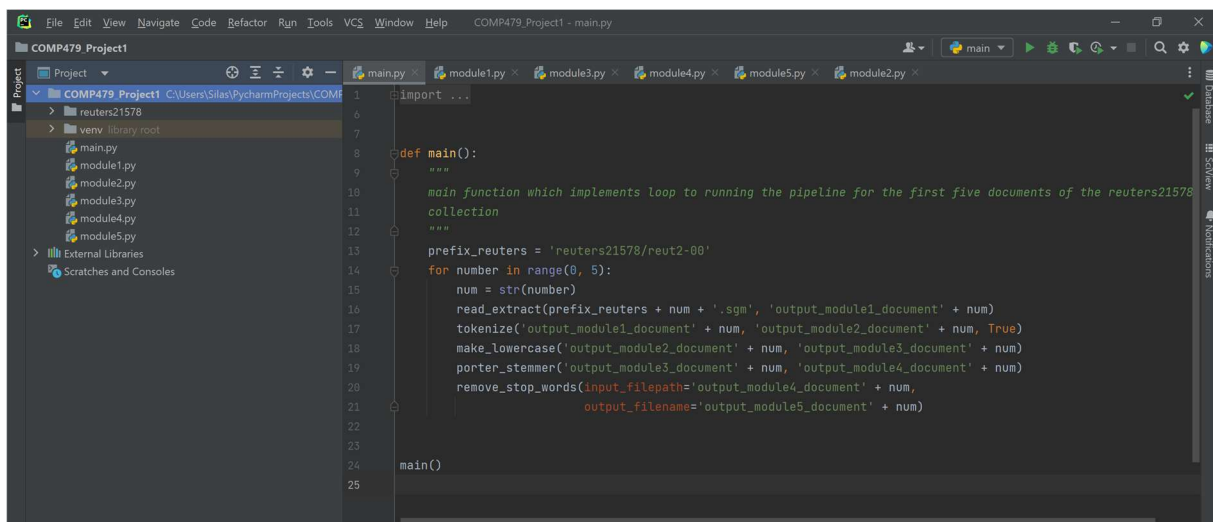


Demo for Project 1 of COMP 479 by Silas Kalinowski ID: 40256077

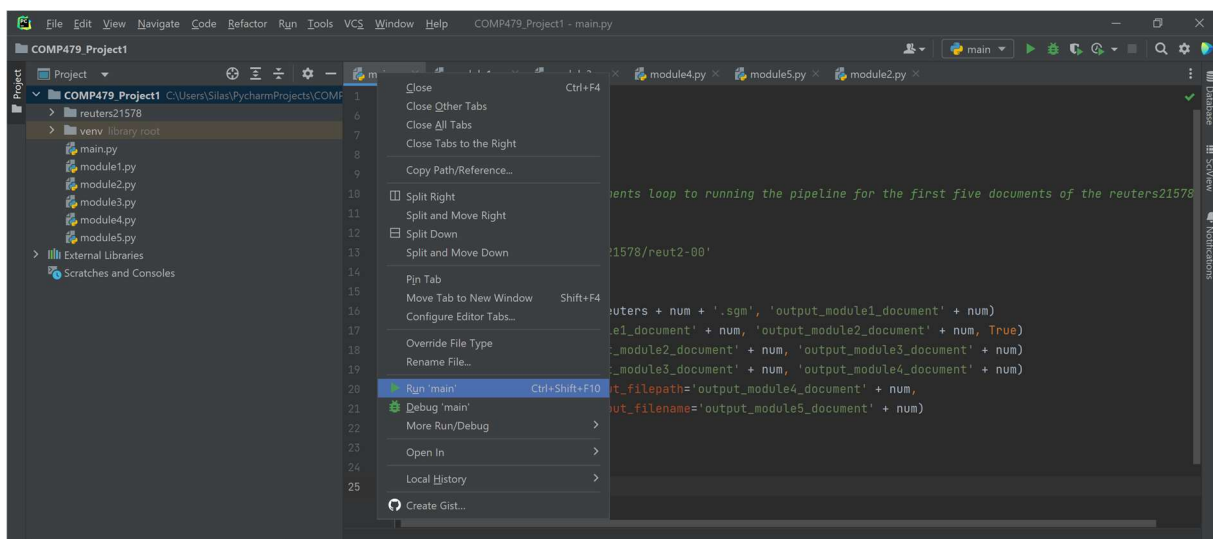
The program can be started by running the main.py file in a Python environment. For the program to work all modules and all input files need to be in the local directory. Also, the program uses NLTK, which needs to be downloaded. For the stemmer and module 5 nltk.download('stopwords') might have to be used. The output files are saved in the local directory and are named in the following manner: output_moduleMODULENUMBER_documentDOCUMENTNUMBER.

