ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ

за курсом «Основи Об’єктно-орієнтованого програмування»

студент(а/ки) групи ПС-20-2 <Калиновської Дарини Віталіївни>

кафедра комп’ютерних технологій,

ґ ДНУ 2021/2022

**1. Постановка задачі**

Номер в академгрупі — 7, в трійковій системі — 0021.

Вхідні дані - Параметри командного рядка:

1. Послідовність слів тексту

Якщо користувач не задав параметри командного рядка, програма входить в діалог з користувачем і питає про ім’я файлу, з якого треба прочитати вхідні дані.

II. Перша функція другого модуля упорядковує надану їй через список параметрів послідовність за критерієм:

2. Спочатку «гарні», потім «погані» (числа, слова, рядки матриці).

Критерій – це функція, передана також як параметр для функції упорядкування.

III. Друга функція другого модуля має для заданої послідовності (або матриці) обчислити і повернути:

0. Два результати: кількість елементів та середнє арифметичне (для тексту – довжин слів).

IV. Програма мусить видати результати у текстовий файл, ім’я якого вона отримує:

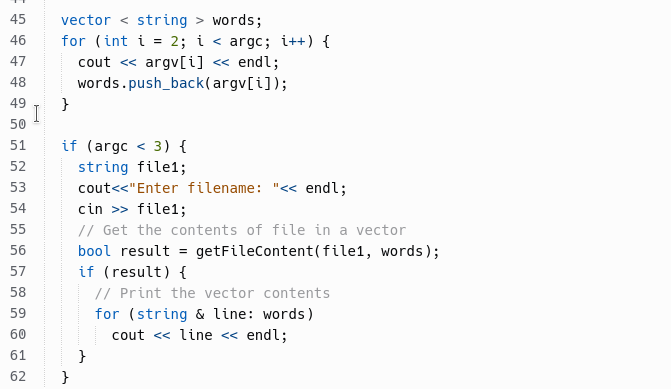
0. З командного рядка.

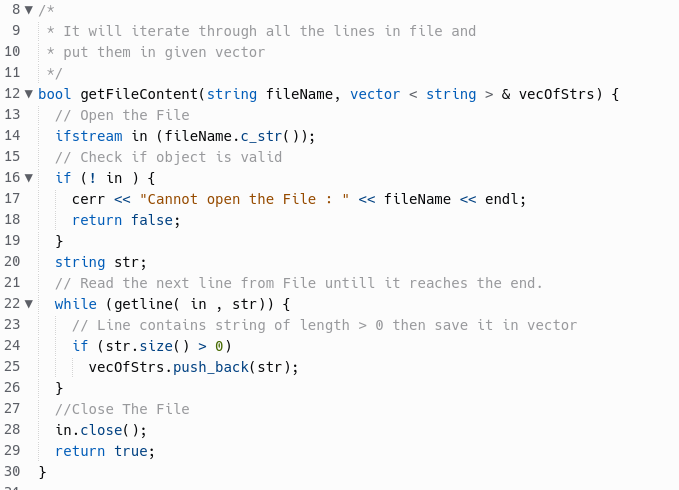
**2. Опис розв’язку**

Створюємо вектор з елементами типа стрінг, заповнюємо його словами, які ми отримаємо з параметрів командного рядка. Якщо користувач не задав параметри командного рядка, програма входить в діалог з користувачем і питає про ім’я файлу, з якого треба прочитати вхідні дані. За допомогою окремої функції зчитуємо дані та заповнюємо наш вектор. Створюємо окремий файл, в який переносимо першу і другу функції. В першій функції сортуємо вектор на «гарні» та «погані» значення(парна та непарна кількість символів). Створивши два нових вектора, упорядковуємо значення головного вектора по черзі перевіряя кількість симолів у словах та розподіляємо їх в перший чи другий вектора. Потім звільняємо головний вектор і записуємо туди нові значення по черзі з першого та другого векторів. Далі створюємо другу функцію, в яку передаємо вже упорядеований вектор та шукаємо кількість елементів та середнє арифметичне довжин слів. Для того щоб повернути два значення функції створюємо глобальний масив на два елемента та передаємо в нього результати. Викликаємо обидві функції в головному файлі, та записуємо результати їх обчислень в вихідний файл.Наявність імені вихідного файлу в командному рядку перевіряється на початку програми.

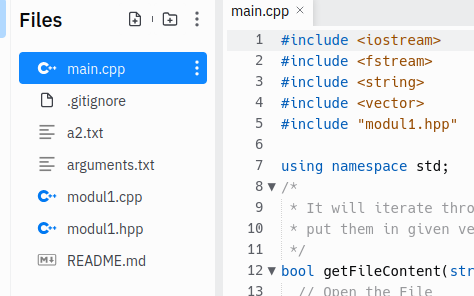
**3.Вихідний текст програми розв’язку задачі (основні фрагменти з коментарями) . . .**

Створюємо вектор з елементами типа стрінг, та, починаючи з третього елементу (перший це назва програми, другий імʼя вихідного файлу згідно з заданням IV(0)) заповнюємо вектор. Якщо елементів меньше трьох — питаємо у користувача імʼя файлу з якого треба прочитати вхідні дані і, за допомогою функції getFileContent, зчитуємо дані і заповнюємо ними вектор.



