```
public class MergeSortImpl {


 public static void mergeSort(List < Integer > integerList, Integer low, Integer high) {
  if (high > low) {
   Integer mid = (low + high) / 2;
   mergeSort(integerList, low, mid);
   mergesort(integerList, mid + 1, high);
   mergeLists(integerList, low, mid, high);
   System.debug('IntegerList' + integerList);
  }


 }

 public static void mergeLists(List < Integer > integerList, Integer low, Integer mid, Integer high) {
  List < Integer > leftList = new List < Integer > ();
  List < Integer > rightList = new List < Integer > ();


  Integer n1 = mid - low + 1;
  Integer n2 = high - mid;
  //populate leftList & rightList
  for (Integer i = 0; i < n1; i++) {
   leftList.add(integerList[i + low]);
  }


  for (Integer j = 0; j < n2; j++) {
```

```
  rightList.add(integerList[mid + 1 + j]);
}

Integer i = 0;
Integer j = 0;
Integer k = 0;

while (i < n1 && j < n2) {

  if (leftList[i] <= rightList[j]) {
    integerList[k] = leftList[i];
    i++;
    k++;
  } else {
    integerList[k] = rightList[j];
    j++;
    k++;
  }

}

while (i < n1) {
  integerList[k] = leftList[i];
  i++;
  k++;
```

```
    }


    while (j < n2) {
      integerList[k] = rightList[j];
      j++;
      k++;
    }
  }
}
```