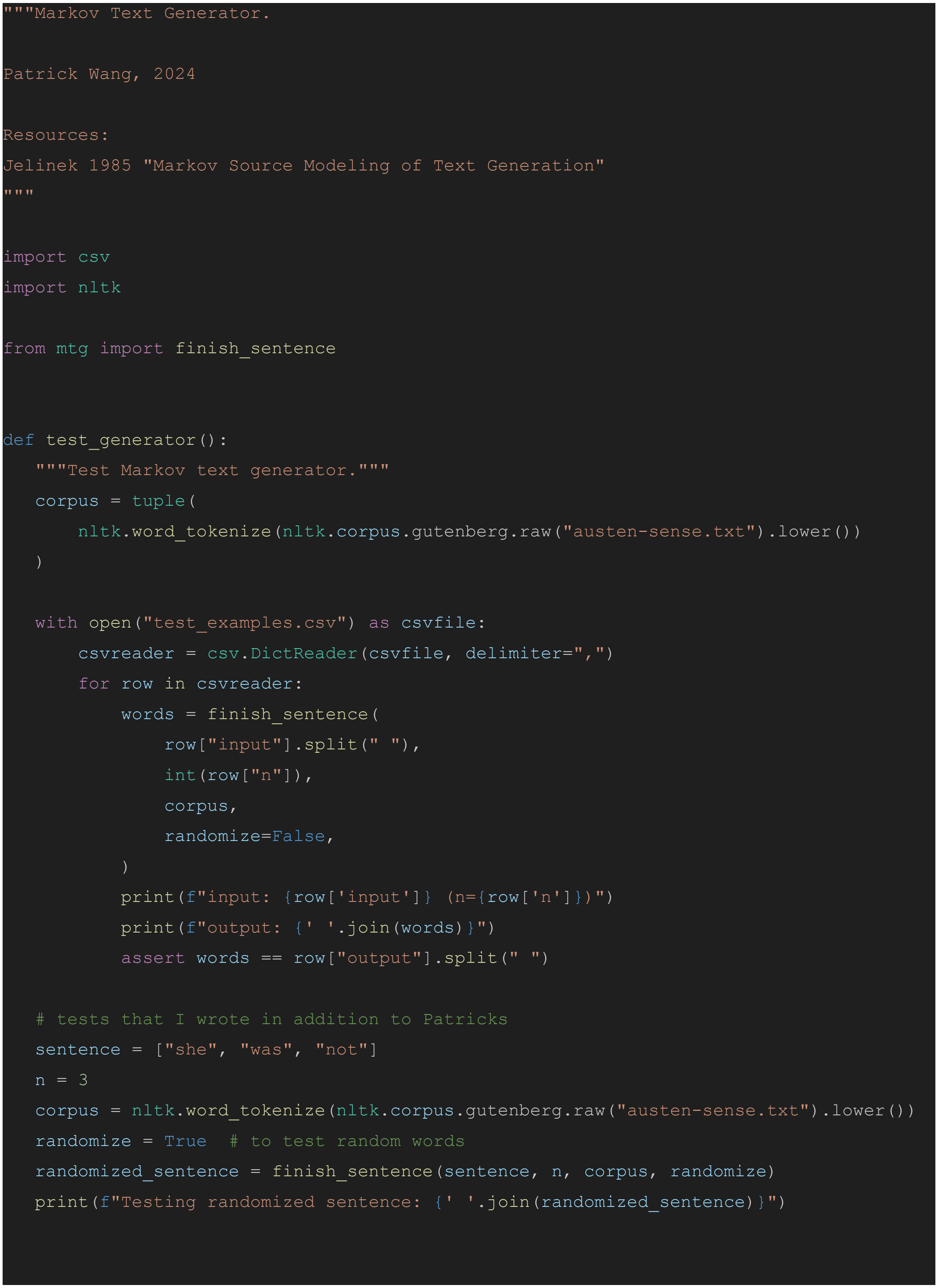
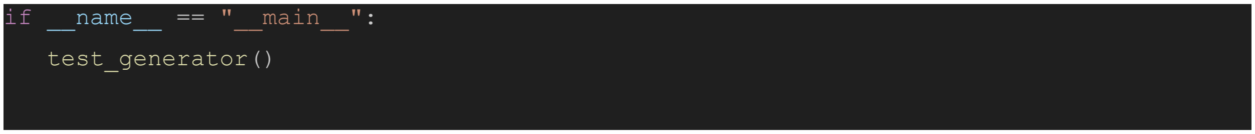


```
"""Markov Text Generator.

Patrick Wang, 2024

Resources:
Jelinek 1985 "Markov Source Modeling of Text Generation"
"""

import csv
import nltk

from mtg import finish_sentence


def test_generator():
    """Test Markov text generator."""
    corpus = tuple(
        nltk.word_tokenize(nltk.corpus.gutenberg.raw("austen-sense.txt").lower())
    )

    with open("test_examples.csv") as csvfile:
        csvreader = csv.DictReader(csvfile, delimiter=",")
        for row in csvreader:
            words = finish_sentence(
                row["input"].split(" "),
                int(row["n"]),
                corpus,
                randomize=False,
            )
            print(f"input: {row['input']} (n={row['n']})")
            print(f"output: {' '.join(words)}")
            assert words == row["output"].split(" ")

    # tests that I wrote in addition to Patricks
    sentence = ["she", "was", "not"]
    n = 3
    corpus = nltk.word_tokenize(nltk.corpus.gutenberg.raw("austen-sense.txt").lower())
    randomize = True  # to test random words
    randomized_sentence = finish_sentence(sentence, n, corpus, randomize)
    print(f"Testing randomized sentence: {' '.join(randomized_sentence)}")
```



```python
if __name__ == "__main__":
   test_generator()
```

OUTPUT

```
(base) ➜  Markov_Text_Gen_Assignment2 /opt/miniconda3/bin/python
/Users/kaylahaeusssler/Documents/NLP/Markov_Text_Gen_Assignment2/te
st_
mtg.py
input: she was not (n=2)
output: she was not be a very well , and the
input: she was not (n=3)
output: she was not in the world , and the two
input: i would ask her (n=3)
output: i would ask her reason , and the two miss
input: i would ask her (n=4)
output: i would ask her reason for thinking so , because
input: they were sorry (n=4)
output: they were sorry to see them , and always openly
input: they were sorry (n=5)
output: they were sorry to see them .
Testing randomized sentence: she was not jennings decided said her reading
and so
```