GEBZE TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

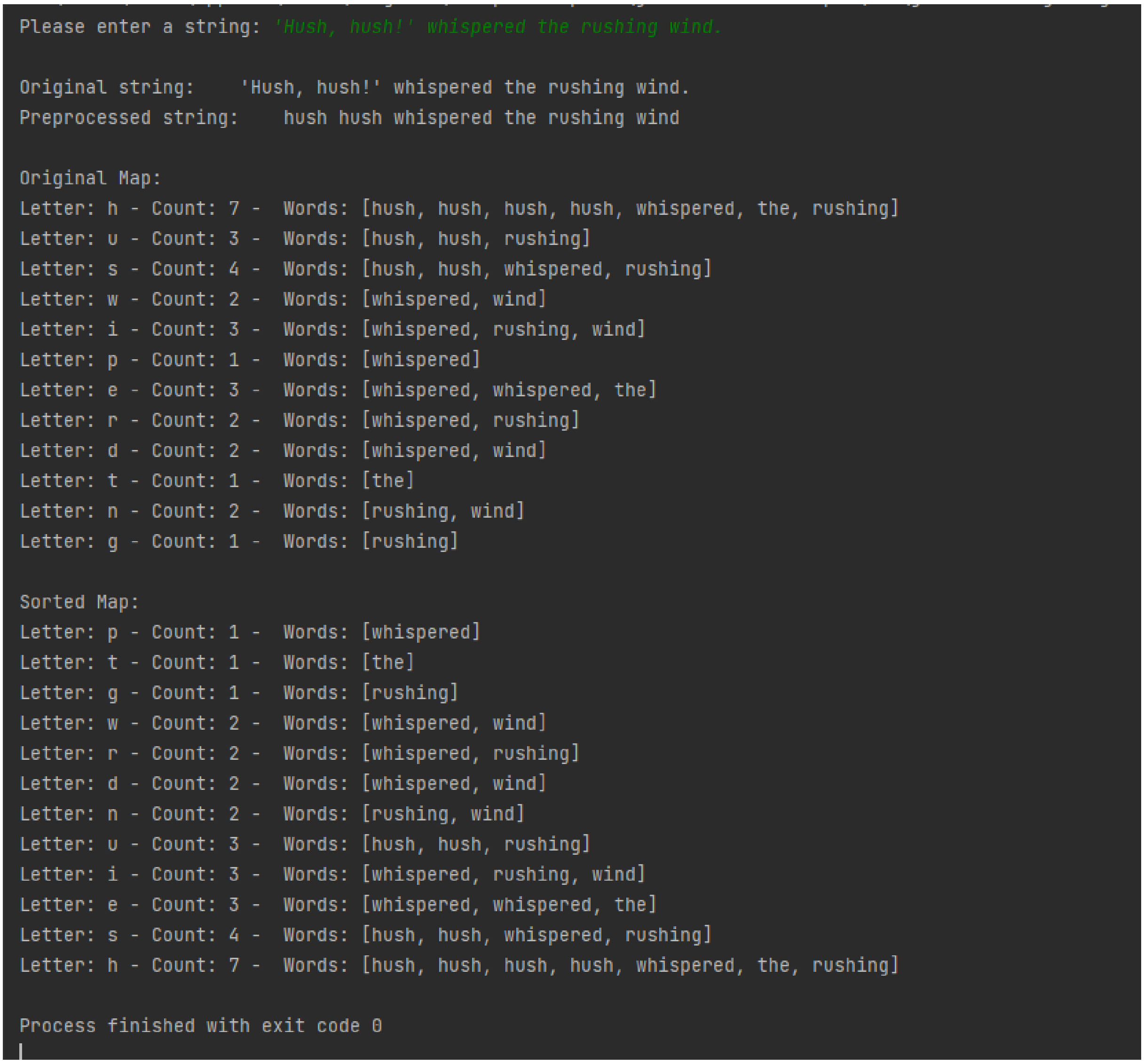
CSE 222 - SPRING 2023

HOMEWORK 6 REPORT

