

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

Звіт

з лабораторної роботи № 1 з дисципліни
«Мультипарадигменне програмування»

«Імперативне програмування»

Виконав: ПП-02, Зусько В. Ю.

Київ 2021

Завдання 1

Текст:

Обчислювальна задача тут тривіальна: 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.

Опис роботи:

Перш за все встановлюються початкові параметри для роботи програми. Наступним кроком зчитується текст з файлу. Зчитаний текст розбивається на масив слів, зважаючи на пробіли та переноси рядка. Наступним етапом є нормалізація: за допомогою двох циклів ми зводимо усі слова до нижнього регістру, враховуючи їхній номер в таблиці ASCII, та ігноруємо усі слова, довжина яких менша за задану. Далі ми рахуємо кількість повторів кожного слова, використовуючи структуру, що містить кількість повторів слова та саме слово. Це реалізовано проходження по масиву слів двічі: якщо слово нове — додаємо до масиву частот, якщо частота цього слова вже була записана — збільшуємо її на одиницю. Настала черга бульбашкового сортування та виводу результатів аналізу.

Приклад роботи:

The image shows a screenshot of the Microsoft Visual Studio Debug Console. The console has a dark background with white text. At the top, the title bar reads "Microsoft Visual Studio Debug Console". Below the title bar, the input text is "White tigers live mostly in India" and "Wild lions live mostly in Africa". The output shows the frequency of words: "live - 2", "mostly - 2", "white - 1", "tigers - 1", "india - 1", "wild - 1", "lions - 1", and "africa - 1".

```
Microsoft Visual Studio Debug Console
White tigers live mostly in India
Wild lions live mostly in Africa

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

Завдання 2

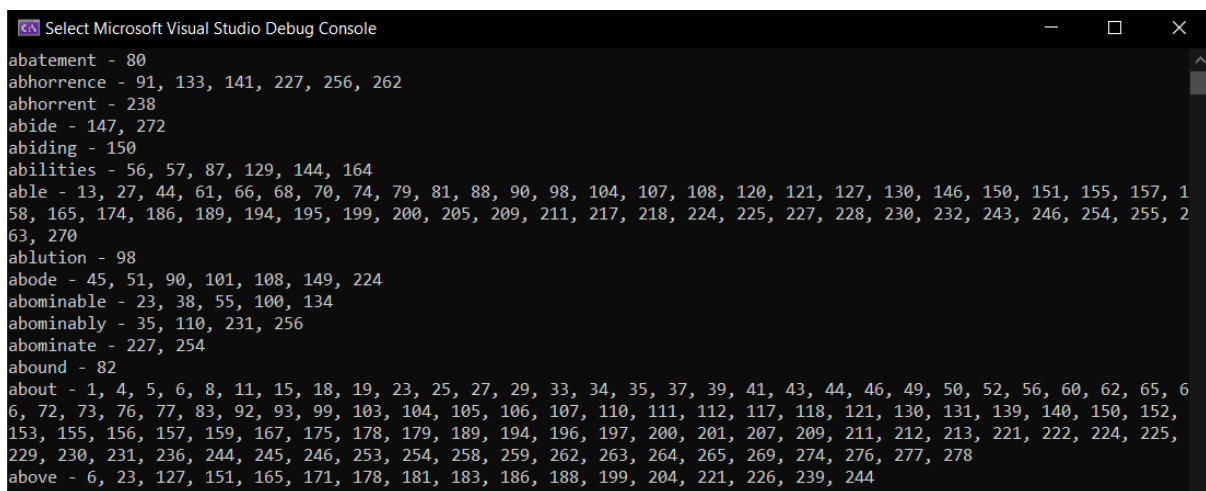
Текст:

Тепер, нам потрібно виконати задачу, що називається словниковим індексуванням. 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.

Опис роботи:

По-перше, встановлюємо налаштування програми. Зчитуємо текст з файлу. Далі зчитаний текст розбиваємо на слова. На цьому етапі ми зводимо слова до нижнього регістру та ігноруємо слова, що не пройшли фільтрацію. Наступним етапом йде індексація слів: ми проходимося по кожному слову та шукаємо його серед інших, записуючи у відповідну структуру номери сторінок (рахуються за допомогою формули). Таким чином, двома циклами, ми виконуємо завдання словникового індексування. Тепер сортуємо список індексів у алфавітному порядку бульбашковим сортуванням використовуючи порівняння рядків. Виводимо результати, попередньо об'єднавши масив сторінок, на яких трапляється слово, у рядок циклом.

Приклад роботи:



```
Select Microsoft Visual Studio Debug Console
abatement - 80
abhorrence - 91, 133, 141, 227, 256, 262
abhorrent - 238
abide - 147, 272
abiding - 150
abilities - 56, 57, 87, 129, 144, 164
able - 13, 27, 44, 61, 66, 68, 70, 74, 79, 81, 88, 90, 98, 104, 107, 108, 120, 121, 127, 130, 146, 150, 151, 155, 157, 158, 165, 174, 186, 189, 194, 195, 199, 200, 205, 209, 211, 217, 218, 224, 225, 227, 228, 230, 232, 243, 246, 254, 255, 263, 270
ablution - 98
abode - 45, 51, 90, 101, 108, 149, 224
abominable - 23, 38, 55, 100, 134
abominably - 35, 110, 231, 256
abominate - 227, 254
abound - 82
about - 1, 4, 5, 6, 8, 11, 15, 18, 19, 23, 25, 27, 29, 33, 34, 35, 37, 39, 41, 43, 44, 46, 49, 50, 52, 56, 60, 62, 65, 66, 72, 73, 76, 77, 83, 92, 93, 99, 103, 104, 105, 106, 107, 110, 111, 112, 117, 118, 121, 130, 131, 139, 140, 150, 152, 153, 155, 156, 157, 159, 167, 175, 178, 179, 189, 194, 196, 197, 200, 201, 207, 209, 211, 212, 213, 221, 222, 224, 225, 229, 230, 231, 236, 244, 245, 246, 253, 254, 258, 259, 262, 263, 264, 265, 269, 274, 276, 277, 278
above - 6, 23, 127, 151, 165, 171, 178, 181, 183, 186, 188, 199, 204, 221, 226, 239, 244
```

Висновок

Виконавши цю лабораторну роботу ми познайомились з імперативною моделлю програмування та виконали дві задачі, що пов'язані з аналізом тексту, використовуючи цю парадигму. Програми успішно виконували поставленні завдання. Також, ми краще зрозуміли переваги утиліт, що надають сучасні мови програмування та недоліки імперативної парадигми. Завдяки роботі ми оглянули спосіб написання програм, який використовували покоління програмістів та впевнилися в складності цієї праці.