Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра систем штучного інтелекту



**Звіт**

**Про виконання лабораторних та практичних робіт блоку №5**

На тему : «Файли. Бінарні Файли. Символи і Рядкові Змінніта Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.»

З дисципліни «Основи програмування»

**Виконала:**

Студентка групи ШІ-12

Куць Софія Тарасівна

Львів – 2024 р.

Тема роботи: Файли. Бінарні Файли. Символи і Рядкові Змінніта Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

Мета роботи : Освоїти роботу з файлами різних типів, зокрема текстовими та бінариними файлами, символами й рядковими змінними , а також методами стандартної бібліотеки для маніпулювання файлами. Розвинути навички створення й використання бібліотек для обробки даних із файлів, а також алгоритмічні вміння обробляти й аналізувати дані, збережені в різних форматах

Виконання завдань:

1)VNS Lab 6

Задано рядок, що складається із символів. Символи поєднуються в слова.

Слова одне від одного відокремлюються одним або декількома пробілами.

Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів.

Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку

рядка у відповідності зі своїм варіантом. Знайти для рядка довжину найдовшого слова.

#include <iostream>

#include <cstring>

using namespace std;

int findLongestWordLength(const char\* s) {

int maxLength = 0;

int currentLength = 0;

for (int i = 0; s[i] != '.'; ++i) {

if (s[i] != ' ') {

++currentLength;

} else {

if (currentLength > maxLength) {

maxLength = currentLength;

}

currentLength = 0;

}

}

if (currentLength > maxLength) {

maxLength = currentLength;

}

return maxLength;

}

int main() {

char s[256];

cout << "Enter the string (end with a dot): ";

cin.getline(s, 256);

int longestWordLength = findLongestWordLength(s);

cout << "The length of the longest word is: " << longestWordLength << endl;

return 0;

}

Результат:



Час виконання 1 год

2) VNS Lab 8

Сформувати двійковий файл із елементів, заданої у варіанті структури,

роздрукувати його вмістиме, виконати знищення й додавання елементів у

відповідності зі своїм варіантом, використовуючи для пошуку елементів що

знищуються чи додаються, функцію. Формування, друк, додавання й знищення

елементів оформити у вигляді функцій. Передбачити повідомлення про

помилки при відкритті файлу й виконанні операцій вводу/виводу.

Структура "Держава":

- назва;

- державна мова;

- грошова одиниця;

- курс валюти відносно $.

Знищити елемент із зазначеною назвою, додати 2 елементи в кінець файлу.

#include <iostream>

#include <fstream>

#include <cstring>

struct State {

char name[50];

char language[50];

char currency[50];

float exchangeRate;

};

// Функція для створення початкового файлу

void createFile(const char\* filename) {

std::ofstream outFile(filename, std::ios::binary);

if (!outFile) {

std::cerr << "Помилка при створенні файлу.\n";

return;

}

State states[] = {

{"Україна", "українська", "гривня", 0.036},

{"США", "англійська", "долар", 1.0},

{"Японія", "японська", "йена", 0.007}

};

for (size\_t i = 0; i < 3; ++i) {

outFile.write(reinterpret\_cast<const char\*>(&states[i]), sizeof(State));

}

outFile.close();

}

// Функція для друку вмісту файлу

void printFile(const char\* filename) {

std::ifstream inFile(filename, std::ios::binary);

if (!inFile) {

std::cerr << "Помилка при відкритті файлу для читання.\n";

return;

}

State state;

while (inFile.read(reinterpret\_cast<char\*>(&state), sizeof(State))) {

std::cout << "Назва: " << state.name

<< ", Мова: " << state.language

<< ", Валюта: " << state.currency

<< ", Курс: " << state.exchangeRate << "\n";

}

inFile.close();

}

// Функція для видалення елемента за назвою

void deleteState(const char\* filename, const char\* stateName) {

std::ifstream inFile(filename, std::ios::binary);

if (!inFile) {

std::cerr << "Помилка при відкритті файлу для читання.\n";

return;