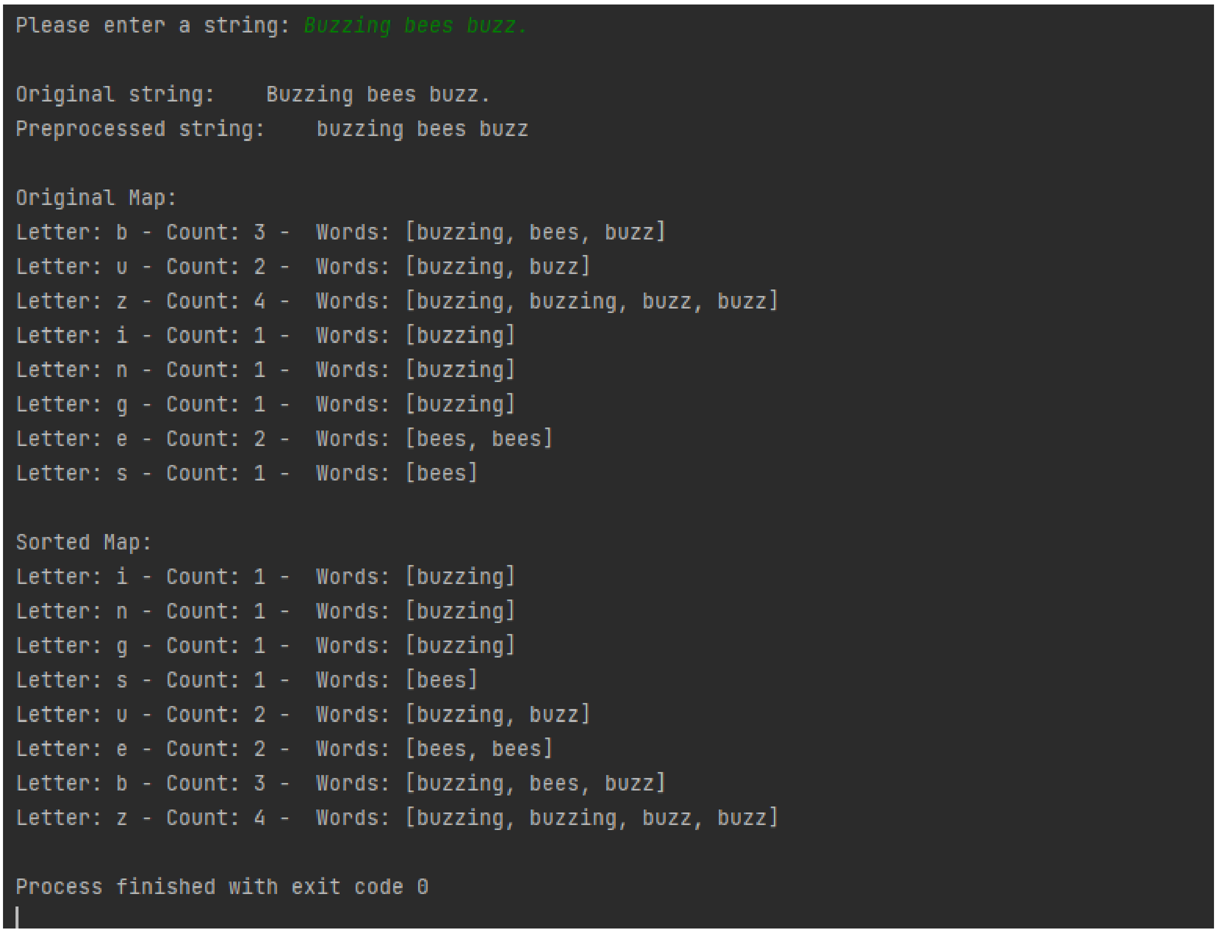
ETHEM MENGAASLAN

200104004016

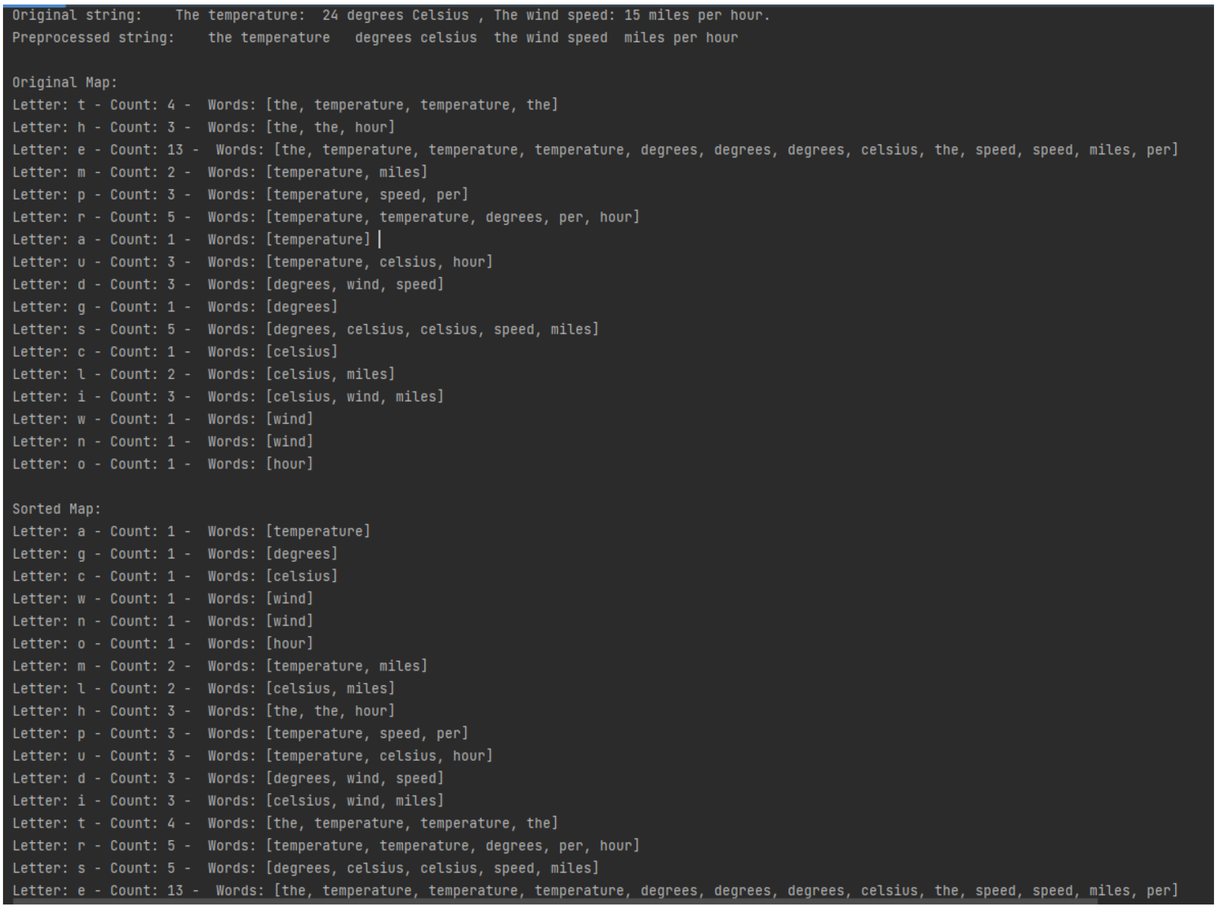
OUTPUT SAMPLES:



