UKA TARSADIA UNIVERSITY

BABU MADHAV INSTITUTE OF INFORMATION TECHNOLOGY

M.SC.(IT)

NATURAL LANGUAGE PROCESSING

Assignment No: 1

Student Enrolment Number : 202206100110061

Student Name : Divya bhaveshbhai ghori

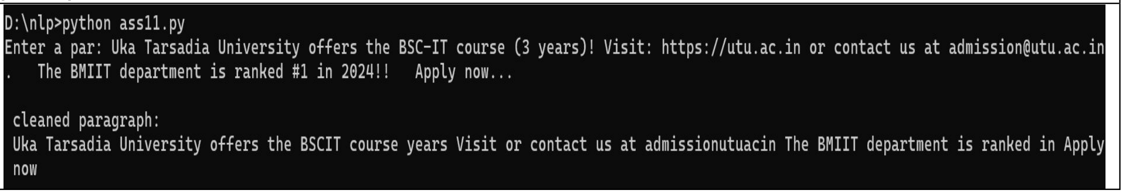
Question No: 1

Basic Text Cleaning
Objective: Clean a paragraph input using Python.
• Take user input of a paragraph.
• Convert to lowercase.
• Remove punctuation, digits, special characters, URLs, emails, and extra spaces.
• Display cleaned text.

Code:

```
import re
import string

par = input("Enter a par: ")

par = par.translate(str.maketrans('', '', string.punctuation))

par = re.sub(r'\d+', '', par)

par = re.sub(r'[^a-zA-Z0-9\s]', '', par)

par = re.sub(r'http\S+|www\.\S+', '', par)

par = re.sub(r'\S+@\S+', '', par)

par = re.sub(r'\s+', ' ', par).strip()

print(par)
```

Output:



```
D:\nlp>python ass11.py
Enter a par: Uka Tarsadia University offers the BSC-IT course (3 years)! Visit: https://utu.ac.in or contact us at admission@utu.ac.in
.   The BMIIT department is ranked #1 in 2024!!   Apply now...

 cleaned paragraph:
 Uka Tarsadia University offers the BSCIT course years Visit or contact us at admissionutuacin The BMIIT department is ranked in Apply
 now
```

Question No: 2

Tokenization (Sentence & Word)

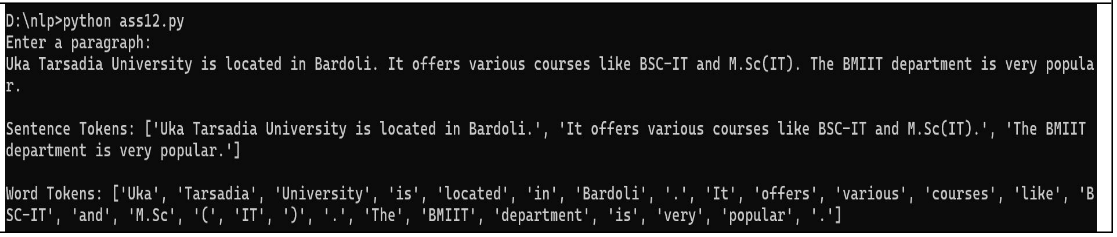
Objective: Apply sentence and word tokenization using NLTK.

- Input a multi-sentence paragraph.
- Tokenize into sentences.
- Tokenize each sentence into words.
- Display both sentence and word tokens clearly.

Code:

```
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize

paragraph = input("Enter a paragraph:\n")

sentences = sent_tokenize(paragraph)
print("\nSentence Tokens:", sentences)

words = word_tokenize(paragraph)

print("\nWord Tokens:" , words)
```

Output:



```
D:\nlp>python ass12.py
Enter a paragraph:
Uka Tarsadia University is located in Bardoli. It offers various courses like BSC-IT and M.Sc(IT). The BMIIT department is very popula
r.

Sentence Tokens: ['Uka Tarsadia University is located in Bardoli.', 'It offers various courses like BSC-IT and M.Sc(IT).', 'The BMIIT
department is very popular.']

Word Tokens: ['Uka', 'Tarsadia', 'University', 'is', 'located', 'in', 'Bardoli', '.', 'It', 'offers', 'various', 'courses', 'like', 'B
SC-IT', 'and', 'M.Sc', '(', 'IT', ')', '.', 'The', 'BMIIT', 'department', 'is', 'very', 'popular', '.']
```

Question No: 3

Stop Word Removal

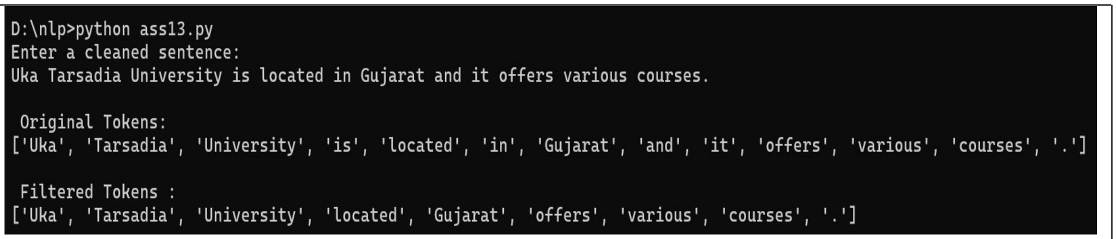
Objective: Filter meaningful words from a sentence.

- Input a cleaned sentence.
- Use NLTK to remove stop words.
- Print original and filtered tokens.

Code:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

sentence = input("Enter a cleaned sentence:\n")

wordtokens = word_tokenize(sentence)
stopwords = set(stopwords.words('english'))
filteredtokens = [word for word in wordtokens if word not in stopwords]

print("\n Original Tokens:")
print(wordtokens)

print("\n Filtered Tokens :")
print(filteredtokens)
```

Output:



```
D:\nlp>python ass13.py
Enter a cleaned sentence:
Uka Tarsadia University is located in Gujarat and it offers various courses.

 Original Tokens:
['Uka', 'Tarsadia', 'University', 'is', 'located', 'in', 'Gujarat', 'and', 'it', 'offers', 'various', 'courses', '.']

 Filtered Tokens :
['Uka', 'Tarsadia', 'University', 'located', 'Gujarat', 'offers', 'various', 'courses', '.']
```

Question No: 4

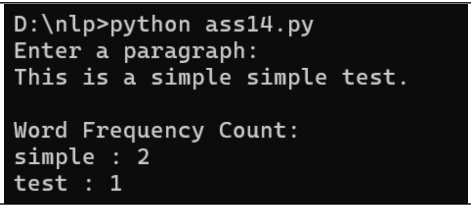
Word Frequency Count
Objective: Analyze word usage in text.

- Take input of a paragraph.
- Clean and tokenize the text.
- Remove stop words.
- Count and display frequency of each word using collections.Counter.

Code:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from collections import Counter
import string

paragraph = input("Enter a paragraph:\n")

tokens = word_tokenize(paragraph)
tokens = [word.lower() for word in tokens if word.isalnum()]

stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word not in stop_words]

word_freq = Counter(filtered_tokens)

print("\nWord Frequency Count:")

for word in word_freq:
    print(word, ":", word_freq[word])
```

Output:



```
D:\nlp>python ass14.py
Enter a paragraph:
This is a simple simple test.

Word Frequency Count:
simple : 2
test : 1
```

Question No: 5

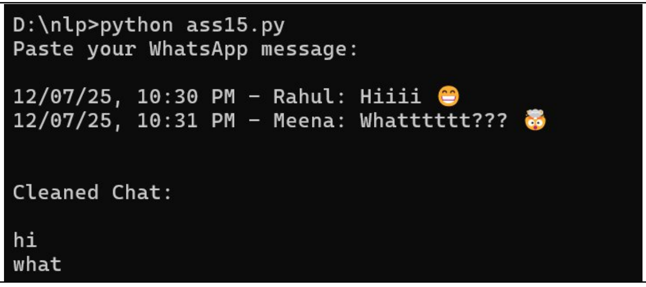
You are given a copied text from a WhatsApp chat. The text contains emojis, timestamps, sender names, repeated characters like “gooooood”, and random special characters.
Instructions:
Clean the chat to extract only useful message text.