The testing of the program starts automatically when running the main file. The result of the tests is printed in the console. The following screenshots display the whole process of running the pipeline and testing it.

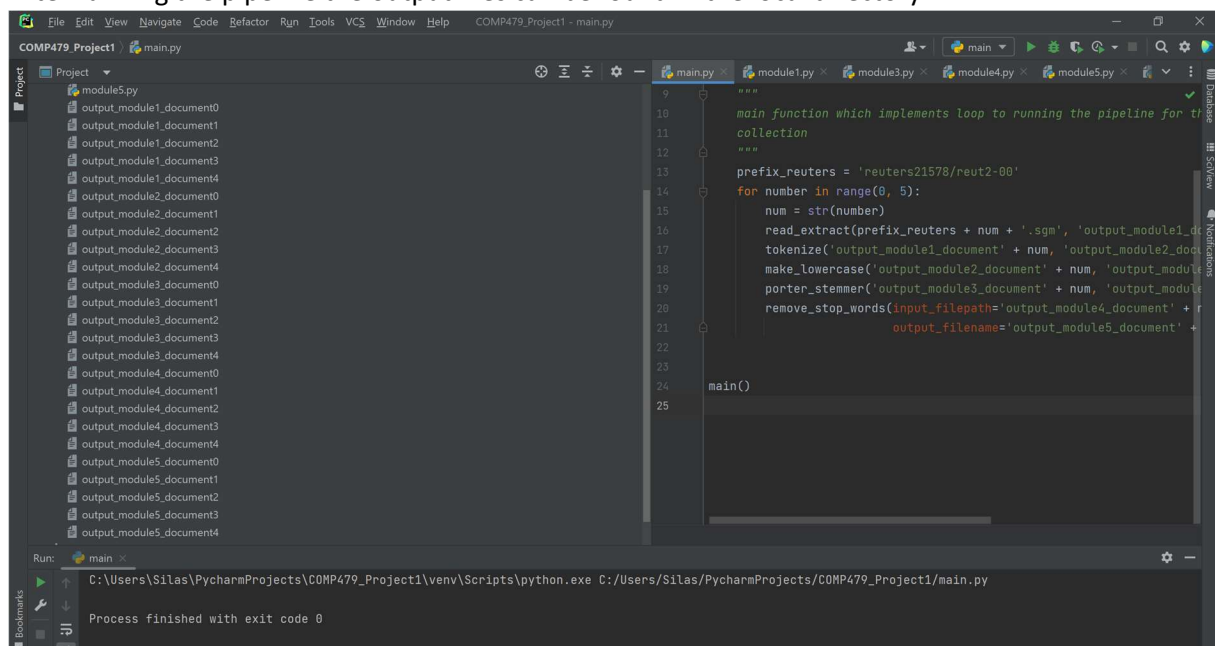
This screenshot shows the directory and main module of the pipeline before running the pipeline.



