QUESTION 1 : COMPARISON OF SPACY AND NLTK AS NATURAL LANGUAGE PROCESSING LIBRARIES

1. **PROCESSING TIMES** :
NLTK processes and manipulates strings since it takes in strings as input and returns strings (in some form) as output to perform nlp tasks. It has specific methods for each task, for eg. nltk.pos_tag for 'Part of Speech' tagging and nltk.word_tokenize for 'Tokenization'. Thus, when I run each method on the text corpus, NLTK 'appears' to take comparatively less time to process individual methods.

However, Spacy follows an object oriented approach by parsing the text to return a 'document' object which has attributes and methods for various NLP tasks such as document.lemma_ for lemmatization, document.tag_ etc. It only takes relatively more time than NLTK initially in processing the text and storing it as an object. But any later tasks such as tokenization, lemmatization and part of speech tagging are much faster when compared to NLTK.
A summary of the processing times :

	Tokenization	Stemming/ Lemmatization	Part of Speech Tagging	Remove Punctuation	Remove Stop Words
NLTK	0:00:14.828852	0:00:22.554763	0:01:07.512428	0:00:00.630917	0:05:11.533861
SpaCy	0:02:25.799867				

I defined functions that process text by first removing punctuation and stop words and then perform tokenization, lemmatization and part of speech tagging.
The total time taken by the function for NLTK was 5 minutes 56 seconds and that for SpaCy was 2 minutes 25 seconds

2. **EASE OF PARALLELISATION** : It is easier to parallely process tasks in NLTK library and it results in a considerable reduction in processing time. This can be done using the standard 'multiprocessing' library. However, the multiprocessing library does not work for SpaCy due to problems with pickling tokens. SpaCy does provide a way for efficient processing - grouping documents for processing into chunks manually, so that instead of only passing one document to nlp each time, a sequence of documents can be streamed through spaCy's **nlp.pipe()**

3. **PERFORMANCE** : When tokenizing, NLTK splits the text on any non alphabetic character whereas SpaCy splits in a way that is more intuitive (For eg. splits **"wasn't" i**nto '**was**' and '**n't**' but keeps **U.K.** intact. It also keeps emails intact)

4. **USE CASES** : NLTK is great for learning and research purposes since it provides options to choose between algorithms to implement for a given task (for eg. it has 9 different stemming libraries). However SpaCy is useful for developers since it implements the best and most efficient algorithm for a given task. NLTK also has functionality for multiple languages whereas Spacy is limited to only English.