Міністерство освіти і науки України
Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Мультипарадигмене програмування

ЗВІТ

до лабораторних робіт

Виконав

студент ІП-01 Берлінський Я.В. _____

(№ групи, прізвище, ім’я, по батькові) _____

Прийняв ас. Очеретяний О. К. _____

(посада, прізвище, ім’я, по батькові) _____

Київ 2022

1. Завдання лабораторної роботи

Практична робота складається із трьох завдань, які самі по собі є досить простими. Але, оскільки задача - зрозуміти, як писали код наші славні пращури у 1950-х, ми введемо кілька обмежень:

Заборонено використовувати функції

Заборонено використовувати цикли

Для виконання потрібно взяти мову, що підтримує конструкцію GOTO

Завдання 1:

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.

Ось такий вигляд матимуть ввід і відповідно вивід результату програми:

Input:

White tigers live mostly in India

Wild lions live mostly in Africa

Output:

live - 2

mostly - 2

africa - 1

india - 1

lions - 1

tigers - 1

white - 1

wild – 1

Завдання 2:

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. Припустимо, що сторінка являє собою послідовність із 45 рядків. Наприклад, якщо взяти книгу Pride and Prejudice, перші кілька записів індексу будуть:

abatement - 89

abhorrence - 101, 145, 152, 241, 274, 281

abhorrent - 253

abide - 158, 292

2. Опис використаних технологій

Для виконання цього завдання було обрано мову C++, адже вона підтримує конструкцію goto.

3. Опис алгоритму

Завдання 1:

1. Відкрити файл.
2. Цикл зчитування рядків. Якщо рядок не зчитано, то вихід з циклу:
    1. Цикл проходу по всім символам слова:
        1. Якщо символ у верхньому регістрі – понизити.
        2. Цикл проходу по всім елементам масиву сепараторів:
            1. Якщо символ відповідає сепаратору – запам’ятати, вийти з циклу.
        3. Якщо символ є сепаратором, додати нове слово до масиву.
    2. Перевірити слово, чи є воно «стоп-словом» із попередньо визначеного масиву «стоп-слів».
3. Цикл проходу по словам:
    1. Якщо слово раніше не перевірялось, занести його до масиву унікальних слів.
    2. Цикл проходу по словам:
        1. Якщо слова відповідають одне одному, збільшити лічильник на 1.
    3. Занести лічильник у відповідну позицію цілочисельного масиву лічильників кожного слова.
4. Сортування масиву лічильників та відповідна перестановка слів у масиві унікальних слів.
5. Цикл проходу по унікальним словам:
    1. Вивід слова та його лічильника на відповідній позиції.

Завдання 2:

1. Відкрити файл.
2. Цикл зчитування рядків. Якщо рядок не зчитано, то вихід з циклу:
    1. Збільшити номер рядку, якщо номер рядка дорівнює кількості рядків на сторінці, збільшити номер сторінки, обнулити рядок.
    2. Цикл проходу по всім символам слова:
        1. Якщо символ у верхньому регістрі – понизити.
        2. Цикл проходу по всім елементам масиву сепараторів:
            1. Якщо символ відповідає сепаратору – запам’ятати, вийти з циклу.
        3. Якщо символ є сепаратором і слова немає у масиві унікальних слів, додати нове слово до масиву унікальних

слів, інакше, занести номер сторінки у цілочисельний масив сторінок на відповідну позицію слова у масиві слів.

3. Цикл проходу по словам:
    1. Якщо позиція maxOccurence у цілочисельному масиві номерів сторінок не дорівнює 1, видалити слово з масиву та підмасив з номерами сторінок з цілочисельного масиву номерів сторінок.
4. Сортування масиву слів та відповідна перестановка номерів сторінок.
5. Цикл проходу по унікальним словам:
    1. Вивід слова.
    2. Цикл проходу по підмасиву номерів сторінок відповідного слова:
        1. Вивід номеру сторінки.

4. **Опис програмного коду**

5. **Тестування програми**

Завдання №1

```
0) one - 4
1) six - 3
2) two - 2
3) three - 1
4) four - 1
5) five - 1
6) twelve - 1
Program ended with exit code: 0
```

Завдання №2

```
5436) stays - 25

5437) steadfast - 143

5438) steadfastly - 89, 161, 254

5439) steadily - 61, 175, 183

5440) steadiness - 178

5441) steady - 12, 49, 81, 84, 101, 105, 108, 111, 131, 158, 188, 237

5442) stem - 224

5443) step - 6, 17, 89, 100, 181, 196, 203, 212, 218, 219, 224, 225, 230, 250, 251

5444) stepped - 245

5445) stepping - 155

5446) steps - 44, 128, 152, 189, 221

5447) steward - 77, 191, 199, 281

5448) stiffly - 276

5449) stiffness - 12, 93

5450) stiles - 25
```