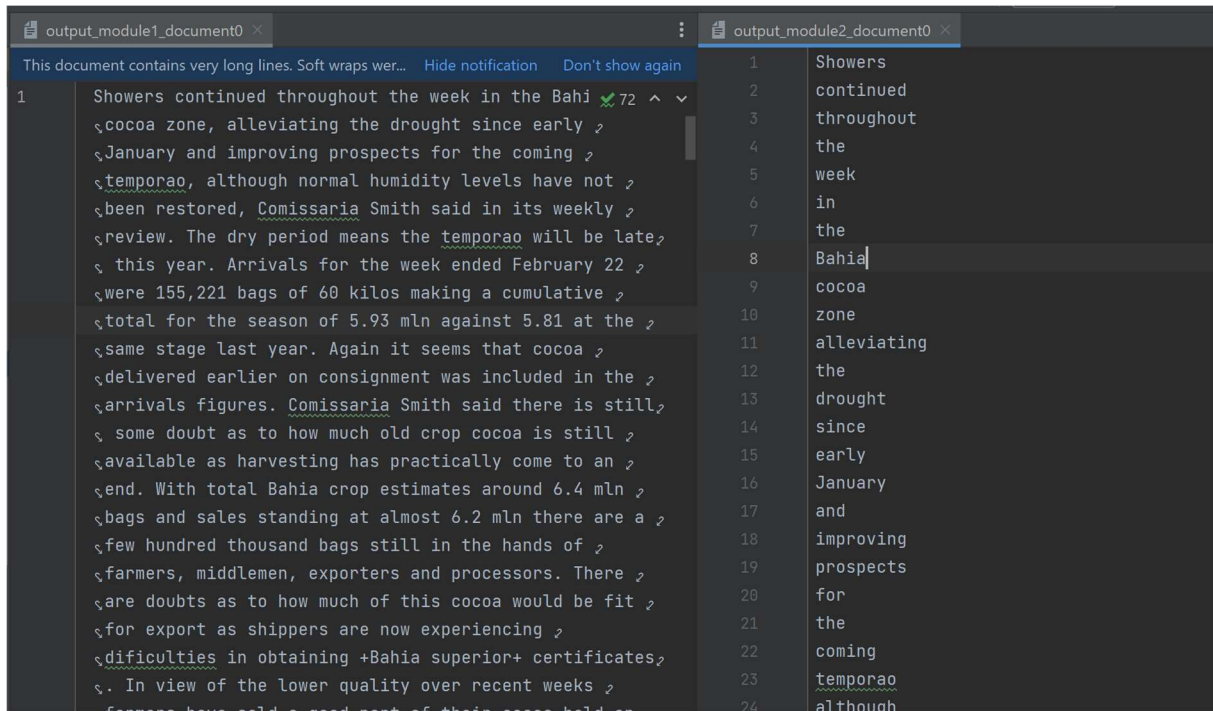
By running the main module the pipeline will process the first five files of the reuters21578 collection automatically.



After running the pipeline the output files can be found in the local directory.

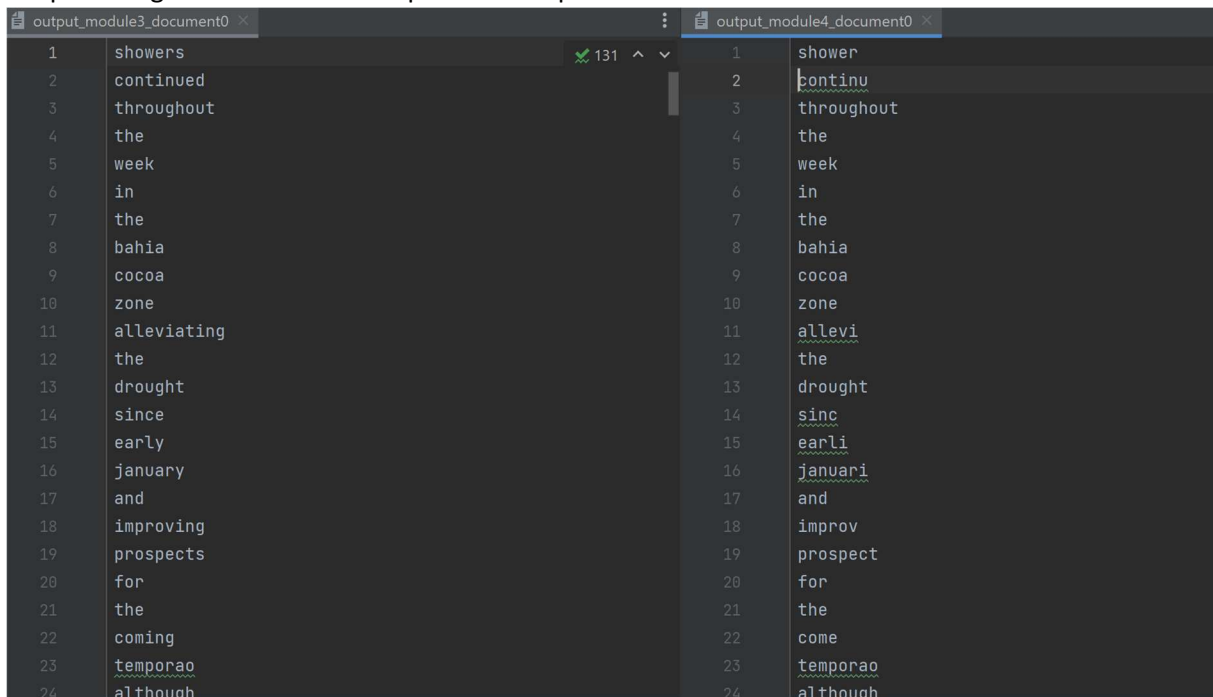


These are the output files of module 1 and 2 for document 0. You can see that module 1 properly extracted the articles' bodies and module 2 tokenized the text and wrote every token in a separate line.



