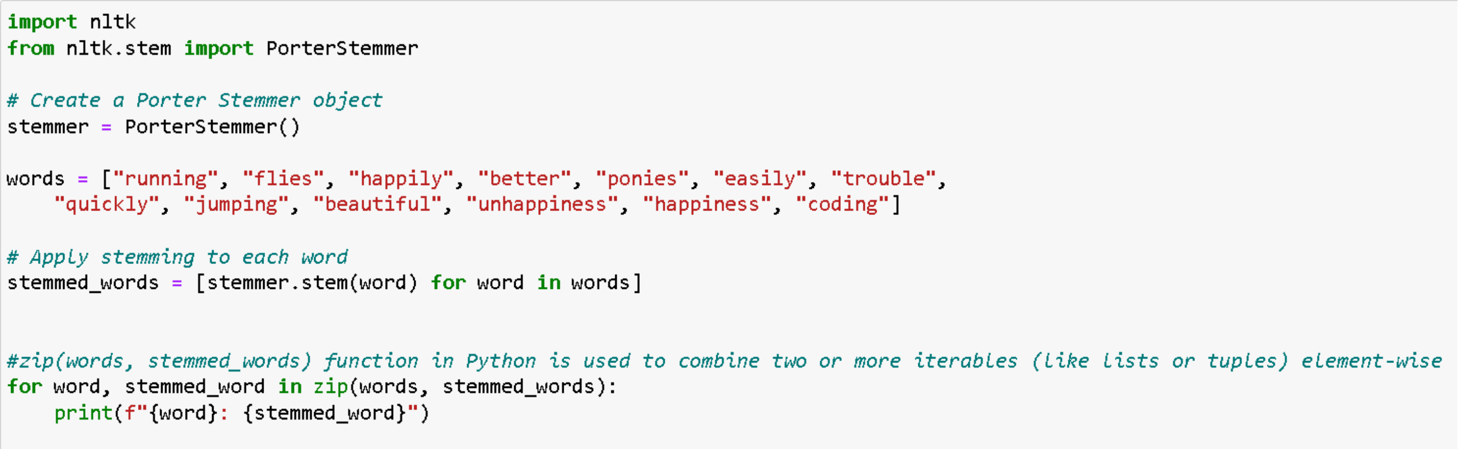
Porter Stemmer Algorithm

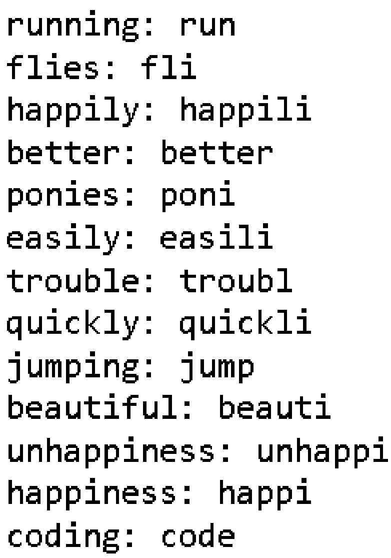
1)Using nltk

Input :



```python
import nltk
from nltk.stem import PorterStemmer

# Create a Porter Stemmer object
stemmer = PorterStemmer()

words = ["running", "flies", "happily", "better", "ponies", "easily", "trouble",
    "quickly", "jumping", "beautiful", "unhappiness", "happiness", "coding"]

# Apply stemming to each word
stemmed_words = [stemmer.stem(word) for word in words]


#zip(words, stemmed_words) function in Python is used to combine two or more iterables (like lists or tuples) element-wise
for word, stemmed_word in zip(words, stemmed_words):
    print(f"{word}: {stemmed_word}")
```

