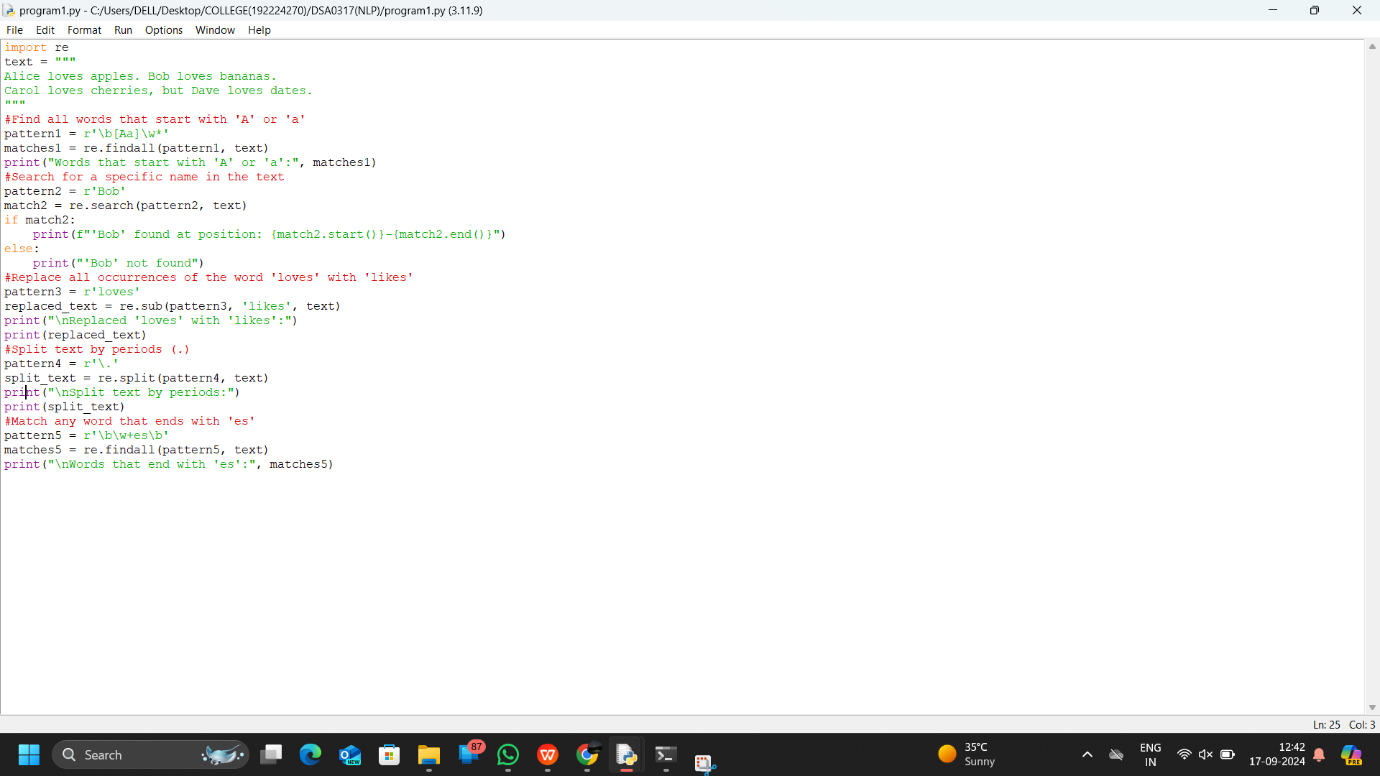
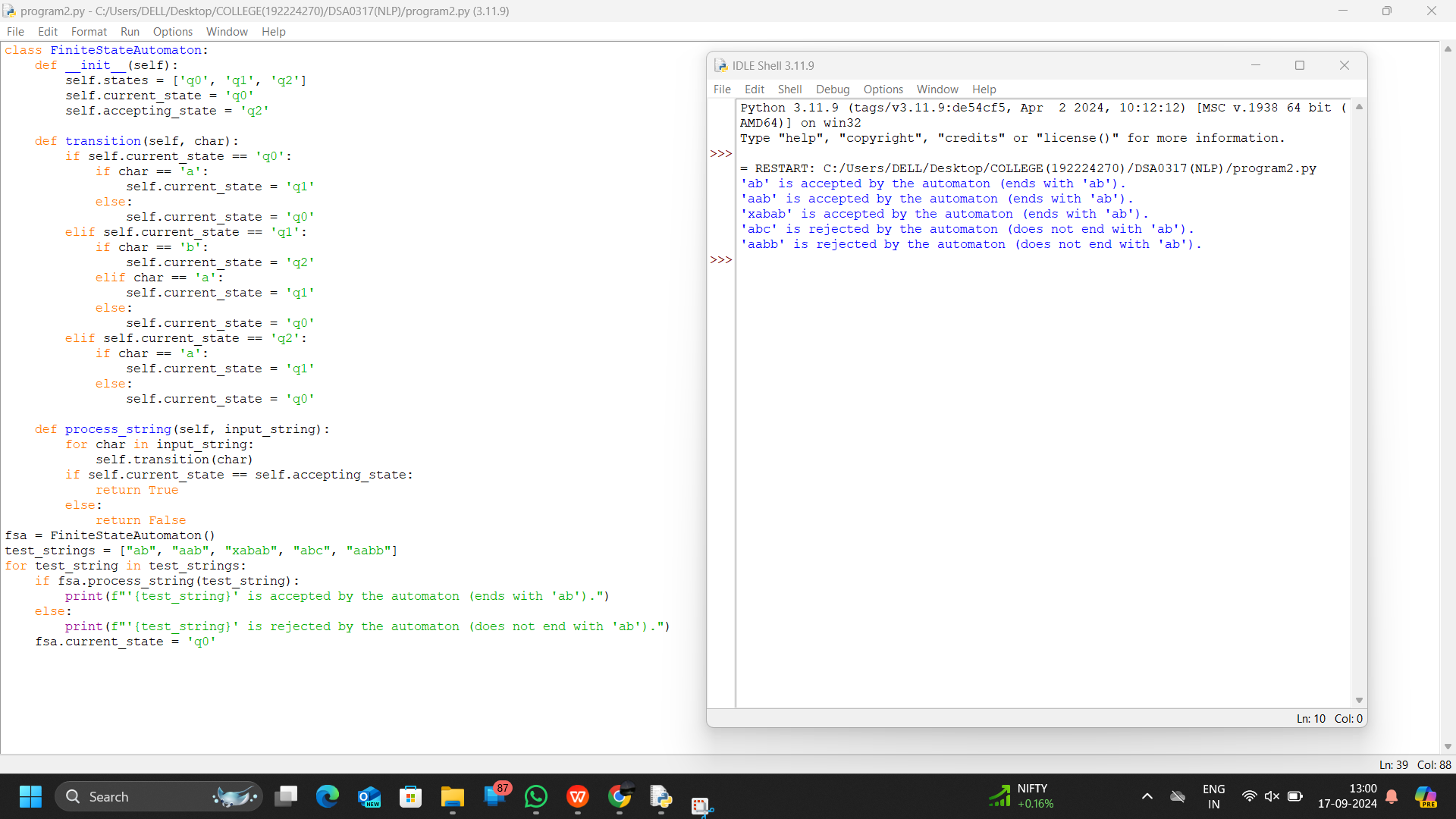
**SATHYA NARAYANAN E (192224252)**

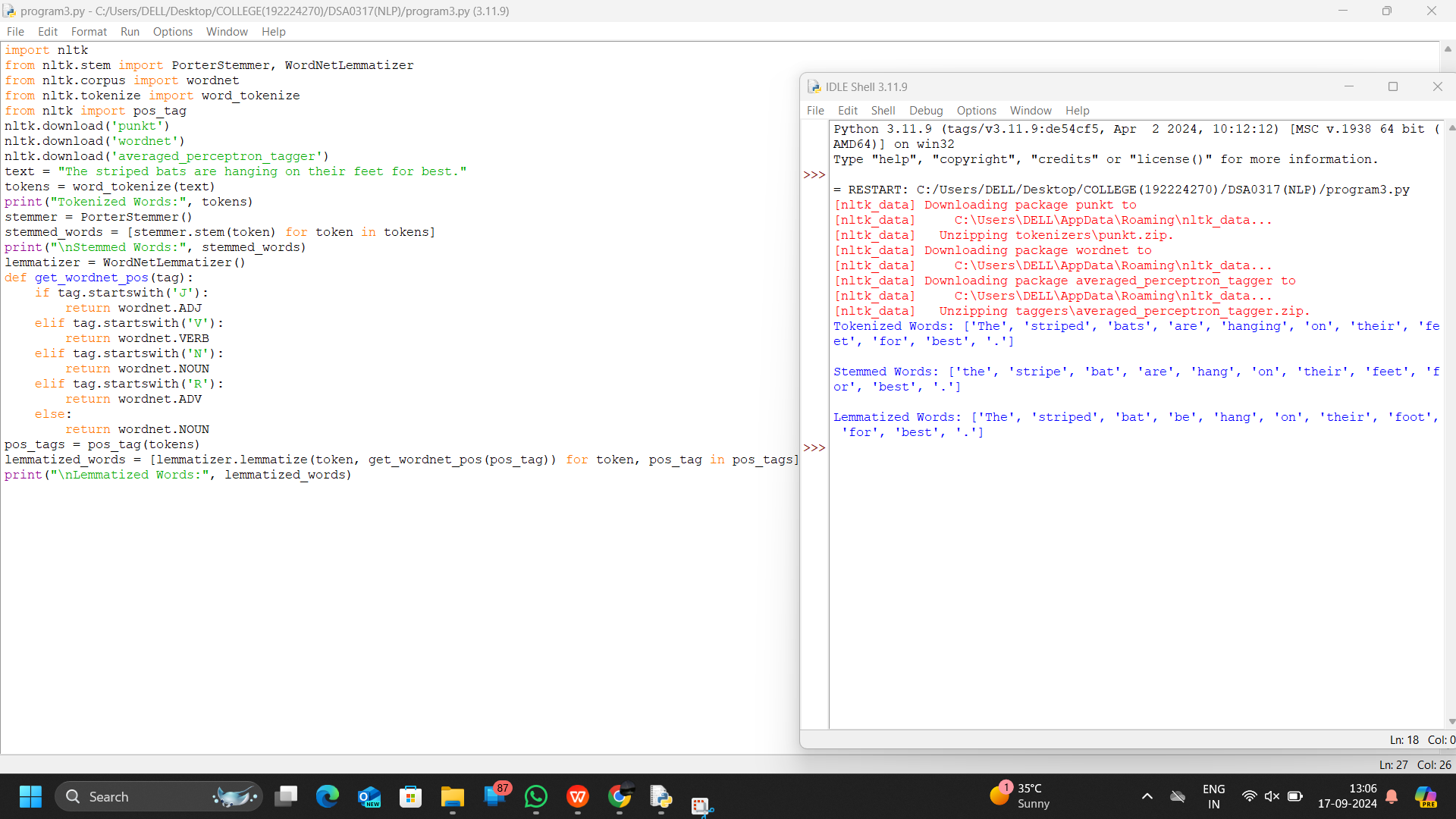
1. Write program demonstrates how to use regular expressions in Python to match and search for patterns in text.



1. Implement a basic finite state automaton that recognizes a specific language or pattern. In this example, we'll create a simple automaton to match strings ending with 'ab' using python.



