CSOC 2430: HW1

Overview:

In this assignment you are required to create a text parser in Java/C++. Given a input text file you need to parse it and answer a set of frequency related questions.

Technical Requirement of Solution:

You are required to do this *ab initio* (bare-bones from scratch). This means, your solution cannot use any library methods in Java except the ones listed below (or equivalent library functions in C++).

- `String.split()` and other String operations can be used wherever required.
- You can use any Regular Expression related facilities (`java.util.regex`) to match target words and phrases.
- You are also allowed to use different variants of array and list based built-in data structures such as `Array` , `List` , `ArrayList` , `Vector` .
- Standard file IO facilities for reading/writing, such as `BufferedReader` .

Create as many files, intermediate array-based data structures as you wish, and allocate as much heap memory from JVM as you need. BUT you are allowed to read the input file **EXACTLY ONCE** to answer ALL the questions. You however can use any internal array based representation of the whole file to do multiple rounds of processing if needed after having read it EXACTLY ONCE.

Suggested programming language Java. However considering this is the very first assignment, you can use C/C++ provided similar constructs and rules are followed and no library functions that leverage hash and maps are used.

For the following questions, list all matching output if there are ties

1. List the most frequent word(s) in the whole file and its frequency.
2. List the 3rd most frequent word(s) in the whole file and its frequency.
3. List the word(s) with the *highest frequency* in a sentence across all sentences in the whole file, also print its frequency and the corresponding sentence.
4. List sentence(s) with the maximum no. of occurrences of the word "**the**" in the entire file and also list the corresponding frequency.
5. List sentence(s) with the maximum no. of occurrences of the word "**of**" in the entire file and also list the corresponding frequency.
6. List sentence(s) with the maximum no. of occurrences of the word "**was**" in the entire file and also list the corresponding frequency.
7. List sentence(s) with the maximum no. of occurrences of the phrase "**but the**" in the entire file and also list the corresponding frequency.
8. List sentence(s) with the maximum no. of occurrences of the phrase "**it was**" in the entire file and also list the corresponding frequency.
9. List sentence(s) with the maximum no. of occurrences of the phrase "**in my**" in the entire file and also list the corresponding

frequency.

Implementation Detail:

Inputs

The program has two arguments:

- The first argument: path to the input text file.
- The second argument: name prefix for the output files

For example:

```
$ java HW1  "./input.txt" "output"
```

input file: A text document. Assume each newline (\n) defines a paragraph. Each period (.) defines end of an sentence. Or if a sentence is the last in a paragraph and doesn't have an explicit period (.), its end marker is the same as a newline. Each space within a sentence (character '32') define the word delimiter. **The assignment is case insensitive so you must transform and work in lower case.**

Outputs: Click here to download sample input and answer files. As grading is automated, your output must conform to the following specifications. For each of the 9 questions you must create one single output file. So your program should produce 9 output files each time you run it. If a question has multiple output (multiple sentences/words/…) you should print each sentence in a new line. Do not print them on the same line! The order of the sentences/words/phrases is not important. However, the order of the output file name must be matching the order of the questions. For example, given prefix "output", the output file of the first question should be `output1.txt` and for the second question it is `output2.txt` . The output format depends on the question type and it must be:

- For **question 1 and 2**:

  word:frequency *e.g.*

  `the:10`

- For **question 3**:

  word:frequency:sentence *e.g.*

  `the:9:you see watson he explained in the early hours of the morning...`

- For **question 4-9**:

  word:frequency:sentence *e.g.*

  `was:2:then it was withdrawn as suddenly as it appeared...`

  `was:2:the 4 a week was a lure which must draw him and...`

Other details

- For Word related questions, Words are defined as whole word separated by a space. So "there" does not count towards the frequence of "the". Whereas for phrases you are required to consider substrings as well, for example "within my" will also count towards the frequence of "in my"
- When printing the sentences, don't print out the period (.) at the end
- When there is a tie, print all words/sentences with the maximum frequency

Run the program on Linux:

Create a directory on the Linux server, its name must be hw1

```
$ mkdir hw1
```

Change your current directory to the hw1

```
$ cd hw1
```

Run the shell script to compile the program

```
$ sh compile_java.sh
```

Evaluate your results with the true results

```
$ sh test_java.sh
```