**Worksheet 19.1**

# Non-recursive Merge and MergeSort

**void** mergeSort (**ArrayList <Integer>** numbers, **int** first, **int** last){

**int** mid = (first + last) / 2;

selectionSort (numbers, first, mid);

selectionSort (numbers, mid+1, last);

merge(numbers, first, mid, last);

}

Use the above non-recursive *mergeSort* method to analyze the changes made to an ArrayList of Integers - referred to as *numbers*, which contains the following integers: 56 32 24 74 29 84 13 65 70 52 16 44 37. The parameter *first* receives the value of 0 since it is the index of the first integer of the array: 56. The parameter last receives the value of 12 since it is the index of the last integer of the array: 37.

Explain the result of each of the following lines of this method and the specific impact on the ArrayList *numbers:*

**int** mid = (first + last) / 2;

selectionSort (numbers, first, mid);

selectionSort (numbers, mid+1, last);

merge (numbers, first, mid, last);

The last line of the *mergeSort* method (above) calls the *merge* method that is discussed in section A of the Student Lesson. Assume that a new array of integers has had each of its two halves already sorted (using the calls to the selectionSort method) and looks as follows. Rewrite the integers using the *merge* method on the next line with arrows to reflect the order of their movement.

12 23 28 31 47 65 3 15 18 35 44 59

\_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_