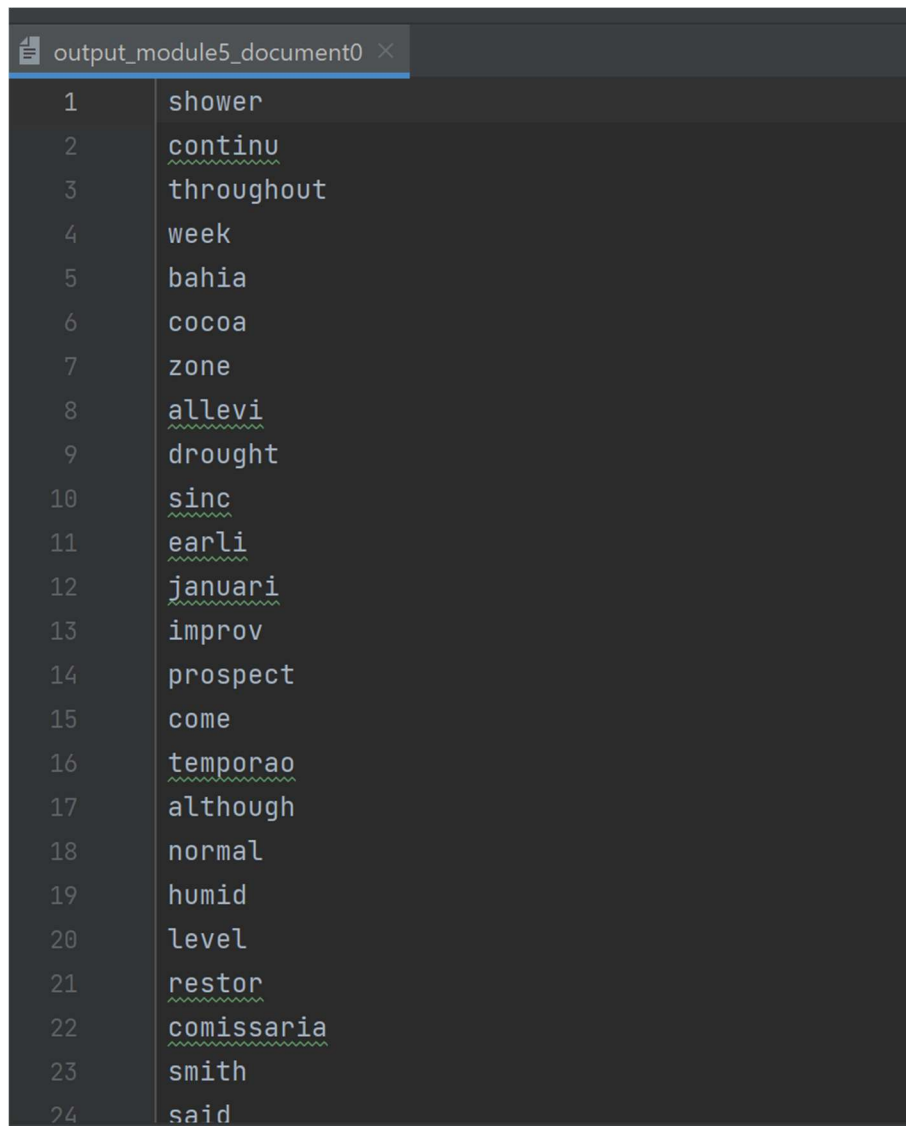
The screenshot shows two side-by-side code editor windows. The left window, titled 'output_module1_document0', displays a long paragraph of text extracted from a document. The text is wrapped and includes words like 'Showers', 'cocoa', 'Bahia', 'drought', and 'temporao'. The right window, titled 'output_module2_document0', shows the same text tokenized into individual words, one per line, in lowercase. The tokens are: 1 Showers, 2 continued, 3 throughout, 4 the, 5 week, 6 in, 7 the, 8 Bahia, 9 cocoa, 10 zone, 11 alleviating, 12 the, 13 drought, 14 since, 15 early, 16 January, 17 and, 18 improving, 19 prospects, 20 for, 21 the, 22 coming, 23 temporao, 24 although.