Remove timestamps, sender names, emojis, repeated characters (e.g., reduce "sooooo" to "soo"), special symbols.
Display cleaned messages line by line.

Code:

```
import re

print("Paste your WhatsApp message:\n")
lines = []
while True:
    line = input()
    if line == "":
        break
    lines.append(line)

print("\nCleaned Chat:\n")
for line in lines:
    line = re.sub(r"\d+/\d+/\d+, \d+:\d+ [AP]M - .*?:", "", line)
    line = re.sub(r"[^\x00-\x7F]+", " ", line)
    line = re.sub(r"[^a-zA-Z\s]", " ", line)
    line = re.sub(r"\s+", " ", line)
    line = re.sub(r"(.)\1{1,}", r"\1", line)
    line = line.lower().strip()
    if line:
        print(line)
```

Output:



```
D:\nlp>python ass15.py
Paste your WhatsApp message:

12/07/25, 10:30 PM - Rahul: Hiiii 😁
12/07/25, 10:31 PM - Meena: Whatttttt??? 🤯


Cleaned Chat:

hi
what
```

Question No: 6

Find the Most Meaningful Word

- Ask user to input any paragraph or story (min 5 lines).
- Clean and preprocess the text.
- Remove stop words and punctuation.
- Count word frequencies.
- Identify and print the top 3 most frequent meaningful words.

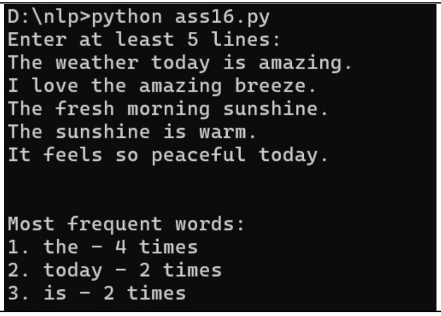
Input paragraph:
“The weather today is amazing. I love the amazing breeze and the fresh morning sunshine. The sunshine is warm and peaceful.”
Expected Output
Most frequent meaningful words:
1. sunshine - 2 times
2. amazing - 2 times
3. breeze - 1 time

Code:

```
from collections import Counter
```

```
import string

while True:
    print("Enter at least 5 lines:")

    text = ''
    count = 0

    while True:
        line = input()
        if line == '':
            break
        text += line + ' '
        count += 1

    if count < 5:
        print("\nYou entered less than 5 lines. Enter again.\n")
        continue
    else:
        break

text = text.lower()
for ch in string.punctuation:
    text = text.replace(ch, '')

words = text.split()
word_counts = Counter(words).most_common(3)

print("\nMost frequent words:")
for i, (word, freq) in enumerate(word_counts, start=1):
    print(f"{i}. {word} - {freq} time{'s' if freq > 1 else ''}")
```

Output:



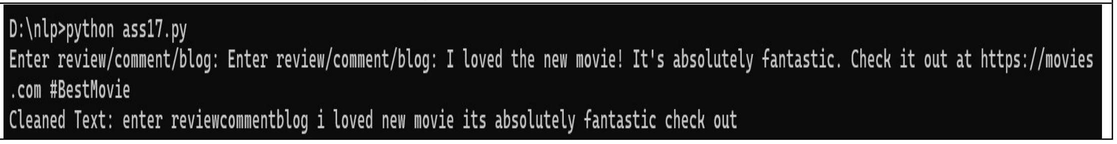
```
D:\nlp>python ass16.py
Enter at least 5 lines:
The weather today is amazing.
I love the amazing breeze.
The fresh morning sunshine.
The sunshine is warm.
It feels so peaceful today.


Most frequent words:
1. the - 4 times
2. today - 2 times
3. is - 2 times
```