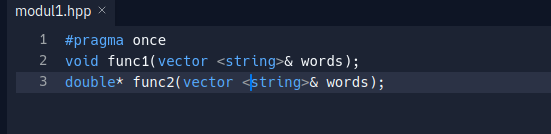


Далі створюємо окремий файл modul1.cpp та заголовочний файл modul1.hpp, і підключаємо їх у головний.

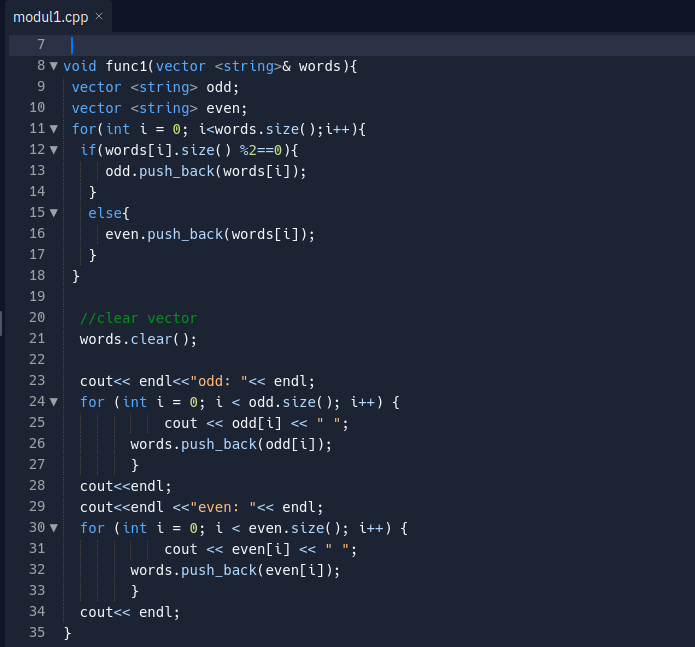
****

**3.**

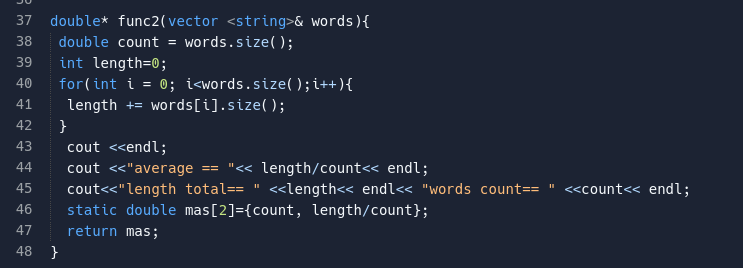
Прописуємо заголовкі функцій в modul1.hpp:



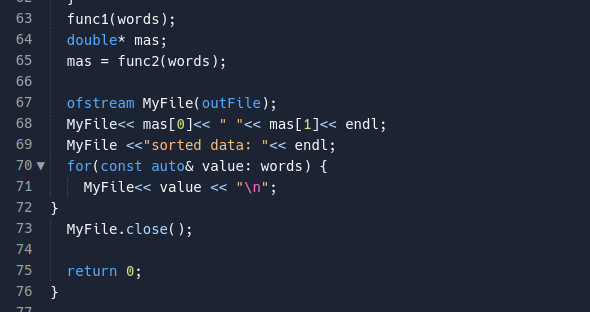
В файлі modul1.cpp створюємо функцію func1 яка упорядковує вектор words на гарні і погані (з парною або непарною кількістю елементів).

****

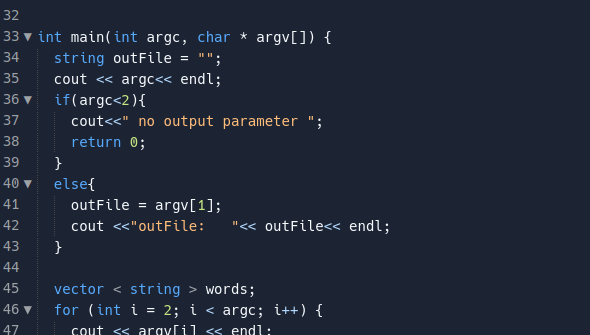
В цьому ж файлі створюємо другу функцію яка обчислює і, за допомогою масиву static double mas[2], повертає два результати: кількість елементів та середнє арифметичне довжин слів.