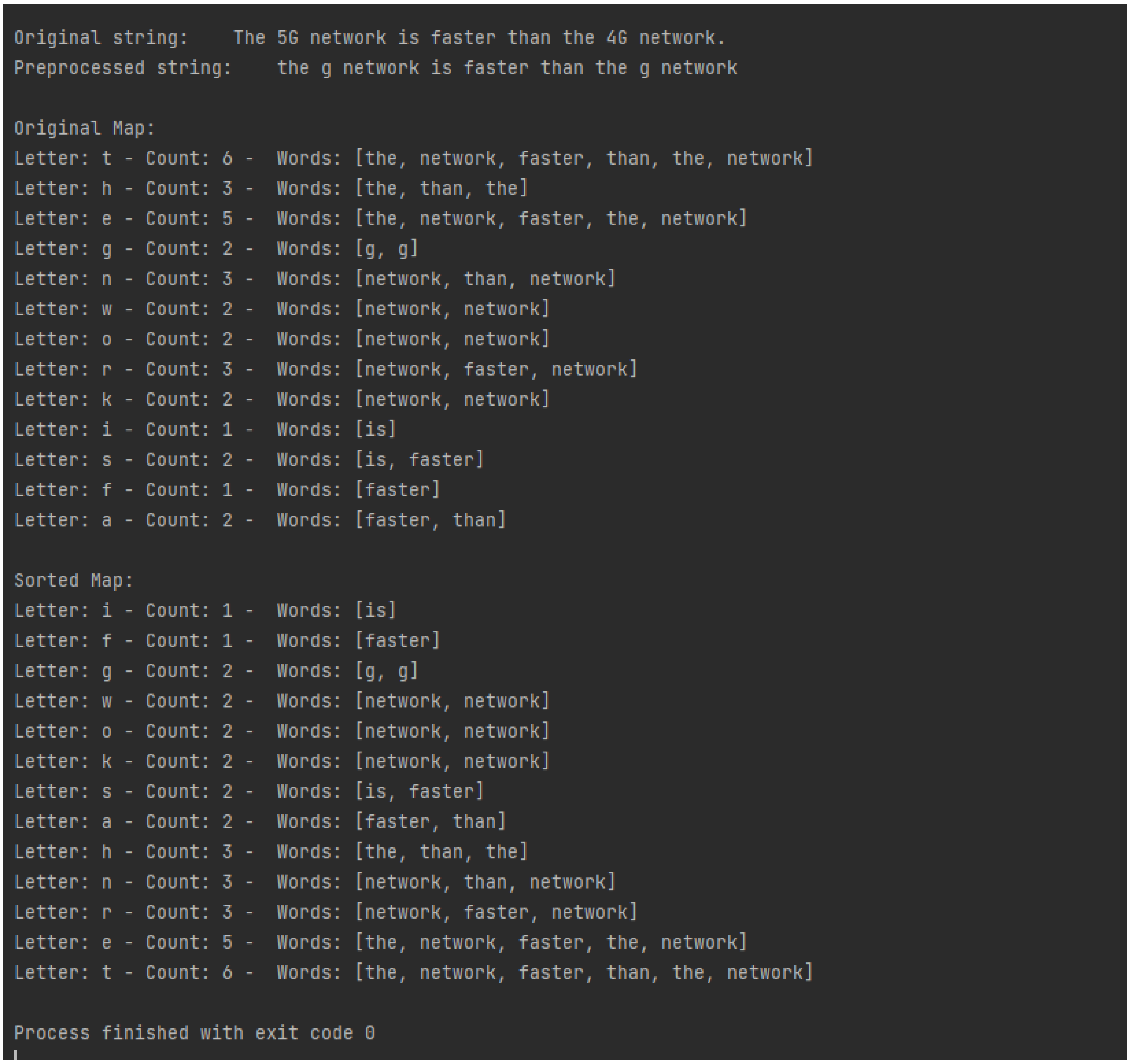
```
Please enter a string: 'Hush, hush!' whispered the rushing wind.

Original string:    'Hush, hush!' whispered the rushing wind.
Preprocessed string:    hush hush whispered the rushing wind

Original Map:
Letter: h - Count: 7 -  Words: [hush, hush, hush, hush, whispered, the, rushing]
Letter: u - Count: 3 -  Words: [hush, hush, rushing]
Letter: s - Count: 4 -  Words: [hush, hush, whispered, rushing]
Letter: w - Count: 2 -  Words: [whispered, wind]
Letter: i - Count: 3 -  Words: [whispered, rushing, wind]
Letter: p - Count: 1 -  Words: [whispered]
Letter: e - Count: 3 -  Words: [whispered, whispered, the]
Letter: r - Count: 2 -  Words: [whispered, rushing]
Letter: d - Count: 2 -  Words: [whispered, wind]
Letter: t - Count: 1 -  Words: [the]
Letter: n - Count: 2 -  Words: [rushing, wind]
Letter: g - Count: 1 -  Words: [rushing]

Sorted Map:
Letter: p - Count: 1 -  Words: [whispered]
Letter: t - Count: 1 -  Words: [the]
Letter: g - Count: 1 -  Words: [rushing]
Letter: w - Count: 2 -  Words: [whispered, wind]
Letter: r - Count: 2 -  Words: [whispered, rushing]
Letter: d - Count: 2 -  Words: [whispered, wind]
Letter: n - Count: 2 -  Words: [rushing, wind]
Letter: u - Count: 3 -  Words: [hush, hush, rushing]
Letter: i - Count: 3 -  Words: [whispered, rushing, wind]
Letter: e - Count: 3 -  Words: [whispered, whispered, the]
Letter: s - Count: 4 -  Words: [hush, hush, whispered, rushing]
Letter: h - Count: 7 -  Words: [hush, hush, hush, hush, whispered, the, rushing]

Process finished with exit code 0
```



```
Please enter a string: Buzzing bees buzz.

Original string:    Buzzing bees buzz.
Preprocessed string:    buzzing bees buzz

Original Map:
Letter: b - Count: 3 -  Words: [buzzing, bees, buzz]
Letter: u - Count: 2 -  Words: [buzzing, buzz]
Letter: z - Count: 4 -  Words: [buzzing, buzzing, buzz, buzz]
Letter: i - Count: 1 -  Words: [buzzing]
Letter: n - Count: 1 -  Words: [buzzing]
Letter: g - Count: 1 -  Words: [buzzing]
Letter: e - Count: 2 -  Words: [bees, bees]
Letter: s - Count: 1 -  Words: [bees]

