**Iterative Algorithm: Merge Sort**

Below is a method which will sort an array of integers in ascending order using the Iterative Merge sort. Please copy this method into your own personal file to test it with the array, consisting of 50+ integers. (You can randomly roll these integers, but make sure you print the array unsorted first prior)

|  |
| --- |
| **public** **static** **void** mergeSort(**int**[] numList) {  **int** high = numList.length - 1;  **int** low = 0;    **int**[] temp = Arrays.copyOf(numList, numList.length); //    **for** (**int** size = 1; size <= high - low; size = 2 \* size) { //  **for** (**int** i = low; i < high; i += 2 \* size) {//  **int** begin = i;  **int** mid = i + size - 1;  **int** end = Integer.min(i + 2 \* size - 1, high);    merge(numList, temp, begin, mid, end); //  }  }  }    **public** **static** **void** merge(**int**[] numList, **int**[] temp, **int** start, **int** mid, **int** end) {  **int** k = start, i = start, j = mid + 1;      **while** (i <= mid && j <= end) { //  **if**(numList[i] < numList[j]) { //  temp[k++] = numList[i++];  }  **else** {  temp[k++] = numList[j++];  }  }    **while** (i < numList.length && i <= mid) {//  temp[k++] = numList[i++]; //  }    **for** (i = start; i <= end; i++) { //  numList[i] = temp[i];  }  } |

1. Please add developer comments at the points indicated with a **“//”**. These comments should briefly outline what each line of code is for, and what it accomplishes.
2. In the box below, please provide a written description of how this sort moves through an array sorting it. This Description should highlight the process, it does not need to outline every single adjustment that is made.

|  |
| --- |
|  |

1. Can you please the adjustments necessary to sort the array in descending order instead of ascending order. Please highlight your adjustments in **red bold text.** (can be done in original box)