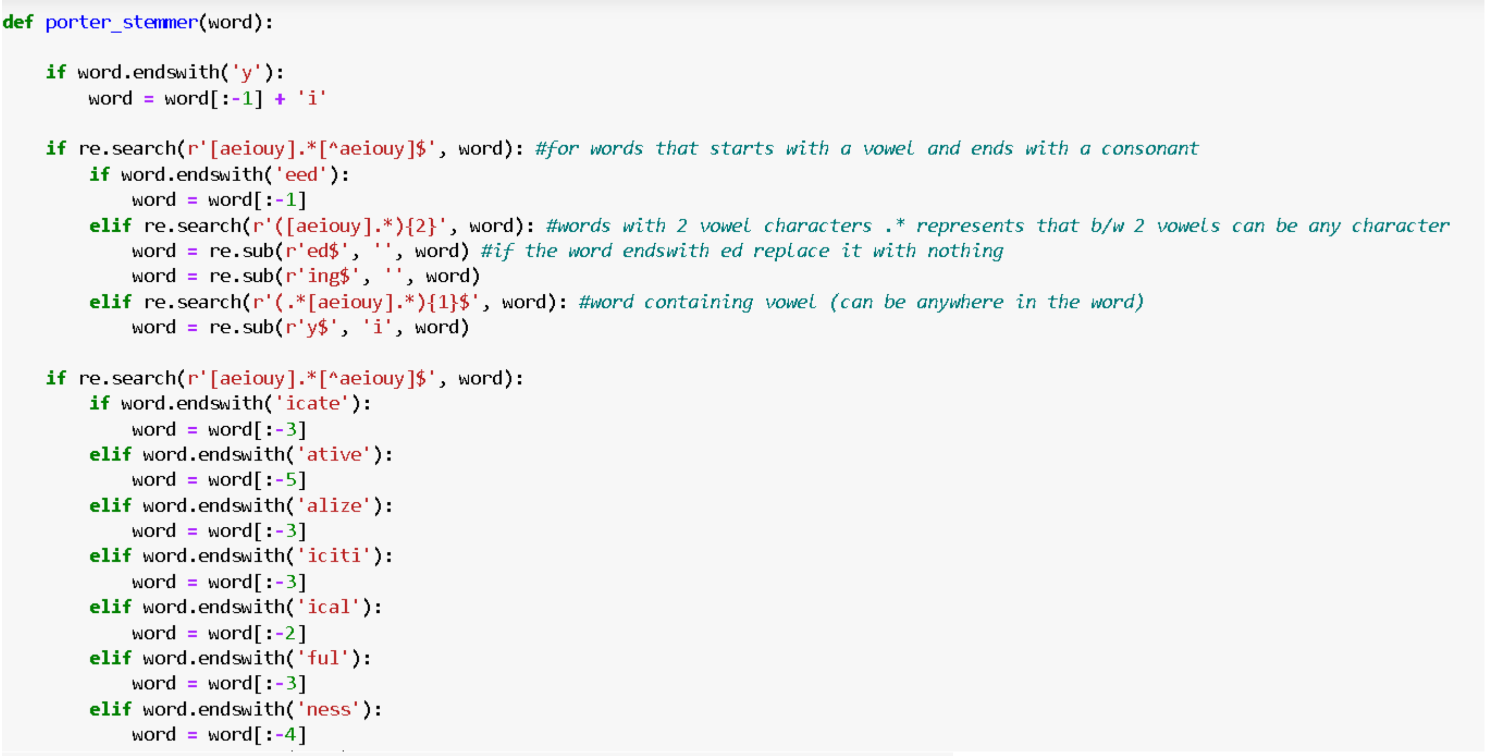
Output :



