In [ ]:

```
# انا رفعت الملف يوم الربوع الي فات وما كان ضابط بالعربي
# عدلته اليوم ورجعت رفعته
```

In [1]:

```
import nltk

from nltk.stem import PorterStemmer
Stemmerporter = PorterStemmer()
Stemmerporter.stem("understanding")
```

Out[1]:

```
'understand'
```

In [5]:

```
# التمرين الثاني
import nltk
from nltk.stem import PorterStemmer
Stemmerporter = PorterStemmer()

list=['dogs', 'programming', 'programs', 'programmed',
      'cakes', 'indices', 'matrices']
for word in list:
    print(f'{word} \t -> {Stemmerporter.stem(word)}'.expandtabs(15))
```

```
dogs            -> dog
programming     -> program
programs        -> program
programmed      -> program
cakes           -> cake
indices         -> indic
matrices        -> matric
```

In [9]:

```
#التمرين الثالث

from nltk.tokenize import TreebankWordTokenizer

sentence = 'A stemmer for English operating on the stem cat should identify such strings as

list = nltk.word_tokenize(sentence)

for word in list:
    print(f'{word} \t -> {Stemmerporter.stem(word)}'.expandtabs(15))
```

```
A               -> A
stemmer         -> stemmer
for             -> for
English         -> english
operating       -> oper
on              -> on
the             -> the
stem            -> stem
cat             -> cat
should          -> should
identify        -> identifi
such            -> such
strings         -> string
as              -> as
cats            -> cat
,               -> ,
catlike         -> catlik
,               -> ,
and             -> and
catty           -> catti
.               -> .
A               -> A
stemming        -> stem
algorithm       -> algorithm
might           -> might
also            -> also
reduce          -> reduc
the             -> the
words           -> word
fishing         -> fish
,               -> ,
fished          -> fish
,               -> ,
and             -> and
fisher          -> fisher
to              -> to
the             -> the
stem            -> stem
fish            -> fish
.               -> .
The             -> the
stem            -> stem
need            -> need
not             -> not
be              -> be
a               -> a
word            -> word
```

```
,               -> ,
for             -> for
example         -> exampl
the             -> the
Porter          -> porter
algorithm       -> algorithm
reduces         -> reduc
,               -> ,
argue           -> argu
,               -> ,
argued          -> argu
,               -> ,
argues          -> argu
,               -> ,
arguing         -> argu
,               -> ,
and             -> and
argus           -> argu
to              -> to
the             -> the
stem            -> stem
argu            -> argu
.               -> .
```

In [ ]:

```
# Porter: Most commonly used stemmer without a doubt,also one of the most gentle stemmers.
#of the few stemmers that actually has Java support
```

In [11]:

```
from nltk.tokenize import sent_tokenize, word_tokenize
sentence = '''A stemmer for English operating on the stem cat should identify such strings


tokenized_words = word_tokenize(sentence)
tokenized_sentence = []
for word in tokenized_words:
    tokenized_sentence.append(Stemmerporter.stem(word))
tokenized_sentence = " ".join(tokenized_sentence)
tokenized_sentence
```

Out[11]:

```
'A stemmer for english oper on the stem cat should identifi such string as c
at , catlik , and catty.a stem algorithm might also reduc the word fish , fi
sh , and fisher to the stem fish the stem need not be a word , for exampl th
e porter algorithm reduc , argu , argu , argu , argu , and argu to the stem
argu .'
```

In [12]:

```
# التمرين الرابع
import nltk
from nltk.stem import SnowballStemmer

StemmerSnowball = SnowballStemmer('english')

list=['dogs', 'programming', 'programs', 'programmed',
      'cakes', 'indices', 'matrices']
for word in list:
    print(StemmerSnowball.stem(word))
```

```
dog
program
program
program
cake
indic
matric
```

In [13]:

```
from nltk.stem.isri import ISRIStemmer

st = ISRIStemmer()
w = 'حركات'

print(st.stem(w))
```

```
حرك
```

In [18]:

```
file=open("C:\\Users\\user\\Desktop\\Faisal.txt")
Sentences= file.read()
def stemSentence(sentence):
    token_words=word_tokenize(sentence)
    token_words
    stem_sentence=[]
    for word in token_words:
        stem_sentence.append(Stemmerporter.stem(word))
        stem_sentence.append(" ")


    return "".join(stem_sentence)

print(Sentences)

zz=stemSentence(Sentences)
print(zz)
```

```
3700472 فيصل سامي الحربي

3700472 فيصل سامي الحربي
```

In [19]:

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

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

Out[19]:

```
True
```

In [20]:

```
# Lemmatization
# التمرين الخامس
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
sentence1 = "He was running and eating at same time. He has bad habit of swimming after pla

punctuations="?:!.,;"
sentence_words = nltk.word_tokenize(sentence1)
for word in sentence_words:
    if word in punctuations:
        sentence_words.remove(word)
sentence_words
print("{0:20}{1:20}".format("Word","Lemma"))
for word in sentence_words:
    print ("{0:20}{1:20}".format(word,wordnet_lemmatizer.lemmatize(word)))
```

```
Word                Lemma
He                  He
was                 wa
running             running
and                 and
eating              eating
at                  at
same                same
time                time
He                  He
has                 ha
bad                 bad
habit               habit
of                  of
swimming            swimming
after               after
playing             playing
long                long
hours               hour
in                  in
the                 the
Sun                 Sun
```

In [21]:

```
for word in sentence_words:
     print ("{0:20}{1:20}".format(word,wordnet_lemmatizer.lemmatize(word, pos="v")))
```

```
He                  He
was                 be
running             run
and                 and
eating              eat
at                  at
same                same
time                time
He                  He
has                 have
bad                 bad
habit               habit
of                  of
swimming            swim
after               after
playing             play
long                long
hours               hours
in                  in
the                 the
Sun                 Sun
```

In [25]:

```
# الكلمة هيا خجول
# التمرين السادس
file=open("C:\\Users\\user\\Desktop\\Faisal.txt")
from nltk.stem.isri import ISRIStemmer

st = ISRIStemmer()

Sentences= file.read()

def stemSentence(sentence):
    token_words=word_tokenize(sentence)
    token_words
    stem_sentence=[]
    for word in token_words:
        stem_sentence.append(st.stem(word))
        stem_sentence.append(" ")
    return "".join(stem_sentence)


x = stemSentence(Sentences)
print(x)
```

خجل

In [ ]:

```

```