МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра "Системи автоматизованого проектування"

Звіт

до лабораторної роботи №8

на тему: ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.СТРУКТУРНЕ ПРОГРАМУВАННЯ МОВОЮ PYTHON (частина2).

з дисципліни "Комп'ютерна лінгвістика"

Виконала:
студентка групи ПРЛм-11
Гарбуз Л.В.
Прийняв:
викладач
Дупак Б.П.

Львів 2015

Мета роботи: вивчення основ програмування на мові Python. Вивчення основ структурного програмування мовою Python. Повторення та закріплення знань отриманих при виконанні попередніх лабораторних робіт. Покращення загальних навичок у програмуванні.

Тексти програм на мові *Python*.

Варіант – 3

3. Створити список списків слів [[‘’,’’,’’…], [[‘’,’’,’’…], [[‘’,’’,’’…],…] (наприклад текст складається з речень, які складаються з стрічок). Здійснити операцію присвоювання text2 = text1[:], та здійснити операцію присвоювання нового значення одному зі слів (text1[1][1] = 'Monty'). Перевірити як ці операції вплинули на text2. Результат письмово пояснити.

```
>>> text1=[['This','is','rainy','day','.'],['Here','comes','the','rain','again',
'.']]
>>> text2=text1[:]
>>> text1[1][1]='goes'
>>> text1
[['This', 'is', 'rainy', 'day', '.'], ['Here', 'goes', 'the', 'rain', 'again', '
.']]
>>> text2
[['This', 'is', 'rainy', 'day', '.'], ['Here', 'goes', 'the', 'rain', 'again', '
.']]
```

6. Написати програму для створення двовимірного масиву word_vowels елементами якого є набори. Програма повинна обробити список слів і додати результати обробки до word_vowels[l][v] де l – довжина слова, v – кількість голосних у слові.

```
import numpy
import re
list = ['Ramstein','Hardkiss','Royksopp']
word_vowels=numpy.zeros(shape=(len(list), 2))
for i in range(len(list)):
    for j in range(1):
        word_vowels[i,j]=len(list[i])
        vocals = re.findall(r'[aoeiu]',list[i])
        word_vowels[i,j+1]=len(''.join(vocals))
print word_vowels
```

```
>>>
[[ 8.  3.]
 [ 8.  2.]
 [ 8.  2.]]
>>>
```

11. Гематрія – метод виявлення прихованого змісту слів на основі порівняння чисел, які відповідають словам. Слова з однаковими числами мають

однаковий зміст. Число слова визначається сумуванням чисел, як відповідають його літерам. Написати функцію decode() для обробки тексту, яка випадковим чином замінює слова на їх Гематрія-еквіваленти. Чи вдалося виявити "прихований зміст" тексту? (Використовувати letter_vals з попередньої задачі)

```
import nltk
text='Come as you are, as you were, as I want you to be'
def decode(text):
    import random, string
    s=0
    text_s=text.lower().split()
    word=random.choice(text_s)
    letter_vals = {'a':1, 'b':2, 'c':3, 'd':4, 'e':5, 'f':80, 'g':3,
                   'h':8,'i':10, 'j':10, 'k':20, 'l':30, 'm':40, 'n':50, 'o':70,
                   'q':100, 'r':200, 's':300, 't':400, 'u':6, 'v':6, 'w':800, 'x
                   'z':7}
    for w in letter_vals.keys():
        if w in word:
            s=s+letter_vals[w]
    for i in text_s:
        if i==word:
            text_s[text_s.index(i)]=str(s)
            print string.join(text_s)
decode(text)
decode(text)
```

```
>>>
come as you are, as you 1005 as i want you to be
come as you 206 as you were, as i want you to be
>>>
```

14. Написати функцію, яка обробляє список слів (з дублюванням слів) і повертає список слів (без дублювання) відсортований в порядку спадання їх частоти.

```
>>> import nltk
>>> from nltk import *
>>> def sortuvannia(text):
        text=nltk.FreqDist(text.lower().split())
        return text.keys()

>>> s='The true story of Whitey Bulger, the brother of a state senator and the m
ost infamous violent criminal in the history of South Boston,who became an FBI i
nformant to take down a Mafia family invading his turf'
>>> sortuvannia(s)
['the', 'of', 'a', 'an', 'and', 'became', 'boston,who', 'brother', 'bulger,', 'c
riminal', 'down', 'family', 'fbi', 'his', 'history', 'in', 'infamous', 'informan
t', 'invading', 'mafia', 'most', 'senator', 'south', 'state', 'story', 'take', '
to', 'true', 'turf', 'violent', 'whitey']
>>>
```

15. Написати функцію, яка приймає текст і словник, як аргументи і повертає набір слів, які є у тексті але відсутні у словнику. Аргументи повинні бути

представлені, як списки стрічок. Чи може функція мати один рядок при використанні set.difference()?

```
>>> s=['The', 'true', 'story', 'of', 'Whitey', 'Bulger',',', 'the', 'brother', '
of', 'a', 'state', 'senator', 'and', 'the', 'most', 'infamous', 'violent', 'crim
inal', 'in', 'the', 'history', 'of', 'South', 'Boston', ',', 'who', 'became', 'a
n', 'FBI', 'informant', 'to', 'take', 'down', 'a', 'Mafia', 'family', 'invading'
, 'his', 'turf']
>>> dict=['story','criminal','senator','family','true','history','state']
>>> def diff(text,dic):
        return set(text).difference(dic)

>>> diff(s,dict)
set(['and', 'South', 'invading', 'an', 'down', 'violent', 'in', 'informant', ','
, 'to', 'take', 'his', 'who', 'most', 'Bulger', 'The', 'the', 'infamous', 'a', '
Whitey', 'of', 'brother', 'became', 'Boston', 'turf', 'FBI', 'Mafia'])
>>>
```

17. В NLTK реалізовано алгоритм Левінштейна для порівняння стрічок. Спробуйте скористатись цим модулем nltk.edit_dist(). Яким чином в цьому модулі використовується динамічне програмування? Який підхід використовується знизу-вверх чи зверху-вниз? Пояснити письмово.

```
import nltk
print help(nltk.edit_distance)
word1 = 'go'
word2 = 'run'
print nltk.edit_distance(word1,word2)
print nltk.edit_distance('must','have')
```

```
>>>
Help on function edit_distance in module nltk.metrics.distance:

edit_distance(s1, s2)
    Calculate the Levenshtein edit-distance between two strings.
    The edit distance is the number of characters that need to be
    substituted, inserted, or deleted, to transform s1 into s2.  For
    example, transforming "rain" to "shine" requires three steps,
    consisting of two substitutions and one insertion:
    "rain" -> "sain" -> "shin" -> "shine".  These operations could have
    been done in other orders, but at least three steps are needed.

    :param s1, s2: The strings to be analysed
    :type s1: str
    :type s2: str
    :rtype int

None
3
4
```

Динамічне програмування відбувається в декілька етапів. Алгоритм визначає довжину стрічок і відстань між ними. Використовується підхід знизу-вверх

Висновок: на цій лабораторній роботі я у своїх програмах використала основи структурного програмування мовою Python, повторила та закріпила знання отримані при виконанні попередніх лабораторних робіт, покращила загальні навички у програмуванні.