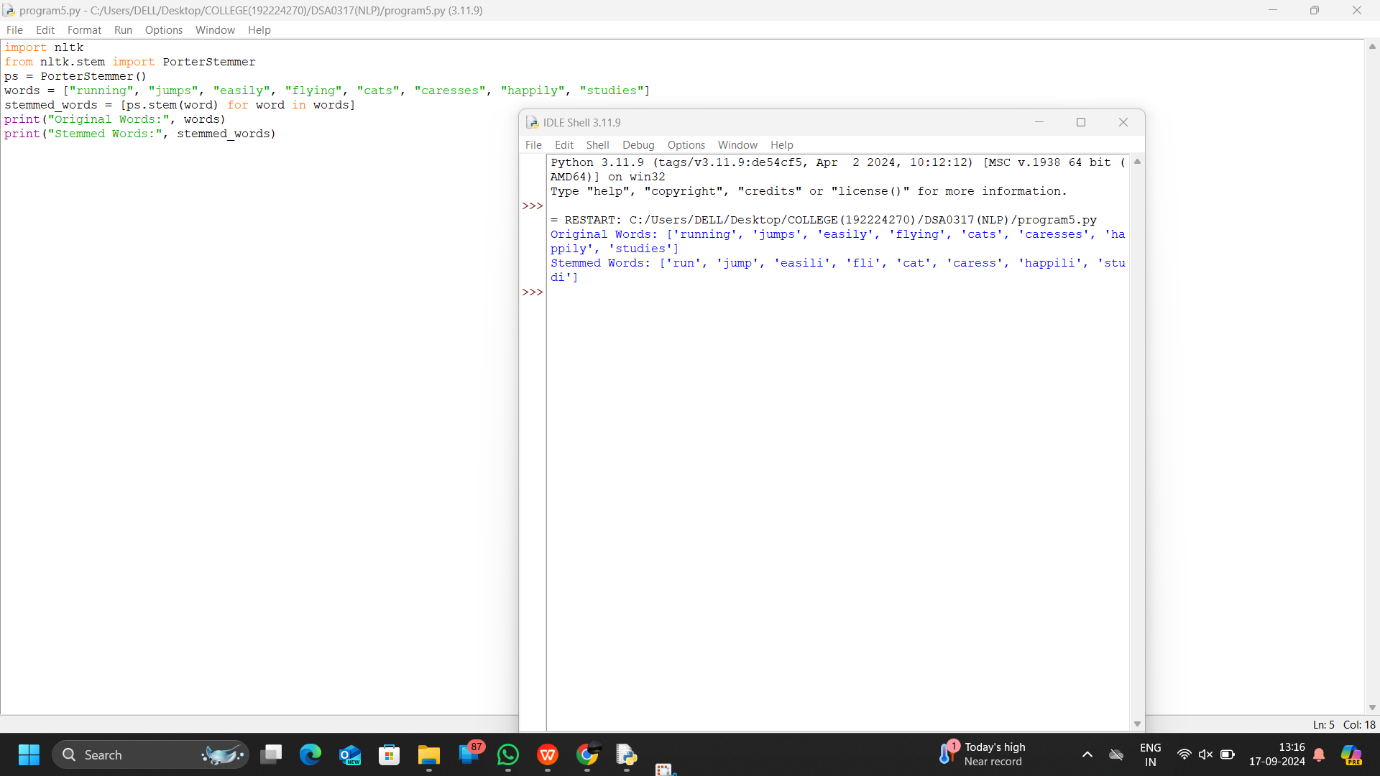
1. Write program demonstrates how to perform morphological analysis using the NLTK library in Python.



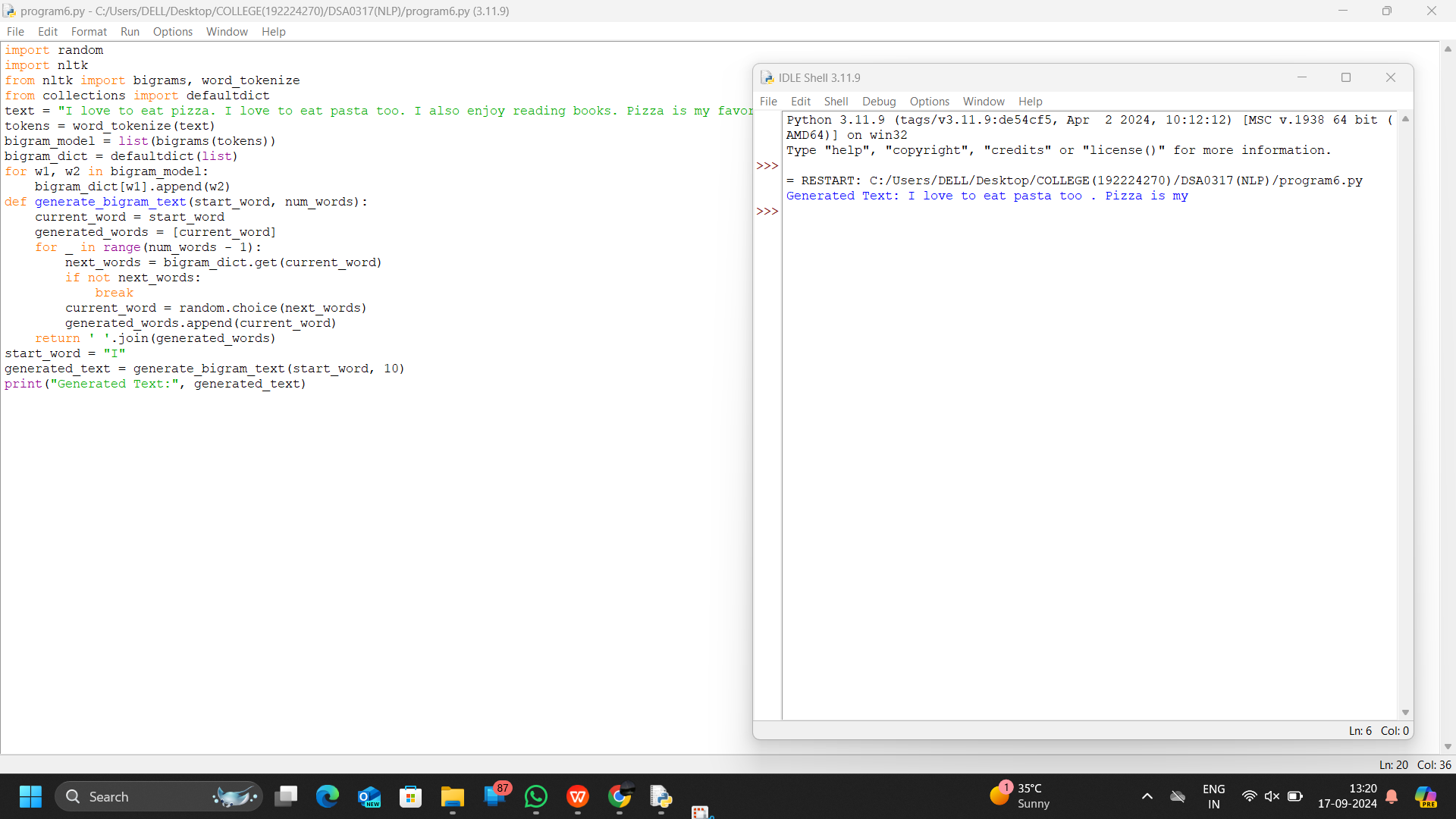
1. Implement a finite-state machine for morphological parsing. In this example, we'll create a simple machine to generate plural forms of English nouns using python.



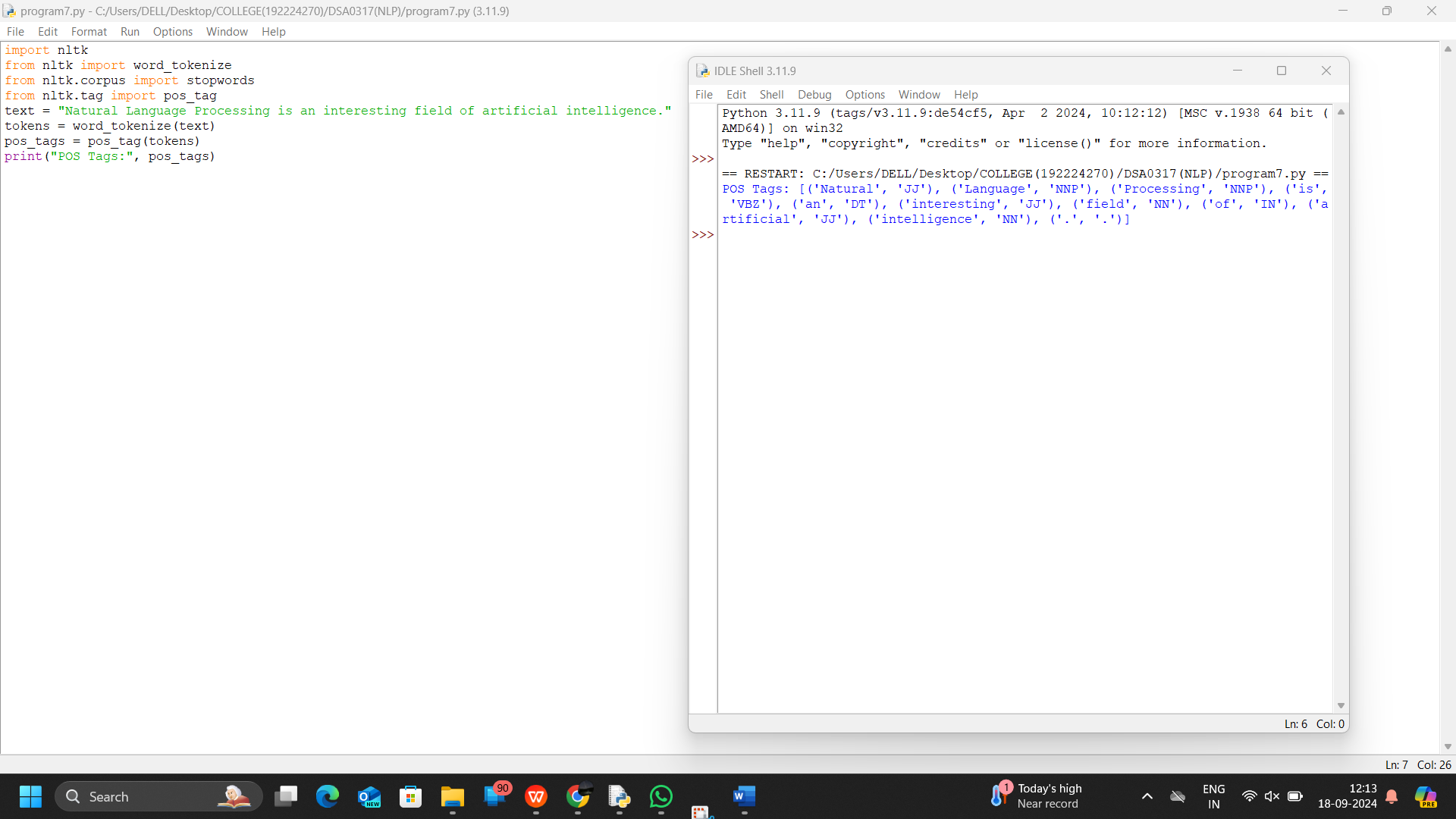
1. Use the Porter Stemmer algorithm to perform word stemming on a list of words using python libraries.



