# Task Sheet PR03
# Streamlining the System

## Preliminary Notes:

All notes and requirements of PR02 still apply! Keep in mind the new test cases which are included in the new template file. Adding the latest functionalities must not break the old ones – they must still work correctly. This includes old unit tests.

## Task 1 – Stemming

1. Implement a stemming functionality, using the algorithm description found in the file `stemming_algorithm.txt`, which is available on the Moodle page! Make it possible for the user to search with stemming enabled[1]!
2. Naturally, the use of methods from NLP libraries such as `nltk` is strictly prohibited! However, using `re` is permitted.

## Task 2 – Vector Space Model

1. Implement the search using the Vector Space Model with inverted lists! Make it available to the user as an alternative search method!
2. Use *tf.idf* for the generation of term weights! The generation of the query vector should be done according to Salton/Buckley (1988).
3. Use the base algorithm with inverted lists as it was presented in the lecture!

### Hints:

- You can assume that the user will only input valid queries, which are terms separated by one space character each.
- It is not necessary to implement this method with linear search. Only do it with inverted lists.

## Task 3 – Evaluation

1. Implement the calculation of precision and recall for queries that contain the terms listed in a provided ground truth file. The updated template on the Moodle page contains two ground truth files that you can use[2].

[1] This means that the user can now search with stopword removal and/or with stemming enabled. Any combination should be possible.

[2] Each ground truth file corresponds to one of the collections extracted from web resources. The source texts are Aesop's Fables and Grimm's Fairy Tales from Project Gutenberg, which you are already familiar with from the PR02 test cases.

2. Make it possible for the Boolean model too, that queries can contain more than one term. Here, multiple terms should be combined by conjunction. For queries that consist of more than one search term from the ground truth, the calculation of precision and recall should also work!
3. The precision and recall values should be printed below the results after every search query. If the calculation is not possible, the program must not crash! You may return a non-value, such as -1.

**Make sure that your solution considers all requirements listed in this file and upload it on Moodle until the specified deadline!**

**Be sure to check that the files you submit work with the latest unit tests!**