

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Кафедра інформатики та програмної
інженерії**



Звіт до лабораторної роботи №1

з курсу

«Мультипарадигмненне програмування»

студента 2 курсу

групи ІТ-01

Девіцький Іван Денисович

Викладач:

Очеретяний О.К.

Завдання

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
```

Опис алгоритму:

Спочатку посимвольно зчитуємо текст з файлу, що знаходиться за заданим шляхом. Проводимо перевірку на великі літери та нормалізуємо їх. Далі перевіряємо, чи слово не є списку ігнорованих слів. Якщо слово є новим, додаємо його до кінцевого списку із лічильником рівним 1, а якщо воно вже є у списку, збільшуємо лічильник на 1. При додаванні слова перевіряється умова, що масив не переповниться, і у випадку, коли довжина масиву є близькою до його початкового розміру, копіюємо його у новий масив із більшим розміром. Далі сортуємо кінцевий список сортуванням бульбашкою та виводимо результат у консоль.

Результат виконання програми:



```
Run:      
C:/Users/Ivan/RiderProjects/mp_lab1/mp_lab1/bin/Debug/net6.0/mp_lab1.exe  
live - 2  
mostly - 2  
white - 1  
tigers - 1  
india - 1  
wild - 1  
lions - 1  
africa - 1  
Process finished with exit code 0.
```

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

Опис алгоритму:

Спочатку построково зчитуємо текст з файлу, що знаходиться за заданим шляхом. Якщо кількість строк стає рівною, заданою нами кількістю для однієї сторінки, збільшуємо лічильник сторінок на 1. Проходимо кожну строку посимвольно за алгоритмом з першого завдання, також додаємо нові слова, або записуємо сторінку, якщо слово вже було знайдено хоча б один раз. Виконуємо сортування бульбашкою та виводимо результат у консоль:

Результат виконання програми:

```
Run: [play button] [stop button] [download icon] [trash icon]
C:/Users/Ivan/RiderProjects/mp_lab1/mp_lab1/bin/Debug/net6.0/mp_lab1.exe
able - 5
about - 0, 2, 3, 4, 6, 7, 8
above - 2
absent - 9
absolutely - 4, 7
abuse - 1
accept - 2, 9
accepted - 9
accidental - 4
accomplished - 3
accomplishments - 6
accosted - 7
account - 1
ached - 9
acknowledge - 5
acknowledged - 0, 4
acknowledgment - 9
acquaint - 9
acquaintance - 2, 4
acquainted - 2, 3, 4, 5
acquired - 4
act - 2, 5
acting - 5
activity - 9
```

Висновок:

У ході виконання лабораторної роботи було створено дві програми, що вирішують завдання задані в умові. Не було використано майже ніяких вбудованих функцій мови програмування, циклів, а також програма у більшій частині побудована на використанні конструкції `goto`. Ця лабораторна робота дала зрозуміти, які засоби для програмування були у наших предків.