}

std::ofstream outFile("temp.bin", std::ios::binary);

if (!outFile) {

std::cerr << "Помилка при створенні тимчасового файлу.\n";

inFile.close();

return;

}

State state;

bool found = false;

while (inFile.read(reinterpret\_cast<char\*>(&state), sizeof(State))) {

if (std::strcmp(state.name, stateName) != 0) {

outFile.write(reinterpret\_cast<const char\*>(&state), sizeof(State));

} else {

found = true;

}

}

inFile.close();

outFile.close();

if (!found) {

std::cerr << "Держава з назвою \"" << stateName << "\" не знайдена.\n";

}

std::remove(filename);

std::rename("temp.bin", filename);

}

// Функція для додавання нових елементів у файл

void addStates(const char\* filename, const State\* newStates, size\_t count) {

std::ofstream outFile(filename, std::ios::binary | std::ios::app);

if (!outFile) {

std::cerr << "Помилка при відкритті файлу для запису.\n";

return;

}

for (size\_t i = 0; i < count; ++i) {

outFile.write(reinterpret\_cast<const char\*>(&newStates[i]), sizeof(State));

}

outFile.close();

}

// Головна функція

int main() {

const char\* filename = "states.bin";

// Створення файлу

createFile(filename);

// Друк вмісту файлу

std::cout << "Початковий вміст файлу:\n";

printFile(filename);

// Видалення елемента

deleteState(filename, "США");

// Друк після видалення

std::cout << "\nВміст файлу після видалення \"США\":\n";

printFile(filename);

// Додавання нових елементів

State newStates[] = {

{"Німеччина", "німецька", "євро", 1.1},

{"Франція", "французька", "євро", 1.1}

};

addStates(filename, newStates, 2);

// Друк після додавання

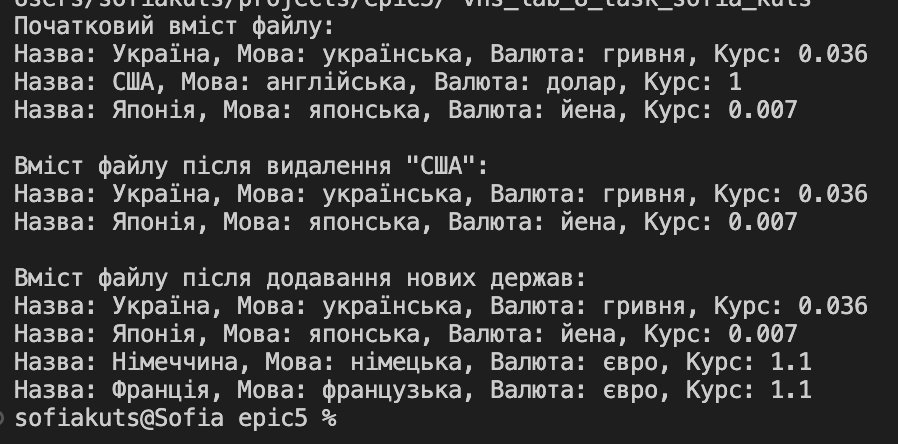
std::cout << "\nВміст файлу після додавання нових держав:\n";

printFile(filename);

return 0;

}

Результат:



Час виконання 2,5 год

3) VNS Lab 9

Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього

Інформацію

1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких є однакові слова.

2) Визначити кількість голосних букв в останньому рядку файлу F2.