Sorted Map:
Letter: i - Count: 1 -  Words: [buzzing]
Letter: n - Count: 1 -  Words: [buzzing]
Letter: g - Count: 1 -  Words: [buzzing]
Letter: s - Count: 1 -  Words: [bees]
Letter: u - Count: 2 -  Words: [buzzing, buzz]
Letter: e - Count: 2 -  Words: [bees, bees]
Letter: b - Count: 3 -  Words: [buzzing, bees, buzz]
Letter: z - Count: 4 -  Words: [buzzing, buzzing, buzz, buzz]

Process finished with exit code 0
```



```
Original string:    The temperature:  24 degrees Celsius , The wind speed: 15 miles per hour.
Preprocessed string:    the temperature   degrees celsius  the wind speed  miles per hour

Original Map:
Letter: t - Count: 4 -  Words: [the, temperature, temperature, the]
Letter: h - Count: 3 -  Words: [the, the, hour]
Letter: e - Count: 13 -  Words: [the, temperature, temperature, temperature, degrees, degrees, degrees, celsius, the, speed, speed, miles, per]
Letter: m - Count: 2 -  Words: [temperature, miles]
Letter: p - Count: 3 -  Words: [temperature, speed, per]
Letter: r - Count: 5 -  Words: [temperature, temperature, degrees, per, hour]
Letter: a - Count: 1 -  Words: [temperature]
Letter: u - Count: 3 -  Words: [temperature, celsius, hour]
Letter: d - Count: 3 -  Words: [degrees, wind, speed]
Letter: g - Count: 1 -  Words: [degrees]
Letter: s - Count: 5 -  Words: [degrees, celsius, celsius, speed, miles]
Letter: c - Count: 1 -  Words: [celsius]
Letter: l - Count: 2 -  Words: [celsius, miles]
Letter: i - Count: 3 -  Words: [celsius, wind, miles]
Letter: w - Count: 1 -  Words: [wind]
Letter: n - Count: 1 -  Words: [wind]
Letter: o - Count: 1 -  Words: [hour]

Sorted Map:
Letter: a - Count: 1 -  Words: [temperature]
Letter: g - Count: 1 -  Words: [degrees]
Letter: c - Count: 1 -  Words: [celsius]
Letter: w - Count: 1 -  Words: [wind]
Letter: n - Count: 1 -  Words: [wind]
Letter: o - Count: 1 -  Words: [hour]
Letter: m - Count: 2 -  Words: [temperature, miles]
Letter: l - Count: 2 -  Words: [celsius, miles]
Letter: h - Count: 3 -  Words: [the, the, hour]
Letter: p - Count: 3 -  Words: [temperature, speed, per]