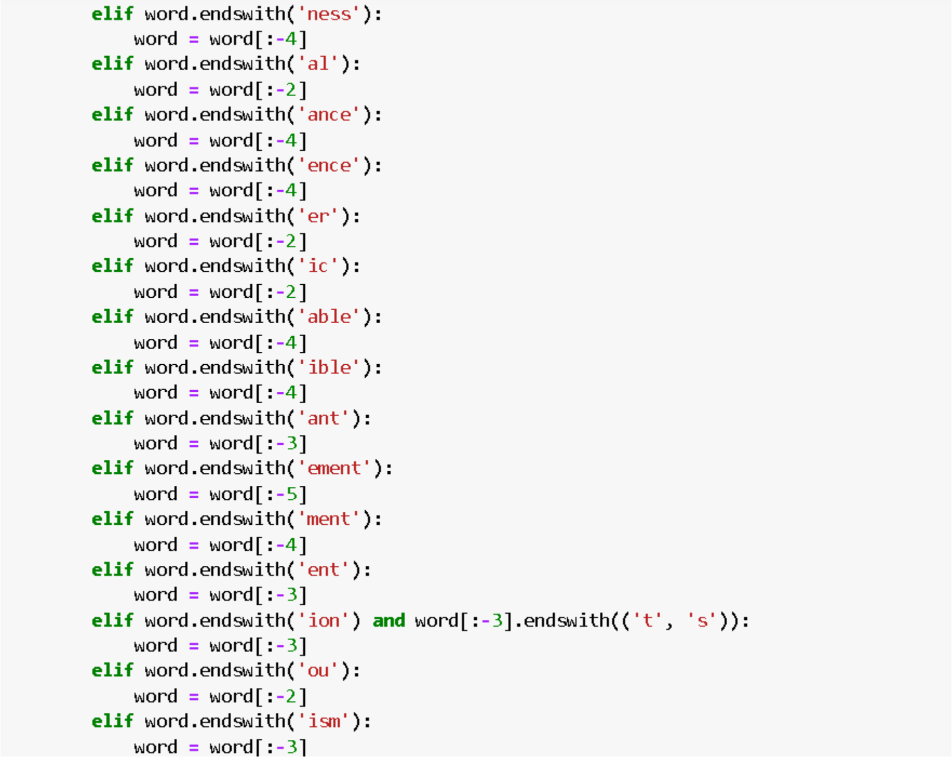
```
running: run
flies: fli
happily: happili
better: better
ponies: poni
easily: easili
trouble: troubl
quickly: quickli
jumping: jump
beautiful: beauti
unhappiness: unhappi
happiness: happi
coding: code
```

2) Without Using NLTK

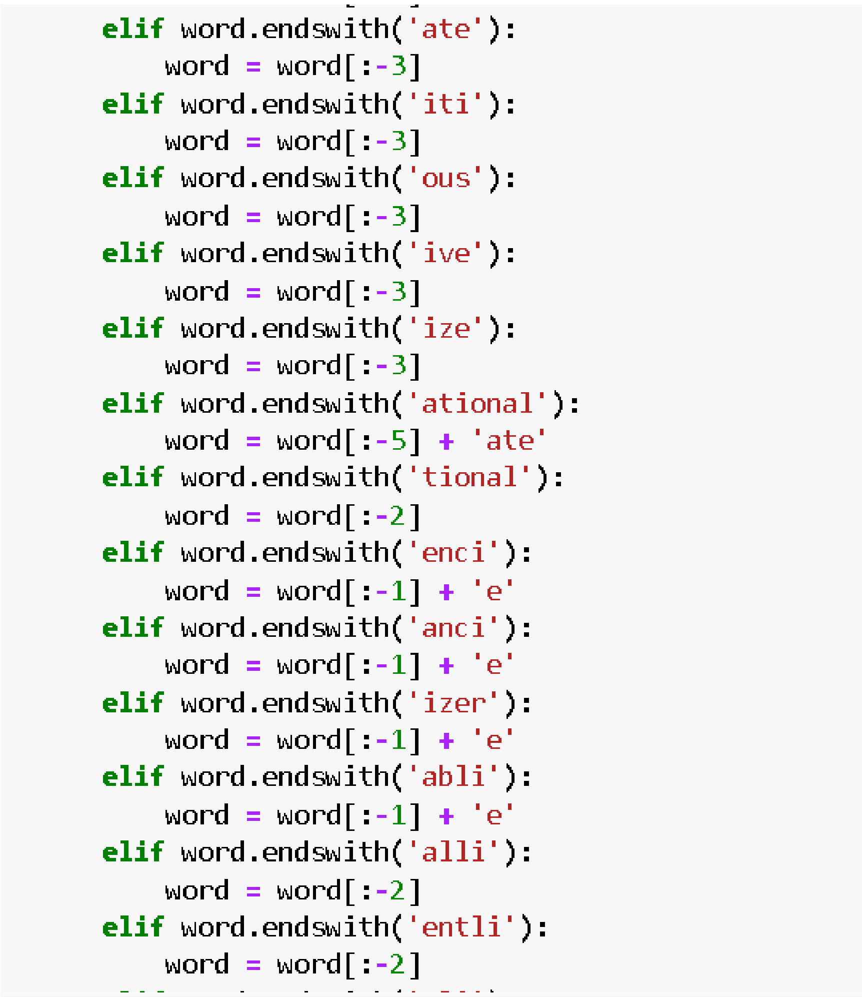
Input :