#include <iostream>

#include <fstream>

#include <string>

#include <set>

#include <sstream>

#include <cctype>

bool hasDuplicateWords(const std::string& line) {

std::set<std::string> words;

std::istringstream stream(line);

std::string word;

while (stream >> word) {

if (words.count(word)) {

return true;

}

words.insert(word);

}

return false;

}

int countVowels(const std::string& line) {

int count = 0;

const std::string vowels = "AEIOUaeiouАЕіИОУЮЯаеиоуюя";

for (size\_t i = 0; i < line.size(); ++i) {

if (vowels.find(line[i]) != std::string::npos) {

++count;

}

}

return count;

}

void createFileF1(const char\* filename) {

std::ofstream outFile(filename);

if (!outFile) {

std::cerr << "Помилка при створенні файлу F1.\n";

return;

}

outFile << "Це перший рядок файлу.\n";

outFile << "Це другий рядок з повтором слово слово.\n";

outFile << "Один раз використане слово.\n";

outFile << "Ще один рядок без дублікатів.\n";

outFile << "Це рядок, де дубль слово слово ще раз.\n";

outFile << "Цей рядок унікальний.\n";

outFile << "Це тестовий рядок.\n";

outFile << "Слово слово тут є двічі.\n";

outFile << "Це останній рядок з дублем дублем.\n";

outFile << "Рядок без дублікатів.\n";

outFile.close();

}

void copyDuplicateLines(const char\* filenameF1, const char\* filenameF2) {

std::ifstream inFile(filenameF1);

if (!inFile) {

std::cerr << "Помилка при відкритті файлу F1 для читання.\n";

return;

}

std::ofstream outFile(filenameF2);

if (!outFile) {

std::cerr << "Помилка при створенні файлу F2.\n";

inFile.close();

return;

}

std::string line;

while (std::getline(inFile, line)) {

if (hasDuplicateWords(line)) {

outFile << line << '\n';

}

}

inFile.close();

outFile.close();

}

int countVowelsInLastLine(const char\* filenameF2) {

std::ifstream inFile(filenameF2);

if (!inFile) {

std::cerr << "Помилка при відкритті файлу F2 для читання.\n";

return 0;

}

std::string line, lastLine;

while (std::getline(inFile, line)) {

lastLine = line;

}

inFile.close();

return countVowels(lastLine);

}

int main() {

const char\* filenameF1 = "F1.txt";

const char\* filenameF2 = "F2.txt";

// Створення файлу F1

createFileF1(filenameF1);

// Копіювання рядків з дубльованими словами у F2

copyDuplicateLines(filenameF1, filenameF2);

// Визначення кількості голосних в останньому рядку F2

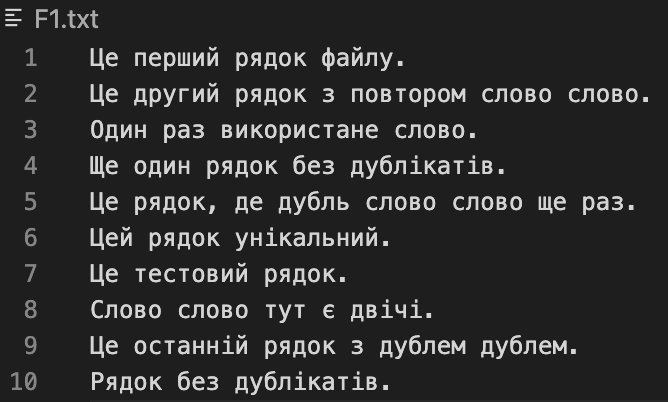
int vowelsCount = countVowelsInLastLine(filenameF2);

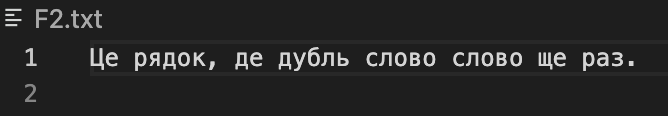
std::cout << "Кількість голосних букв в останньому рядку файлу F2: " << vowelsCount << "\n";

return 0;

}

Результат:





Час виконання: 3 год

4) Self practice

Вам дано масив a з N цілих чисел.  
Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4].  
Після цього оберніть посортовану версію масиву a на K, тобто при K=3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3].  
Виведіть результат.

**Input**

У першому рядку цілі числа N та K

У другому рядку N цілих чисел - елементи масиву a.