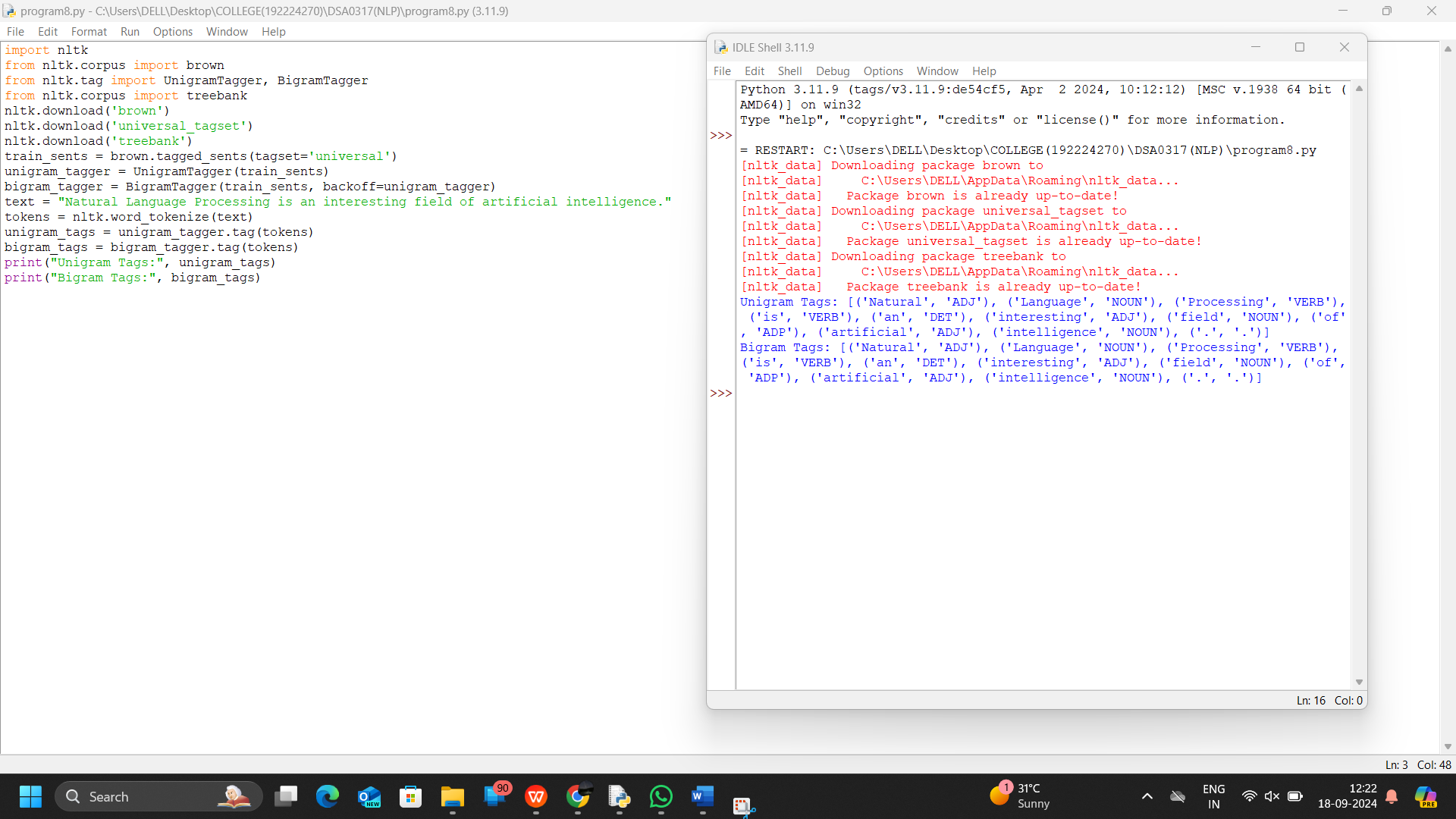
1. Implement a basic N-gram model for text generation. For example, generate text using a bigram model using python.



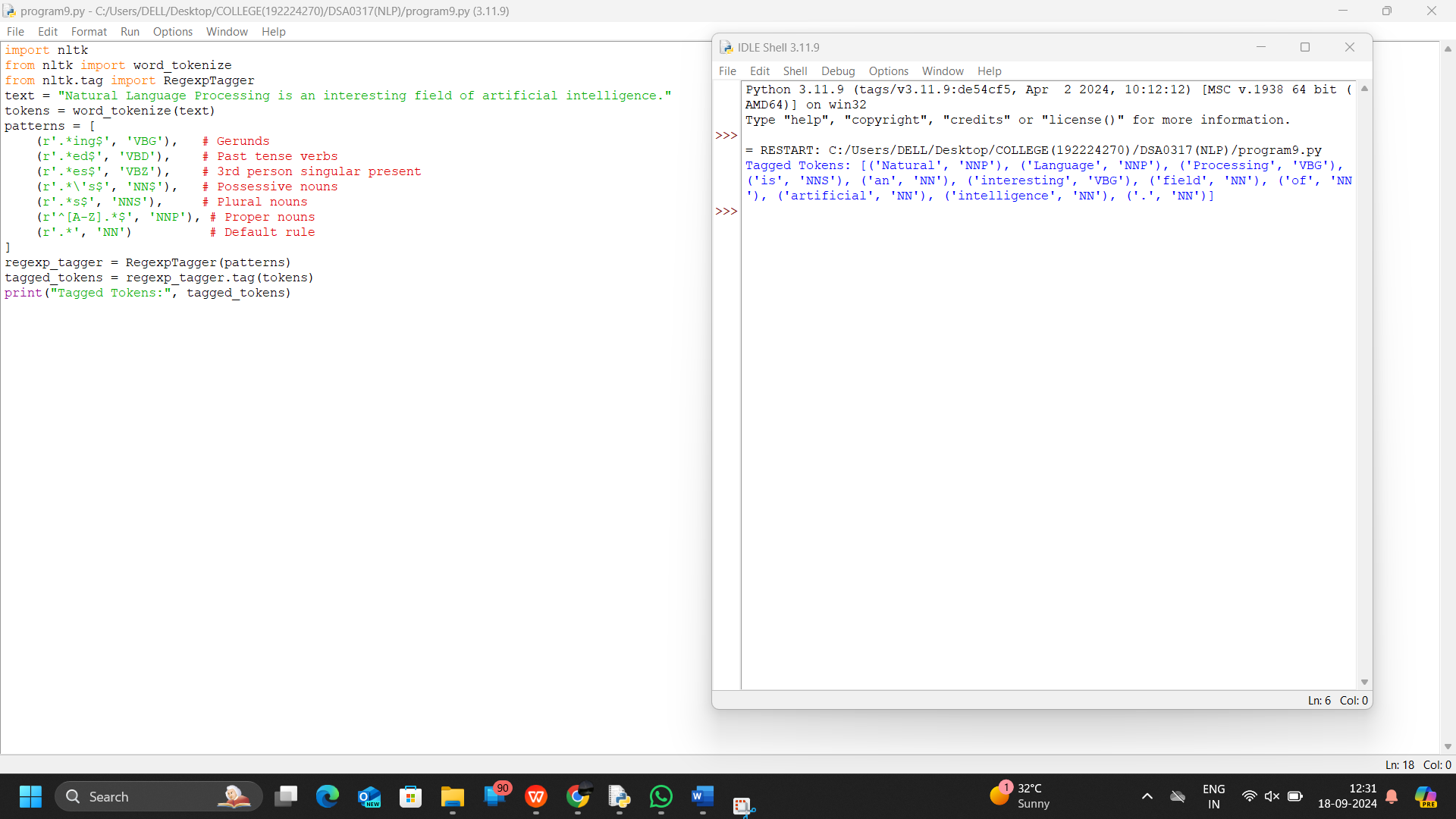
1. Write program using the NLTK library to perform part-of-speech tagging on a text.



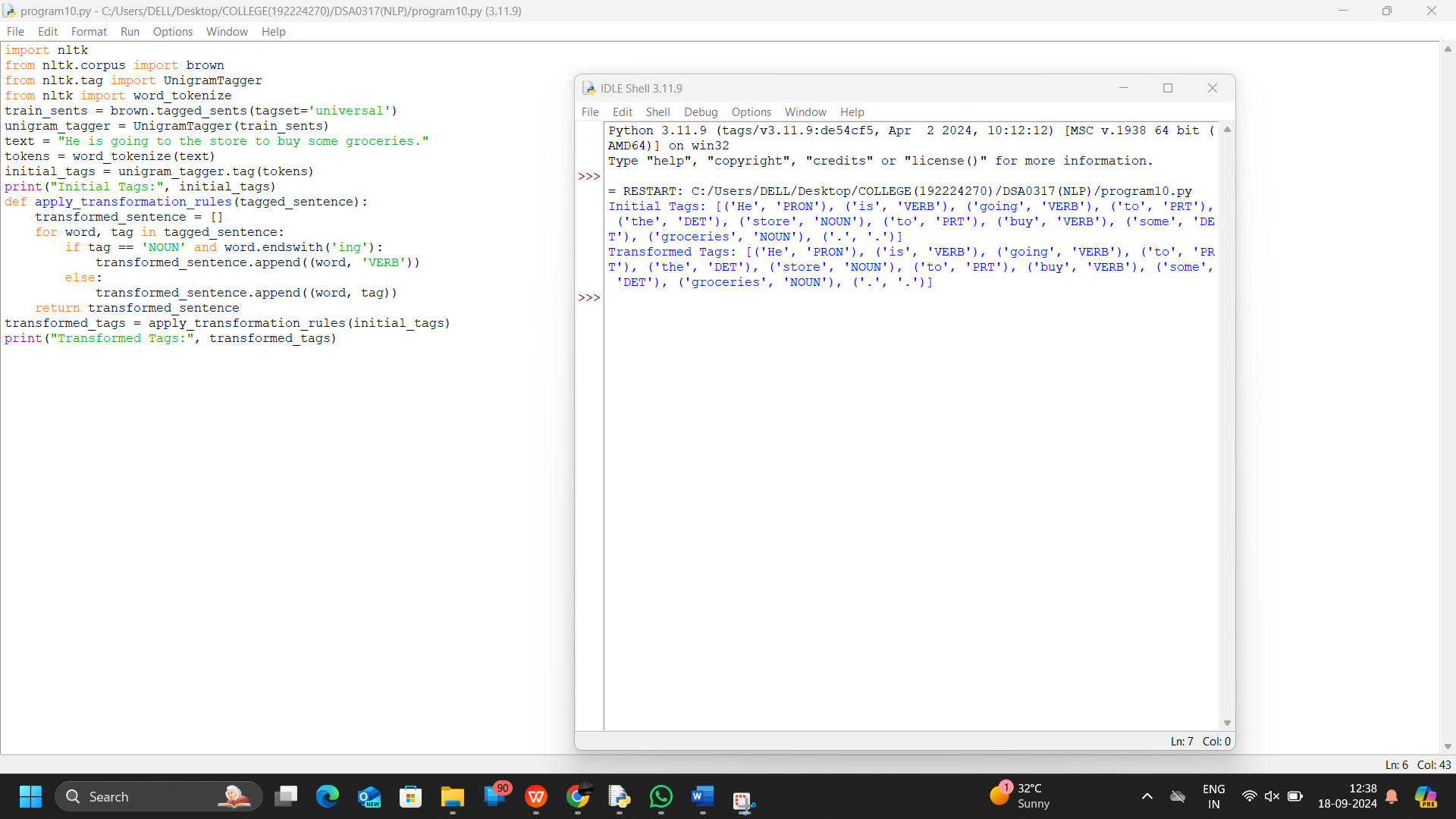
1. Implement a simple stochastic part-of-speech tagging algorithm using a basic probabilistic model to assign POS tags using python.



