Name: \_\_\_Daniel Grimshaw\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Merge Sort Assignment**

1. How many calls on the mergeSortHelper method are generated by a call to sort a list of length 32? Include the first call of mergeSortHelper in the mergeSort method.

63

2. Suppose the following arrays have been declared:

int[] numbers2 = {37, 29, 19, 48, 23, 55, 74, 12};

mergeSortHelper(elements, 0, 7,temp);

--- mergeSortHelper(elements, 0, 3, temp);

------ mergeSortHelper(elements, 0,1,temp);

---------mergeSortHelper(elements, 0, 0, temp);

---------mergeSortHelper(elements, 1, 1, temp);

---------merge(elements, 0, 0, 1, temp);

{29, 37, 19, 48, 23, 55, 74, 12}

------mergeSortHelper(elements, 2, 3, temp);

---------mergeSortHelper(elements, 2, 2, temp);

---------mergeSortHelper(elements, 3, 3, temp);

---------merge(elements, 2, 2, 3, temp);

{29, 37, 19, 48, 23, 55, 74, 12}

------merge(elements, 0,1,3,temp)

int[] numbers3 = {8, 5, -9, 14, 0, -1, -7, 3};

int[] numbers4 = {15, 56, 24, 5, 39, -4, 27};

For the 3 arrays above, trace the complete execution of the merge sort algorithm. Show the mergeSortHelper calls and the merge calls in order, indented and bold, as shown in the example below and show state of the elements array after each call to the merge method as also shown in the example below. Sort the array in ascending order. Show your work below.

Example: int[] numbers1 = {63, 9, 45, 72, 27, 18, 54};

mergeSortHelper(elements, 0, 6, temp);

---mergeSortHelper(elements, 0, 3, temp);

------mergeSortHelper(elements, 0, 1, temp);

---------mergeSortHelper(elements, 0, 0, temp);

---------mergeSortHelper(elements, 1, 1, temp);

---------merge(elements, 0, 0, 1, temp);

{**9, 63,** 45, 72, 27, 18, 54}

------mergeSortHelper(elements, 2, 3, temp);

---------mergeSortHelper(elements, 2, 2, temp);

---------mergeSortHelper(elements, 3, 3, temp);

---------merge(elements, 2, 2, 3, temp);

{9, 63, **45, 72**, 27, 18, 54}

------merge(elements, 0, 1, 3, temp);

{**9, 45, 63, 72**, 27, 18, 54}

---mergeSortHelper(elements, 4, 6, temp);

------mergeSortHelper(elements, 4, 5, temp);

---------mergeSortHelper(elements, 4, 4, temp);

---------mergeSortHelper(elements, 5, 5, temp);

---------merge(elements, 4, 4, 5, temp);

{9, 45, 63, 72, **18, 27,** 54}

------mergeSortHelper(elements, 6, 6, temp);

------merge(elements, 4, 5, 6, temp);

{9, 63, 45, 72, **18, 27, 54**}

---merge(elements, 0, 3, 6, temp);

**{9, 18, 27, 45, 54, 63, 72}**

3. Write a program that reads a series of input lines from a file and sorts them into alphabetical order. The program should use the merge sort algorithm so that it efficiently sorts even a large file. See file SortNamesFramework.java. Name your file LastnameMerge.java. Also available is the text file names.txt.

Output based on input file names.txt is below.

What is the input file? names.txt  
  
Chandler, Adam Jr  
Chandler, Arabella "Babe"  
Chandler, Colby  
Chandler, Krystal  
Chandler, Marian  
Chandler, Stuart

Codahy, Livia  
Colby, Liza  
Cortlandt, Palmer  
Cortlandt, Opal  
Devane, Aidan  
Dillon, Amanda  
Dillon, Janet  
English, Brooke  
Fargate, Myrtle  
Frye, Derek  
Grey, Maria  
Hayward, David  
Henry, Del  
Henry, Di  
Jefferson, Kelsey  
Kane, Erica  
Keefer, Julia  
Lavery, Ryan  
Lavery, Spike  
Lavery,k Jonathan  
Madden, Josh  
Martin, James  
Martin, Joseph  
Martin, Ruth  
McDermott, Annie  
Montgomery, Barbara  
Montgomery, Bianca  
Montgomery, Jackson  
Montgomery, Lily  
Montgomery, Reggie  
Montgomery, Sean  
Santos, Hayley  
Santos, Mateo  
Slater, Kendall  
Slater, Zach  
Smythe, Greenlee  
Stone, Mary Frances  
Warner, Anita