

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Кафедра САПР

ЗВІТ

до лабораторної роботи № 5

на тему:

ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ
ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ
ПОЧАТКОВА ОБРОБКА ТЕКСТІВ ПРИРОДНОЮ МОВОЮ
з дисципліни “Комп’ютерна лінгвістика”

Виконала:

Студентка групи ПРЛм-12

Рибчак Х. В.

Перевірив:

Асистент кафедри САПР

Дупак Б. П.

Львів 2015

МЕТА РОБОТИ

Вивчення основ програмування на мові Python. Вивчення методів роботи з файлами на локальних дисках та з Інтернету. Використання Юнікоду при обробці текстів. Нормалізація текстів, стемінг, лематизація та сегментація.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Корпуси текстів та тексти з Інтернету є важливими джерелами даних для здійснення лінгвістичних досліджень. Звичайно, якщо дослідник має власноруч зібрані тексти, то потрібні засоби для доступу до них.

Частина електронних книжок з Project Gutenberg розповсюджується разом з NLTK у вигляді корпусу текстів. Для використання інших текстів з цього проекту можна переглянути каталог 25000 електронних книжок за адресою <http://www.gutenberg.org/catalog/> та встановити адресу (URL) потрібного текстового файлу в ASCII кодуванні. 90% текстів в Project Gutenberg є англійською мовою, але він включає також тексти більше ніж 50-ма іншими мовами (каталонська, китайська, датська, фінська, французька, німецька, італійська, португальська, іспанська...)

Більшість текстів в Інтернеті є у вигляді HTML документів (файлів). Інтернет сторінки можна зберігати на диску у вигляді файлів і доступатися до них. Python також дозволяє працювати Інтернет сторінками безпосередньо використовуючи функцію `urlopen`.

Для читання локальних файлів необхідно використовувати вбудовану функцію Python `open()` та `read()` метод. Для перевірки чи дійсно файл є в потрібній директорії у графічному інтерфейсі IDLE використовується команда *Open* з пункту меню *File*.

Для вводу тексту з клавіатури (при взаємодії користувача з програмою) потрібно використати функцію `raw_input()`. Після збереження введеного тексту у змінній з ним можна працювати як зі звичайною стрічкою.

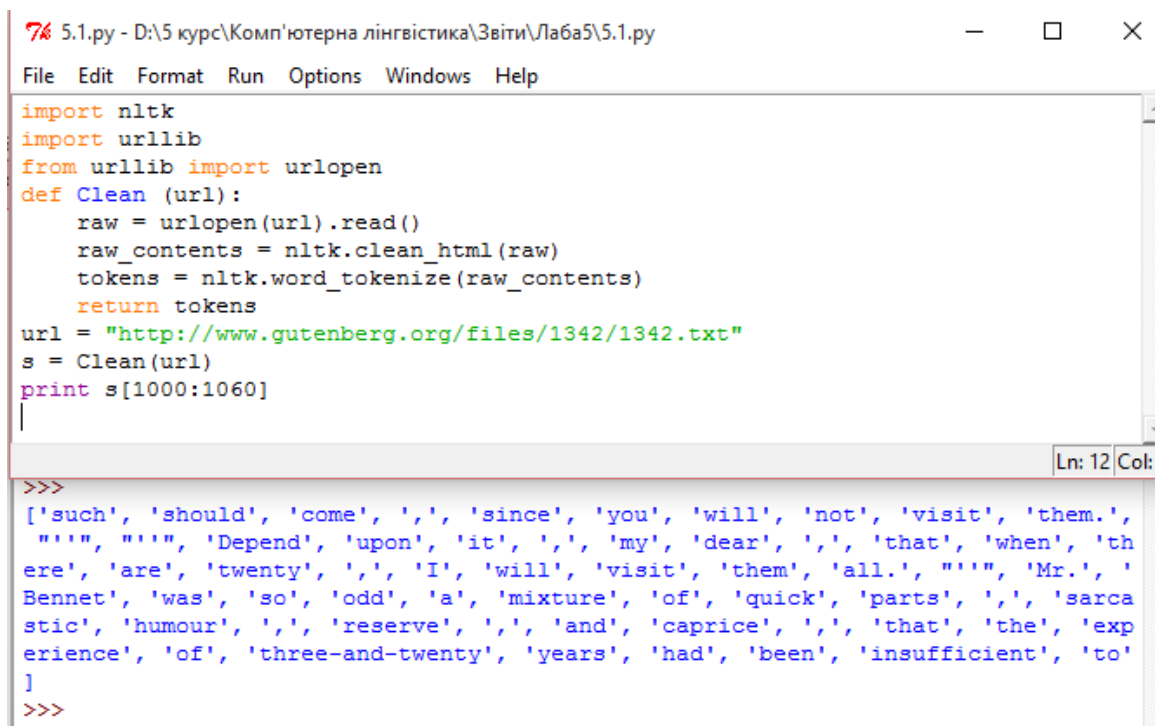
Юнікод, (англ. *Unicode*) — це промисловий стандарт розроблений, щоб зробити можливим для текстів і символів (графічних знаків) всіх писемних систем світу узгоджене представлення (репрезентацію) і обробку комп'ютерами. Юнікод підтримує більш ніж мільйон символів. Кожному символу ставиться у відповідність число, яке називають кодовою точкою. В Python кодові точки записуються у вигляді `\uXXXX` , де `XXXX` - чотири символи шістнадцяткового числа.