****

Далі в головній функції main викликаємо func1 i func2, та виводимо результати в створений файл, імʼя якого програма отримує з командного рядка.

****

Якщо в програму не передали жодного значення на початку, програма завершує свою роботу.

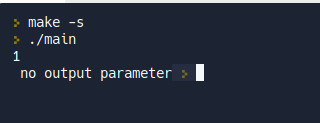
****

**4. Опис інтерфейсу програми (керівництво користувача)**

Програма отримує назву вихідного файлу та дані вектора з командного рядка. Перший параметр — імʼя вихідного файлу, обовʼязковий. Другий параметр і далі — вхідні дані, не обовʼязкові параметри. Якщо вхідні дані не були задані програма входить в діалог із користувачем та питає про ім’я файлу, з якого треба прочитати вхідні дані. Зчитавши дані, програма обчислює шукані значення і передає їх у вихідний файл. Якщо на початку імʼя вихідного файлу було незадано, програма завершує роботу.

**5. Опис тестових прикладів**

Тест 1, в командному рядку немає ніяких значень



Тест 2, в командному рядку:

імʼя вихідного файлу out.txt,

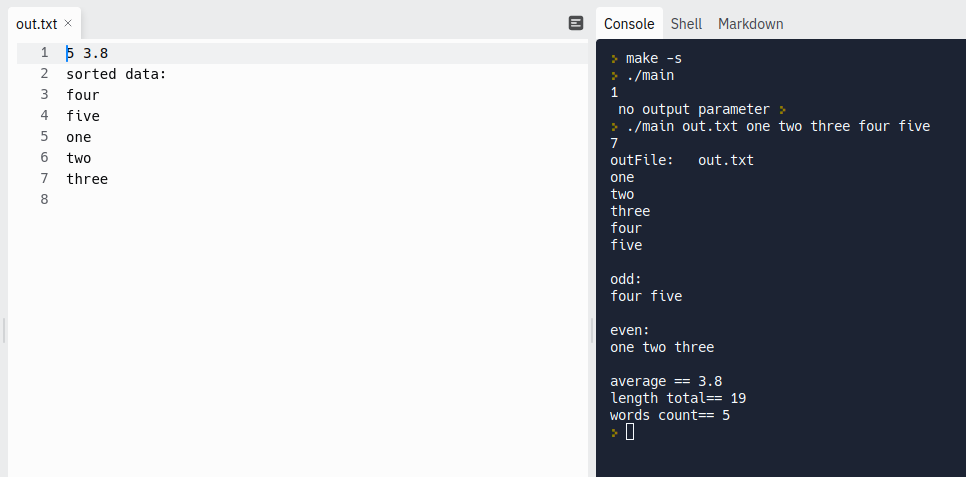
значення: one

two

three

four

five



Тест 3. в командному рядку:

імʼя вихідного фaйлу output.txt

значення : січень

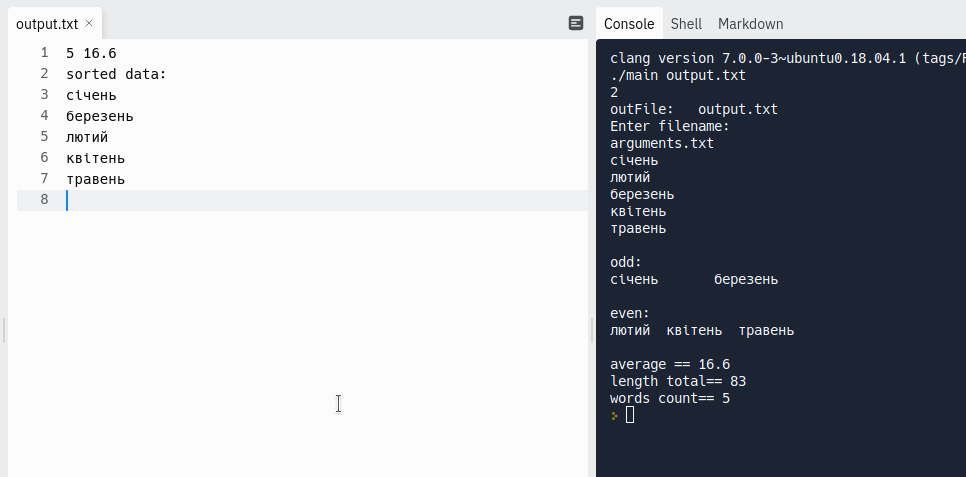
лютий

березень

квітень

травень

отримуємо з файла arguments.txt



**6. Аналіз помилок (опис усунення зауважень)**

Спочатку перша функція другого модулю не сортувала вектор на гарні і погані і не передавала значення упорядкованого вектора в вихідний файл, а просто шукала і виводила їх на екран. Також імʼя вихідного файлу питалось в діалозі з корисувачем, а не зчитувалось з командного рядка.