Question No: 7	Create Your Own Text Cleaning Function Students must create a function clean_text(text) that takes raw text and: • Removes emails, URLs, hashtags, numbers, punctuation. • Converts text to lowercase. • Removes stop words. Then test it on any sample review/comment/blog they find.

Code:

```
import re
import string

stop_words = {'a', 'an', 'the', 'is', 'in', 'on', 'and', 'or', 'at', 'to', 'for',
              'with', 'of', 'this', 'that', 'it', 'as', 'was', 'but', 'by', 'are', 'be'}

def clean_text(text):
    text = text.lower()
    text = re.sub(r'\S+@\S+', '', text)
    text = re.sub(r'http\S+|www\.\S+', '', text)
    text = re.sub(r'#\w+', '', text)
    text = re.sub(r'\d+', '', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    words = text.split()
    filtered_words = [word for word in words if word not in stop_words]
    return ' '.join(filtered_words)

inp = input("Enter review/comment/blog: ")
print("Cleaned Text:", clean_text(inp))
```

Output:



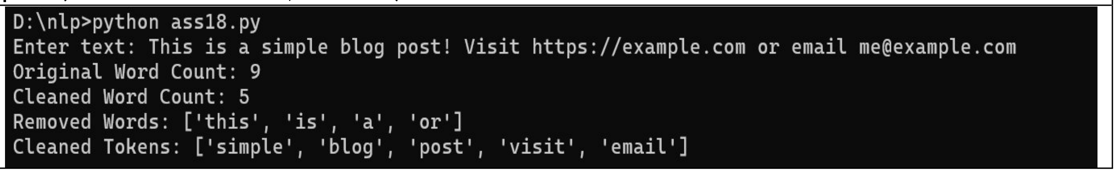
```
D:\nlp>python ass17.py
Enter review/comment/blog: Enter review/comment/blog: I loved the new movie! It's absolutely fantastic. Check it out at https://movies
.com #BestMovie
Cleaned Text: enter reviewcommentblog i loved new movie its absolutely fantastic check out
```

Question No: 8	Compare Text Before and After Cleaning • Input: A raw paragraph with noise (punctuation, stop words, symbols, etc.) • Process: Clean it using preprocessing steps. • Output: o Original word count o Cleaned word count o Removed words list o Final cleaned tokens

Code:

```
import re
import string

stop_words = {'a', 'an', 'the', 'is', 'in', 'on', 'and', 'or', 'at', 'to', 'for',
              'with', 'of', 'this', 'that', 'it', 'as', 'was', 'but', 'by', 'are', 'be'}

def clean(text):
    text = text.lower()
    text = re.sub(r'\S+@\S+', '', text)
    text = re.sub(r'http\S+|www\.\S+', '', text)
    text = re.sub(r'#\w+', '', text)
    text = re.sub(r'\d+', '', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    words = text.split()
    cleaned = [w for w in words if w not in stop_words]
    removed = [w for w in words if w in stop_words]
    return words, cleaned, removed

text = input("Enter text: ")
original, cleaned, removed = clean(text)

print("Original Word Count:", len(original))
print("Cleaned Word Count:", len(cleaned))
print("Removed Words:", removed)
print("Cleaned Tokens:", cleaned)
```

Output:



```
D:\nlp>python ass18.py
Enter text: This is a simple blog post! Visit https://example.com or email me@example.com
Original Word Count: 9
Cleaned Word Count: 5
Removed Words: ['this', 'is', 'a', 'or']
Cleaned Tokens: ['simple', 'blog', 'post', 'visit', 'email']
```