В межах програми обробка стрічок Unicode відбувається аналогічно до звичайних стрічок. Однак, коли Unicode символи зберігаються у файл або виводяться на екран, вони повинні бути закодовані, як потік байт. Деякі кодування (такі як ASCII та Latin-2) використовують один байт для представлення одної кодової точки і відповідно підтримують невеликий набір символів Unicode, достатній для одної мови. Інші кодування (такі як UTF-8) використовують послідовності байтів і можуть представити весь набір символів Unicode.

ТЕКСТИ ПРОГРАМ НА МОВІ PYTHON

ВАРІАНТ №8

1. Напишіть функцію, яка приймає адресу URL, як аргумент, і повертає те що міститься за цією адресою з видаленням HTML розмітки. Використовувати urllib.urlopen для доступу до контенту наступним чином `raw_contents = urllib.urlopen('http://www.nltk.org/').read()`.



```
7% 5.1.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба5\5.1.py
File Edit Format Run Options Windows Help

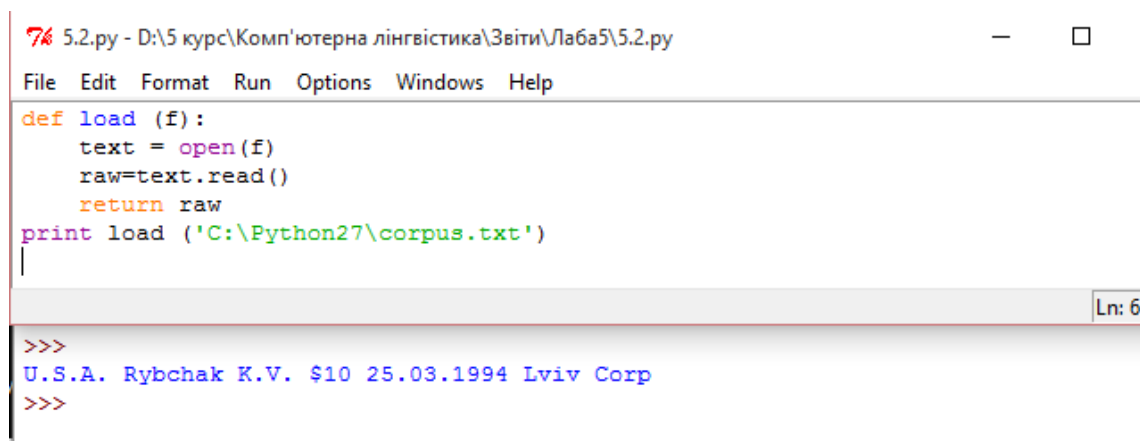
import nltk
import urllib
from urllib import urlopen
def Clean (url):
    raw = urlopen(url).read()
    raw_contents = nltk.clean_html(raw)
    tokens = nltk.word_tokenize(raw_contents)
    return tokens
url = "http://www.gutenberg.org/files/1342/1342.txt"
s = Clean(url)
print s[1000:1060]
|

Ln: 12 Col: 1

>>>
['such', 'should', 'come', ',', 'since', 'you', 'will', 'not', 'visit', 'them.',
'', '', 'Depend', 'upon', 'it', ',', 'my', 'dear', ',', 'that', 'when', 'th
ere', 'are', 'twenty', ',', 'I', 'will', 'visit', 'them', 'all.', '', 'Mr.', '
Bennet', 'was', 'so', 'odd', 'a', 'mixture', 'of', 'quick', 'parts', ',', 'sarca
stic', 'humour', ',', 'reserve', ',', 'and', 'caprice', ',', 'that', 'the', 'exp
erience', 'of', 'three-and-twenty', 'years', 'had', 'been', 'insufficient', 'to'
]
>>>
```

Рис. 1. Текст програми №1.

2. Збережіть деякий текст у файлі corpus.txt. Визначити функцію load(f) для читання файлу, назва якого є її аргументом і повертає стрічку, яка містить текст з файлу.



```
7% 5.2.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба5\5.2.py
File Edit Format Run Options Windows Help

def load (f):
    text = open(f)
    raw=text.read()
    return raw
print load ('C:\Python27\corpus.txt')
|

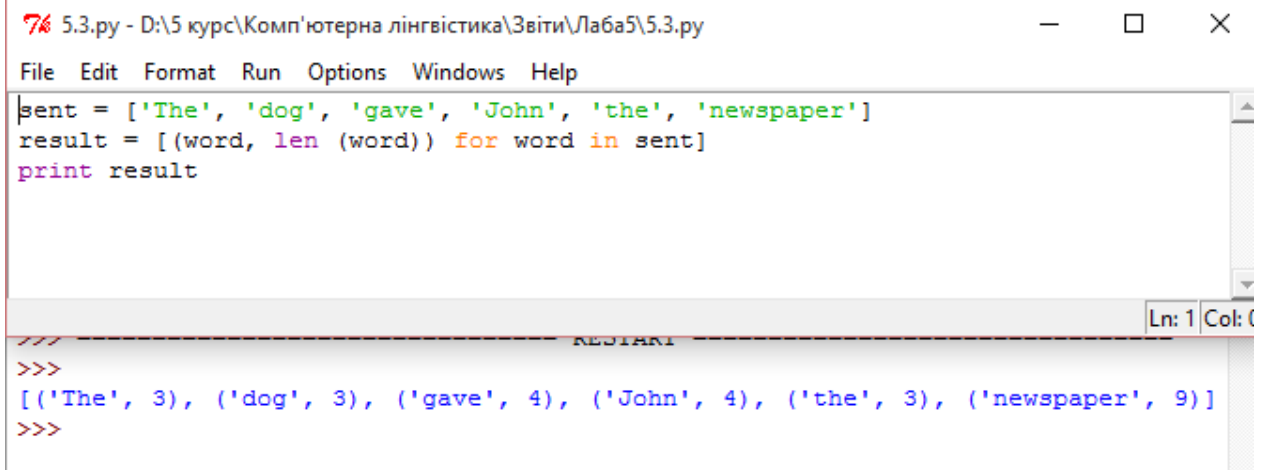
Ln: 6

>>>
U.S.A. Rybchak K.V. $10 25.03.1994 Lviv Corp
>>>
```

Рис. 2. Текст програми №2.

3. Перепишіть наступний цикл як list comprehension:

```
>>> sent = ['The', 'dog', 'gave', 'John', 'the', 'newspaper']
>>> result = []
>>> for word in sent:
...     word_len = (word, len(word))
...     result.append(word_len)
>>> result
[('The', 3), ('dog', 3), ('gave', 4), ('John', 4), ('the', 3), ('newspaper', 9)]
```



The screenshot shows a Python IDE window titled "5.3.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба5\5.3.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The editor contains the following code:

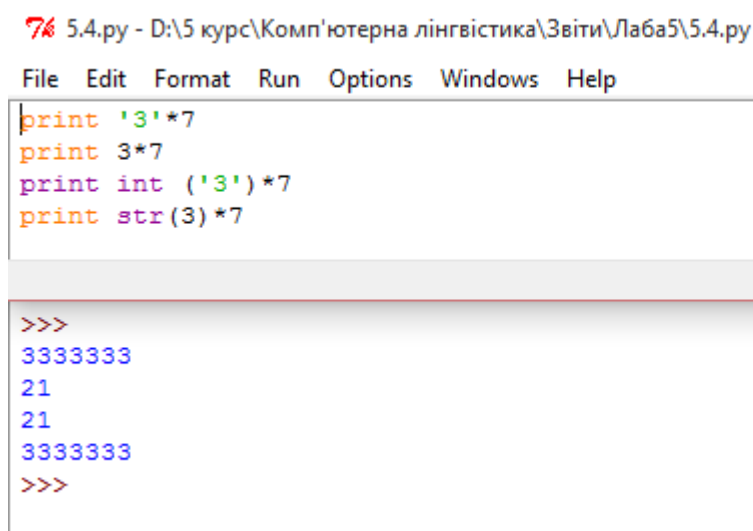
```
sent = ['The', 'dog', 'gave', 'John', 'the', 'newspaper']
result = [(word, len(word)) for word in sent]
print result
```

The output console shows the result of the list comprehension:

```
>>>
[('The', 3), ('dog', 3), ('gave', 4), ('John', 4), ('the', 3), ('newspaper', 9)]
>>>
```

Рис. 3. Текст програми №3.

