a. Sentence tokenization, word tokenization, Part of speech Tagging and chunking of user defined text.

Code:

```
import nltk
from nltk import tokenize
nltk.download('punkt_tab')
from nltk import tag
from nltk import chunk
nltk.download('averaged_perceptron_tagger_eng')
nltk.download('maxent_ne_tagger_tab')
nltk.download('words')

para = "today you will learn NLTK"
sents = tokenize.sent_tokenize(para)
print("\nSentence Tokenization\n======\n", sents)

# word tokenization
print("\nword tokenization\n======\n")
for index in range(len(sents)):
  words = tokenize.word_tokenize(sents[index])
  print(words)

#POS tagging
tagged_words = []
for index in range(len(sents)):
  tagged_words.append(tag.pos_tag(words))

print("\nPOS tagging\n======\n", tagged_words)

#chunking
```

```
tree = []
for index in range(len(sents)):
  tree.append(chunk.ne_chunk(tagged_words[index]))

print("\nChunking\n======\n", tagged_words)
print(tree)
```

Output:

```
= RESTART: D:\MSc.IT\Sem_4\NLP\NLP_Practical\NLP_Prac_Codes\nlp_code_p8_a.py
[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\acer\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     C:\Users\acer\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
[nltk_data] Error loading maxent_ne_tagger_tab: Package
[nltk_data]     'maxent_ne_tagger_tab' not found in index
[nltk_data] Downloading package words to
[nltk_data]     C:\Users\acer\AppData\Roaming\nltk_data...
[nltk_data]   Package words is already up-to-date!
```

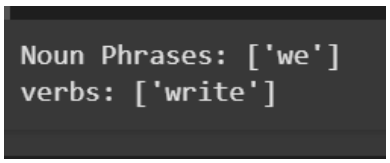
```
Sentence Tokenization
======
 ['today you will learn NLTK']

word tokenization
======

['today', 'you', 'will', 'learn', 'NLTK']

POS tagging
======
 [[('today', 'NN'), ('you', 'PRP'), ('will', 'MD'), ('learn', 'VB'), ('NLTK', 'N
NP')]]

Chunking
======
 [[('today', 'NN'), ('you', 'PRP'), ('will', 'MD'), ('learn', 'VB'), ('NLTK', 'N
NP')]]
[Tree('S', [('today', 'NN'), ('you', 'PRP'), ('will', 'MD'), ('learn', 'VB'), Tr
ee('ORGANIZATION', [('NLTK', 'NNP')])])]
```

b. Named Entity Recognition of user defined text .

Code:

```
import spacy
nlp = spacy.load("en_core_web_sm")
text = "we are writing practical"
doc = nlp(text)
print("Noun Phrases:", [chunk.text for chunk in doc.noun_chunks])
print("verbs:", [token.lemma_ for token in doc if token.pos_ == "VERB"])
```

Output:



c. Named Entity Recognition with diagram using NLTK corpus - treebank

Code:

```
import nltk
nltk.download('treebank')
from nltk.corpus import treebank_chunk
treebank_chunk.tagged_sents() [0]
treebank_chunk.chunked_sents() [0]
#for Google colab
#treebank_chunk.chunked_sents() [0].pretty_print()
#for Idle
treebank_chunk.chunked_sents() [0].draw()
```

Output: