Lab Programming Assignment #1

Due: Friday, September 18, 2020, by 11:59 pm PDT

Points possible: 20

Submission Requirements:

- You can work in teams up to 3. If working in teams, only one person should submit the required file.
  - If working in a team, adhere to social distancing and collaborate remotely over the phone and internet.
- You should submit just one file, and that file's name should be:
  - **ModifiedInsertionSort.java**
- Submitting a file with any other name will get you points deducted. Do not zip the file either.
- Because the Java file name should be what's listed above, the top level class name within it should also be named **ModifiedInsertionSort**.
- Within the java file, include your name (at the top as a comment). If working in a team, the full names of all team members must also be within the Java file (at the top as a comment).
- In addition, all team member names need to be listed on the submission in BeachBoard, as a comment.
- Submit the Java file to BeachBoard. No email submissions will be accepted.
- The project must be in the Java programming language.
- I should be able to both pass in arguments and run your program via the command line.
- Ensure the Java keyword "package" doesn't exist in your java file before you submit your code. That will prevent me from properly testing your code. So remove it before submitting your code.

Program Requirements:

Modify the insertion sort algorithm such that:

1. The algorithm must be able to sort a list of lower cased words (strings) in non-decreasing alphabetical order.
2. It should be able to support the sorting of up to 20 words.
   a. I will only test up to 20 lower cased words.
   b. I will only test words with at least two letters.
3. The last letter in the word should NOT be considered in the sorting. However, the full word still needs to be outputted in the sorted list.
4. In addition to the sorted list, it must also output the number of comparisons for each word (when it's a key) from j=2 to n.
   a. The output of comparison information must also be in order of the original list from top down.
   b. In case of duplicate words, I still want to see each word's number of comparisons.
   c. For further clarity, see the examples below.

I will be testing your code via the command line so it is a requirement that your code be able to take in arguments and execute via the command line. Failure to do this will result in zero credit.

Comment your code enough so that in case it doesn't work, I can see what you're trying to do and possibly give you partial credit.

Examples:

```
C:\MyFolder> java ModifiedInsertionSort apple test going zebra aardvark
aardvark apple going test zebra
test 1
going 2
zebra 1
aardvark 4

C:\MyFolder> java ModifiedInsertionSort andy ands anda and boats boatc boat an sea
an and andy ands anda boat boats boatc sea
ands 1
anda 1
and 3
boats 1
boatc 1
boat 3
an 7
sea 1

C:\MyFolder> java ModifiedInsertionSort check checkz check checks checks check
check check check checkz checks checks
checkz 1
check 2
checks 1
checks 1
check 4
```

Where "C:\MyFolder" is just an example of where the program is being executed from. Your program should not output anything else other than the requirements above. Output should be in the above format (as shown in the examples) as well.

Failure to follow the above requirements may result in substantial point deduction, up to and including receiving zero credit. I recommend you read the instructions more than once to ensure your program meets the requirements.

Grading Guidelines:

- Does the program meet the requested requirements/criteria?
- Are the submission instructions followed?
- Does your code compile?
- Does your code pass my test cases? Note that I do not share my test cases.
- Ensure your code works with duplicate words (as you can see in an example above).