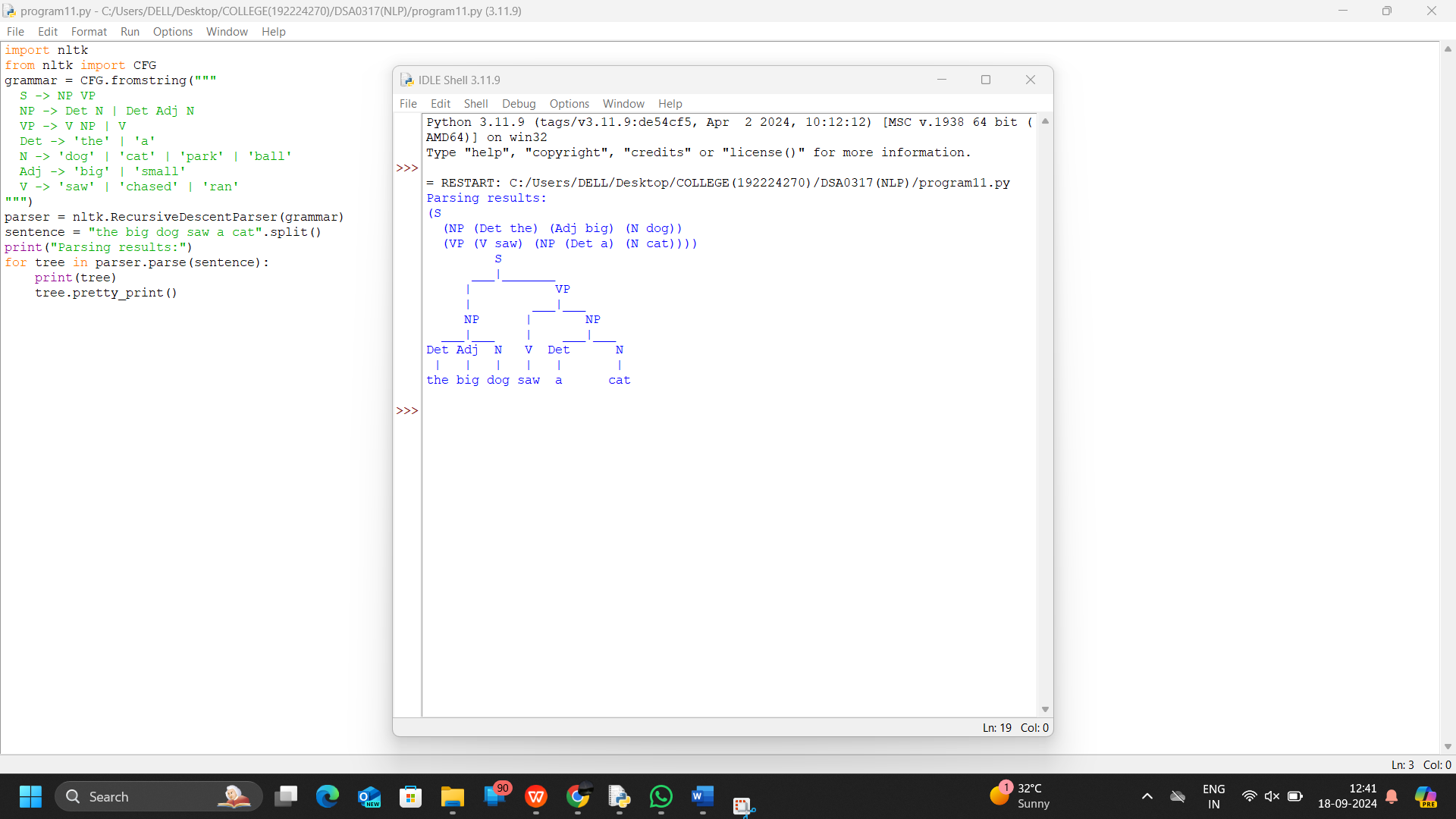
1. Implement a rule-based part-of-speech tagging system using regular expressions using python.



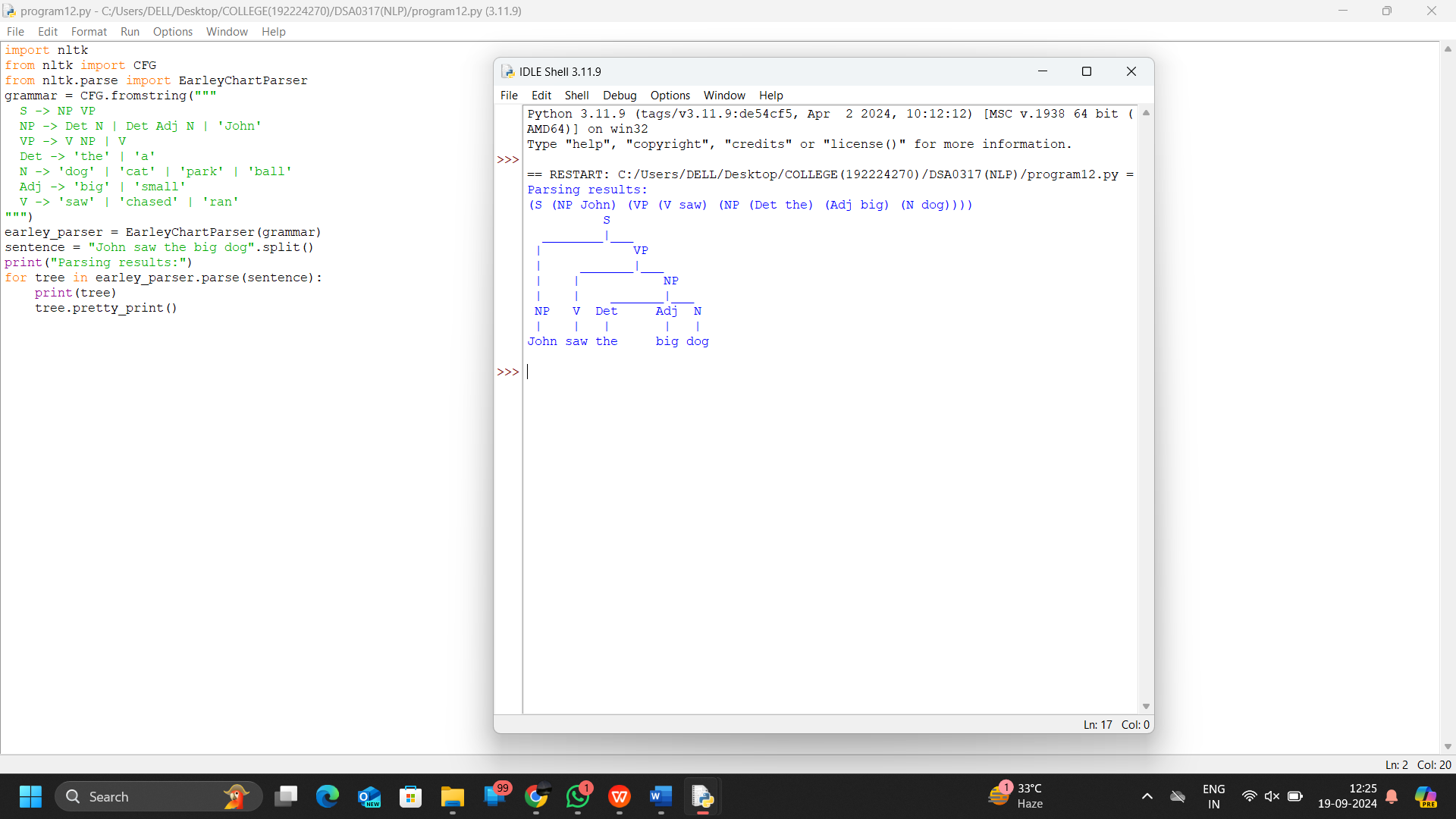
1. Implement transformation-based tagging using a set of transformation rules, apply a simple rule to tag words using python.



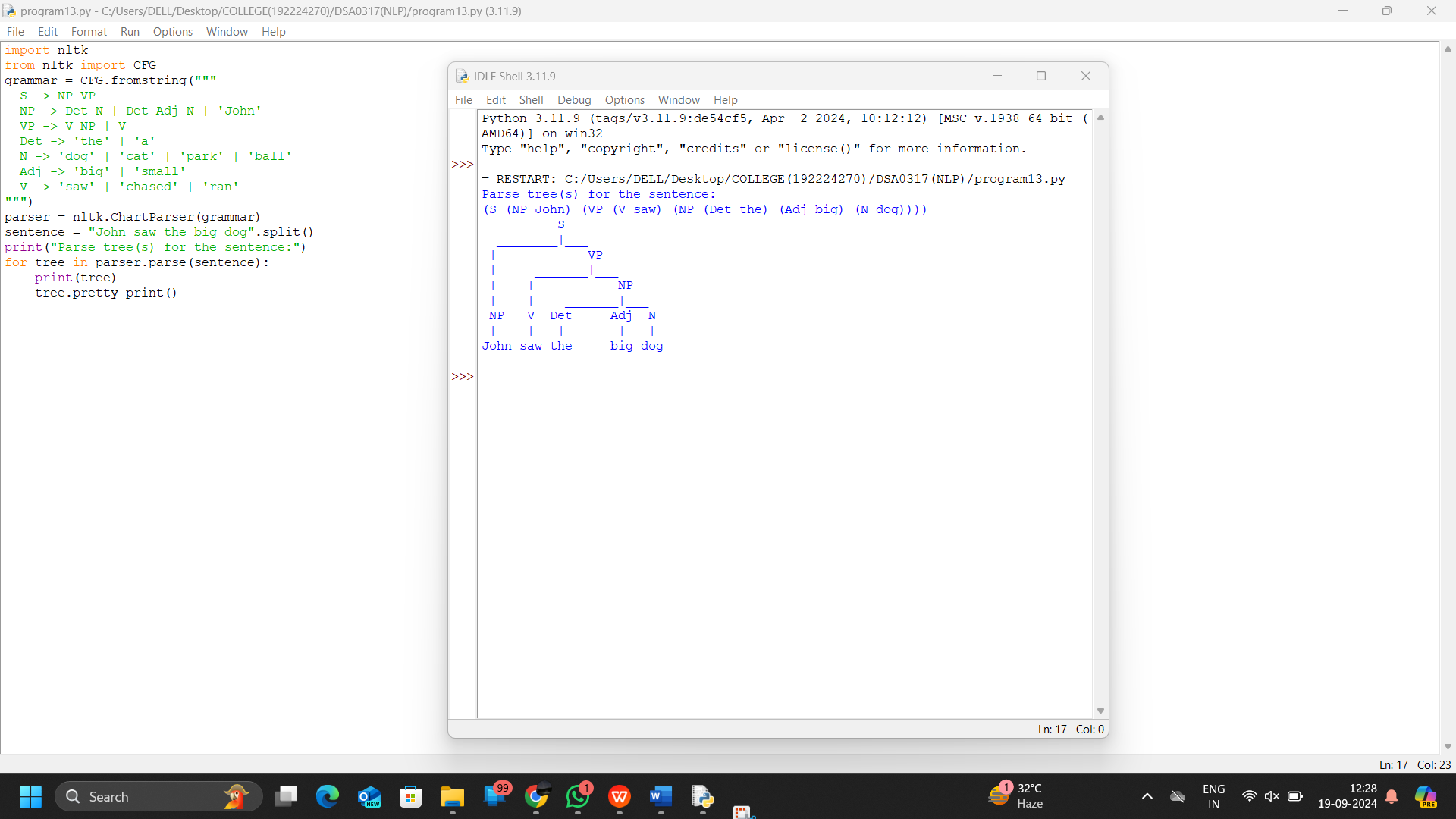
1. Implement a simple top-down parser for context-free grammars using python.