Letter: u - Count: 3 -  Words: [temperature, celsius, hour]
Letter: d - Count: 3 -  Words: [degrees, wind, speed]
Letter: i - Count: 3 -  Words: [celsius, wind, miles]
Letter: t - Count: 4 -  Words: [the, temperature, temperature, the]
Letter: r - Count: 5 -  Words: [temperature, temperature, degrees, per, hour]
Letter: s - Count: 5 -  Words: [degrees, celsius, celsius, speed, miles]
Letter: e - Count: 13 -  Words: [the, temperature, temperature, temperature, degrees, degrees, degrees, celsius, the, speed, speed, miles, per]
```



```
Original string:    The 5G network is faster than the 4G network.
Preprocessed string:    the g network is faster than the g network

Original Map:
Letter: t - Count: 6 -  Words: [the, network, faster, than, the, network]
Letter: h - Count: 3 -  Words: [the, than, the]
Letter: e - Count: 5 -  Words: [the, network, faster, the, network]
Letter: g - Count: 2 -  Words: [g, g]
Letter: n - Count: 3 -  Words: [network, than, network]
Letter: w - Count: 2 -  Words: [network, network]
Letter: o - Count: 2 -  Words: [network, network]
Letter: r - Count: 3 -  Words: [network, faster, network]
Letter: k - Count: 2 -  Words: [network, network]
Letter: i - Count: 1 -  Words: [is]
Letter: s - Count: 2 -  Words: [is, faster]
Letter: f - Count: 1 -  Words: [faster]
Letter: a - Count: 2 -  Words: [faster, than]

Sorted Map:
Letter: i - Count: 1 -  Words: [is]
Letter: f - Count: 1 -  Words: [faster]
Letter: g - Count: 2 -  Words: [g, g]
Letter: w - Count: 2 -  Words: [network, network]
Letter: o - Count: 2 -  Words: [network, network]
Letter: k - Count: 2 -  Words: [network, network]
Letter: s - Count: 2 -  Words: [is, faster]
Letter: a - Count: 2 -  Words: [faster, than]
Letter: h - Count: 3 -  Words: [the, than, the]
Letter: n - Count: 3 -  Words: [network, than, network]
Letter: r - Count: 3 -  Words: [network, faster, network]
Letter: e - Count: 5 -  Words: [the, network, faster, the, network]
Letter: t - Count: 6 -  Words: [the, network, faster, than, the, network]

Process finished with exit code 0
```