These are the output files of module 3 and 4 for document 0. You can see that module 3 successfully made all the tokens lowercase and module 4 successfully stemmed all the tokens. Both modules keep the original structure of the previous output files.



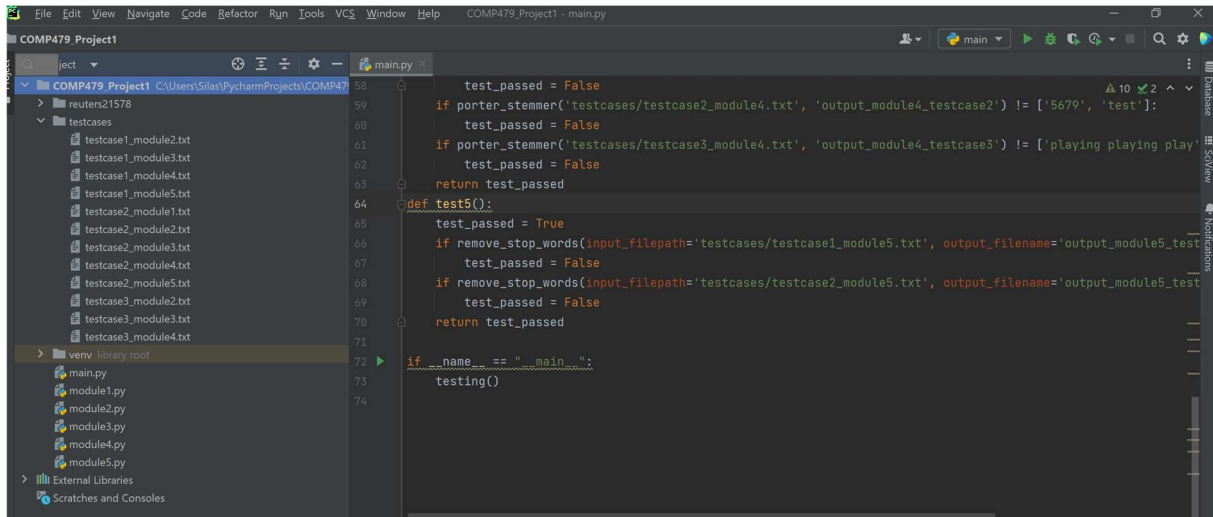
The screenshot shows two side-by-side code editor windows. The left window, titled 'output_module3_document0', displays the tokens from the previous window, all converted to lowercase. The tokens are: 1 showers, 2 continued, 3 throughout, 4 the, 5 week, 6 in, 7 the, 8 bahia, 9 cocoa, 10 zone, 11 alleviating, 12 the, 13 drought, 14 since, 15 early, 16 january, 17 and, 18 improving, 19 prospects, 20 for, 21 the, 22 coming, 23 temporao, 24 although. The right window, titled 'output_module4_document0', shows the tokens from the previous window, all converted to lowercase and stemmed. The tokens are: 1 shower, 2 continu, 3 throughout, 4 the, 5 week, 6 in, 7 the, 8 bahia, 9 cocoa, 10 zone, 11 allevi, 12 the, 13 drought, 14 sinc, 15 earli, 16 januari, 17 and, 18 improv, 19 prospect, 20 for, 21 the, 22 come, 23 temporao, 24 although.

This screenshot shows the output file of the module 5 for document 0. You can see that some stop words were removed for example the word 'for' was removed because it is in the English stop words list of NLTK.



