## Write a program in Java implementing the merge sort algorithm

```
class MergeSort
{
  void merge(int arr[], int l, int m, int r)
  {
     int n1 = m - l + 1;
     int n2 = r - m;
    /* Create temp arrays */
    int L[] = new int [n1];
    int R[] = new int [n2];
    /*Copy data to temp arrays*/
    for (int i=0; i<n1; ++i)
       L[i] = arr[l + i];
    for (int j=0; j<n2; ++j)
       R[j] = arr[m + 1 + j];
    int i = 0, j = 0;
          int k = 1;
    while (i < n1 \&\& j < n2)
       if (L[i] <= R[j])
         arr[k] = L[i];
         j++;
       else
```

```
arr[k] = R[j];
      j++;
    k++;
  while (i < n1)
    arr[k] = L[i];
    j++;
    k++;
  while (j < n2)
    arr[k] = R[j];
    j++;
    k++;
void sort(int arr[], int l, int r)
{
  if (| < r)
  {
    int m = (1+r)/2;
    sort(arr, I, m);
    sort(arr, m+1, r);
    merge(arr, I, m, r);
```

```
static void printArray(int arr[])
                                      int n = arr.length;
                                      for (int i=0; i<n; ++i)
                                                        System.out.print(arr[i] + " ");
                                      System.out.println();
workspace-spring-tool-suite 4-4.15.TRIEASE - First/arCPS/Merge-Softawa abring now file Edit Source Refactor Navigate Search Project Run Window Help 1 - Search Project Run Window Help 2 - Search Project Run Window Help 3 - Search Run Window Help 3 - Search Run W
          /* Copy data to temp arrays */
for (int i = 0; i < n1; ++i)
    L[i] = arr[1 + i];
for (int j = 0; j < n2; ++j)
    R[j] = arr[m + 1 + j];
                                                                                                                                                                                                                                       /* Merge the temp arrays */
                                                                                                                                                                                                                                        // Initial indexes of first and second subarrays int i = 0, j = 0;
                                                                                                                                                                                                                                      // Initial index of merged <u>subarray</u> array
int k = 1;
while (i < n1 & s j < n2) {
    if (L[i] < - R[j]) {
        arr[k] = L[i];
        i++;
    } else {
        arr[k] = R[j];
        j++;
    }
```

/\* Copy remaining elements of L[] if any \*/
while (i < n1) {
 arr[k] = L[i];</pre>

🔡 Q 📦 🔎 🥃 🥲 🥦