DESIGN ANALYSIS AND ALGORITHMS

LAB EXPERIMENT 2

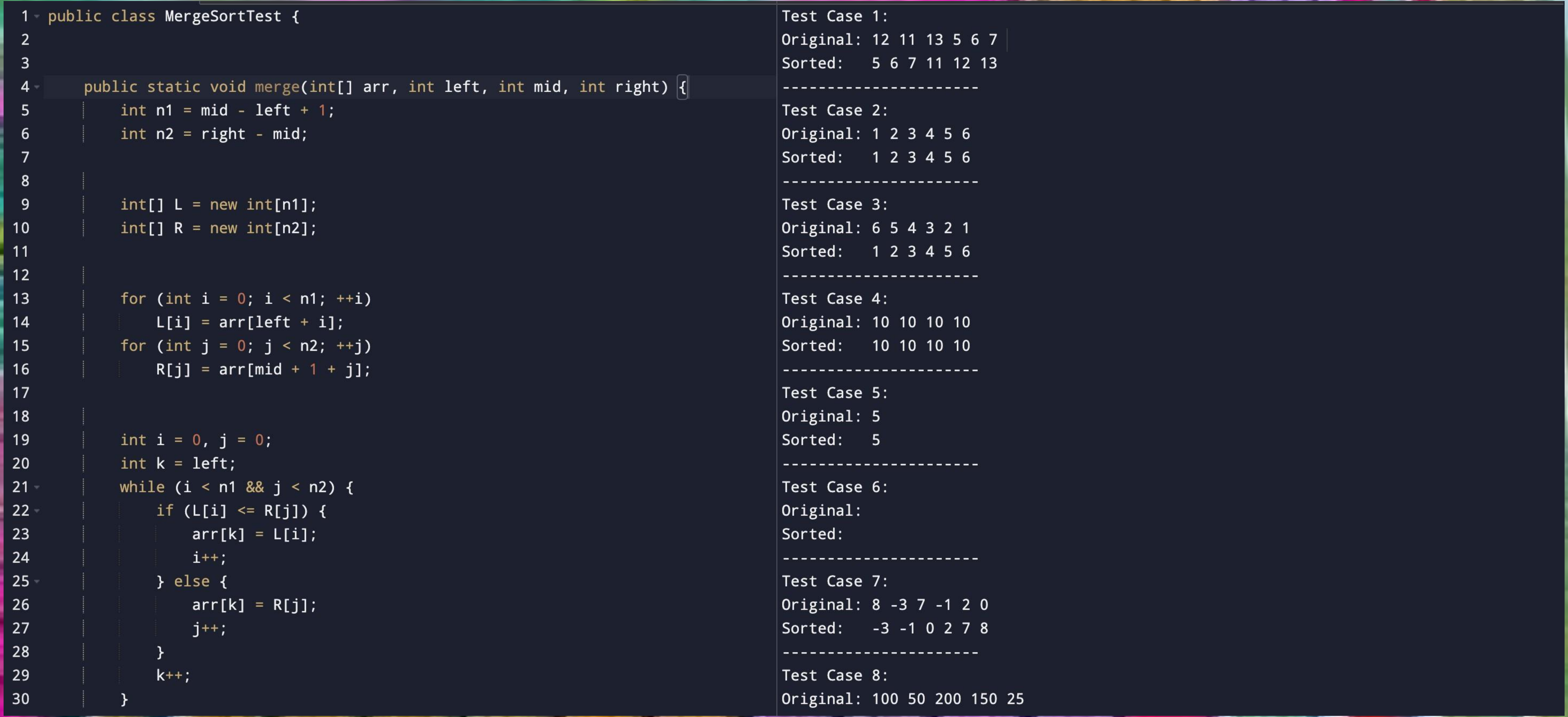
NAME: ARYAN MAAN
SAPID: 590015512
BATCH: 34
SUBMITTED TO : MR. ARYAN GUPTA

CODE:

```
public class MergeSortTest {
   public static void merge(int[] arr, int left, int mid, int right) {
        int n1 = mid - left + 1;
        int n2 = right - mid;
        int[] L = new int[n1];
        int[] R = new int[n2];

        for (int i = 0; i < n1; ++i)
            L[i] = arr[left + i];
        for (int j = 0; j < n2; ++j)
            R[j] = arr[mid + 1 + j];

        int i = 0, j = 0;
        int k = left;
        while (i < n1 && j < n2) {
            if (L[i] <= R[j]) {
                arr[k] = L[i];
                i++;
            } else {
                arr[k] = R[j];
                j++;
            }
            k++;
        }
while (i < n1) {
            arr[k] = L[i];
            i++;
            k++;
        }

        while (j < n2) {
            arr[k] = R[j];
            j++;
            k++;
        }
    }

    public static void mergeSort(int[] arr, int left, int right) {
        if (left < right) {
            int mid = (left + right) / 2;

            mergeSort(arr, left, mid);
            mergeSort(arr, mid + 1, right);

            merge(arr, left, mid, right);
        }
    }
```

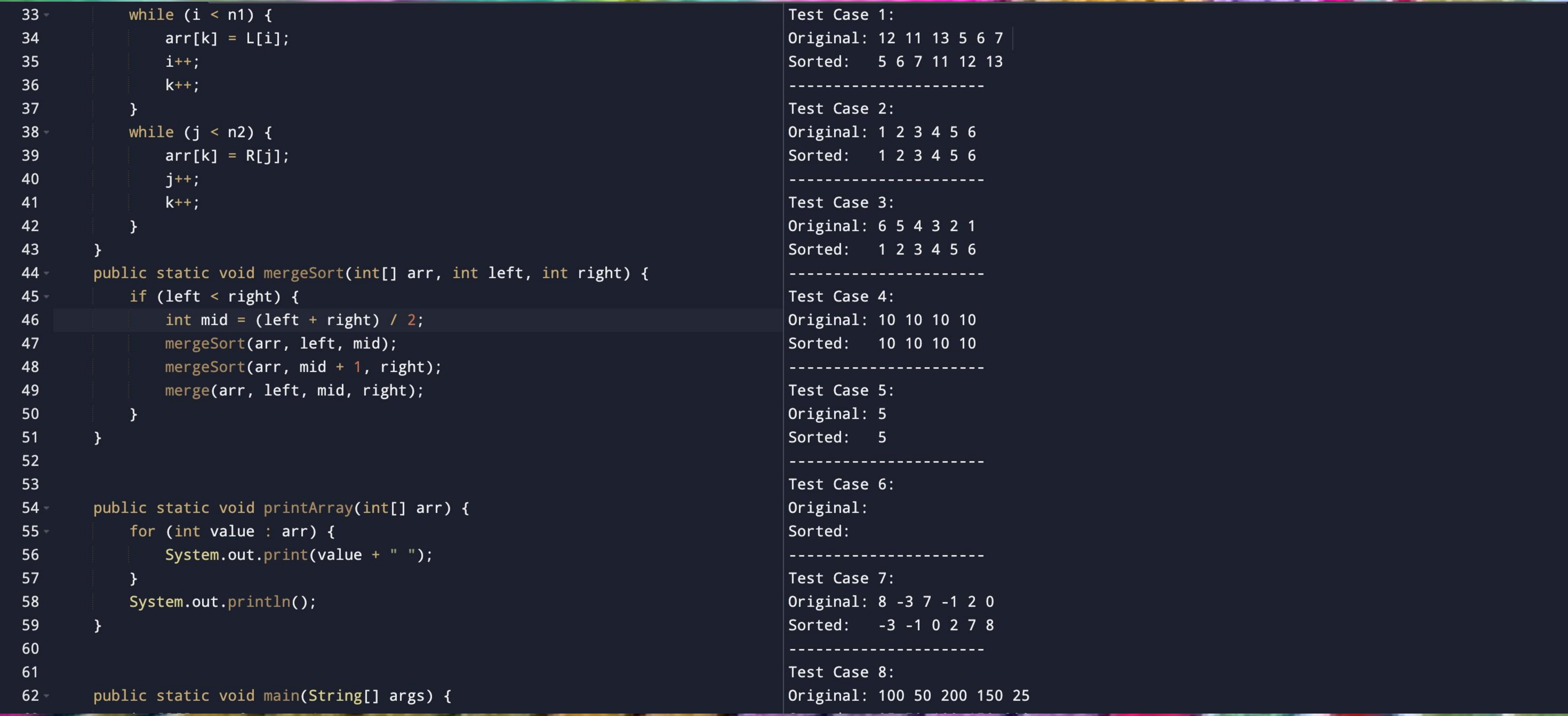
```
public static void printArray(int[] arr) {
        for (int value : arr) {
            System.out.print(value + " ");
        }
        System.out.println();
    }


    public static void main(String[] args) {
        int[][] testCases = {
            {12, 11, 13, 5, 6, 7},
            {1, 2, 3, 4, 5, 6},
            {6, 5, 4, 3, 2, 1},
            {10, 10, 10, 10},
            {5},
            {},
            {8, -3, 7, -1, 2, 0},
            {100, 50, 200, 150, 25},
            {3, 3, 2, 1, 2, 1},
            {9, 1, 8, 2, 7, 3, 6, 4, 5}
        };

        for (int i = 0; i < testCases.length; i++) {
            System.out.println("Test Case " + (i + 1) + ":");
            int[] arr = testCases[i];
            System.out.print("Original: ");
            printArray(arr);

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

            System.out.print("Sorted:   ");
            printArray(arr);
            System.out.println("-------------------------");
        }
    }
}
```

```
OUTPUT:
Test Case 1:
Original: 12 11 13 5 6 7
Sorted:   5 6 7 11 12 13
----------------------
Test Case 2:
Original: 1 2 3 4 5 6
Sorted:   1 2 3 4 5 6
----------------------
Test Case 3:
Original: 6 5 4 3 2 1
Sorted:   1 2 3 4 5 6
----------------------
Test Case 4:
Original: 10 10 10 10
Sorted:   10 10 10 10
----------------------
Test Case 5:
Original: 5
Sorted:   5
----------------------
Test Case 6:
Original:
Sorted:
----------------------
Test Case 7:
Original: 8 -3 7 -1 2 0
Sorted:   -3 -1 0 2 7 8
----------------------
Test Case 8:
Original: 100 50 200 150 25
Sorted:   25 50 100 150 200
----------------------
Test Case 9:
Original: 3 3 2 1 2 1
Sorted:   1 1 2 2 3 3
----------------------
Test Case 10:
Original: 9 1 8 2 7 3 6 4 5
Sorted:   1 2 3 4 5 6 7 8 9
----------------------
```



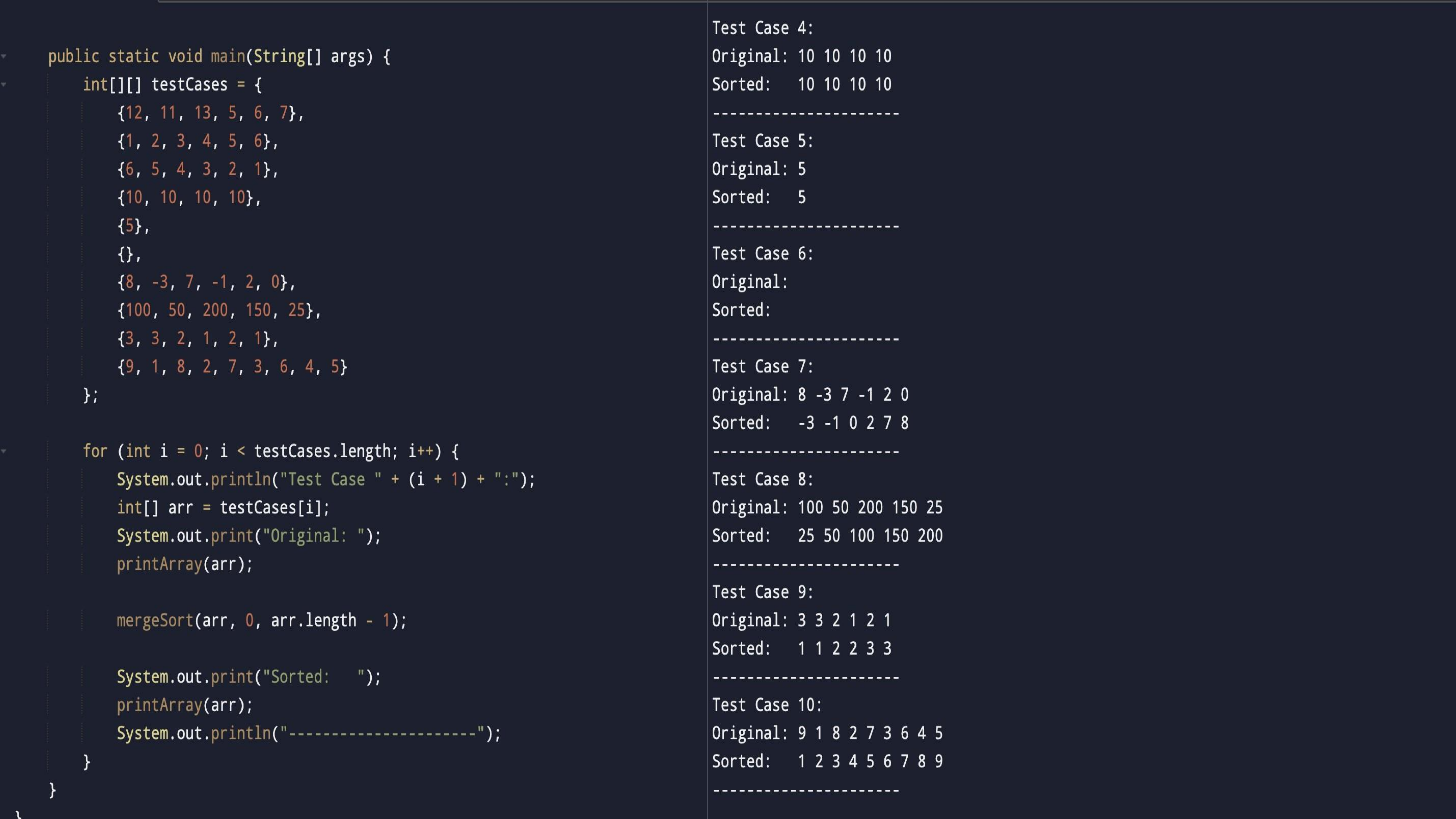
```java
public class MergeSortTest {


    public static void merge(int[] arr, int left, int mid, int right) {
        int n1 = mid - left + 1;
        int n2 = right - mid;


        int[] L = new int[n1];
        int[] R = new int[n2];


        for (int i = 0; i < n1; ++i)
            L[i] = arr[left + i];
        for (int j = 0; j < n2; ++j)
            R[j] = arr[mid + 1 + j];


        int i = 0, j = 0;
        int k = left;
        while (i < n1 && j < n2) {
            if (L[i] <= R[j]) {
                arr[k] = L[i];
                i++;
            } else {
                arr[k] = R[j];
                j++;
            }
            k++;
        }
```

```
Test Case 1:
Original: 12 11 13 5 6 7
Sorted:   5 6 7 11 12 13
----------------------
Test Case 2:
Original: 1 2 3 4 5 6
Sorted:   1 2 3 4 5 6
----------------------
Test Case 3:
Original: 6 5 4 3 2 1
Sorted:   1 2 3 4 5 6
----------------------
Test Case 4:
Original: 10 10 10 10
Sorted:   10 10 10 10
----------------------
Test Case 5:
Original: 5
Sorted:   5
----------------------
Test Case 6:
Original:
Sorted:
----------------------
Test Case 7:
Original: 8 -3 7 -1 2 0
Sorted:   -3 -1 0 2 7 8
----------------------
Test Case 8:
Original: 100 50 200 150 25
```



