NAME – Ritik Gupta

REG. NO. – 18BCE0154

WEB MINING

ASSIGNMENT-1

Write a python program for following questions:

a) **Take dynamic input from user as a paragraph (give input >= 25 lines) and remove punctuations first (only). Then print resulting paragraph (after removing punctuations) with all lower case**

CODE:

```
print("\nReg. No- 18BCE0154\n")
import string
print("INPUT:\n")
para = input()
table = str.maketrans(dict.fromkeys(string.punctuation))
new_para = para.translate(table)
new_para = new_para.lower()
print("\nOUTPUT:-\n")
print(new_para)
print("\nReg. No- 18BCE0154")
```

OUTPUT:

In [1]:
```
print("\nReg. No- 18BCE0154\n")
import string
print("INPUT:\n")
para = input()
table = str.maketrans(dict.fromkeys(string.punctuation))
new_para = para.translate(table)
new_para = new_para.lower()
print("\nOUTPUT:-\n")
print(new_para)
print("\nReg. No- 18BCE0154")
```

```
Reg. No- 18BCE0154

INPUT:

I am come amongst you, as you see, at this time, not for my recreation and disport, but being resolved, in the mid
st and heat of the battle, to live and die amongst you all; to lay down for my God, and for my kingdom, and my peo
ple, my honour and my blood, even in the dust. I know I have the body but of a weak and feeble woman; but I have t
he heart and stomach of a king, and of a king of England too, and think foul scorn that Parma or Spain, or any pri
nce of Europe, should dare to invade the borders of my realm; to which rather than any dishonour shall grow by me,
I myself will take up arms, I myself will be your general, judge, and rewarder of every one of your virtues in the
field.

OUTPUT:-

i am come amongst you as you see at this time not for my recreation and disport but being resolved in the midst an
d heat of the battle to live and die amongst you all to lay down for my god and for my kingdom and my people my ho
nour and my blood even in the dust i know i have the body but of a weak and feeble woman but i have the heart and
stomach of a king and of a king of england too and think foul scorn that parma or spain or any prince of europe sh
ould dare to invade the borders of my realm to which rather than any dishonour shall grow by me i myself will take
up arms i myself will be your general judge and rewarder of every one of your virtues in the field

Reg. No- 18BCE0154
```

b) Afterwards, remove stop words and print the whole paragraph again.

CODE:

```
print("\nReg. No- 18BCE0154\n")

import string

import nltk

nltk.download('stopwords')
```

```
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
print("INPUT:\n")
para = input()
table = str.maketrans(dict.fromkeys(string.punctuation))
new_para = para.translate(table)
new_para = new_para.lower()
example_sent = new_para
stop_words = set(stopwords.words('english'))
print("\n STOP WORDS:\n",stop_words)
word_tokens = word_tokenize(example_sent)
filtered_sentence = [w for w in word_tokens if not w in stop_words]
print("\nfiltered paragraph:\n"," ".join(filtered_sentence))
print("\nReg. No- 18BCE0154")
```

OUTPUT:

In [1]:
```
print("\nReg. No- 18BCE0154\n")
import string
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
print("INPUT:\n")
para = input()
table = str.maketrans(dict.fromkeys(string.punctuation))
new_para = para.translate(table)
new_para = new_para.lower()
example_sent = new_para
stop_words = set(stopwords.words('english'))
print("\n STOP WORDS:\n",stop_words)
word_tokens = word_tokenize(example_sent)
filtered_sentence = [w for w in word_tokens if not w in stop_words]
print("\nfiltered paragraph:\n"," ".join(filtered_sentence))
print("\nReg. No- 18BCE0154")
```

```
Reg. No- 18BCE0154

[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\LENOVO\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\LENOVO\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
INPUT:

I am come amongst you, as you see, at this time, not for my recreation and disport, but being resolved, in the midst and heat o
f the battle, to live and die amongst you all; to lay down for my God, and for my kingdom, and my people, my honour and my bloo
d, even in the dust. I know I have the body but of a weak and feeble woman; but I have the heart and stomach of a king, and of
a king of England too, and think foul scorn that Parma or Spain, or any prince of Europe, should dare to invade the borders of
my realm; to which rather than any dishonour shall grow by me, I myself will take up arms, I myself will be your general, judg
e, and rewarder of every one of your virtues in the field.

 STOP WORDS:
 {'does', 'him', 'you', 'down', 'there', "couldn't", "didn't", 'at', 's', 'are', 'each', 'own', 'weren', 'will', 'what', "you'r
e", 'shouldn', 'any', 'how', 'y', 'because', "shan't", 'but', 'few', 'why', 'can', 'doesn', 'about', 'above', "you've", 'be',
"weren't", 'on', "she's", 'to', 'its', 'once', 'just', 'had', 'theirs', 'have', 'again', "hasn't", "you'd", 'under', 'is', 'aft
er', 'as', 'the', 'off', 'myself', 'too', 'an', 'than', 'our', 'd', 're', "aren't", 'while', 'don', 'hadn', 'very', 'until',
't', 'yourself', 'didn', 'ours', 'shan', 'did', 'your', 'am', 'itself', "you'll", 'she', "doesn't", 'ain', 'all', 'only', 've',
'we', 'his', 'these', 'no', 'where', 'nor', 'mightn', 'them', 'not', 'if', 'm', 'aren', 'some', 'ma', 'o', 'was', 'won', 'll',
'they', "mightn't", 'herself', 'from', 'up', "won't", 'my', 'this', 'over', 'themselves', "hadn't", 'who', 'mustn', 'out', 'i',
'by', 'and', 'hasn', 'when', 'been', "should've", 'here', 'he', 'ourselves', 'couldn', 'with', 'wasn', 'that', 'it', 'other',
"it's", 'were', 'further', 'then', 'now', 'me', 'whom', "that'll", 'such', 'of', "isn't", 'below', 'most', "needn't", 'for', 't
hrough', 'isn', 'yours', "haven't", 'which', 'same', 'needn', 'yourselves', 'between', 'both', 'haven', 'into', 'their', 'would
n', 'in', 'doing', "mustn't", 'her', 'against', 'has', 'so', "don't", 'hers', "wouldn't", 'having', 'being', 'those', "should
n't", 'do', 'a', 'before', 'should', 'or', 'himself', 'more', "wasn't", 'during'}

filtered paragraph:
 come amongst see time recreation disport resolved midst heat battle live die amongst lay god kingdom people honour blood even
dust know body weak feeble woman heart stomach king king england think foul scorn parma spain prince europe dare invade borders
realm rather dishonour shall grow take arms general judge rewarder every one virtues field

Reg. No- 18BCE0154
```

c) Collect those words that occur only once in the paragraph and print them.

CODE:

```
print("\nReg. No- 18BCE0154\n")
import string
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
print("INPUT:\n")
para = input()
table = str.maketrans(dict.fromkeys(string.punctuation))
new_para = para.translate(table)
new_para = new_para.lower()
example_sent = new_para
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(example_sent)
filtered_sentence = [w for w in word_tokens if not w in stop_words]
S=" ".join(filtered_sentence)
l=S.split(" ")
print("\nfiltered paragraph:\n",S)
print("\nLIST OF WORDS APPEARING ONLY ONCE IN THE FILTERED PARAGRAPH:-")
for j in l:
    if S.count(j)==1 and j.isalpha():
```

print(j,end=", ")

print("\nReg. No- 18BCE0154")

In [2]:
```
print("\nReg. No- 18BCE0154\n")
import string
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
print("INPUT:\n")
para = input()
table = str.maketrans(dict.fromkeys(string.punctuation))
new_para = para.translate(table)
new_para = new_para.lower()
example_sent = new_para
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(example_sent)
filtered_sentence = [w for w in word_tokens if not w in stop_words]
S=" ".join(filtered_sentence)
l=S.split(" ")
print("\nfiltered paragraph:\n",S)
print("\nLIST OF WORDS APPEARING ONLY ONCE IN THE FILTERED PARAGRAPH:-")
for j in l:
    if S.count(j)==1 and j.isalpha():
        print(j,end=", ")
print("\nReg. No- 18BCE0154")
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\LENOVO\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\LENOVO\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

INPUT:

I am come amongst you, as you see, at this time, not for my recreation and disport, but being resolved, in the midst and heat o
f the battle, to live and die amongst you all; to lay down for my God, and for my kingdom, and my people, my honour and my bloo
d, even in the dust. I know I have the body but of a weak and feeble woman; but I have the heart and stomach of a king, and of
a king of England too, and think foul scorn that Parma or Spain, or any prince of Europe, should dare to invade the borders of
my realm; to which rather than any dishonour shall grow by me, I myself will take up arms, I myself will be your general, judg
e, and rewarder of every one of your virtues in the field.

filtered paragraph:
 come amongst see time recreation disport resolved midst heat battle live die amongst lay god kingdom people honour blood even
dust know body weak feeble woman heart stomach king king england think foul scorn parma spain prince europe dare invade borders
realm rather dishonour shall grow take arms general judge rewarder every one virtues field

LIST OF WORDS APPEARING ONLY ONCE IN THE FILTERED PARAGRAPH:-
come, see, time, recreation, disport, resolved, midst, heat, battle, live, die, lay, god, kingdom, people, blood, even, dust, k
now, body, weak, feeble, woman, heart, stomach, england, think, foul, scorn, parma, spain, prince, europe, dare, invade, border
s, realm, rather, dishonour, shall, grow, take, arms, general, judge, rewarder, every, one, virtues, field,
Reg. No- 18BCE0154

d) Collect those words that occur only twice in the paragraph and print them.

CODE:

```
print("\nReg. No- 18BCE0154\n")
import string
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
print("INPUT:\n")
para = input()
table = str.maketrans(dict.fromkeys(string.punctuation))
new_para = para.translate(table)
new_para = new_para.lower()
example_sent = new_para
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(example_sent)
filtered_sentence = [w for w in word_tokens if not w in stop_words]
S=" ".join(filtered_sentence)
```

```
l=set(S.split(" "))
print("\nfiltered paragraph:\n",S)
print("\nLIST OF WORDS APPEARING ONLY TWICE IN THE FILTERED PARAGRAPH:-")
for j in l:
    if S.count(j)==2 and j.isalpha():
        print(j,end=", ")
print("\nReg. No- 18BCE0154")
```

OUTPUT:

```
In [3]: print("\nReg. No- 18BCE0154\n")
        import string
        import nltk
        nltk.download('stopwords')
        nltk.download('punkt')
        from nltk.corpus import stopwords
        from nltk.tokenize import word_tokenize
        print("INPUT:\n")
        para = input()
        table = str.maketrans(dict.fromkeys(string.punctuation))
        new_para = para.translate(table)
        new_para = new_para.lower()
        example_sent = new_para
        stop_words = set(stopwords.words('english'))
        word_tokens = word_tokenize(example_sent)
        filtered_sentence = [w for w in word_tokens if not w in stop_words]
        S=" ".join(filtered_sentence)
        l=set(S.split(" "))
        print("\nfiltered paragraph:\n",S)
        print("\nLIST OF WORDS APPEARING ONLY TWICE IN THE FILTERED PARAGRAPH:-")
        for j in l:
            if S.count(j)==2 and j.isalpha():
                print(j,end=", ")
        print("\nReg. No- 18BCE0154")

        [nltk_data] Downloading package stopwords to
        [nltk_data]     C:\Users\LENOVO\AppData\Roaming\nltk_data...
        [nltk_data]   Package stopwords is already up-to-date!
        [nltk_data] Downloading package punkt to
        [nltk_data]     C:\Users\LENOVO\AppData\Roaming\nltk_data...
        [nltk_data]   Package punkt is already up-to-date!

        Reg. No- 18BCE0154
```

```
INPUT:

I am come amongst you, as you see, at this time, not for my recreation and disport, but being resolved, in the midst and heat o
f the battle, to live and die amongst you all; to lay down for my God, and for my kingdom, and my people, my honour and my bloo
d, even in the dust. I know I have the body but of a weak and feeble woman; but I have the heart and stomach of a king, and of
a king of England too, and think foul scorn that Parma or Spain, or any prince of Europe, should dare to invade the borders of
my realm; to which rather than any dishonour shall grow by me, I myself will take up arms, I myself will be your general, judg
e, and rewarder of every one of your virtues in the field.

filtered paragraph:
 come amongst see time recreation disport resolved midst heat battle live die amongst lay god kingdom people honour blood even
dust know body weak feeble woman heart stomach king king england think foul scorn parma spain prince europe dare invade borders
realm rather dishonour shall grow take arms general judge rewarder every one virtues field

LIST OF WORDS APPEARING ONLY TWICE IN THE FILTERED PARAGRAPH:-
honour, amongst,
Reg. No- 18BCE0154
```

e) **Make a dictionary of these 4 types of words and apply stemming to reduce inflected words to their word stem, base or root form.**

CODE

```
import string
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
print("INPUT:\n")
para = input()
table = str.maketrans(dict.fromkeys(string.punctuation))
new_para = para.translate(table)
new_para = new_para.lower()

example_sent = new_para
stop_words = set(stopwords.words('english'))

word_tokens = word_tokenize(example_sent)
filtered_sentence = [w for w in word_tokens if not w in stop_words]

S=" ".join(filtered_sentence)
l=set(S.split(" "))
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
ps = PorterStemmer()
sentence = S
words = word_tokenize(sentence)
dict={"learn":"to reand and understand","read":"to speak and understand
text","play":"take part in (a sport)","whole":"entire","night":"the period from sunset
to sunrise."}
```

```
for w in words:
    print(w, " : ", ps.stem(w),":Meaning:",dict[ps.stem(w)])
```

OUTPUT

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\LENOVO\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\LENOVO\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
INPUT:

i was learning whole night and then i played
learning  :  learn :Meaning: to reand and understand
whole  :  whole :Meaning: entire
night  :  night :Meaning: the period from sunset to sunrise.
played  :  play :Meaning: take part in (a sport)
```

In [ ]:

In [ ]: