NAME : PAREENITA A.SHIRSATH B.E.A.I.&.D.S. ROLL.NO : 57

PRN : 221101062

NLP EXPERIMENT NO : 04

```
import nltk
import spacy
from nltk.tokenize import word_tokenize
from nltk.corpus import wordnet as wn
```

```
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
True
```

```
nlp = spacy.load("en_core_web_sm")
```

+ Code + Text

```
def analyze_word_nltk(word):
    """Perform basic morphological analysis using NLTK."""
    synsets = wn.synsets(word)
    analysis = []
    for synset in synsets:
        analysis.append({
            'synset': synset.name(),
            'definition': synset.definition(),
            'examples': synset.examples()
        })
    return analysis

def generate_words(root):
    """Generate new words by adding common suffixes."""
    suffixes = ['able', 'er', 'ing', 'ed', 'ly', 's', 'es']
    return [root + suffix for suffix in suffixes]

def analyze_text_spacy(text):
    """Perform advanced morphological analysis using spaCy."""
    doc = nlp(text)
    analysis = []
    for token in doc:
        analysis.append({
            'word': token.text,
            'lemma': token.lemma_,
            'POS': token.pos_,
            'tag': token.tag_,
            'dep': token.dep_
        })
    return analysis


def main():
    text = "Indian Railways"

    words = word_tokenize(text)

    print("Morphological Analysis using NLTK:\n")
    for word in words:
        print(f"Analysis for '{word}':")
        analysis_nltk = analyze_word_nltk(word)
        for entry in analysis_nltk:
            print(f"  Synset: {entry['synset']}")
            print(f"  Definition: {entry['definition']}")
            print(f"  Examples: {entry['examples']}")
        print()

    print("Word Generation:\n")
    root_word = "play"
    new_words = generate_words(root_word)
    print(f"Generated words based on '{root_word}': {new_words}")
    print()

    print("Morphological Analysis using spaCy:\n")
    analysis_spacy = analyze_text_spacy(text)
    for entry in analysis_spacy:
        print(f"Word: {entry['word']}")
        print(f"  Lemma: {entry['lemma']}")
        print(f"  POS: {entry['POS']}")
        print(f"  Tag: {entry['tag']}")
        print(f"  Dep: {entry['dep']}")
        print()

if __name__ == "__main__":
    main()
```

```
Morphological Analysis using NLTK:

Analysis for 'Indian':
  Synset: indian.n.01
  Definition: a member of the race of people living in America when Europeans arrived
  Examples: []
  Synset: indian.n.02
  Definition: a native or inhabitant of India
  Examples: []
  Synset: amerind.n.01
  Definition: any of the languages spoken by Amerindians
  Examples: []
  Synset: indian.a.01
  Definition: of or relating to or characteristic of India or the East Indies or their peoples or languages or cultures
  Examples: ['the Indian subcontinent', 'Indian saris']
  Synset: indian.a.02
  Definition: of or pertaining to American Indians or their culture or languages
  Examples: ['Native American religions', 'Indian arrowheads']

Analysis for 'Railways':
  Synset: railway.n.01
  Definition: line that is the commercial organization responsible for operating a system of transportation for trains that pull pa
  Examples: []
  Synset: railroad_track.n.01
  Definition: a line of track providing a runway for wheels
  Examples: ['he walked along the railroad track']

Word Generation:

Generated words based on 'play': ['playable', 'player', 'playing', 'played', 'playly', 'plays', 'playes']

Morphological Analysis using spaCy:

Word: Indian
  Lemma: Indian
  POS: PROPN
  Tag: NNP
  Dep: compound

Word: Railways
  Lemma: Railways
  POS: PROPN
  Tag: NNPS
  Dep: ROOT
```