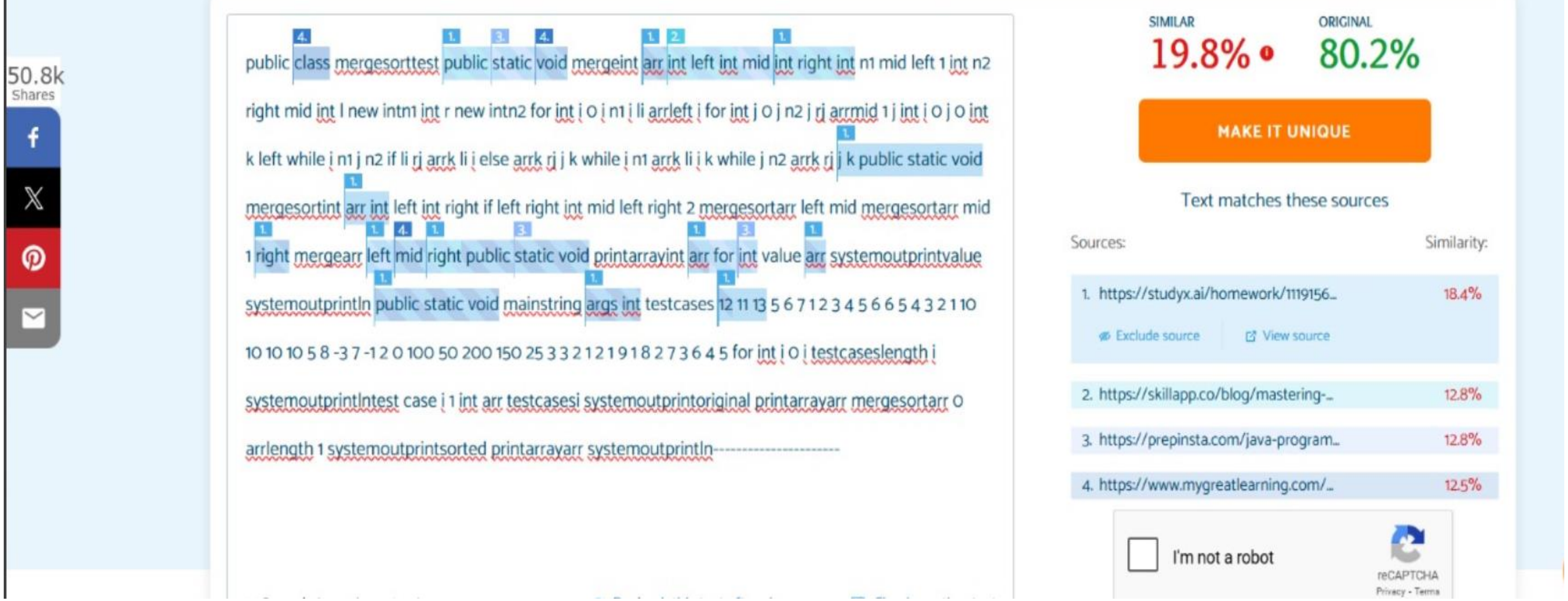
```java
        while (i < n1) {
            arr[k] = L[i];
            i++;
            k++;
        }
        while (j < n2) {
            arr[k] = R[j];
            j++;
            k++;
        }
    }
    public static void mergeSort(int[] arr, int left, int right) {
        if (left < right) {
            int mid = (left + right) / 2;
            mergeSort(arr, left, mid);
            mergeSort(arr, mid + 1, right);
            merge(arr, left, mid, right);
        }
    }


    public static void printArray(int[] arr) {
        for (int value : arr) {
            System.out.print(value + " ");
        }
        System.out.println();
    }


    public static void main(String[] args) {
```

```
Test Case 1:
Original: 12 11 13 5 6 7
Sorted:   5 6 7 11 12 13
----------------------
Test Case 2:
Original: 1 2 3 4 5 6
Sorted:   1 2 3 4 5 6
----------------------
Test Case 3:
Original: 6 5 4 3 2 1
Sorted:   1 2 3 4 5 6
----------------------
Test Case 4:
Original: 10 10 10 10
Sorted:   10 10 10 10
----------------------
Test Case 5:
Original: 5
Sorted:   5
----------------------
Test Case 6:
Original:
Sorted:
----------------------
Test Case 7:
Original: 8 -3 7 -1 2 0
Sorted:   -3 -1 0 2 7 8
----------------------
Test Case 8:
Original: 100 50 200 150 25
```



```java
    public static void main(String[] args) {
        int[][] testCases = {
            {12, 11, 13, 5, 6, 7},
            {1, 2, 3, 4, 5, 6},
            {6, 5, 4, 3, 2, 1},
            {10, 10, 10, 10},
            {5},
            {},
            {8, -3, 7, -1, 2, 0},
            {100, 50, 200, 150, 25},
            {3, 3, 2, 1, 2, 1},
            {9, 1, 8, 2, 7, 3, 6, 4, 5}
        };

        for (int i = 0; i < testCases.length; i++) {
            System.out.println("Test Case " + (i + 1) + ":");
            int[] arr = testCases[i];
            System.out.print("Original: ");
            printArray(arr);

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

            System.out.print("Sorted:   ");
            printArray(arr);
            System.out.println("----------------------");
        }
    }
}
```

```
Test Case 4:
Original: 10 10 10 10
Sorted:   10 10 10 10
----------------------
Test Case 5:
Original: 5
Sorted:   5
----------------------
Test Case 6:
Original:
Sorted:
----------------------
Test Case 7:
Original: 8 -3 7 -1 2 0
Sorted:   -3 -1 0 2 7 8
----------------------
Test Case 8:
Original: 100 50 200 150 25
Sorted:   25 50 100 150 200
----------------------
Test Case 9:
Original: 3 3 2 1 2 1
Sorted:   1 1 2 2 3 3
----------------------
Test Case 10:
Original: 9 1 8 2 7 3 6 4 5
Sorted:   1 2 3 4 5 6 7 8 9
----------------------
```


