

General Description

Welcome, Candidate. Please create a program executable from the command line that when given text(s) will return a list of the 100 most common three word sequences.

For example, if I ran `ruby ./solution.rb moby_dick.txt` the first lines of the result would be:

```
the sperm whale - 85
the white whale - 71
of the whale - 67
```

What to prioritize

- We suggest spending 2 hours meeting the basic requirements and, if you so choose, 2 hours on the extra credit.
- Basic requirements should be met.
- Basic requirements should be tested.
- Code should be well structured. Consider extensibility. Consider readability.
- Your README should include **** how to run your program. **** What would you do next, given more time (if anything)? ****** Are there bugs that you are aware of? ****** How to run the tests provided.

Basic Requirements

- The program accepts as arguments a list of one *or more* file paths (e.g. `./solution.rb file1.txt file2.txt ...`).
- The program also accepts input on stdin (e.g. `cat file1.txt | ./solution.rb`).
- The program outputs a list of the 100 most common three word sequences.
- The program ignores punctuation, line endings, and is case insensitive (e.g. “I love\nsandwiches.” should be treated the same as “(I LOVE SANDWICHES!!)”). Watch out that contractions aren't changed into 2 words (eg. shouldn't should not become shouldn t).
- The program should be tested. Provide a test file for your solution.
- The program should be well structured and understandable.
- The program is capable of processing large files and runs as fast as possible.

Extra Credit

- The program can run in a docker container. In this case providing the container is not necessary but a dockerfile that can be built is required.
- The program is capable of processing large files and remains performant. Think about if the program needed to handle 1,000 Moby Dick's at once. What would you need to do? Consider memory consumption and speed.
- It handles unicode characters(eg. the ü in Süsse or ß in Straße).

Follow up questions:

- What are the steps needed to have this program (or any application) ready for production?
- What would be the advantages or disadvantages of having a program like this running in a distributed system with a container-orchestration platform?
- How would you scale this if you needed to support millions of requests simultaneously? (Imagine the program is complex enough to require it)