**Explanation of the Solution Approach**

1. **Data Extraction (extract\_article function):**

○ I utilized requests and BeautifulSoup libraries to fetch and parse HTML content from URLs.

○ Extracted the article title from <h1> tag and article text from <article> or <p> tags.

2. **Text Analysis (analyze\_text function):**

○ Tokenized the text into words and sentences using nltk tokenizer.

○ Calculated various metrics such as word count, sentence count, average sentence length, percentage of complex words, Fog index, average number of words per sentence, etc.

○ Used nltk.sentiment.vader for sentiment analysis to compute positive, negative, polarity, and subjectivity scores.

○ Identifies personal pronouns using regular expressions.

3. **Output Generation:**

○ Iterates through each article extracted and analyzed.

○ Constructed a DataFrame (df\_results) containing all the computed metrics along with URL\_ID and Title.

**How to Run the .py File to Generate Output**

To run the Python script (main.py for example) and generate the required output:

1. **Ensure Dependencies:**

**2. Prepare Input**

**3. Execute the Script**

**Dependencies Required**

Ensure the following Python libraries are installed:

● pandas: For data manipulation and DataFrame operations.

● requests: For making HTTP requests to fetch web content.

● beautifulsoup4: For parsing HTML content.

● openpyxl: For reading and writing Excel files.

● nltk: For natural language processing tasks such as tokenization and sentiment analysis.