50.8k Shares

public class mergesorttest public static void mergeint arr int left int mid int right int n1 mid left 1 int n2 right mid int l new intn1 int r new intn2 for int i 0 i n1 i li arrleft i for int j 0 j n2 j rj arrmid 1 j int i 0 j 0 int k left while i n1 j n2 if li rj arrk li i else arrk rj j k while i n1 arrk li i k while j n2 arrk rj j k public static void mergesortint arr int left int right if left right int mid left right 2 mergesortarr left mid mergesortarr mid 1 right mergearr left mid right public static void printarrayint arr for int value arr systemoutprintvalue systemoutprintln public static void mainstring args int testcases 12 11 13 5 6 7 1 2 3 4 5 6 6 5 4 3 2 1 10 10 10 10 5 8 -3 7 -1 2 0 100 50 200 150 25 3 3 2 1 2 1 9 1 8 2 7 3 6 4 5 for int i 0 i testcaseslength i systemoutprintlntest case i 1 int arr testcasesi systemoutprintoriginal printarrayarr mergesortarr 0 arrlength 1 systemoutprintsorted printarrayarr systemoutprintln--------------------

SIMILAR 19.8% ORIGINAL 80.2%

MAKE IT UNIQUE

Text matches these sources

Sources:	Similarity:
1. https://studyx.ai/homework/1119156...	18.4%
2. https://skillapp.co/blog/mastering-...	12.8%
3. https://prepinsta.com/java-program...	12.8%
4. https://www.mygreatlearning.com/...	12.5%

Exclude source View source

I'm not a robot

reCAPTCHA
Privacy - Terms