PHASE 1 PRACTICE ASSISTED PROJECT

7. Writing a program in Java implementing the merge sort algorithm

```
package main;

public class MergeSort {
	public static void mergeSort(int[] arr) {
		int n = arr.length;
		if (n <= 1) {
			return; // Base case: the array is already sorted
		}

		// Divide the array into two halves
		int mid = n / 2;
		int[] left = new int[mid];
		int[] right = new int[n - mid];
		System.arraycopy(arr, 0, left, 0, mid);
		System.arraycopy(arr, mid, right, 0, n - mid);

		// Recursively sort the two halves
		mergeSort(left);
		mergeSort(right);

		// Merge the sorted halves
		merge(arr, left, right);
	}

	public static void merge(int[] arr, int[] left, int[] right) {
		int n1 = left.length;
		int n2 = right.length;
		int i = 0, j = 0, k = 0;

		// Merge the elements from left and right arrays into the
original array in sorted order
		while (i < n1 && j < n2) {
			if (left[i] <= right[j]) {
				arr[k++] = left[i++];
			} else {
				arr[k++] = right[j++];
			}
		}

		// Copy the remaining elements from left array, if any
		while (i < n1) {
			arr[k++] = left[i++];
		}

		// Copy the remaining elements from right array, if any
		while (j < n2) {
			arr[k++] = right[j++];
		}
	}

	public static void main(String[] args) {
		int[] arr = {64, 34, 25, 12, 22, 11, 90};

		System.out.println("Original array: ");
		printArray(arr);
```

```
            mergeSort(arr);

            System.out.println("Sorted array: ");
            printArray(arr);
      }

      public static void printArray(int[] arr) {
            for (int i = 0; i < arr.length; i++) {
                  System.out.print(arr[i] + " ");
            }
            System.out.println();
      }
}
```

OUTPUT-