EX.NO.: 08

ANALYSE SENTENCE

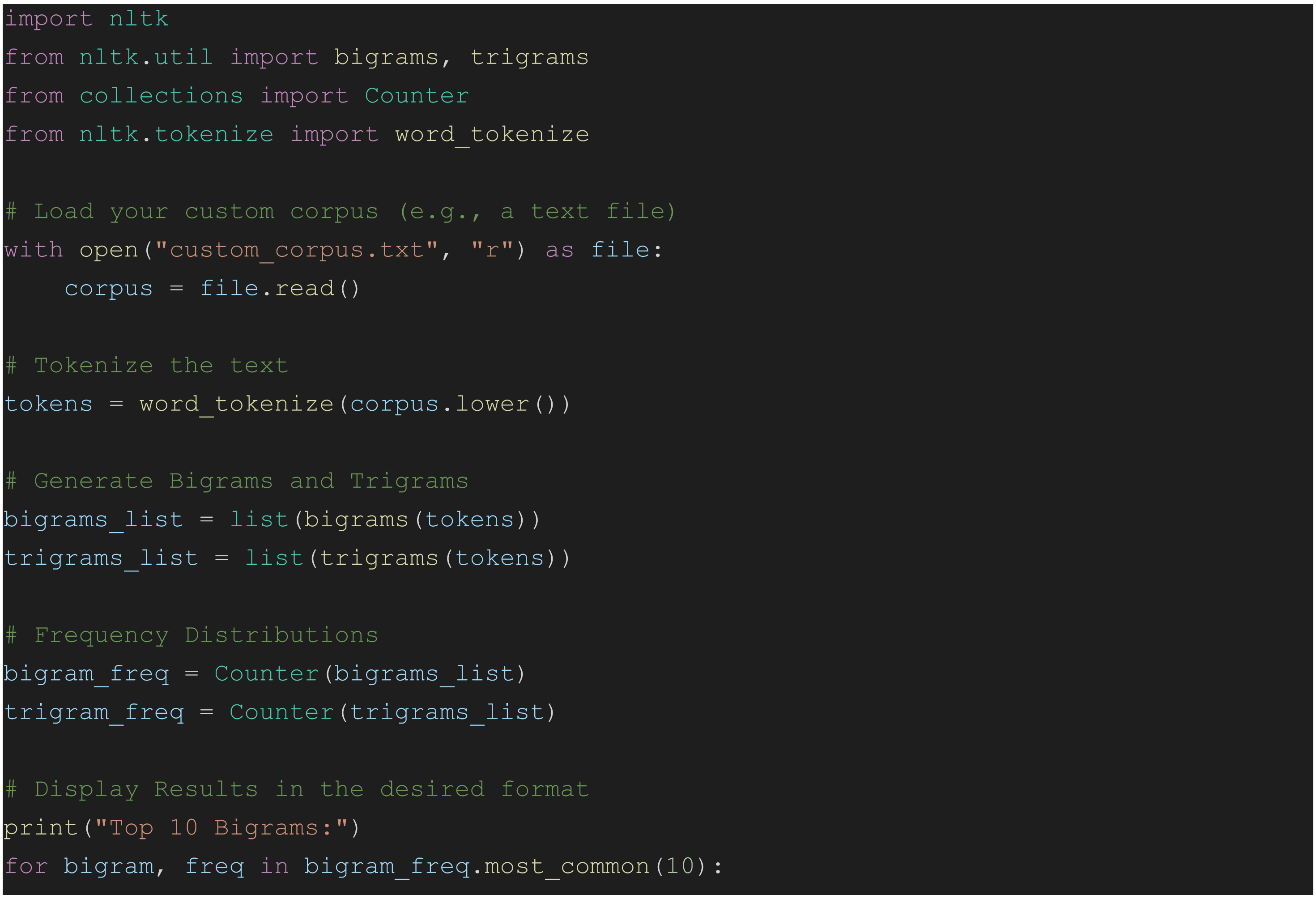
DATE: 24.01.2025

To analyze a given corpus and generate the top 10 most frequent bigrams and trigrams, formatted in a readable manner, to understand word associations and patterns within the text.

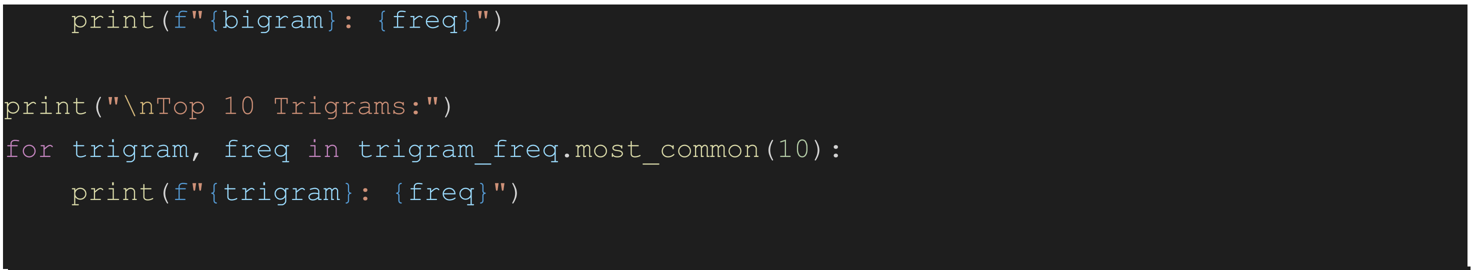
PROCEDURE:

1. Import Necessary Libraries:
   a. nltk, bigrams and trigrams, nltk.util, Counter
2. Load the Corpus:
   a. Open and read the text from a file (e.g., custom_corpus.txt).
   b. Store the text in a variable for processing.
3. Preprocess the Text:
   a. Tokenize the text into individual words using nltk.word_tokenize.
   b. Convert all words to lowercase to ensure case insensitivity in the analysis.
4. Generate Bigrams and Trigrams:
   a. Create a list of bigrams using the bigrams() function from the tokenized words.
   b. Similarly, create a list of trigrams using the trigrams() function.
5. Calculate Frequencies:
   a. Use the Counter class to calculate the frequency of each bigram and trigram in the respective lists.
6. Display Results:
   a. Retrieve the top 10 most frequent bigrams and trigrams using the most_common() method of the Counter object.
   b. Loop through the results and format the output as ('word1', 'word2'): frequency for bigrams and ('word1', 'word2', 'word3'): frequency for trigrams.
7. Run the Program:
   a. Execute the script and observe the results printed in the desired format.

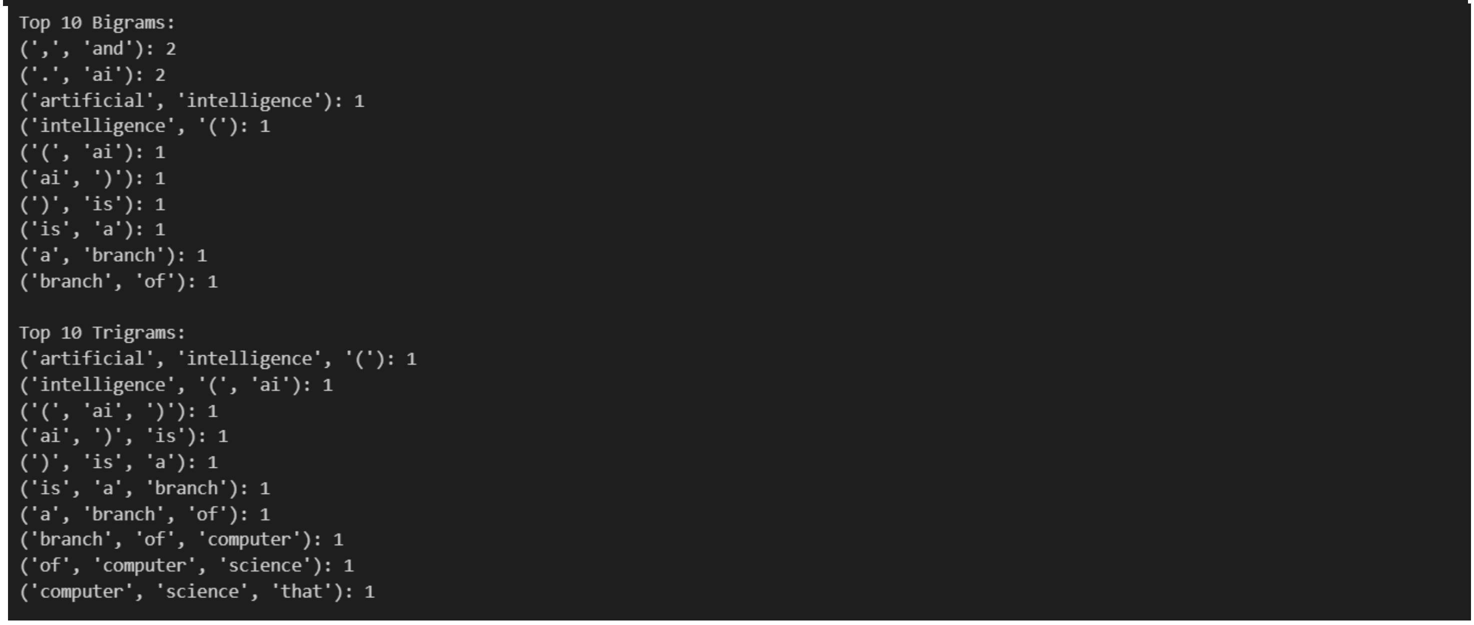
CODE AND OUTPUT



```
import nltk
from nltk.util import bigrams, trigrams
from collections import Counter
from nltk.tokenize import word_tokenize

# Load your custom corpus (e.g., a text file)
with open("custom_corpus.txt", "r") as file:
    corpus = file.read()

# Tokenize the text
tokens = word_tokenize(corpus.lower())

# Generate Bigrams and Trigrams
bigrams_list = list(bigrams(tokens))
trigrams_list = list(trigrams(tokens))

# Frequency Distributions
bigram_freq = Counter(bigrams_list)
trigram_freq = Counter(trigrams_list)

# Display Results in the desired format
print("Top 10 Bigrams:")
for bigram, freq in bigram_freq.most_common(10):
```



```python
    print(f"{bigram}: {freq}")

print("\nTop 10 Trigrams:")
for trigram, freq in trigram_freq.most_common(10):
    print(f"{trigram}: {freq}")
```



```
Top 10 Bigrams:
(',', 'and'): 2
('.', 'ai'): 2
('artificial', 'intelligence'): 1
('intelligence', '('): 1
('(', 'ai'): 1
('ai', ')'): 1
(')', 'is'): 1
('is', 'a'): 1
('a', 'branch'): 1
('branch', 'of'): 1

Top 10 Trigrams:
('artificial', 'intelligence', '('): 1
('intelligence', '(', 'ai'): 1
('(', 'ai', ')'): 1
('ai', ')', 'is'): 1
(')', 'is', 'a'): 1
('is', 'a', 'branch'): 1
('a', 'branch', 'of'): 1
('branch', 'of', 'computer'): 1
('of', 'computer', 'science'): 1
('computer', 'science', 'that'): 1
```