4. Перевірити різницю між стрічками і цілим виконавши наступні дії: "3" * 7 та 3 * 7. Спробуйте здійснити конвертування між стрічками і цілими використавши int("3") та str(3).



The screenshot shows a Python IDE window titled "5.4.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба5\5.4.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The editor contains the following code:

```
print '3'*7
print 3*7
print int('3')*7
print str(3)*7
```

The output console shows the results of these operations:

```
>>>
3333333
21
21
3333333
>>>
```

Рис. 4. Текст програми №4.

5. Що станеться, коли стрічки форматування %6s та %-6s використовується для відображення стрічки довшої ніж 6 символів?

7% 5.5.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба5\5.5.py

File Edit Format Run Options Windows Help

```
print '%6s' % 'love'
print '%-6s' % 'love'
print '%6s' % 'interpreter'
print '%-6s' % 'interpreter'
```

```
>>>
love
love
interpreter
interpreter
>>>
```

Рис. 5. Текст програми №5.

7. Створіть файл, який буде містити слова та їх частоту записані в окремих рядках через пробіл (fuzzy 53). Прочитайте цей файл використовуючи open(filename).readlines(). Розділіть кожен рядок на дві частини використовуючи split(), і перетворіть число в ціле значення використовуючи int(). Результат повинен бути у вигляді списку: [['fuzzy', 53], ...].

7% 5.7.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба5\5.7.py

File Edit Format Run Options Windows Help

```
from __future__ import division
import nltk, re, pprint
file1 = open('C:\Python27\corpus.txt').readlines()
print file1

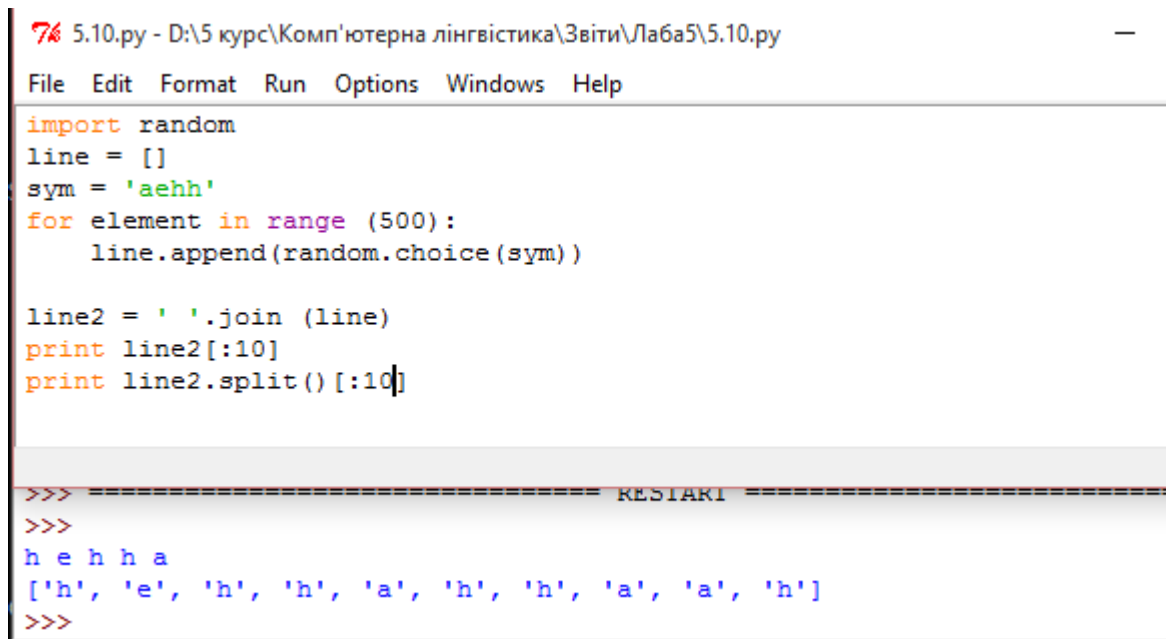
text = file1
list = []
for element in text:
    list.append ([element.split() [0], int(element.split() [1])])

print list
```

```
>>>
['dog 10\n', 'cat 7\n', 'snow 5']
[['dog', 10], ['cat', 7], ['snow', 5]]
>>>
```

Рис. 6. Текст програми №7.

