МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

### *Кафедра “Системи автоматизованого проектування”*



Звіт

до лабораторної роботи №5

на тему: ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮ­ВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.ПОЧАТКОВА ОБРОБКА ТЕКСТІВ ПРИРОДНЬОЮ МОВОЮ.

ОСНОВИ ПРОГРАМУВАННЯ НА МОВІ PYTHON(частина 2) ”

з дисципліни “Комп’ютерна лінгвістика”

Виконала:

студентка групи ПРЛм-11

Свіляк О.Ю.

Прийняв:

викладач

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 сегментизація.

**Теоретичні відомості**

Юнікод – це промисловий стандарт, розроблений для того, щоб зробити можливим для текстів і символів (графічних знаків) усіх 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.

У межах програми обробка стрічок Unicode відбувається аналогічно до звичайних стрічок. Однак, коли Unicode-символи зберігаються у файл або виводяться на екран, вони мають бути закодовані, як потік байтів. Деякі кодування (такі як ASCII та Latin-2) використовують 1 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.

Текст у файлах є в певному кодуванні, і потрібен певний механізм для перетворення його в Unicode. Такий механізм називають декодування. Навпаки записати Unicode-символи у файл або вивести на екран можна тільки попередньо перетворивши їх у потрібне кодування. Таке перетворення називають кодуванням.

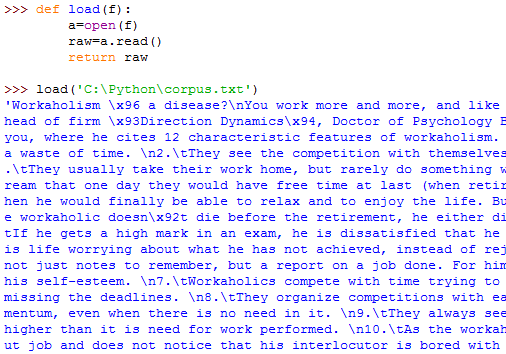
**Тексти програм на мові Python**

**Варіант 15**

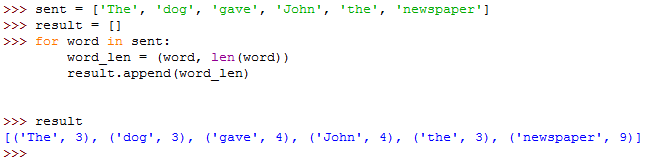
**5.1.** Напишіть функцію, яка приймає адресу URL як аргумент і повертає те, що міститься за цією адресою з видаленням HTML-розмітки. Використовувати *urllib.urlopen* для доступу до контенту наступним чином: *raw\_contents = urllib.urlopen('http://www.nltk.org/').read().*

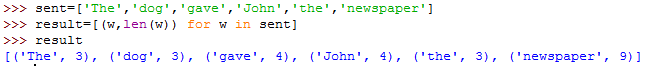


**5.2.** Збережіть деякий текст у файлі *corpus.txt*. Визначити функцію *load(f)* для читання файлу, назва якого є її аргументом і повертає стрічку, яка містить текст з файлу.



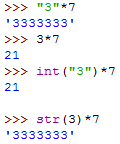
**5.3.** Перепишіть наступний цикл як *list comprehension*:

****

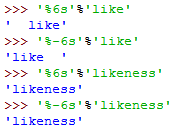


**5.4.** Перевірити різницю між стрічками і цілим виконавши наступні дії: *"3" \* 7 та 3 \* 7*. Спробуйте здійснити конвертування між стрічками і цілими використавши *int("3")* та *str(3).*

Змінна "3" є стрічкового характеру, тому помноживши її на 7, отримуємо у 7 разів довшу стрічку. Відповідно функція *int("3")* перетворює стрічкову змінну в цілочисельну, а функція *str(3)* перетворює цілочисельну змінну в стрічкову.

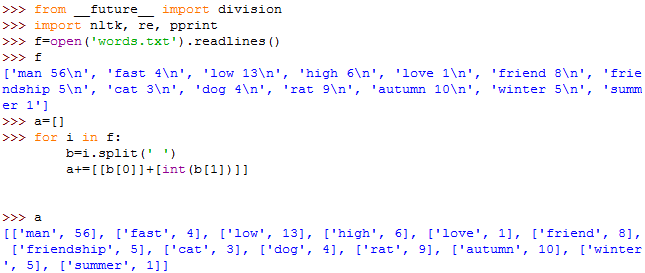


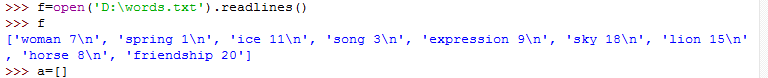
**5.5.** Що станеться, коли стрічки форматування *%6s* та *%-6s* використовуються для відображення стрічки, довшої ніж 6 символів?

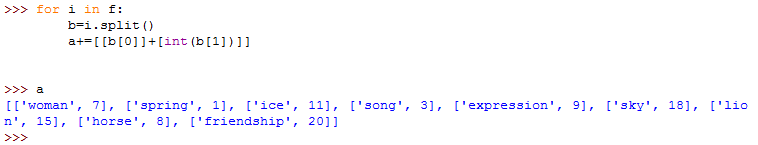


Стрічка *%6s* для стрічок довжиною до шести символів виконує форматування по лівому краю (тобто додає пробіли), а стрічка *%-6s* – форматування по правому краю. Коли ж стрічка довша за 6 символів, форматування не відбувається.

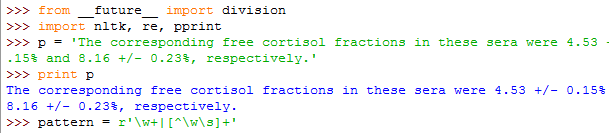
**5.7.** Створіть файл, який буде містити слова та їх частоту записані в окремих рядках через пробіл (fuzzy 53). Прочитайте цей файл використовуючи open(filename).readlines(). Розділіть кожну стрічку на дві частини використовуючи split(), і перетворіть число в ціле значення використовуючи int(). Результат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

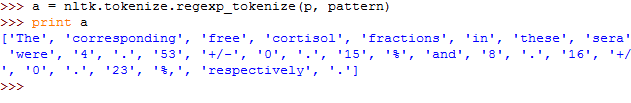






**5.11.** Здійсніть аналіз числового виразу в наступному реченні з корпуса MedLine: The corresponding free cortisol fractions in these sera were 4.53 +/- 0.15% and 8.16 +/- 0.23%, respectively. Чи можна сказати, що 4.53 +/- 0.15% це три окремих слова? Чи це одне складне слово? Чи це дев’ять слів "four 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?

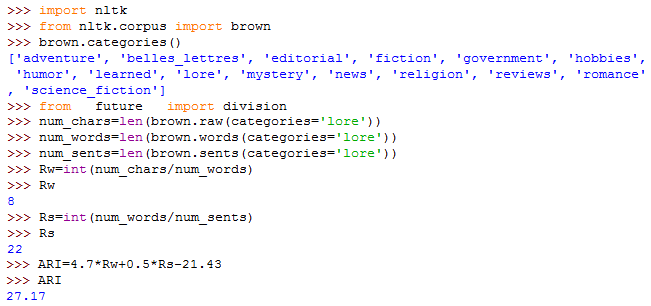


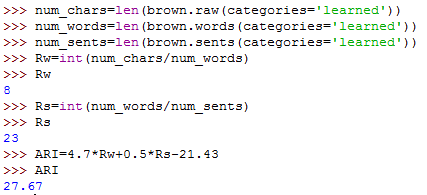


Програма розглядає кожен символ та число окремо. Це не правильно, тому що розривати

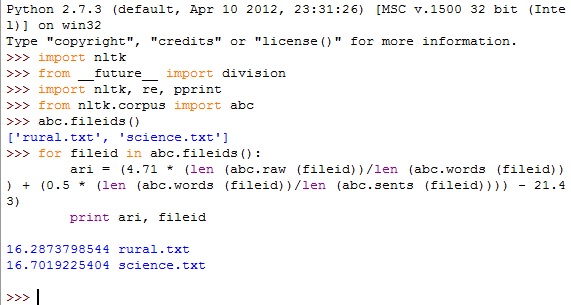
числа 4.53 та 0.15 не можна. Також не можна відділяти % від числа.

**5.12.** Міра оцінки читабельності використовується для оцінки складності тексту для читання. Нехай, *μw* – середня кількість літер у слові, та *μs* – середнє значення кількості слів у реченні в певному тексті. *Automated Readability Index* (ARI) тексту визначається згідно виразу: *4.71 μw + 0.5 μs - 21.43.* Визначити значення ARI для різних частин корпуса *Brown Corpus*, включаючи частину *f (popular lore)* та *j (learned).* Використовуйте *nltk.corpus.brown.words()* для знаходження послідовності слів та *nltk.corpus.brown.sents()* для знаходження послідовності речень.





**5.14.** Доступіться до текстів ABC Rural News та ABC Science News з корпуса (nltk.corpus.abc). Знайдіть значення для оцінки читабельності текстів (аналогічно до задачі №12). Використовуйте Punkt для поділу тексту на окремі речення.

****

**Висновок:** на цій лабораторній роботі я вивчила методи роботи з файлами на локальних дисках та з Інтернету, а також методи використання Юнікоду під час обробки текстів. Також я ознайомився з поняттями нормалізації текстів, стемінгу, лематизації та сегментації і закріпив свої знання та навички на практиці, виконавши найпоширеніші задачі.