```java
import java.util.Arrays;

public class MergeSort {

	public static void main(String[] args) {

		int[] arr = new int[10]; // за умовами завдання створюємо масив на 100 елементів
		for (int i = 0; i < arr.length; i++) {
			arr[i] = ((int) (Math.random() * 11)); // циклічно наповнюємо масив випадковими значеннями
		}
		System.out.println(Arrays.toString(arr)); // для наочності виводимо у консоль генерований масив із одночасним конвертуванням у стрічку

		int[] sortArr = sort(arr); // створюємо відсортований масив та запускаємо метод сортування, який покищо не написаний
		System.out.println(Arrays.toString(sortArr)); // виводимо у консоль відсортований масив із одночасним конвертуванням у стрічку

	}

	// метод розподілу масиву
	public static int[] sort(int[] arr) {
		if (arr.length < 2) // перевіряємо чи масив не містить один елемент
			return arr;
		int m = arr.length / 2; // визначення середини масиву
		int[] arr1 = Arrays.copyOfRange(arr, 0, m); // створення нового масиву та його наповнення елементами лівої частини вихідного масиву
		int[] arr2 = Arrays.copyOfRange(arr, m, arr.length); // створення нового масиву та його наповнення елементами правої чатини вихідного масиву
		return merge(sort(arr1), sort(arr2)); // виклик методу злиття, який поки що не написаний
	}

	// метод злиття та сортування
	public static int[] merge(int[] arr1, int arr2[]) {
		int n = arr1.length + arr2.length; // визначення кількості елементів у відсортованому масиві
		int[] arr = new int[n]; // вказуємо скільки елементів міститиметься у відсортованому масиві, хоча за умовами задачі цей показник можна вказати відразу
		int i1 = 0;
		int i2 = 0; // вводимо додаткові індекси для обліку поточного елемента лівого та правого масивів відповідно
		for (int i = 0; i < n; i++) { // запускаємо цикл до поки не досягне останнього елемента масиву
			if (i1 == arr1.length) {
				arr[i] = arr2[i2++]; // умова: якщо індекс поточної комірки лівого масиву на черговому циклі ітерації досягнув останнього елементу, то в поточну комірку відсортованого масиву записується поточне значення з правого масиву. операція постфіксного інкремента обовязкова, для переходу до наступного елменту масиву на наступній ітерації
			} else if (i2 == arr2.length) {
				arr[i] = arr1[i1++]; // аналогічна умова, лише для правого масиву
			} else {
				if (arr1[i1] < arr2[i2]) { // визначення який з поточних еоеменів лівого та правого масивів менший та його присвоєння у поточну комірку відсортованого масиву
					arr[i] = arr1[i1++];
```

```
                        } else {
                            arr[i] = arr2[i2++];
                        }
                    }
                }
                return arr; // повертаємо відсортований масив
            }

}
```