Adaptavist

Technical Assessment

Engineering Team Lead

Type of assessment: Take home task

Task 1.

Word count

Instructions

Please implement a command-line application that takes a path to a file as an argument and prints a word count of its contents. The output should consist of a line for each word, with the number of its occurrences in the file. It should be sorted by the number of occurrences starting with the most frequent word.

Input file example:

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Non sodales neque
sodales ut. Gravida arcu ac tortor dignissim convallis aenean et tortor.
Ultricies mi eget mauris pharetra et ultrices neque. Vel eros donec ac odio
tempor orci.

Tellus in metus vulputate eu scelerisque felis. Volutpat diam ut venenatis
tellus in metus vulputate eu. Tempor id eu nisl nunc mi ipsum faucibus
vitae. Sit amet luctus venenatis lectus magna fringilla urna. Habitant
morbi tristique senectus et netus et. Eget felis eget nunc lobortis. Neque
vitae tempus quam pellentesque nec.

Pellentesque nec nam aliquam sem et tortor consequat id porta. Eget dolor
morbi non arcu risus quis. Lacinia quis vel eros donec. Velit laoreet id
donec ultrices tincidunt arcu non sodales. Feugiat in fermentum posuere
urna nec tincidunt. Tortor at risus viverra adipiscing at. In mollis nunc
sed id semper. Semper eget duis at tellus at urna condimentum mattis
pellentesque. Egestas purus viverra accumsan in nisl nisi scelerisque eu. A
cras semper auctor neque vitae tempus quam pellentesque.
```

Output example:

```
foo: 24
bar: 17
...
```

The results should be consistent between two files which contain the exact same words in the same frequency but in a different order. Please provide the solution in an archive together with instructions on how to build it and run it.

Consideration

- You can use **Groovy** or **Java** (if you are not comfortable with either of these you can use another language) to build the solution. Do not implement the solution in multiple languages, we only want a single solution in one language.
- If you believe that the instructions are not precise and you need to make some assumptions please do so but make them clear to us by including them in the instructions sent as part of your solution. If you are uneasy about something and would rather ask a question than make an assumption then you can always get in touch.
- The task is simple on purpose - we're sure that apart from applying for this position you still have other things to do in life, like your current job for example therefore we don't want to ask you to spend an unnecessarily long time on solving complex problems. We are on the other hand interested in seeing what you consider best software engineering practices so please ensure that your solution could be considered "production-ready" and complete and that we can run the solution ourselves. This includes things like testing, good packaging and instructions on how to run etc.
- We should be able to follow your instructions to run the code ourselves with minimal effort and get the expected result.

Next Steps

Your submitted assessment will be reviewed by our team of engineers where they will look at your code and approach, and decide whether to move to the next stage of the process, which will be a panel interview with some of the team. If selected for the next stage of interview, we will discuss your approach to the task and its implementation, and discuss how you would handle changes to the requirements.

Good luck!