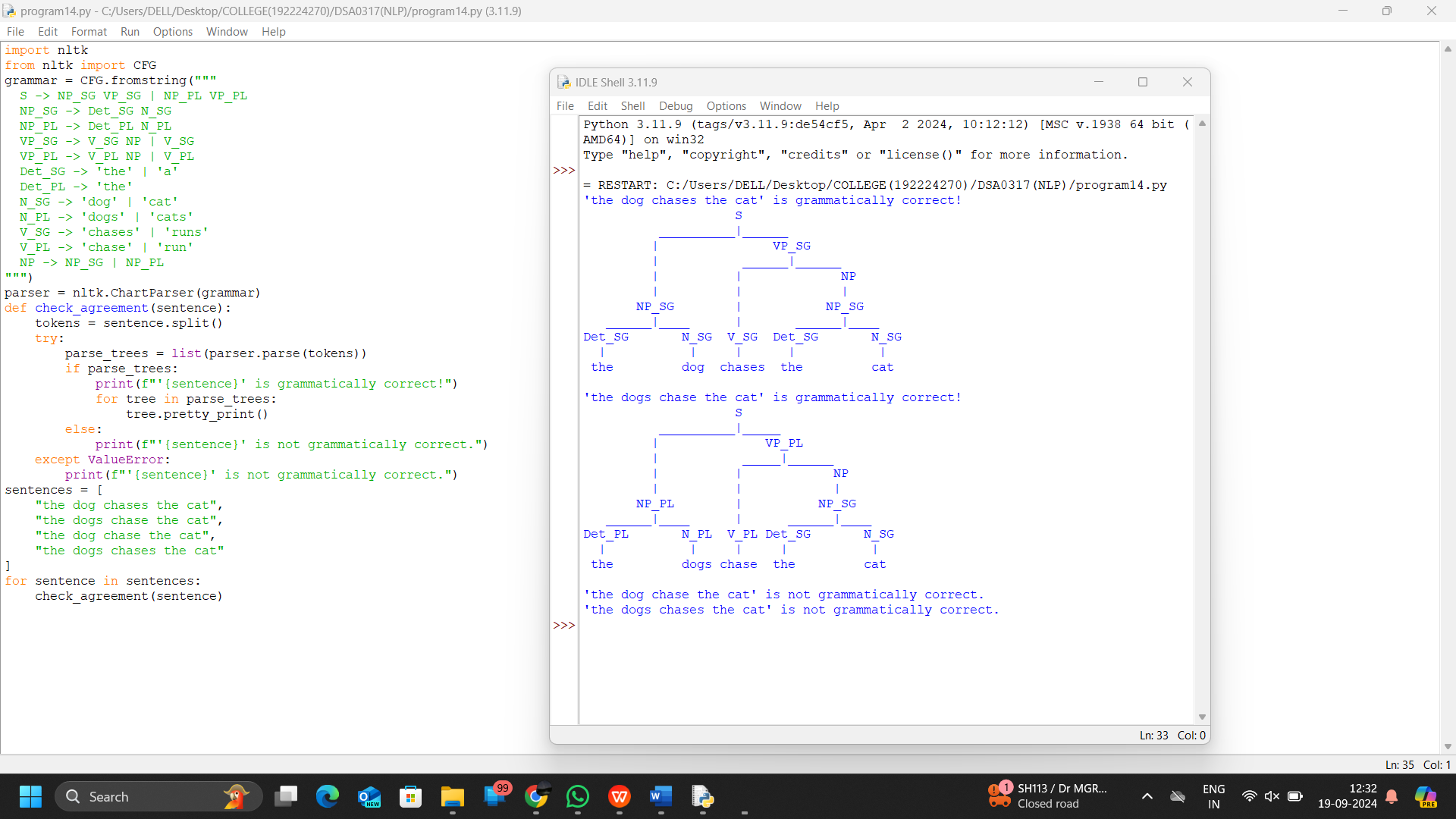
1. Implement an Earley parser for context-free grammars using a simple python program.



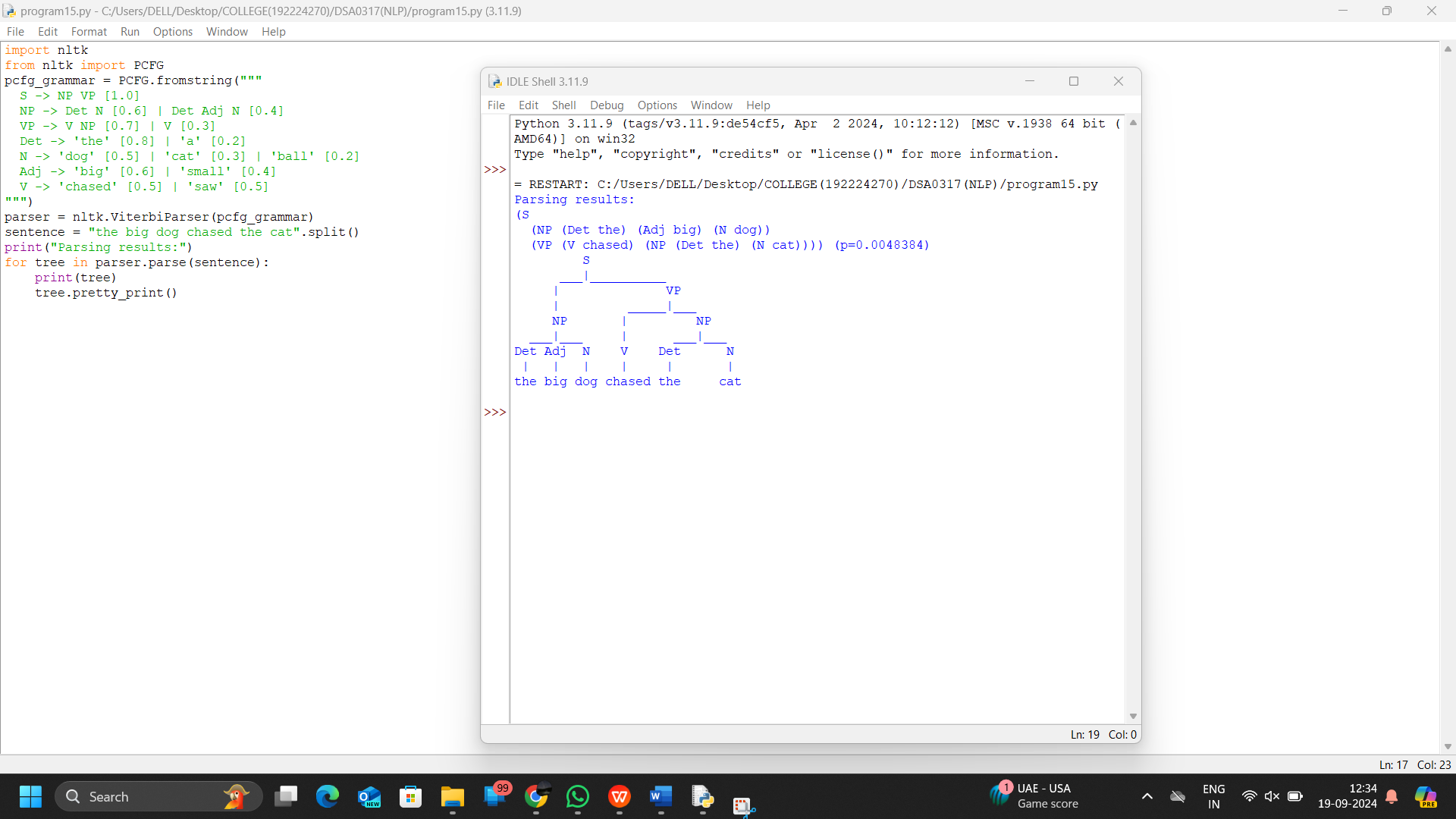
1. Generate a parse tree for a given sentence using a context-free grammar using python program.



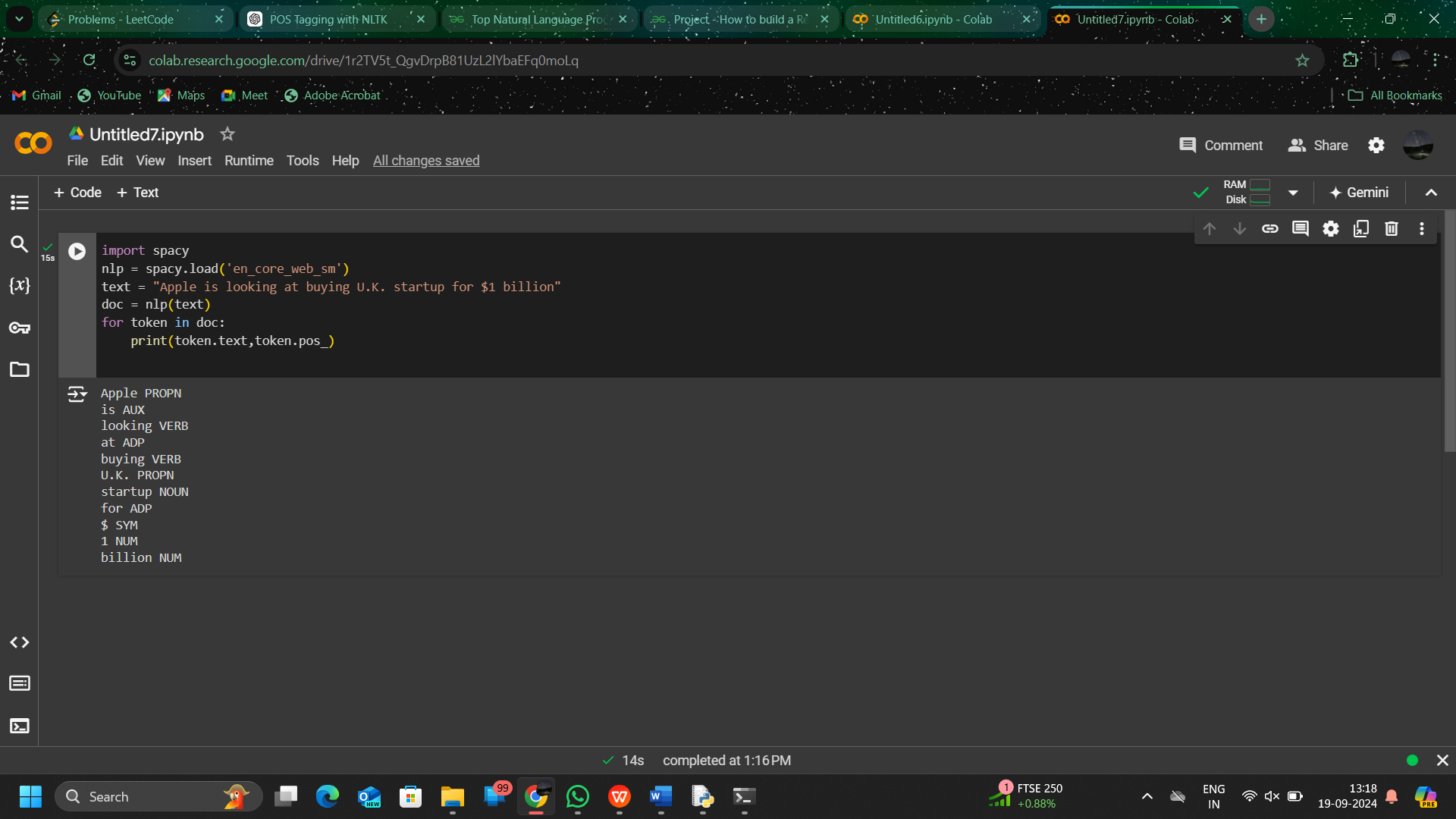
1. Create a program in python to check for agreement in sentences based on a context-free grammar's rules.



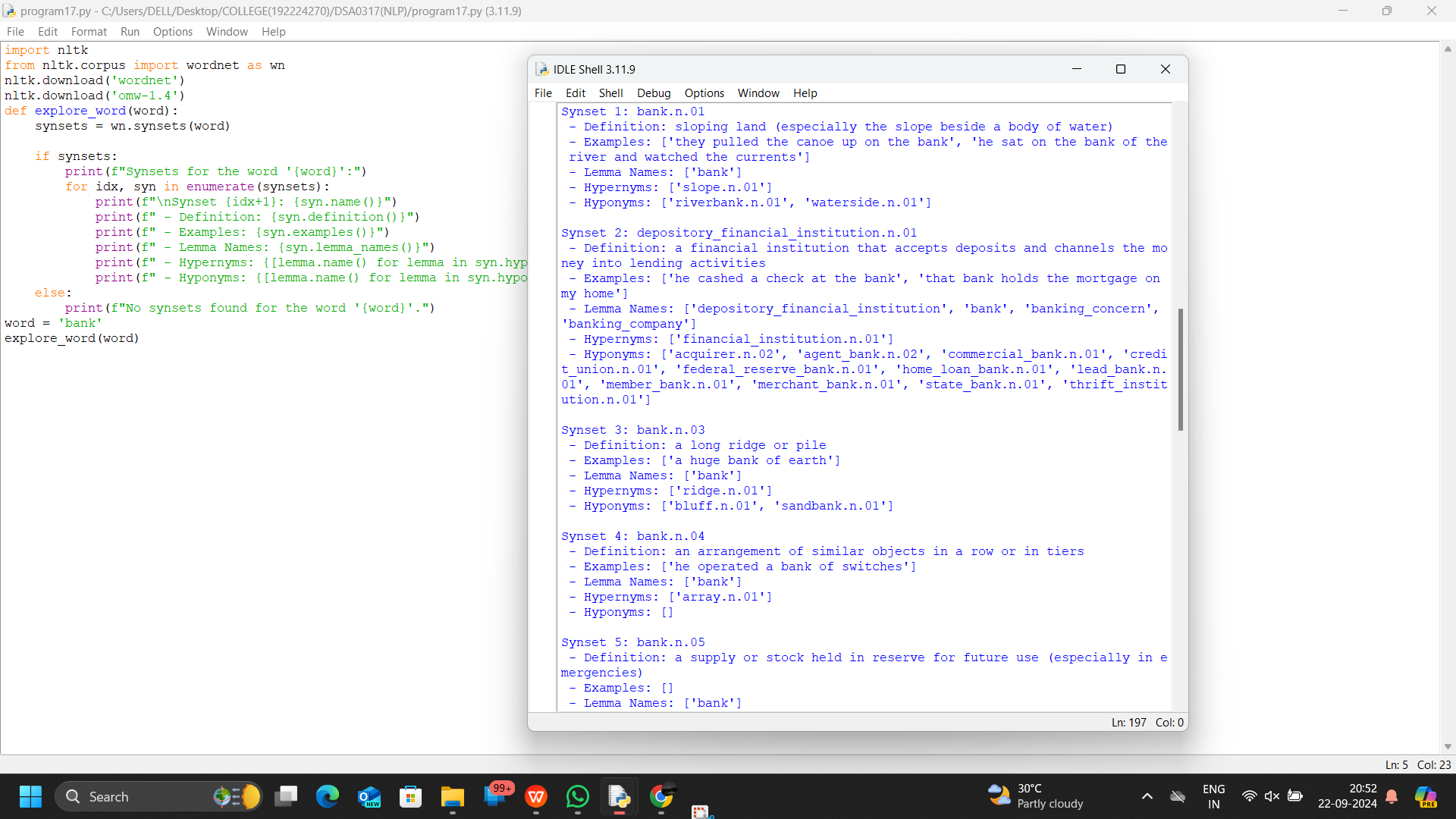
1. Implement probabilistic context-free grammar parsing for a sentence using python.



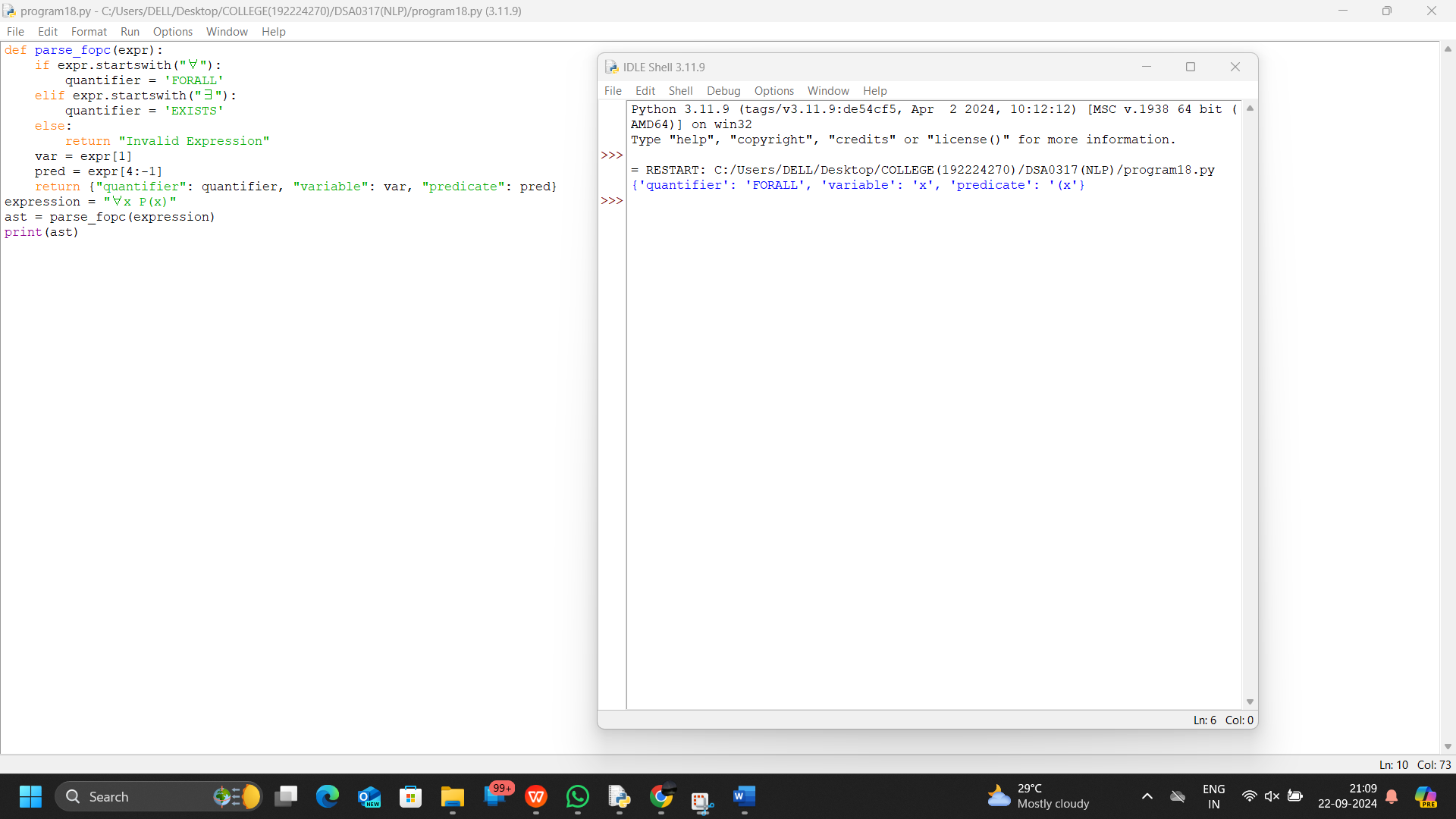
1. Implement a Python program using the SpaCy library to perform Named Entity Recognition (NER) on a given text.



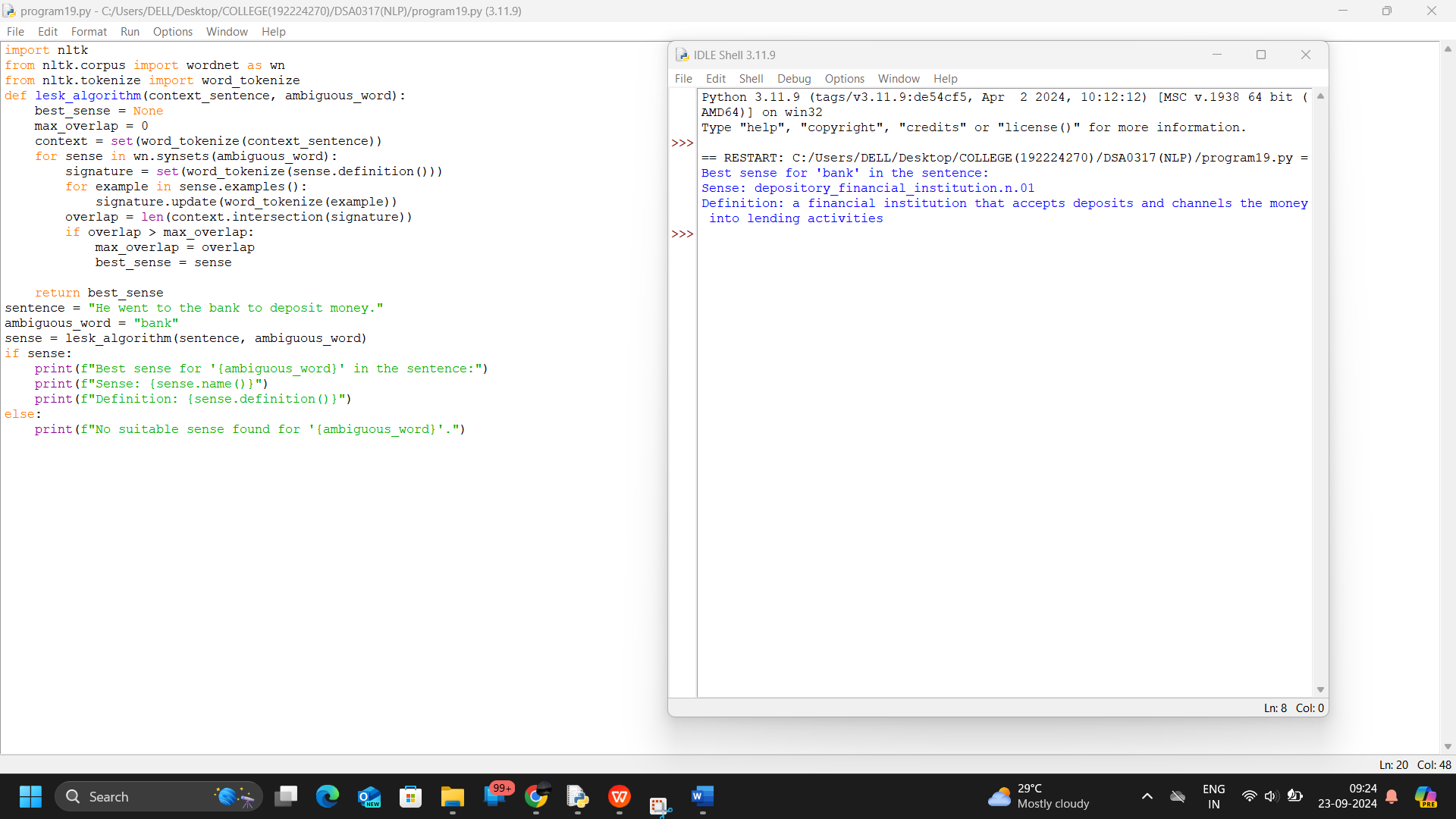
1. Write program demonstrates how to access WordNet, a lexical database, to retrieve synsets and explore word meanings in python.



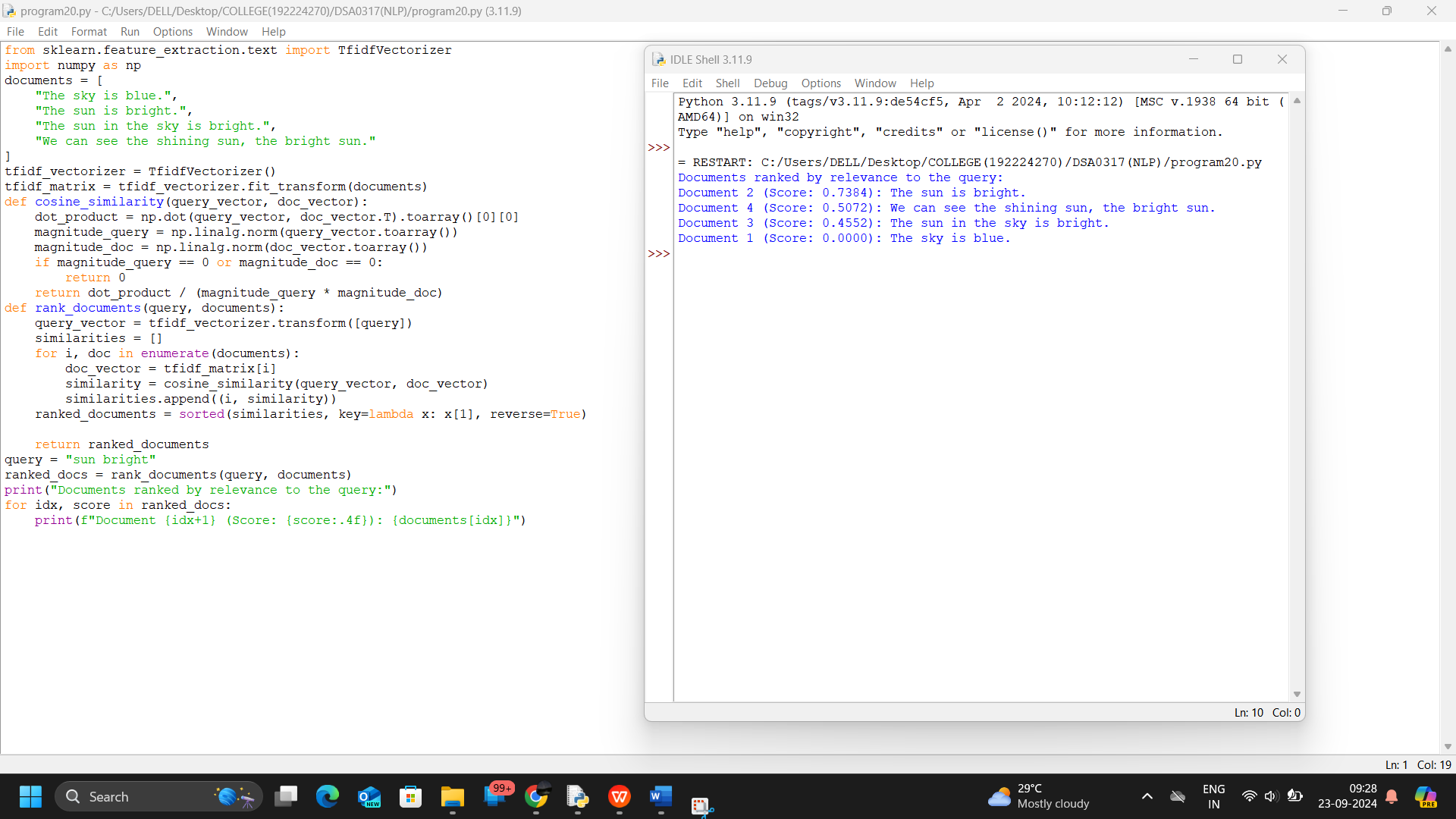
1. Implement a simple FOPC parser for basic logical expressions using python program.



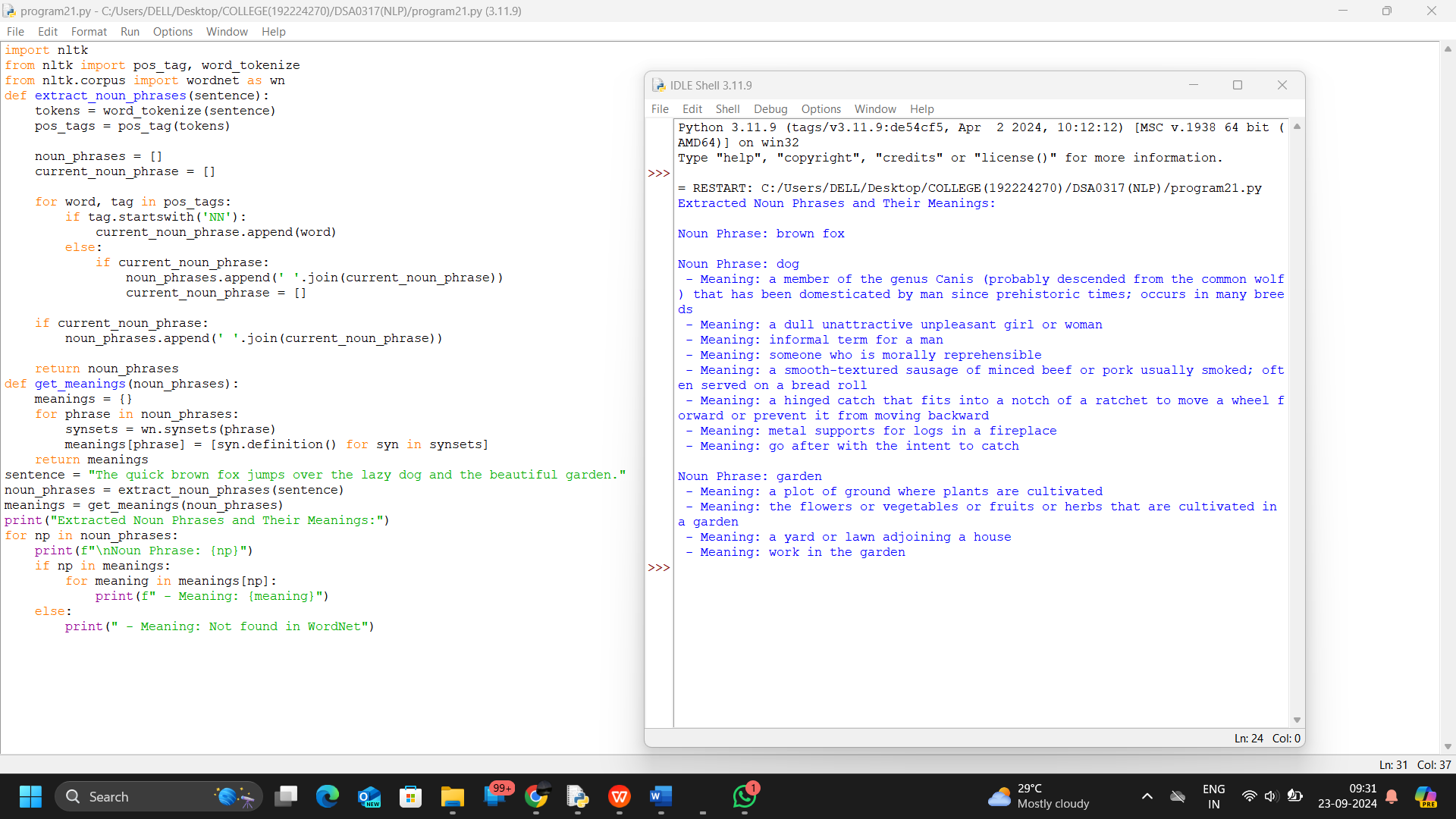
1. Create a program for word sense disambiguation using the Lesk algorithm using python.



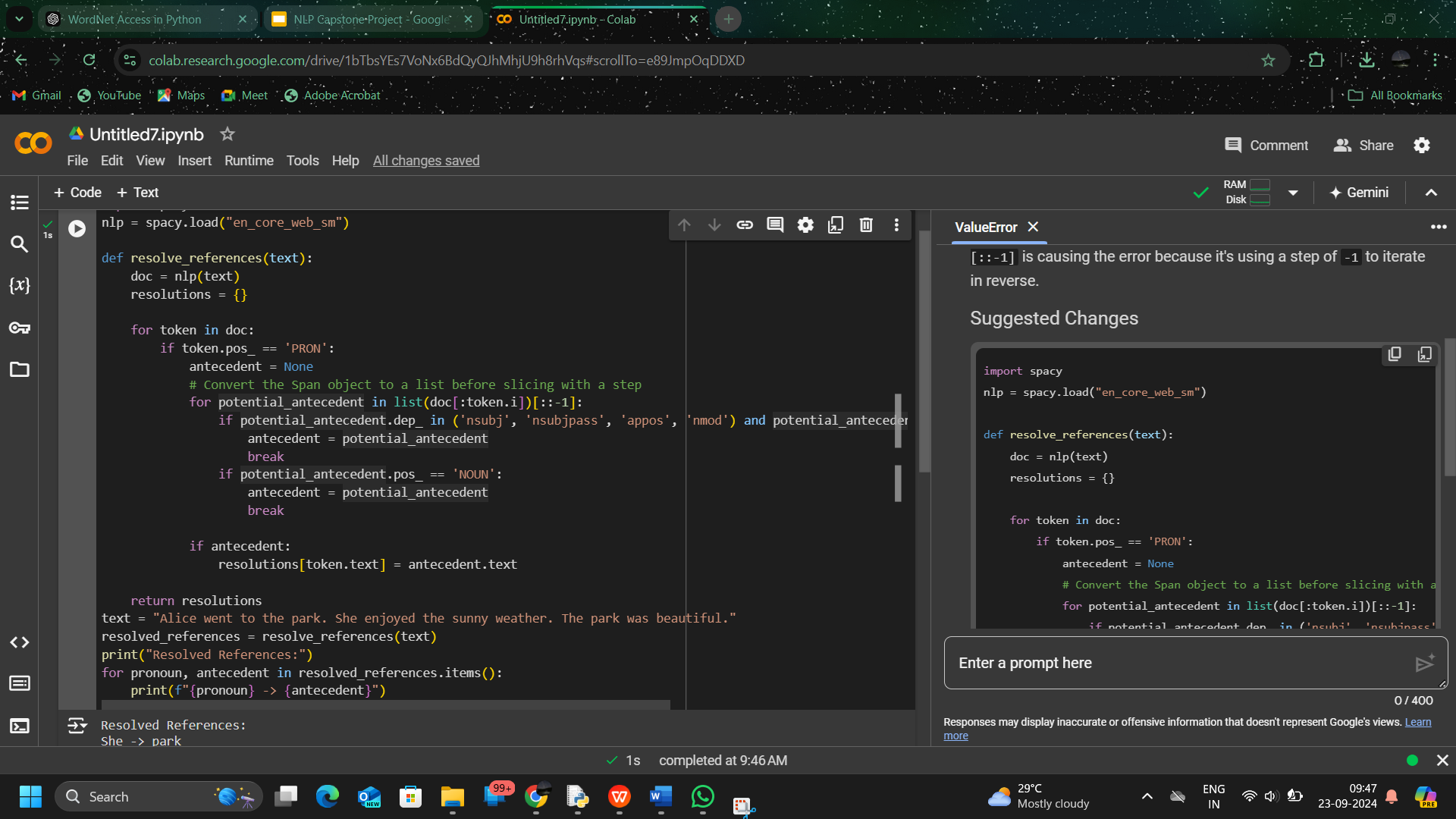
1. Implement a basic information retrieval system using TF-IDF (Term Frequency-Inverse Document Frequency) for document ranking using python.