Introduction:

This report is a documentation for the implementation of the Merge Sort Algorithm on myMap objects in Java. This algorithm is implemented in the mergeSort class as requested in the assignment pdf. The mergeSort class is designed to sort myMap objects by the count values of their entries, in ascending order.

Classes:

1. **info** class: The info class has two variables, count and words. The count variable represents the number of words in the set, which also corresponds to the number of occurrences of the letter. The words variable is an ArrayList of the words in the set. The push method of the info class adds a new word to the set and increments the count. The toString method returns a string representation of an info object, including the array of words with the letter's count value.

2. **mergeSort** class: The mergeSort class is designed to sort myMap objects by the count values of their entries, in ascending order. It has three variables, originalMap, sortedMap, and aux. The originalMap variable represents the original map to be sorted. The sortedMap variable represents the sorted map, and the aux variable is an auxiliary array for storing the keys of the map as strings. The mergeSort method of the mergeSort class sorts the input map by the count values of the entries using the merge sort algorithm. The mergeSortHelper method recursively sorts an array of strings (letters as keys) in accordance with the count values of the map entry related to the string/letter. The merge method merges two sorted subarrays of an array.

3. **myMap** class: The myMap class is designed to create and store a custom map that holds String, info pairs. It has two constructors, one that creates an empty myMap object, and the other that takes a sentence as input and processes it to produce letter-info pairs for the map. The preProcessString method is used to preprocess the input string by removing all non-alphabetic characters and converting the string to lowercase. The resulting preprocessed string is then split into words, and each word is processed to create letter-info pairs. The printMap method is used to print the contents of the myMap.

Sorting Mechanism:

The mergeSort algorithm implemented in the mergeSort class takes a myMap object as input and sorts its entries by the count values of their values(info objects) . The algorithm works by first traversing the original map and putting the characters in it in order, using merge sort to compare and sort the characters/strings (keys) according to their count values (in this part, the corresponding count value is accessed by using the get() method of the map with the current traversed key as the parameter) . After that, it iterates through this array and gets the related key-value pair from the original map and puts it in the sorted map. Since the aux array is sorted, the sortedMap will be filled with entries in a sorted way.
keys.