МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп’ютерних наук

та інформаційних технологій

Звіт

до лабораторної роботи №7

з дисципліни «Комп’ютерна лінгвістика»

«***ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.***

***СТРУКТУРНЕ ПРОГРАМУВАННЯ МОВОЮ PYTHON (частина1)*»**

Виконала:

ст.гр. ПРЛм-12

Щур Н. М.

Перевірив:

Старший викладач кафедри САПР

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.

**Короткі теоретичні відомості**

Вивчити основи структурного програмування мовою Python, повторити та закріпити матеріал, пройдений у попередніх лабораторних роботах, покращити загальні навички програмування.

**Присвоювання.**

Присвоювання – найпростіше поняття програмування, але навіть і йому властиві певні тонкощі.

За допомогою операції присвоєння відбувається копіювання значення виразу іншій змінній, хоча насправді, значення структурованого об’єкту, наприклад такого, як список, це є посилання на цей об’єкт. В наступному прикладі показано, як присвоювання нового значення елементу списку не приводить до зміни його інших елементів:

**Порівняння**

Python підтримує два способи порівняння. Оператор is перевіряє об’єкти на ідентичність. Створивши список з декількох копій одного і того самого об’єкту не складно переконатися, що елементи цього списку не тільки ідентичні, згідно ==, але і є одним і тим самим об’єктом.

Використовуючи функцію id() можна легко зрозуміти відмінності між об’єктами і побачити що елементи списку не є ідентичними - списки мають різні ідентифікатори.

### Умовні твердження(висловлювання)

В частині умов if твердження, не пусті стрічки вважаються «true», а пусті стрічки чи списки вважаються «false» і не обробляються.

Тому, не потрібно використовувати додаткову перевірку if len(element) > 0 в умові.

Для пояснення відмінностей між використанням if...elif та використання декількох if тверджень.

### Операції над послідовностями різних типів

Елементи послідовності s можна обробляти почергово (здійснювати над ними ітерації, ітерувати) різними способами, основні з яких, наведено у таблиці 1.

Способи ітерування елементів послідовностей Таблиця 1:

| Вираз Python | Пояснення |
| --- | --- |
| for item in s | Проітерувати елементи s |
| for item in sorted(s) | Проітерувати впорядковані елементи s |
| for item in set(s) | Проітерувати унікальні елементи s |
| for item in reversed(s) | Проітерувати зворотньо впорядковані елементи s |
| for item in set(s).difference(t) | Проітерувати елементи s, які не входять в t |
| for item in random.shuffle(s) | Проітерувати випадково впорядковані елементи s |

Функції з Таблиці 1. можна поєднувати у різний спосіб. Наприклад, для одержання унікальних елементів s впорядкованих у зворотному порядку необхідно використовувати наступний вираз reversed(sorted(set(s))).

Послідовності різних типів можна перетворювати між собою. Наприклад, tuple(s) – перетворення послідовності будь-якого типу в кортеж, list(s) – перетворення послідовності будь-якого типу в список. Для перетворення списку стрічок в єдину стрічку потрібно використовувати функцію join(), наприклад, ':'.join(words).

**Стиль програмування**

Програмування це більше мистецтво ніж наука. Одна з найгрунтовніших книг по програмуванню, написана Дональдом Кнутом (Donald Knuth) так і називається The Art of Computer Programming. Приклади програм у більшості підручників написані таким чином, щоб не тільки комп’ютери але і люди змогли прочитати і зрозуміти текст програми. При програмуванні важливе значення має стиль програмування, який впливає на розуміння тексту програми і передбачає такі складові, як розміщення тексту програми, вибір процедурного чи декларативного стилю, використання змінних в циклах та багато інших.

**Тексти програм на мові PYTHON**

**Варіант 11**

**Завдання 1, 2, 3, 4, 5, 6, 7, 8, 9, 10**

**Завдання 1.** Знайти в Python's help додаткову інформацію про послідовності. В інтерпретаторі, набрати по черзі help(str), help(list), та help(tuple). На екрані буде відображено повний список функцій властивих кожному з типів. Деякі функції мають спеціальні імена з подвійними підкреслюваннями. Кожній такій функції відповідає і інший запис показаний в документації. Наприклад x.\_\_getitem\_\_(y) відповідає x[y].