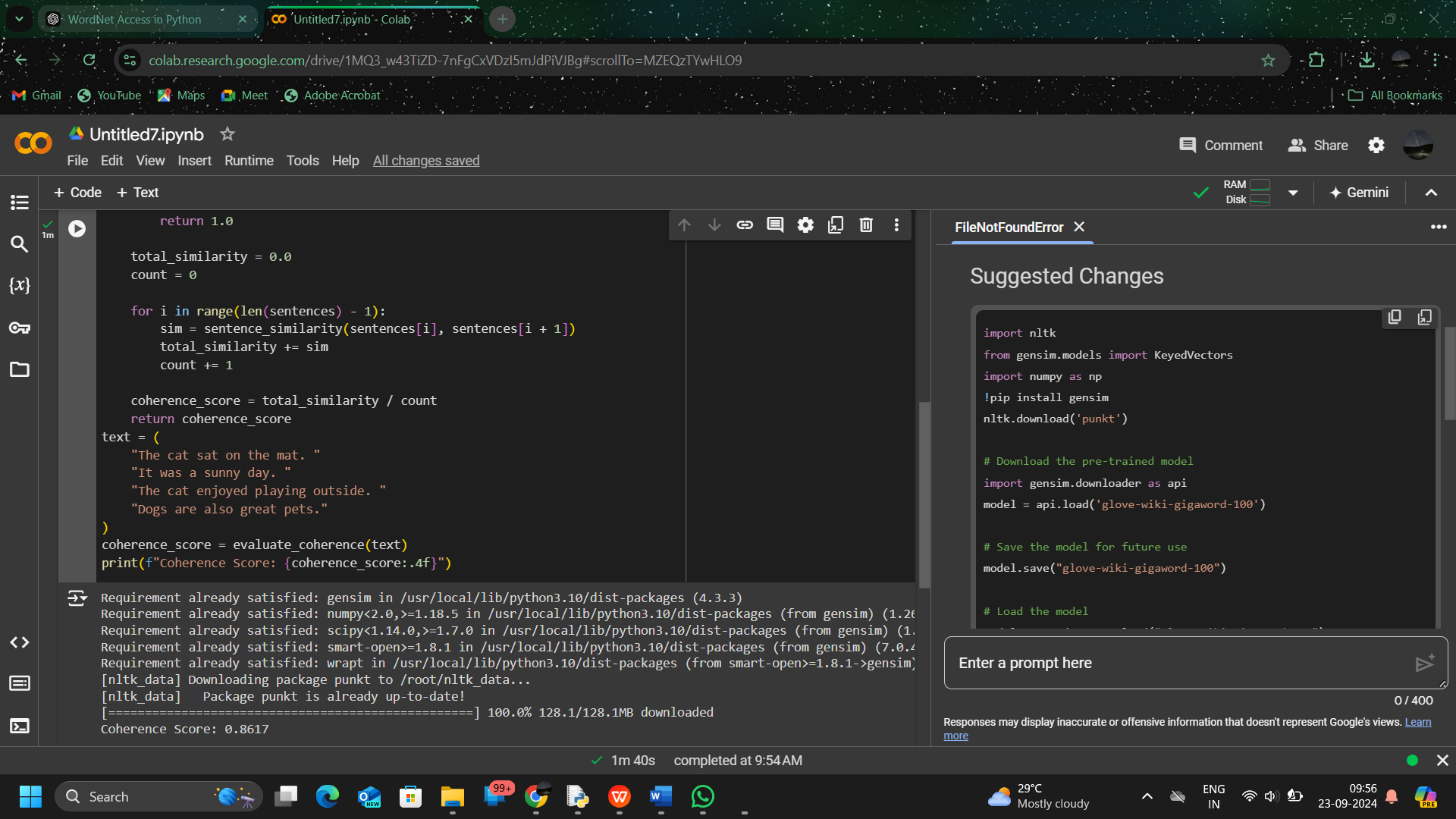
1. Create a python program that performs syntax-driven semantic analysis by extracting noun phrases and their meanings from a sentence.



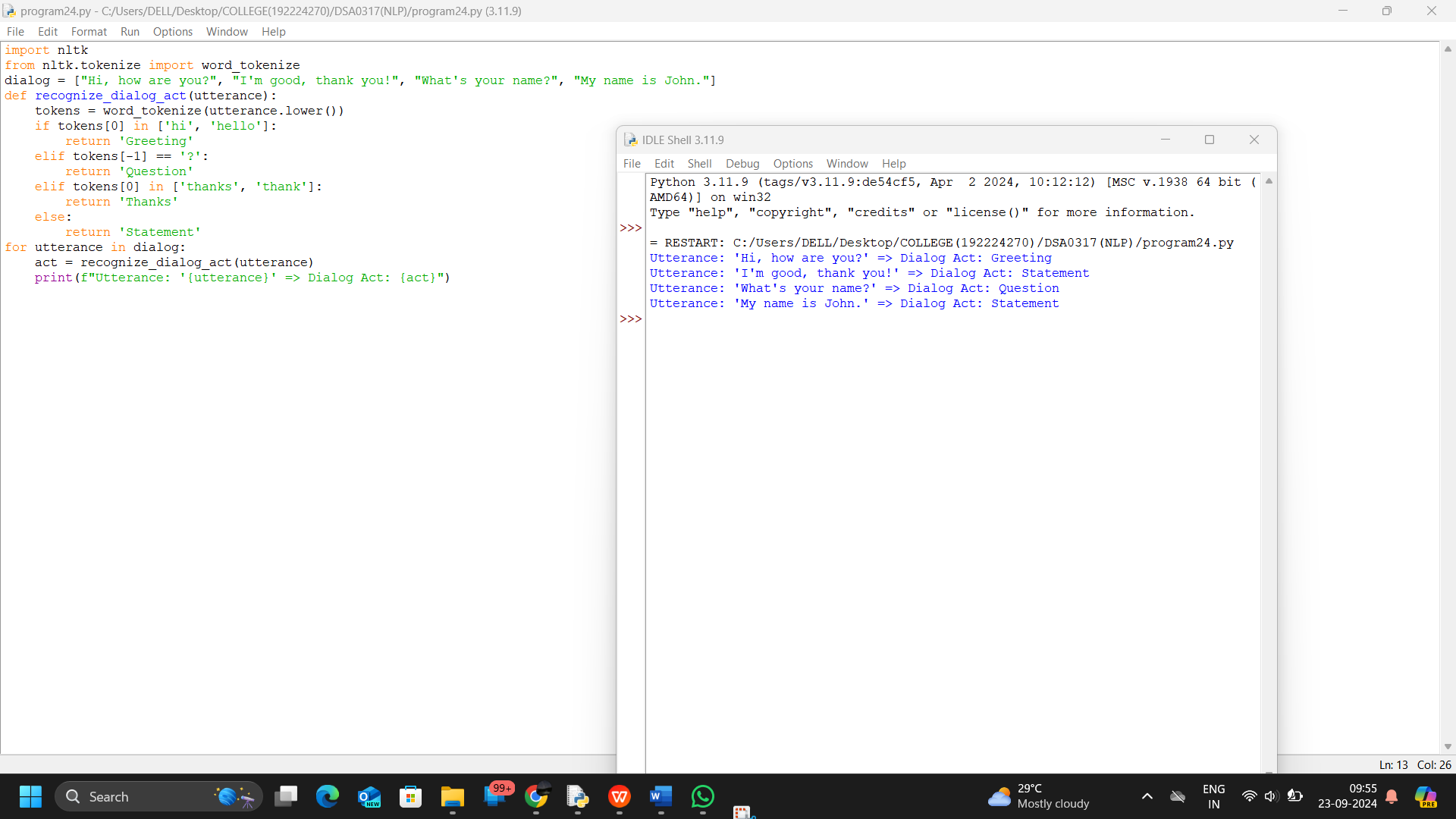
1. Create a python program that performs reference resolution within a text.



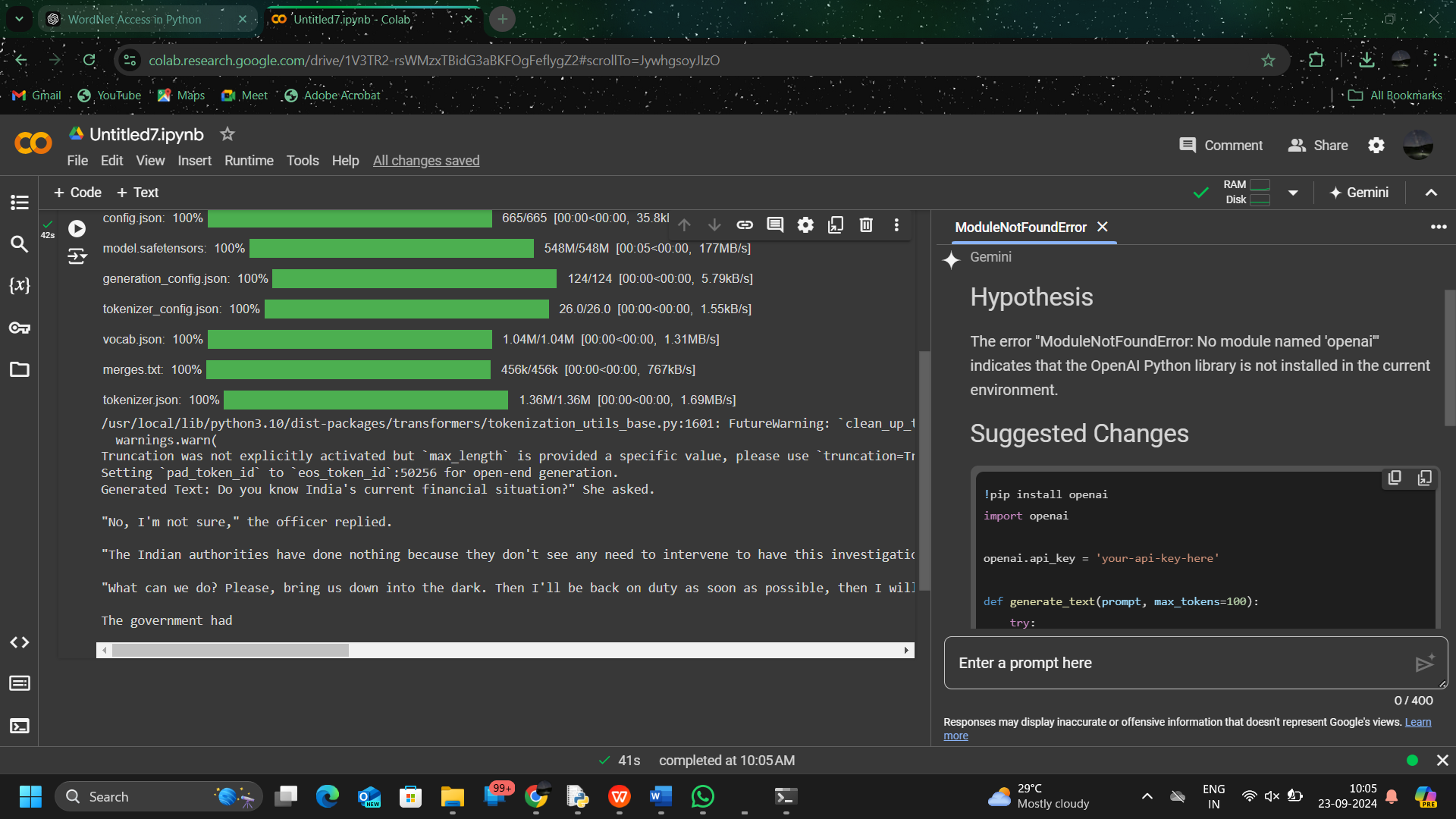
1. Develop a python program that evaluates the coherence of a given text.



1. Create a python program that recognizes dialog acts in a given dialog or conversation.



1. Utilize the GPT-3 model to generate text based on a given prompt. Make sure to install the OpenAI GPT-3 library in python implementation.



1. Implement a machine translation program using the Hugging Face Transformers library, translate English text to French using python.

