CODE 1

```
import nltk
nltk.download('punkt_tab')  # Cần cho sent_tokenize và word_tokenize
nltk.download('wordnet')  # Cần cho wordnet


from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import wordnet

if __name__ == '__main__':
    my_text = "Hello there. Welcome to new semester. How are you? Today is a good

    print(sent_tokenize(my_text))
    print("-" * 20)

    print(word_tokenize(my_text))
    print("-" * 20)

    print(list(my_text))
    print("-" * 20)

    syn = wordnet.synsets("NLP")
    if syn:
        print(syn[0].definition())
    else:
        print("No definition found for NLP.")
```

CODE 2

```
import nltk
sample_text = "We will discuss briefly about the basic syntax, structure and desi

print("Total characters: " + str(len(sample_text)))

# import sentence tokenization
default_st = nltk.sent_tokenize

# apply on the sample_text
sample_sentences = default_st(text=sample_text)

print("Total sentences in sample_text: " + str(len(sample_sentences)))
for i in range(len(sample_sentences)):
    print(" sentence " + str(i + 1) + ": " + sample_sentences[i])


import nltk
from nltk.corpus import gutenberg

# Download the 'gutenberg' corpus
nltk.download('gutenberg')

alice = gutenberg.raw(fileids="carroll-alice.txt")
print ("Total characters: " + str(len(alice)))
print("First 100 characters in the corpus\n")
print(alice[0:100])
# import sentence tokenization
default_st = nltk.sent_tokenize
# apply on the sample_text
alice_sentences = default_st(text=alice)

print("Total sentences in alice: " + str(len(alice_sentences)))
print("First 2 sentences in alice: ")
print("sentence 1: " + alice_sentences[0])
print("sentence 2: " + alice_sentences[1])
```

CODE 3

```
import nltk
sample_text = "the brown fox wasn't that quick and he couldn't win the race."

default_wt = nltk.word_tokenize
words = default_wt(text=sample_text)

print(len(words))
print(words)
print("-" * 20)

whitespace_wt = nltk.WhitespaceTokenizer()
words = whitespace_wt.tokenize(sample_text)

print(len(words))
print(words)
```

CODE 4

CODE 4.1

```
import nltk
sample_text = "The brown fox wasn't that quick and he couldn't win the race. Hey

default_wt = nltk.word_tokenize
words = default_wt(text=sample_text)
print(words)
print("-" * 20)

def remove_characters_after_tokenization(tokens):
    import re
    import string
    pattern = re.compile("[{}]".format(re.escape(string.punctuation)))
    filtered_tokens = list(filter(None, [pattern.sub("",token) for token in token
    return filtered_tokens

filter_list_1 = remove_characters_after_tokenization(words)
print(filter_list_1)
```

CODE 4.2

```
sample_text = "The brown FOX wasn't that quick and he couldn't win the RACE."
# lowercase
print(sample_text.lower())

#uppercase
print(sample_text.upper())
```

CODE 4.3

```
import nltk
nltk.download('stopwords')
english_stopwords = nltk.corpus.stopwords.words("english")
print(len(english_stopwords))
print(english_stopwords)
print("-"*20)

def remove_stopwords(tokens, language="english"):
    stopword_list = nltk.corpus.stopwords.words(language)
    filter_tokens = [token for token in tokens if token not in stopword_list]
    return filter_tokens

print(filter_list_1)
filter_list_2 = remove_stopwords(filter_list_1)
print(filter_list_2)
```

CODE 4.4

```
def remove_repeated_characters(tokens):
    from nltk.corpus import wordnet
    import re
    repeat_pattern = re.compile(r'(\w*)(\w)\2(\w*)')
    match_substitution = r'\1\2\3'
    def replace(old_word):
        if wordnet.synsets(old_word):
            return old_word
        new_word = repeat_pattern.sub(match_substitution, old_word)
        return replace(new_word) if new_word != old_word else new_word

    correct_tokens = [replace(word) for word in tokens]
    return correct_tokens

import nltk
sample_text = "My schooool is reallyyyyy ammaaazinggg"

default_wt = nltk.word_tokenize
tokens = default_wt(text=sample_text)
print(tokens)

sample_tokens = remove_repeated_characters(tokens)
print(sample_tokens)
```

CODE 5

```
from nltk.stem import PorterStemmer
ps = PorterStemmer()

from nltk.stem import PorterStemmer, WordNetLemmatizer

nltk.download('wordnet')
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()

words = ["running", "flies", "easily", "studies"]

print("Stemming:")
for word in words:
    print(word, "->", stemmer.stem(word))

print("\nLemmatization:")
for word in words:
    print(word, "->", lemmatizer.lemmatize(word))
```

CODE 6

```
nltk.download('averaged_perceptron_tagger')
sentence_1 = "NLP is an interesting field of study."
sentence_2 = 'The brown fox is quick and he is jumping over the lazy dog'

words_1 = word_tokenize(sentence_1)
words_2 = word_tokenize(sentence_2)

pos_tags_1 = nltk.pos_tag(words_1)
pos_tags_2 = nltk.pos_tag(words_2)

print("POS tagging:", pos_tags_1)
print("-"*20)
print("POS tagging:", pos_tags_2)
```