10. Модуль random включає функцію choice(), яка випадковим чином вибирає елементи послідовності. Наприклад, choice("aehh ") буде вибирати один з чотирьох символів. Напишіть програму генерації стрічки з 500 випадково вибраних символів "aehh ". Для поєднання елементів в стрічку використовуйте ".join()". Нормалізуйте отриманий результат використовуючи split() та join().



```
7% 5.10.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба5\5.10.py
File Edit Format Run Options Windows Help

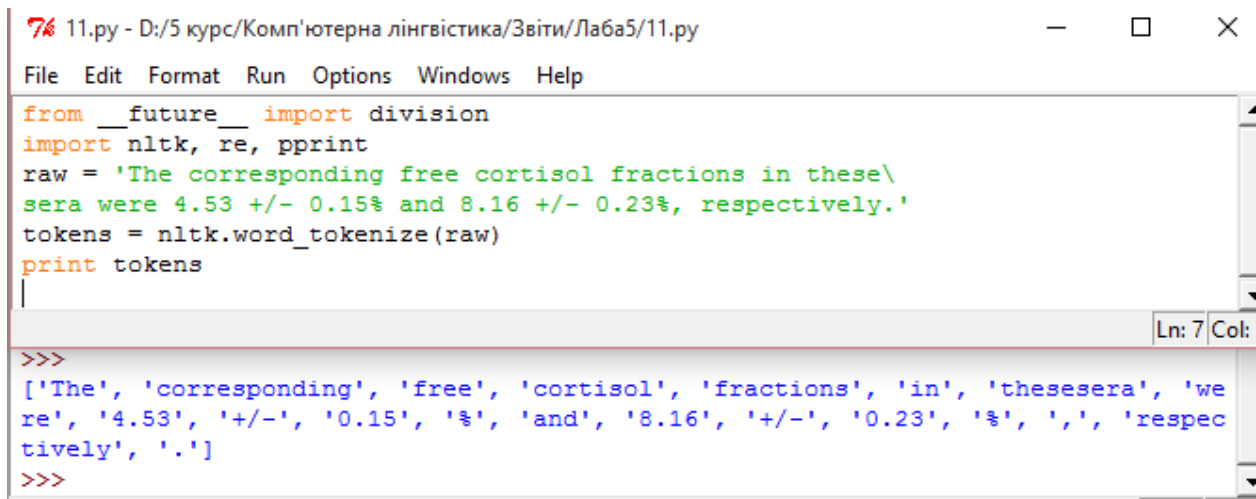
import random
line = []
sym = 'aehh'
for element in range (500):
    line.append(random.choice(sym))

line2 = ' '.join (line)
print line2[:10]
print line2.split()[:10]

>>> ===== RESTART =====
>>>
h e h h a
['h', 'e', 'h', 'h', 'a', 'h', 'h', 'a', 'a', 'h']
>>>
```

Рис. 7. Текст програми №10.

