# Practical No:2

**Title:**

Find the synonym /antonym of a word using WordNet.

**Aim:**

To write a program that finds the synonyms and antonyms of a given word using the WordNet lexical database.

**Pre-requisites:**

1. Understanding of Natural Language Processing (NLP) concepts, particularly lexical semantics.
2. Familiarity with Python and libraries such as nltk (Natural Language Toolkit).
3. Knowledge of WordNet, a large lexical database of English words.
4. Basic understanding of how to use dictionaries and data structures to store word relationships.

---

**Theory:**

**WordNet** is a widely used lexical database in NLP that groups English words into sets of synonyms (called **synsets**) and provides short definitions, usage examples, and other linguistic relationships like antonyms (opposite meanings), hypernyms (broader categories), and hyponyms (subcategories).

**Synonyms:**

- Synonyms are words that have the same or nearly the same meaning as another word. Example: *happy* and *joyful* are synonyms.

**Antonyms:**

- Antonyms are words that have the opposite meaning to another word. Example: *happy* and *sad* are antonyms.

WordNet allows us to query these relationships programmatically. By using the `nltk` library in Python, we can access WordNet to find the synonyms and antonyms of words.

**Steps for Finding Synonyms and Antonyms:**

1. **Input Word:**
    - A word is provided as input.
2. **Using WordNet via nltk:**
    - Use WordNet's synsets to retrieve synonyms.
    - For each synonym, check if an antonym exists and retrieve it if available.
3. **Finding Synonyms:**
    - For the input word, find all synsets using `wordnet.synsets()`.
    - For each synset, retrieve its lemmas (words that are synonymous) using `synset.lemmas()`.
4. **Finding Antonyms:**
    - For each lemma, check if it contains an antonym by using `lemma.antonyms()`.
    - If antonyms are found, store them.
5. **Output:**
    - Display a list of synonyms and antonyms of the input word.

## Conclusion:

Using WordNet through the `nltk` library, we can efficiently find synonyms and antonyms for any given word. Synonym and antonym

extraction is crucial in various NLP tasks, such as paraphrasing, sentiment analysis, and text generation. By leveraging WordNet's semantic relationships, we can better understand the meaning and context of words, making it an essential tool for many language processing applications.