



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:

Філянін Н.С.

Перевірили:

Глушко Б. С.
Очеретяний О. К.

Київ 2022

Мета:

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. 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.
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.

1. Вихідний код

Посилання на [GitHub](#).

2. Алгоритм завдання №1

1. Відбувається зчитування файлу у змінну «text».
2. Ітеруємось по занесеному в змінну тексту, ми замінюємо великі літери на маленькі, прибираємо зайві символи, проводимо заміну переходів на новий рядок на звичайні пробіли.
3. Проводимо фільтрацію від заборонених слів.
4. Рахуємо та записуємо слова, що пройшли перевірку, у масив, із вказанням відповідної кількості повторів кожного з них.
5. За допомогою сортування «Бульбашкою» (Bubble Sort), сортуємо масив за спаданням кількості повторів слів.
6. Виводимо результат у консоль. Вивід припиняється за умови досягнення заданого нами обмеження по кількості слів на вивід.

3. Алгоритм завдання №2

1. Відбувається зчитування файлу у змінну «text».
2. Ітеруємось по занесеному в змінну тексту, ми замінюємо великі літери на маленькі, прибираємо зайві символи, проводимо заміну переходів на новий рядок на звичайні пробіли.
3. Переходом на наступну сторінку вважається досягнення заданих умовних констант у вигляді 255 слів або 1800 символів на сторінку.
4. Рахуємо та записуємо слова у масив, із вказанням відповідної кількості повторів кожного з них та номером сторінок, на яких кожне з них зустрічалося.
5. Відбувається фільтрація, яка прибирає усі слова, що зустрічаються понад 100 разів. В реалізації помічено міткою “label.avoid_words”.
6. За допомогою сортування «Бульбашкою» (Bubble Sort), сортуємо масив за алфавітом.
7. Виводимо отриманий результат у консоль.

4. Приклади роботи:

Вивід завдання 1:

```
Most common words:
```

- 1) pleasure - 6
- 2) pain - 5
- 3) because - 4
- 4) you - 2
- 5) great - 2
- 6) avoids - 2
- 7) itself - 2
- 8) consequences - 2
- 9) obtain - 2
- 10) which - 2

```
Process finished with exit code 0
```

Вивід завдання 2:

```
aliquet - 40 time(s) - [1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 19, 20, 21, 24, 25, 27, 28, 30, 31, 32, 35, 36, 38, 39]
ante - 74 time(s) - [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 36, 37, 38, 39]
aptent - 12 time(s) - [3, 8, 9, 16, 23, 25, 26, 28, 29, 31, 39]
arcu - 48 time(s) - [1, 3, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 21, 22, 23, 24, 25, 26, 27, 29, 31, 32, 33, 35, 37, 38, 39]
auctor - 49 time(s) - [1, 3, 4, 6, 8, 11, 12, 13, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 37, 38, 39]
augue - 62 time(s) - [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 16, 17, 18, 19, 20, 21, 23, 24, 25, 27, 28, 29, 32, 35, 36, 37, 38, 39]
bibendum - 44 time(s) - [1, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 23, 24, 25, 26, 27, 28, 30, 32, 33, 35, 38, 39]
blandit - 45 time(s) - [1, 3, 6, 8, 9, 11, 15, 16, 18, 20, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 34, 35, 38]
class - 12 time(s) - [3, 8, 9, 16, 23, 25, 26, 28, 29, 31, 39]
commodo - 45 time(s) - [2, 5, 7, 8, 9, 10, 11, 13, 14, 15, 18, 19, 20, 22, 24, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 37, 38, 39]
condimentum - 42 time(s) - [1, 2, 4, 5, 6, 8, 10, 11, 13, 14, 15, 16, 17, 18, 21, 22, 23, 25, 28, 29, 30, 31, 32, 35, 36, 37, 38, 39]
congue - 42 time(s) - [2, 3, 5, 6, 8, 10, 12, 13, 16, 18, 19, 20, 23, 24, 25, 27, 29, 30, 32, 35, 36, 37, 38, 39]
consectetur - 44 time(s) - [1, 2, 3, 4, 5, 8, 9, 10, 12, 13, 14, 15, 16, 18, 19, 21, 22, 24, 26, 28, 29, 30, 35, 37, 38, 39]
consequat - 60 time(s) - [2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14, 15, 17, 18, 19, 20, 21, 22, 23, 24, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39]
conubia - 12 time(s) - [3, 8, 9, 10, 16, 23, 25, 26, 28, 29, 31, 39]
convallis - 42 time(s) - [1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 30, 32, 33, 34, 35, 36, 37, 38, 39]
cras - 32 time(s) - [1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 15, 18, 19, 21, 22, 23, 24, 25, 26, 27, 35, 37, 38, 39]
cubilia - 4 time(s) - [6, 11, 13, 23]
curabitur - 39 time(s) - [2, 3, 6, 7, 9, 11, 13, 17, 18, 19, 20, 21, 22, 26, 27, 29, 30, 31, 32, 36, 37]
curae - 4 time(s) - [6, 11, 13, 23]
cursus - 43 time(s) - [1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 15, 16, 19, 22, 23, 25, 27, 28, 30, 31, 32, 35, 36, 37, 39]
dapibus - 46 time(s) - [1, 2, 3, 4, 5, 7, 9, 11, 12, 15, 16, 17, 18, 19, 20, 22, 25, 27, 28, 30, 31, 32, 33, 34, 36, 39, 40]
diam - 72 time(s) - [1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]
dictum - 40 time(s) - [1, 2, 3, 5, 6, 7, 8, 11, 13, 14, 15, 16, 17, 18, 19, 21, 22, 29, 31, 32, 34, 36, 37, 38, 39]
```

Висновки

Під час виконання даної лабораторної роботи, я ознайомився з основами імперативного програмування та конструкцією GOTO.