11. Здійсніть аналіз числового виразу в наступному реченні з корпусу MedLine: The corresponding free cortisol fractions in these sera were 4.53 +/- 0.15% and 8.16 +/- 0.23%, respectively. Чи можна сказати, що 4.53 +/- 0.15% це три окремих слова? Чи це одне складне слово? Чи це дев'ять слів "four point five three, plus or minus fifteen percent"? Чи це взагалі не можна вважати словом? При вирішенні яких задач потрібно вибирати ту чи іншу відповідь?



```
11.py - D:/5 курс/Комп'ютерна лінгвістика/Звіти/Лаба5/11.py
File Edit Format Run Options Windows Help

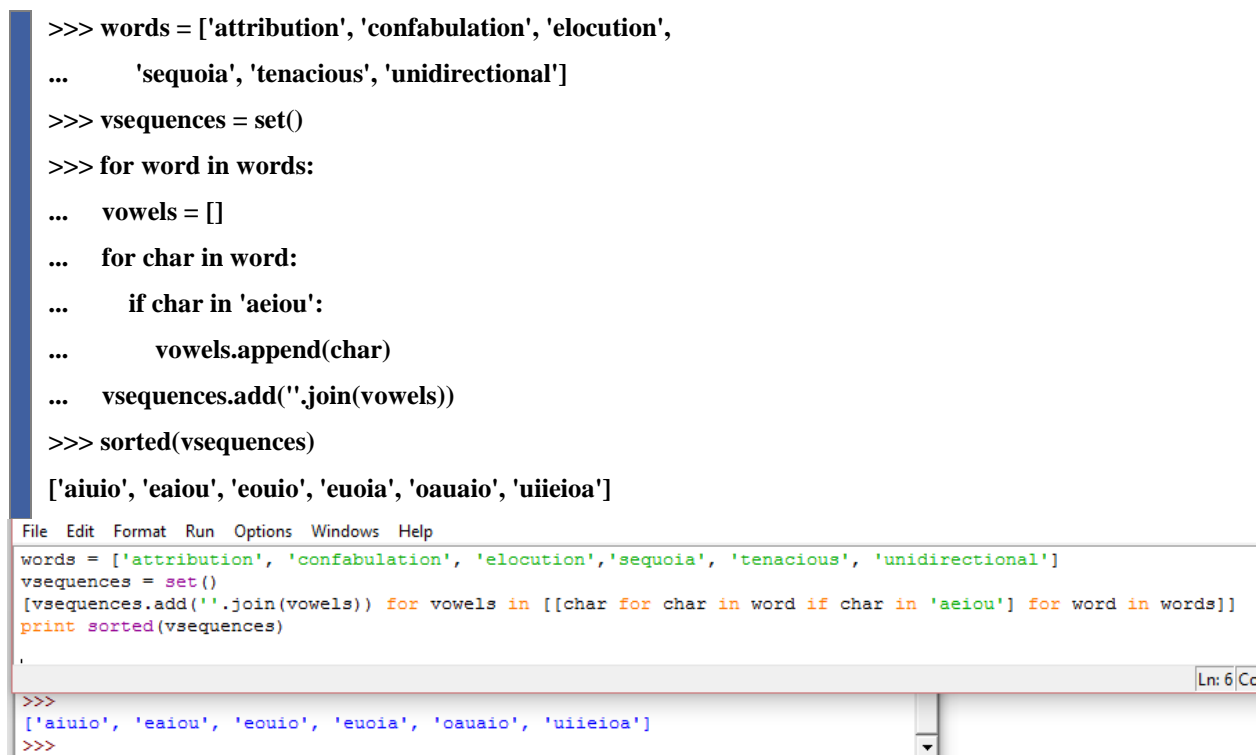
from __future__ import division
import nltk, re, pprint
raw = 'The corresponding free cortisol fractions in these\
sera were 4.53 +/- 0.15% and 8.16 +/- 0.23%, respectively.'
tokens = nltk.word_tokenize(raw)
print tokens

Ln: 7 Col:

>>>
['The', 'corresponding', 'free', 'cortisol', 'fractions', 'in', 'thesesera', 'we', 're', '4.53', '+/-', '0.15%', 'and', '8.16', '+/-', '0.23%', '%', ',', 'respectively', '.']
>>>
```

Рис. 8. Текст програми №11.

15. Перепишіть наступний цикл, як list comprehension:



```
>>> words = ['attribution', 'confabulation', 'elocution',
...          'sequoia', 'tenacious', 'unidirectional']
>>> vsequences = set()
>>> for word in words:
...     vowels = []
...     for char in word:
...         if char in 'aeiou':
...             vowels.append(char)
...     vsequences.add(''.join(vowels))
>>> sorted(vsequences)
['aiuiio', 'eaiau', 'eouio', 'euoia', 'ouaiao', 'uieioa']

File Edit Format Run Options Windows Help
words = ['attribution', 'confabulation', 'elocution', 'sequoia', 'tenacious', 'unidirectional']
vsequences = set()
[vsequences.add(''.join(vowels)) for vowels in [[char for char in word if char in 'aeiou'] for word in words]]
print sorted(vsequences)

Ln: 6 Cc

>>>
['aiuiio', 'eaiau', 'eouio', 'euoia', 'ouaiao', 'uieioa']
>>>
```

Рис. 9. Текст програми №15.

ВИСНОВОК

У цій лабораторній роботі я вивчила основи програмування на Python. Вивчила методи роботи з файлами на локальних дисках та з Інтернету, дізналася про використання Юнікоду при обробці текстів. Ознайомила з такими методами як нормалізація текстів, стемінг, лематизація та сегментація.