// You are given two integer arrays nums1 and nums2, sorted in non-decreasing order, and two integers m and n,
// representing the number of elements in nums1 and nums2 respectively.
// Merge nums1 and nums2 into a single array sorted in non-decreasing order.
// The final sorted array should not be returned by the function, but instead be stored inside the array nums1.
// To accommodate this, nums1 has a length of m + n, where the first m elements denote the elements that should
// be merged, and the last n elements are set to 0 and should be ignored. nums2 has a length of n.
import java.util.Scanner;
public class arrShiftSortMerge {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter size of arr1");
        int n1 = sc.nextInt();
        System.out.println("Enter " + n1 + " elements");
        int[] arr1 = new int[n1];
        for (int i = 0; i < n1; i++)
            arr1[i] = sc.nextInt();
        System.out.println("Enter size of arr2");
        int n2 = sc.nextInt();
        System.out.println("Enter " + n2 + " elements");
        int[] arr2 = new int[n2];
        for (int i = 0; i < n2; i++)
            arr2[i] = sc.nextInt();
        // Create new array tmp with size n1 + n2 and filled with elements of arr1 and zeroes
        int[] tmp = new int[n1+n2];
        for(int i = 0; i < n1+n2; i++) {
            if (i < n1)
                tmp[i] = arr1[i];
            else
                tmp[i] = 0;
        }
        merge(tmp, n1, arr2, n2);
    }

    public static void merge(int[] arr1, int n1, int[] arr2, int n2) {
        int i = n1 - 1;
        int j = n2 - 1;
        int k = n1 + n2 - 1;
        while (j >= 0) {
            if (i >= 0 && arr1[i] > arr2[j])
                arr1[k--] = arr1[i--];
            else
                arr1[k--] = arr2[j--];
        }
        for (i = 0; i < arr1.length; i++)
            System.out.print(arr1[i] + " ");
    }
}
```