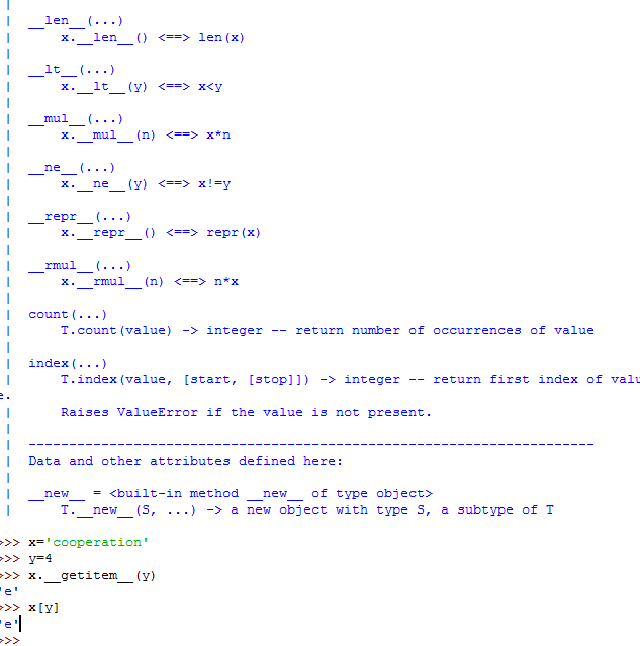


Рис. 1. Завдання 1

**Завдання 2.** Знайти три операції, які можна здійснювати і зі списками та із кортежами. Знайти три операції, які не можна здійснювати над кортежами. Знайдіть коли використання списку замість кортежу приводить до Python помилки.

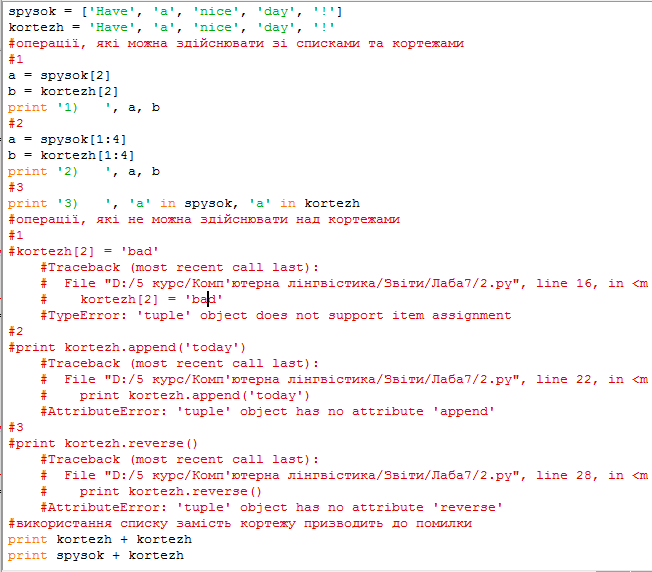


Рис. 2. Завдання 2

**Завдання 3.** Яким чином можна створити кортеж з одного елемента. Продемонструвати два різні способи.

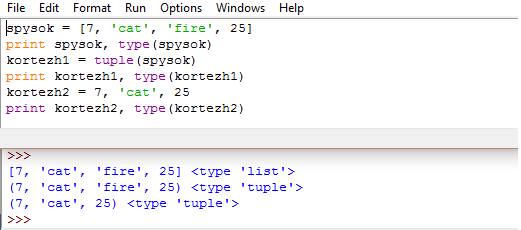


Рис.3. Завдання 3

**Завдання 4.** Створити список words = ['is', 'NLP', 'fun', '?']. Використовуючи операції присвоювання подібні до words[1] = words[2] та тимчасову змінну tmp перетворити цей список в список ['NLP', 'is', 'fun', '!']. Здійснити аналогічні перетворення використовуючи присвоювання в кортежах.

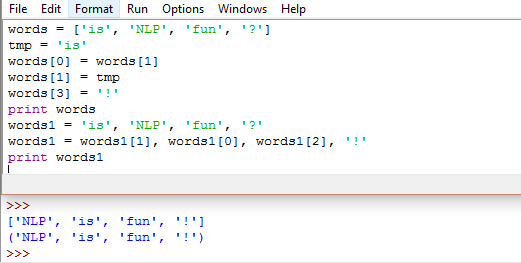


Рис.4. Завдання 4

**Завдання 5.** Прочитати про вбудовану функцію здійснення порівнянь cmp, набравши help(cmp). Продемонструвати чим поведінка цієї функції відрізняється від поведінки операторів порівняння.