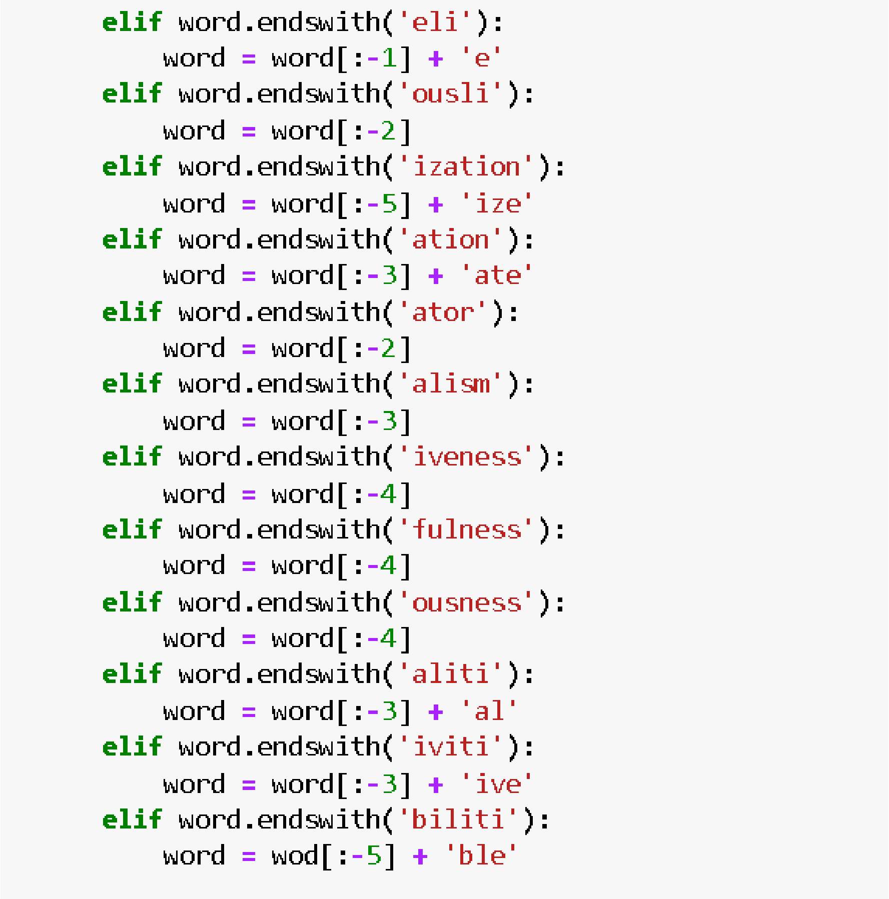
```python
def porter_stemmer(word):

    if word.endswith('y'):
        word = word[:-1] + 'i'

    if re.search(r'[aeiouy].*[^aeiouy]$', word): #for words that starts with a vowel and ends with a consonant
        if word.endswith('eed'):
            word = word[:-1]
        elif re.search(r'([aeiouy].*){2}', word): #words with 2 vowel characters .* represents that b/w 2 vowels can be any character
            word = re.sub(r'ed$', '', word) #if the word endswith ed replace it with nothing
            word = re.sub(r'ing$', '', word)
        elif re.search(r'(.*[aeiouy].*){1}$', word): #word containing vowel (can be anywhere in the word)
            word = re.sub(r'y$', 'i', word)

    if re.search(r'[aeiouy].*[^aeiouy]$', word):
        if word.endswith('icate'):
            word = word[:-3]
        elif word.endswith('ative'):
            word = word[:-5]
        elif word.endswith('alize'):
            word = word[:-3]
        elif word.endswith('iciti'):
            word = word[:-3]
        elif word.endswith('ical'):
            word = word[:-2]
        elif word.endswith('ful'):
            word = word[:-3]
        elif word.endswith('ness'):
            word = word[:-4]
```



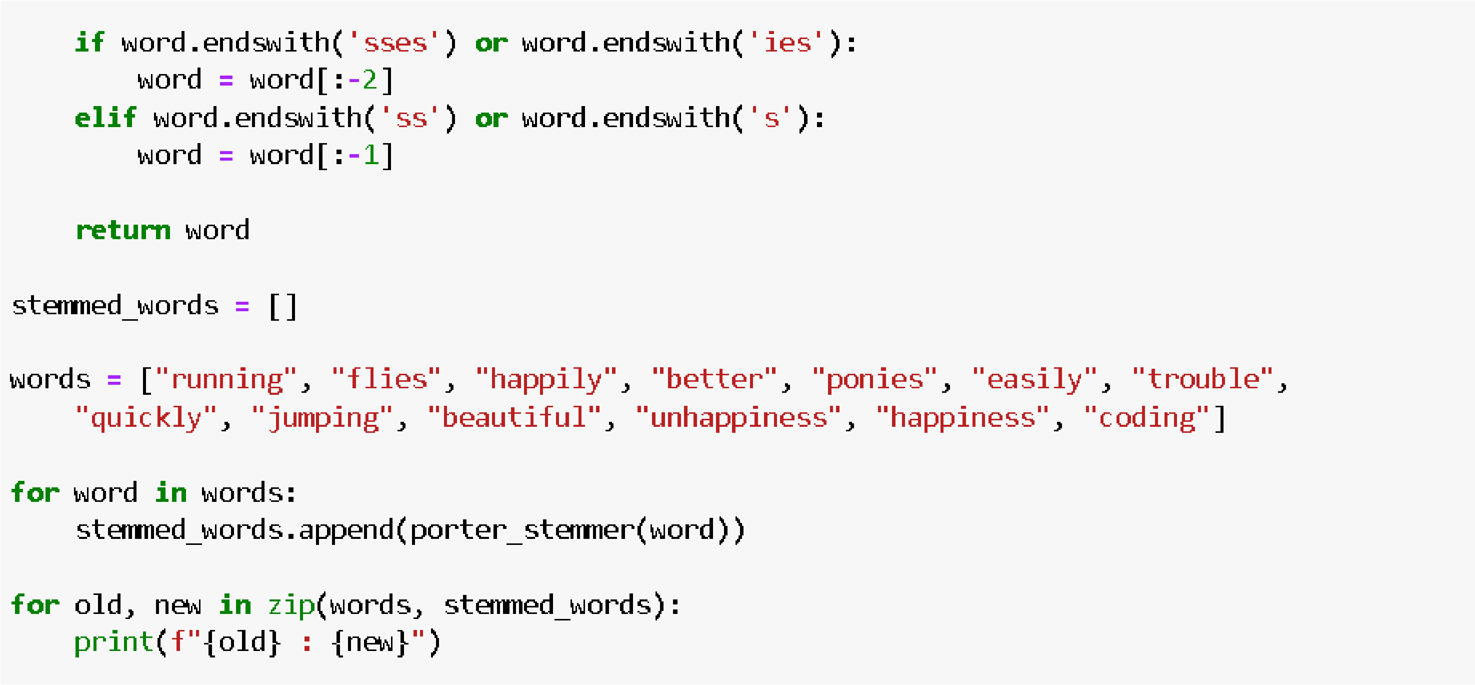
```python
        elif word.endswith('ness'):
            word = word[:-4]
        elif word.endswith('al'):
            word = word[:-2]
        elif word.endswith('ance'):
            word = word[:-4]
        elif word.endswith('ence'):
            word = word[:-4]
        elif word.endswith('er'):
            word = word[:-2]
        elif word.endswith('ic'):
            word = word[:-2]
        elif word.endswith('able'):
            word = word[:-4]
        elif word.endswith('ible'):
            word = word[:-4]
        elif word.endswith('ant'):
            word = word[:-3]
        elif word.endswith('ement'):
            word = word[:-5]
        elif word.endswith('ment'):
            word = word[:-4]
        elif word.endswith('ent'):
            word = word[:-3]
        elif word.endswith('ion') and word[:-3].endswith(('t', 's')):
            word = word[:-3]
        elif word.endswith('ou'):
            word = word[:-2]
        elif word.endswith('ism'):
            word = word[:-3]
```



```
        elif word.endswith('ate'):
            word = word[:-3]
        elif word.endswith('iti'):
            word = word[:-3]
        elif word.endswith('ous'):
            word = word[:-3]
        elif word.endswith('ive'):
            word = word[:-3]
        elif word.endswith('ize'):
            word = word[:-3]
        elif word.endswith('ational'):
            word = word[:-5] + 'ate'
        elif word.endswith('tional'):
            word = word[:-2]
        elif word.endswith('enci'):
            word = word[:-1] + 'e'
        elif word.endswith('anci'):
            word = word[:-1] + 'e'
        elif word.endswith('izer'):
            word = word[:-1] + 'e'
        elif word.endswith('abli'):
            word = word[:-1] + 'e'
        elif word.endswith('alli'):
            word = word[:-2]
        elif word.endswith('entli'):
            word = word[:-2]
```

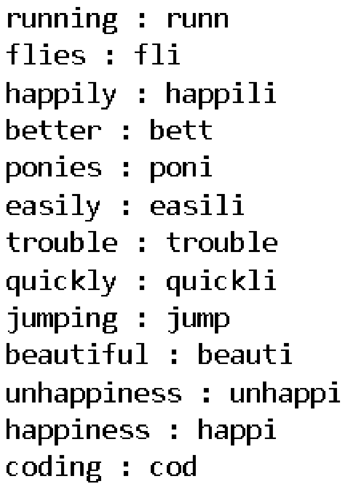


```
        elif word.endswith('eli'):
            word = word[:-1] + 'e'
        elif word.endswith('ousli'):
            word = word[:-2]
        elif word.endswith('ization'):
            word = word[:-5] + 'ize'
        elif word.endswith('ation'):
            word = word[:-3] + 'ate'
        elif word.endswith('ator'):
            word = word[:-2]
        elif word.endswith('alism'):
            word = word[:-3]
        elif word.endswith('iveness'):
            word = word[:-4]
        elif word.endswith('fulness'):
            word = word[:-4]
        elif word.endswith('ousness'):
            word = word[:-4]
        elif word.endswith('aliti'):
            word = word[:-3] + 'al'
        elif word.endswith('iviti'):
            word = word[:-3] + 'ive'
        elif word.endswith('biliti'):
            word = wod[:-5] + 'ble'
```



```
    if word.endswith('sses') or word.endswith('ies'):
        word = word[:-2]
    elif word.endswith('ss') or word.endswith('s'):
        word = word[:-1]

    return word

stemmed_words = []

words = ["running", "flies", "happily", "better", "ponies", "easily", "trouble",
    "quickly", "jumping", "beautiful", "unhappiness", "happiness", "coding"]

for word in words:
    stemmed_words.append(porter_stemmer(word))

for old, new in zip(words, stemmed_words):
    print(f"{old} : {new}")
```

Output :



```
running : runn
flies : fli
happily : happili
better : bett
ponies : poni
easily : easili
trouble : trouble
quickly : quickli
jumping : jump
beautiful : beauti
unhappiness : unhappi
happiness : happi
coding : cod
```

Repo link

https://github.com/Shreyaww/Sem7_NLP/blob/main/Experiment3.ipynb