.Write a python program for chunking using nltk.

INTRODUCTION ::

Chunking is a process of extracting phrases from unstructured text, which means analyzing a sentence to identify the constituents(Noun Groups, Verbs, verb groups, etc.) However, it does not specify their internal structure, nor their role in the main sentence.

It works on top of POS tagging. It uses POS-tags as input and provides chunks as output.

sentence → clauses → phrases → words

Group of words make up phrases and there are five major categories.

- Noun Phrase (NP)
- Verb phrase (VP)
- Adjective phrase (ADJP)

```
#Chunking

import nltk

from nltk.tokenize import word_tokenize

from nltk import pos_tag, RegexpParser




nltk.download('punkt')

nltk.download('averaged_perceptron_tagger')


def chunk_sentence(sentence):

    words = word_tokenize(sentence)

    tagged_words = pos_tag(words)
```

```
    # Define grammar for chunking
    grammar = r"""
    NP: {<DT|JJ|NN.*>+}         # Chunk sequences of DT, JJ, NN
    PP: {<IN><NP>}              # Chunk prepositions followed by NP
    VP: {<VB.*><NP|PP|CLAUSE>+$}  # Chunk verbs and their arguments
    CLAUSE: {<NP><VP>}          # Chunk NP, VP pairs
    """
    parser = RegexpParser(grammar)
    chunked_sentence = parser.parse(tagged_words)

    return chunked_sentence

sentence = "The quick brown fox jumps over the lazy dog"
chunked_sentence = chunk_sentence(sentence)
print(chunked_sentence)
```