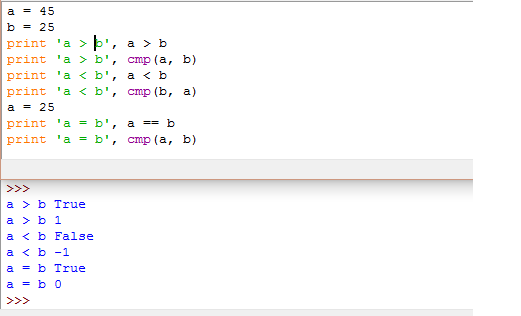


Рис.5. Завдання 5

**Завдання 6.** Написати програму для коректного виділення в тексті n-грамів з врахуванням граничних випадків: n = 1, та n = len(sent)?

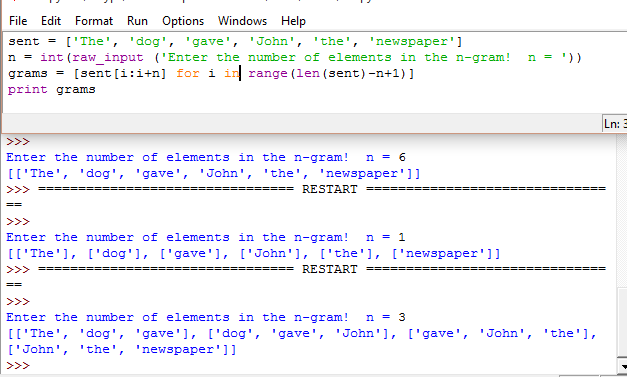


Рис.6. Завдання 6

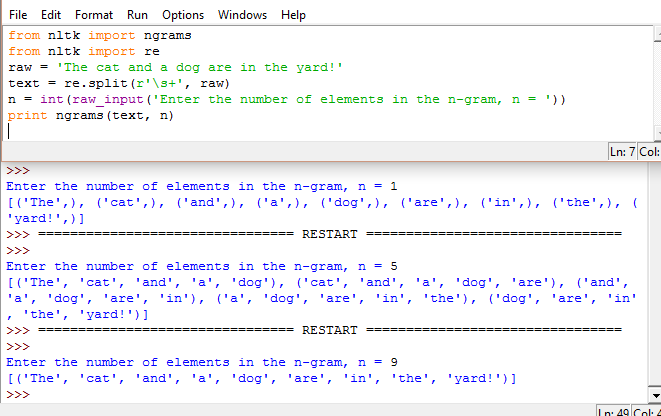


Рис.7. Завдання 6

**Завдання 7.** Використати оператори нерівності для порівняння стрічок, наприклад. 'Monty' < 'Python'. Що станеться, якщо виконати 'Z' < 'a'? Порівняти стрічки,як мають однаковий префікс, наприклад 'Monty' < 'Montague'. Спробувати порівняти структуровані об’єкти, наприклад ('Monty', 1) < ('Monty', 2). Чи отримали очікувані результати?

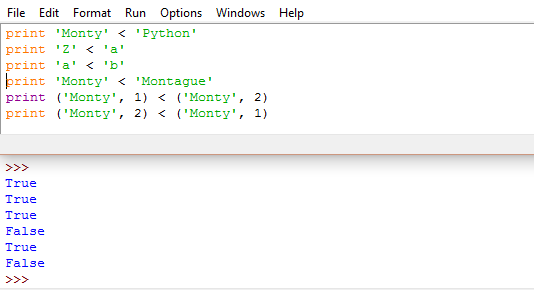


Рис.8. Завдання 7

**Завдання 8.** Написати програму видалення пробілів на початку і в кінці стрічки та для видалення зайвих пробілів між словами. Використовувати split() та join(). Оформити у вигляді функції. Функція повинна містити повну стрічку документування.