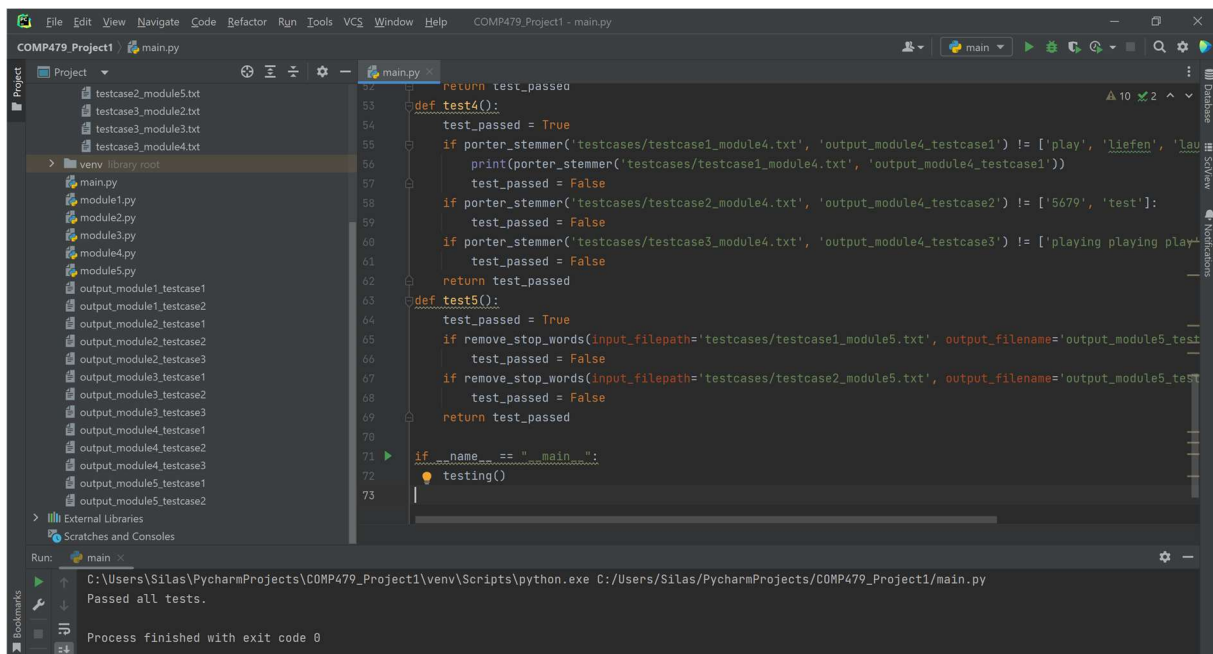
```
output_module5_document0 x
1 shower
2 continu
3 throughout
4 week
5 bahia
6 cocoa
7 zone
8 allevi
9 drought
10 sinc
11 earli
12 januari
13 improv
14 prospect
15 come
16 temporao
17 although
18 normal
19 humid
20 level
21 restor
22 comissaria
23 smith
24 said
```

This screenshot shows how you test the pipeline. I only ran the testing without processing the reuters files in this screenshot. By default, the testing and the processing of the files are both started automatically when running the main.py file.



```
58 test_passed = False
59 if porter_stemmer('testcases/testcase2_module4.txt', 'output_module4_testcase2') != ['5679', 'test']:
60     test_passed = False
61 if porter_stemmer('testcases/testcase3_module4.txt', 'output_module4_testcase3') != ['playing playing play']:
62     test_passed = False
63 return test_passed
64
65 def test5():
66     test_passed = True
67     if remove_stop_words(input_filepath='testcases/testcase1_module5.txt', output_filename='output_module5_testcase1'):
68         test_passed = False
69     if remove_stop_words(input_filepath='testcases/testcase2_module5.txt', output_filename='output_module5_testcase2'):
70         test_passed = False
71     return test_passed
72
73 if __name__ == "__main__":
74     testing()
```

This screenshot shows the output files that are created after testing. Also, if the pipeline passes all tests 'Passed all tests.' is printed in the console.



```
52 return test_passed
53
54 def test4():
55     test_passed = True
56     if porter_stemmer('testcases/testcase1_module4.txt', 'output_module4_testcase1') != ['play', 'liefen', 'lau']:
57         print(porter_stemmer('testcases/testcase1_module4.txt', 'output_module4_testcase1'))
58         test_passed = False
59     if porter_stemmer('testcases/testcase2_module4.txt', 'output_module4_testcase2') != ['5679', 'test']:
60         test_passed = False
61     if porter_stemmer('testcases/testcase3_module4.txt', 'output_module4_testcase3') != ['playing playing play']:
62         test_passed = False
63     return test_passed
64
65 def test5():
66     test_passed = True
67     if remove_stop_words(input_filepath='testcases/testcase1_module5.txt', output_filename='output_module5_testcase1'):
68         test_passed = False
69     if remove_stop_words(input_filepath='testcases/testcase2_module5.txt', output_filename='output_module5_testcase2'):
70         test_passed = False
71     return test_passed
72
73 if __name__ == "__main__":
74     testing()
```

Run: C:\Users\Silas\PycharmProjects\COMP479_Project1\venv\Scripts\python.exe C:\Users\Silas\PycharmProjects\COMP479_Project1/main.py
Passed all tests.
Process finished with exit code 0

I certify that this submission is my original work and meets the Faculty's Expectations of Originality.

Signature

ID: 40256077

Date 2022-09-22

S. Pelimowski