ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ

за курсом «Програмування»

студент(а) групи ПА-22-1

Бердик Роман Вікторович

кафедра комп’ютерних технологій, ДНУ

2023/2024 н.р.

1. Постановка задачі

Метою лабораторної роботи було ознайомлення з інфраструктурою розробки проектів, використання git та налаштуванням IDE, а також створення багатомодульної програми на мові C++. Завдання включало наступні етапи:

Створення проекту "Hello, git!" на мові C++ з прив'язкою до GitHub.

Налаштування середовища розробки та параметрів проекту.

Реалізація багатомодульної програми, яка виконує сортування та обчислення середніх значень для послідовності дійсних чисел.

Збереження результатів у текстовий файл.

Індивідуальне завдання передбачало роботу з послідовністю дійсних чисел, упорядкування її за неспаданням суми цифр дробної частини числа, обчислення середнього арифметичного та середнього гармонічного, а також виведення результатів у текстовий файл.

2. Опис розв’язку

Для реалізації поставленого завдання було створено багатомодульну програму на мові C++, що складається з трьох основних частин:

Головний модуль (main.cpp): відповідає за зчитування даних, виклик функцій сортування та обчислення середніх значень, а також за виведення результатів у текстовий файл.

Модуль для сортування (sorting.cpp, sorting.h): містить функцію для сортування послідовності дійсних чисел за сумою цифр дробної частини.

Модуль для обчислень (calculations.cpp, calculations.h): включає функції для обчислення середнього арифметичного та середнього гармонічного значень для послідовності чисел.

Програма отримує послідовність чисел або з командного рядка, або з файлу, виконує обробку даних і зберігає результати у файл results.txt.

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

main.cpp

#include <iostream>

#include <fstream>

#include <vector>

#include "sorting.h"

#include "calculations.h"

int main(int argc, char\* argv[]) {

std::vector<double> numbers;

if (argc > 1) {

std::ifstream inputFile(argv[1]);

double num;

while (inputFile >> num) {

numbers.push\_back(num);

}

} else {

std::cout << "Enter numbers (separated by space, end with Ctrl+D): ";

double num;

while (std::cin >> num) {

numbers.push\_back(num);

}

}

// Сортування за неспаданням суми цифр дробної частини

sortByFractionalSum(numbers);

// Обчислення середніх значень

double arithmeticMean = calculateArithmeticMean(numbers);

double harmonicMean = calculateHarmonicMean(numbers);

// Виведення результатів у файл

std::ofstream outputFile("results.txt");

outputFile << "Arithmetic mean: " << arithmeticMean << std::endl;

outputFile << "Harmonic mean: " << harmonicMean << std::endl;

std::cout << "Results saved to results.txt" << std::endl;

return 0;

}

sorting.cpp

#include "sorting.h"

#include <cmath>

#include <algorithm>

int sumOfFractionalDigits(double number) {

number = std::fabs(number);

int sum = 0;

number -= static\_cast<int>(number); // Обрізаємо цілу частину

while (number > 0) {

number \*= 10;

sum += static\_cast<int>(number) % 10;

number -= static\_cast<int>(number);

}

return sum;

}

void sortByFractionalSum(std::vector<double>& numbers) {

std::sort(numbers.begin(), numbers.end(), [](double a, double b) {

return sumOfFractionalDigits(a) < sumOfFractionalDigits(b);

});

}

Calculations.cpp

#include "calculations.h"

double calculateArithmeticMean(const std::vector<double>& numbers) {

double sum = 0;

for (double num : numbers) {

sum += num;

}

return sum / numbers.size();

}

double calculateHarmonicMean(const std::vector<double>& numbers) {

double sum = 0;

for (double num : numbers) {

sum += 1 / num;

}

return numbers.size() / sum;

}

4. Опис інтерфейсу програми (керівництво користувача)

Запуск програми: Для запуску програми користувач повинен ввести дані через командний рядок або з файлу. Якщо файл з даними не вказано, програма переходить у діалоговий режим і запитує числа через стандартний ввід.

Введення даних: Числа вводяться через пробіли. Після введення всіх чисел необхідно натиснути Ctrl+D для завершення вводу.

Результати: Після виконання програми результати зберігаються у файл results.txt, який можна відкрити для перегляду середніх значень.

5. Опис тестових прикладів

Тест 1: Введення послідовності чисел: 1.23, 4.56, 7.89.

Результат:

Arithmetic mean: 4.56

Harmonic mean: 2.39

Тест 2: Введення послідовності чисел: 10.75, 20.25, 30.5.

Результат:

Arithmetic mean: 20.5

Harmonic mean: 17.38

6. Аналіз помилок (опис усунення зауважень)

Під час виконання завдання були виявлені наступні помилки:

Помилка обробки порожнього вводу: Якщо не було введено жодного числа, програма не мала механізму обробки порожньої послідовності. Було додано перевірку на порожню послідовність перед виконанням обчислень.

Неточності в обчисленні середнього гармонічного: При обчисленні гармонічного середнього не було враховано можливість нульових значень у послідовності, що призводило до ділення на нуль. Було додано перевірку на нульові значення.