Name: Vinayak V Thayil

Roll No: AM.EN.U4CSE21161

```python
import nltk
from nltk.corpus import gutenberg, brown
from nltk.corpus import wordnet as wn


nltk.download('gutenberg')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('brown')
```

```
[nltk_data] Downloading package gutenberg to /root/nltk_data...
[nltk_data]   Unzipping corpora/gutenberg.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package brown to /root/nltk_data...
[nltk_data]   Unzipping corpora/brown.zip.
True
```

```python
# 1. Gutenberg corpus

print(gutenberg.fileids()) # a

emma_text = gutenberg.raw('austen-emma.txt')
print("Length of emma text:", len(emma_text)) # b

emma_words = nltk.word_tokenize(emma_text)
emma_words.sort()
print("Sorted words in emma text:")
print(emma_words[:10]) # c
print("Length of word list:", len(emma_words)) #d

emma_text = nltk.Text(emma_words)
text_concordance = nltk.ConcordanceIndex(emma_text)
print(text_concordance.print_concordance("affection")) # e

emma_synonyms = set(wn.synsets("love"))
print(emma_synonyms) # f

print(gutenberg.fileids()) # g
```

```
['austen-emma.txt', 'austen-persuasion.txt', 'austen-sense.txt', 'bible-kjv.txt', 'blake-poems.txt', 'bryant-stories.txt', 'burgess-bust
Length of emma text: 887071
Sorted words in emma text:
['!', '!', '!', '!', '!', '!', '!', '!', '!', '!']
Length of word list: 191785
Displaying 25 of 49 matches:
ly affectedly affectedly affecting affection affection affection affection affe
dly affectedly affecting affection affection affection affection affection affe
edly affecting affection affection affection affection affection affection affe
ting affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
tion affection affection affection affection affection affection affection affe
None
{Synset('love.n.04'), Synset('love.n.01'), Synset('love.v.01'), Synset('sexual_love.n.02'), Synset('love.v.02'), Synset('sleep_together.
['austen-emma.txt', 'austen-persuasion.txt', 'austen-sense.txt', 'bible-kjv.txt', 'blake-poems.txt', 'bryant-stories.txt', 'burgess-bust
```

```python
# 2. Brown Corpus

print(brown.categories()) # a

modal_verbs = ['can','may','could','might','must','will']
cfd = nltk.ConditionalFreqDist(
    (genre, word)
    for genre in brown.categories()
    for word in brown.words(categories=genre)
    if word in modal_verbs)

print(cfd.tabulate()) #b
```

```
    ['adventure', 'belles_lettres', 'editorial', 'fiction', 'government', 'hobbies', 'humor', 'learned', 'lore', 'mystery', 'news', 'religic
                    can could  may might  must  will
       adventure     46   151    5    58    27    50
   belles_lettres   246   213  207   113   170   236
        editorial   121    56   74    39    53   233
          fiction    37   166    8    44    55    52
       government   117    38  153    13   102   244
          hobbies   268    58  131    22    83   264
            humor    16    30    8     8     9    13
          learned   365   159  324   128   202   340
             lore   170   141  165    49    96   175
          mystery    42   141   13    57    30    20
             news    93    86   66    38    50   389
         religion    82    59   78    12    54    71
          reviews    45    40   45    26    19    58
          romance    74   193   11    51    45    43
   science_fiction    16    49    4    12     8    16
    None
```

```python
# 3. WordNet

motorcar_synsets = wn.synsets("motorcar")
print(motorcar_synsets) # a

for synset in motorcar_synsets:
    print(synset.name(), ":", synset.definition()) # b

car_synsets = wn.synsets("car")
print("Different meanings of 'car':")
for synset in car_synsets:
    print("-", synset.name(), ":", synset.definition()) # c

motorcar_synset = wn.synset('car.n.01')
print("Hypernyms:", motorcar_synset.hypernyms()) # d
print("Meronyms:", motorcar_synset.part_meronyms()) # d
```

```
    [Synset('car.n.01')]
    car.n.01 : a motor vehicle with four wheels; usually propelled by an internal combustion engine
    Different meanings of 'car':
    - car.n.01 : a motor vehicle with four wheels; usually propelled by an internal combustion engine
    - car.n.02 : a wheeled vehicle adapted to the rails of railroad
    - car.n.03 : the compartment that is suspended from an airship and that carries personnel and the cargo and the power plant
    - car.n.04 : where passengers ride up and down
    - cable_car.n.01 : a conveyance for passengers or freight on a cable railway
    Hypernyms: [Synset('motor_vehicle.n.01')]
    Meronyms: [Synset('accelerator.n.01'), Synset('air_bag.n.01'), Synset('auto_accessory.n.01'), Synset('automobile_engine.n.01'), Synset('
```