



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Лабораторна робота №1

Мультипарадигмненне програмування

«Імперативне програмування»

Виконав

студент групи ІТ-03:

Карванський Н. Р.

Перевірили:

Глушко Б. С.

Очеретяний О. К.

Київ 2021

Завдання:

1. Для текстового файлу ми хочемо відобразити N (наприклад, 25) найчастіших слів і відповідну частоту їх повторення, упорядковано за зменшенням. Слід обов'язково нормалізувати використання великих літер і ігнорувати стоп-слова, як «the», «for» тощо. Щоб все було просто, ми не піклуємося про порядок слів з однаковою частотою повторень. Ця обчислювальна задача відома як term frequency.
2. Для текстового файлу виведіть усі слова в алфавітному порядку разом із номерами сторінок, на яких Ці слова знаходяться. Ігноруйте всі слова, які зустрічаються більше 100 разів. Припустимо, що сторінка являє собою послідовність із 45 рядків.

Обмеження:

- Заборонено використовувати функції
- Заборонено використовувати цикли
- Для виконання потрібно взяти мову, що підтримує конструкцію GOTO

1. Вихідний код

[GitHub](#)

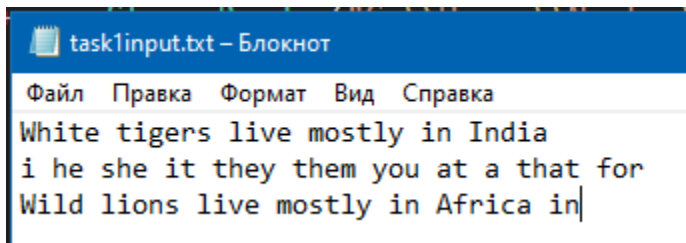
2. Використані технології

Для виконання поставлених задач була обрана мова програмування C#. Спочатку планувалось зробити все на Java, але не зважаючи на наявність у ній конструкції GOTO, вона є лише зарезервованим терміном і не може використовуватись. Також були спроби написати на Python, але виникли проблеми з IDE. Оскільки C# деякими своїми аспектами нагадує Java, завдання було виконано на ній.

3.1. Опис алгоритму роботи 1 завдання

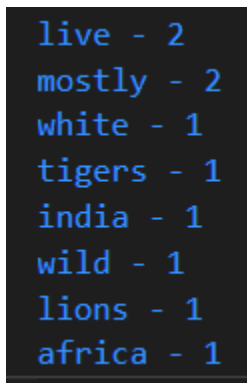
1. Записати вміст усього файлу в одну стрічку, вилучивши непотрібні символи на перетворивши усі букви в малі та додавши змінну, яка позначить кінець файлу.
2. Додати слова зі стрічки у масив, відфільтрувавши стоп-слова.
3. Здійснити підрахунок слів та записати у ще один масив, координата слова в масиві стрічок-слів відповідає координаті її повторювань у новоствореному масиві.
4. Відсортувати масив у порядку спадання кількості повторювань.
5. Вивести результат.

Вхідні дані:



```
task1input.txt - Блокнот
Файл  Правка  Формат  Вид  Справка
White tigers live mostly in India
i he she it they them you at a that for
Wild lions live mostly in Africa in
```

Приклад виконання:



```
live - 2
mostly - 2
white - 1
tigers - 1
india - 1
wild - 1
lions - 1
africa - 1
```

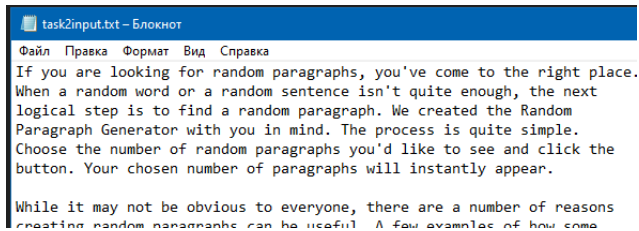
3.2. Опис алгоритму роботи 2 завдання

1. Посимвольно прочитати текст, прибравши зайві символи, звівши усі букви в малі й пропускаючи стоп-слова. Порахувати кількість повторювань й записати у масив-словник, який містить дані про слова

(саме слово, кількість повторювань, масив сторінок, на який слово зустрічається).

2. Відсортувати утворений масив відповідно до алфавітного порядку букв у словах.
3. Вивести результат.

Вхідні дані:



```
task2input.txt - Блокнот
Файл  Правка  Формат  Вид  Справка
If you are looking for random paragraphs, you've come to the right place.
When a random word or a random sentence isn't quite enough, the next
logical step is to find a random paragraph. We created the Random
Paragraph Generator with you in mind. The process is quite simple.
Choose the number of random paragraphs you'd like to see and click the
button. Your chosen number of paragraphs will instantly appear.

While it may not be obvious to everyone, there are a number of reasons
creating random paragraphs can be useful. A few examples of how some
```

Приклад роботи:

```
can: 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2
causing: 1
challenge: 1, 1
challenges: 1, 1
changed: 2
choose: 1
chosen: 1
click: 1
come: 1
common: 1
complete: 1, 1
completely: 1, 2
connection: 1
create: 1, 1, 2
created: 1, 2, 2

original: 1
paragraph: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2
paragraphs: 1, 1, 1, 1, 1, 1, 2
people: 1
picker: 2
place: 1, 1
please: 2, 2
process: 1
productive: 1
programmer: 2

about: 1
above: 2
added: 2
already: 1
also: 1, 1
```

У вихідному коді містяться певні коментарі щодо коду, для пояснень, які не були зазначені у звіті.

Висновок

Під час виконання даної лабораторної роботи я навчився реалізовувати рішення певних задач без використання циклів та сучасних методів програмування. Натомість, були використані такі конструкції як GOTO та мітки, які були єдиним рішенням для перших програмістів.