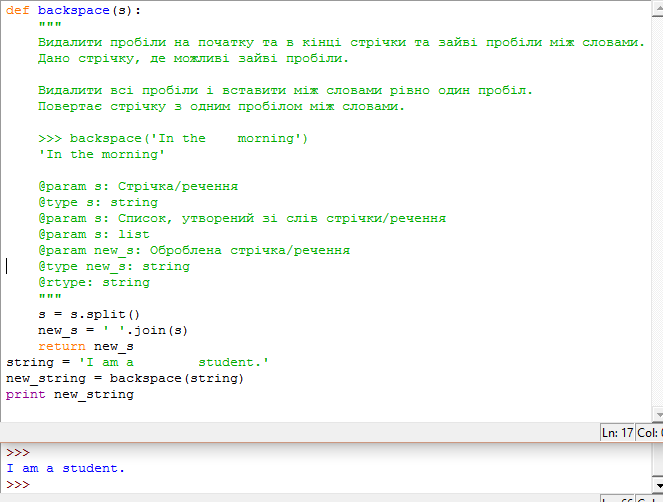


Рис.9. Завдання 8

**Завдання 9.** Написати програму видалення пробілів на початку і в кінці стрічки та для видалення зайвих пробілів між словами. Використовувати re.sub(). Оформити у вигляді функції. Функція повинна містити повну стрічку

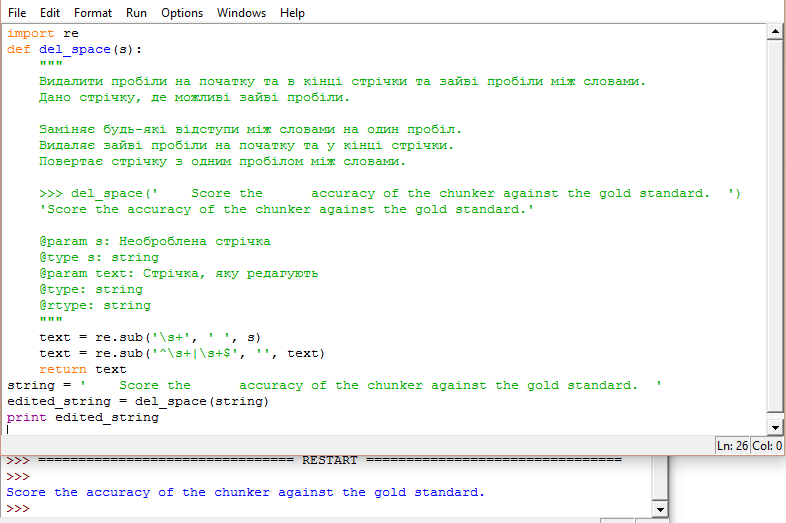


Рис.10. Завдання 9

**Завдання 10.** Написати програму сортування слів за їх довжиною. Визначити допоміжну функцію cmp\_len, яка буде використовувати функцію cmp для порівняння довжин слів. Функція повинна містити повну стрічку документування.

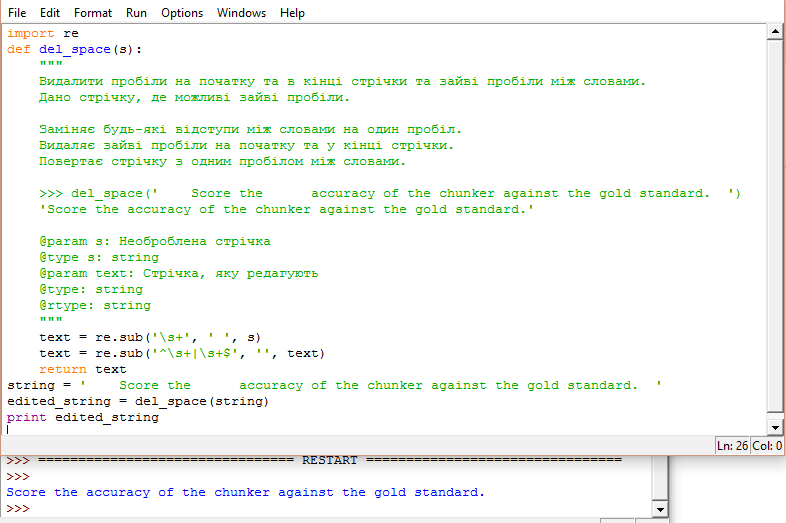
****

Рис.11. Завдання 10

**Висновок:** під час роботи я вивчила основи програмування на мові *Python* та основи структурного програмування на цій мові. Також повторила та закріпила отриманні вже знання. Покращила загальні навички у програмуванні.