**Output**

У першому рядку ціле число N - розмір множини a  
У наступному рядку N цілих чисел - множина a

#include <iostream>

#include <vector>

#include <algorithm>

int main() {

int n, k;

std::cin >> n >> k;

std::vector<int> a(n);

for (int i = 0; i < n; ++i) {

std::cin >> a[i];

}

// Видалення повторів

std::sort(a.begin(), a.end());

a.erase(std::unique(a.begin(), a.end()), a.end());

// Поворот на K

std::rotate(a.begin(), a.begin() + (k % a.size()), a.end());

// Виведення результату

std::cout << a.size() << std::endl;

for (size\_t i = 0; i < a.size(); ++i) {

std::cout << a[i] << " ";

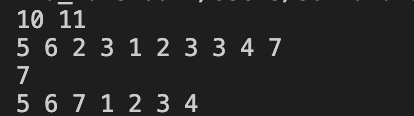
}

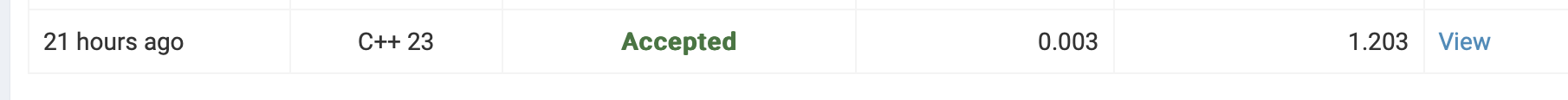
std::cout << std::endl;

return 0

}

Результат:





Час виконання : 45 хв

5) Algotester Lab 4

Вам дано 2 цілих чисел масиви, розміром N та M.

Ваше завдання вивести:

1. Різницю N-M

2. Різницю M-N

3. Їх перетин

4. Їх обєднання

5. Їх симетричну різницю

# Input

У першому рядку ціле число N - розмір масиву 1

У другому рядку N цілих чисел - елементи масиву 1

У третьому рядку ціле число M - розмір масиву 2

У четвертом рядку M цілих чисел - елементи масиву 2

# Output

Вивести результат виконання 5 вищезазначених операцій у форматі:

У першому рядку ціле число N - розмір множини

У наступному рядку N цілих чисел - посортована у порядку зростання множина

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

void print\_vector(const vector<int>& v) {

cout << v.size() << endl;

for (size\_t i = 0; i < v.size(); ++i) {

cout << v[i] << " ";

}

cout << endl;

}

int main() {

int n, m;

cin >> n;

vector<int> a(n);

for (int i = 0; i < n; ++i) cin >> a[i];

cin >> m;

vector<int> b(m);

for (int i = 0; i < m; ++i) cin >> b[i];

// Сортуємо масиви

sort(a.begin(), a.end());

sort(b.begin(), b.end());

// 1. Різниця N-M

vector<int> diff\_nm;

set\_difference(a.begin(), a.end(), b.begin(), b.end(), back\_inserter(diff\_nm));

print\_vector(diff\_nm);

// 2. Різниця M-N

vector<int> diff\_mn;

set\_difference(b.begin(), b.end(), a.begin(), a.end(), back\_inserter(diff\_mn));

print\_vector(diff\_mn);

// 3. Перетин

vector<int> intersection;

set\_intersection(a.begin(), a.end(), b.begin(), b.end(), back\_inserter(intersection));

print\_vector(intersection);

// 4. Об'єднання

vector<int> union\_set;

set\_union(a.begin(), a.end(), b.begin(), b.end(), back\_inserter(union\_set));

print\_vector(union\_set);

// 5. Симетрична різниця

vector<int> sym\_diff;

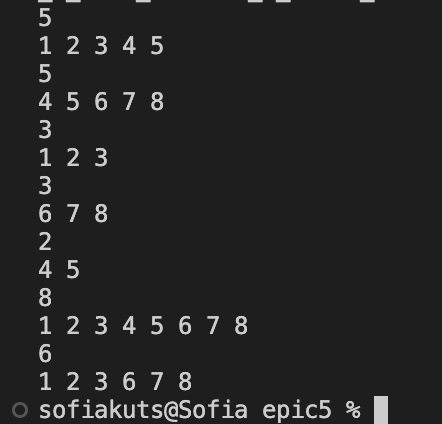
set\_symmetric\_difference(a.begin(), a.end(), b.begin(), b.end(), back\_inserter(sym\_diff));

print\_vector(sym\_diff);

return 0;

}

Результат :





Час виконання 1 год

6) Algotester Lab 6

Вам дано N слів та число K.

Ваше завдання перечислити букви в словах, які зустрічаються в тексті більше-рівне ніж K разів (саме слово, не буква!).

Великі та маленькі букви вважаються однаковими, виводити необхідно малі, посортовані від останьої до першої у алфавіті. Букву потрібно виводити лише один раз.

У випадку якщо таких букв немає - вивести "Empty!".

# Input

Цілі числа N та K - загальна кількість слів та мінімальна кількість слів щоб враховувати букви цього слова в результаті.

N стрічок s

# Output

У першому рядку ціле число M - кількість унікальних букв

У другому рядку унікальні букви через пробіли

#include <iostream>

#include <vector>

#include <map>

#include <set>

#include <algorithm>

#include <cctype>

using namespace std;

int main() {

int N, K;

cin >> N >> K;

map<string, int> wordCount;

vector<string> words(N);

for (int i = 0; i < N; ++i) {

cin >> words[i];

for (size\_t j = 0; j < words[i].size(); ++j) {

words[i][j] = tolower(words[i][j]);

}

wordCount[words[i]]++;

}

set<char> uniqueLetters;

for (map<string, int>::iterator it = wordCount.begin(); it != wordCount.end(); ++it) {

if (it->second >= K) {

for (size\_t j = 0; j < it->first.size(); ++j) {

uniqueLetters.insert(it->first[j]);

}

}

}

if (uniqueLetters.empty()) {

cout << "Empty!" << endl;

} else {

cout << uniqueLetters.size() << endl;

vector<char> result(uniqueLetters.begin(), uniqueLetters.end());

sort(result.rbegin(), result.rend());

for (size\_t i = 0; i < result.size(); ++i) {

cout << result[i] << " ";

}

cout << endl;

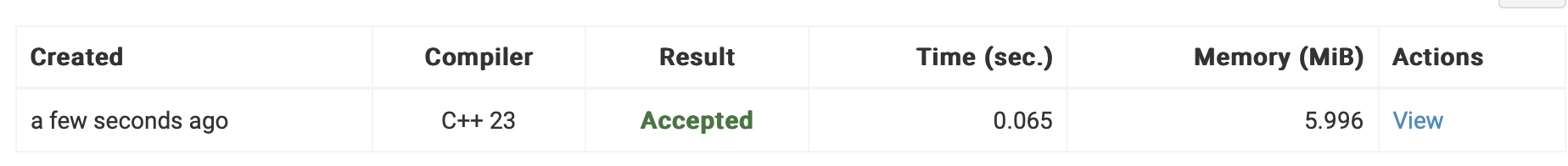
}

return 0;

}

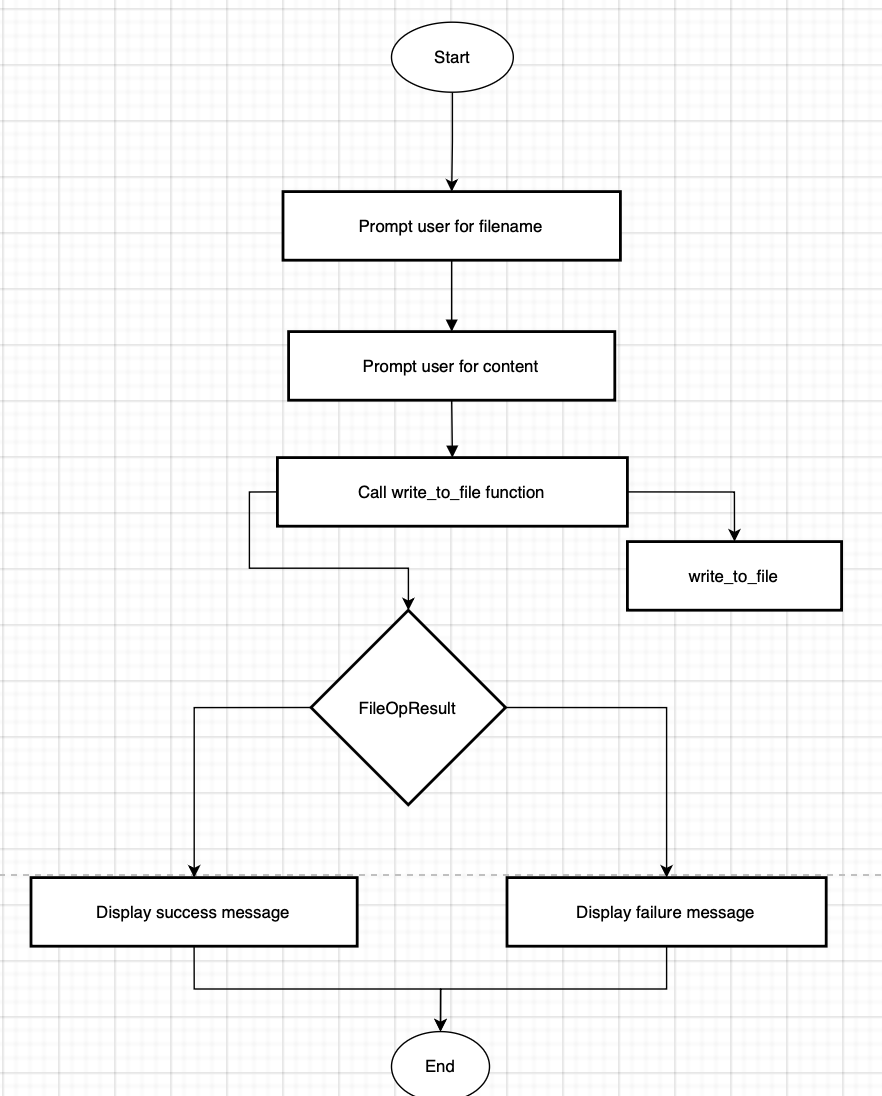
Резульат:

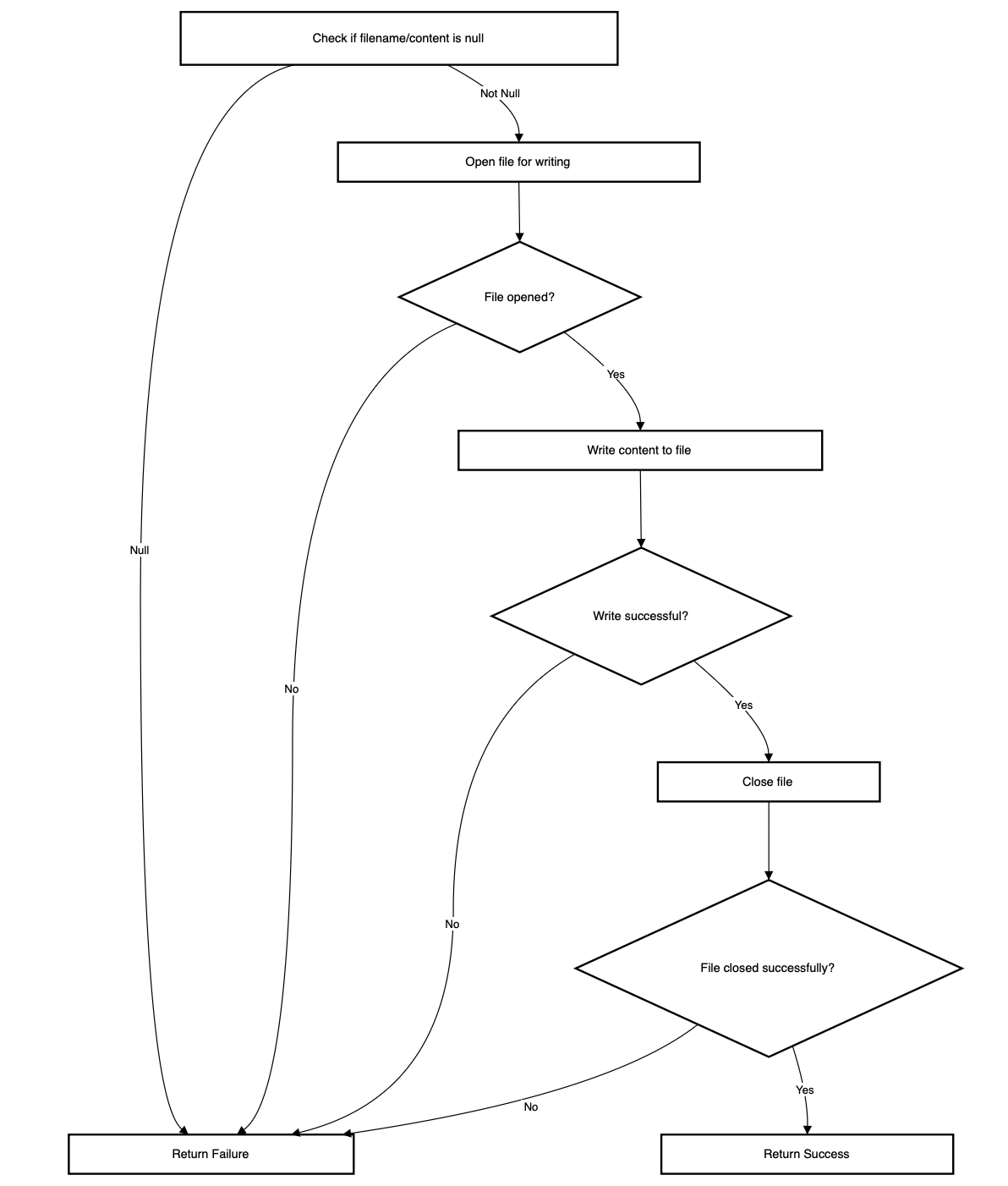




Час виконання 40 хв

7) Class practice work





#include <iostream>

#include <fstream>

#include <string>

enum FileOpResult {

Success,

Failure

};

FileOpResult write\_to\_file(const char \*name, const char \*content) {

if (name == nullptr || content == nullptr) {

return Failure; // Перевірка на нульові вказівники

}

std::ofstream outfile(name, std::ios::out | std::ios::trunc); // Відкриття файла для запису з перезаписом

if (!outfile) {

return Failure; // Не вдалося відкрити файл

}

outfile << content; // Запис вмісту в файл

if (!outfile.good()) {

return Failure; // Перевірка на помилки запису

}

outfile.close(); // Закриття файла

if (!outfile.good()) {

return Failure; // Помилка при закритті файла

}

return Success;

}

int main() {

std::string filename, content;

std::cout << "Enter the filename: ";

std::getline(std::cin, filename);

std::cout << "Enter the content to write: ";

std::getline(std::cin, content);

FileOpResult result = write\_to\_file(filename.c\_str(), content.c\_str());

if (result == Success) {

std::cout << "File written successfully.\n";

} else {

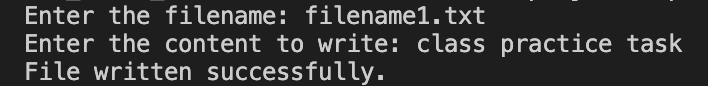
std::cout << "Failed to write to file.\n";

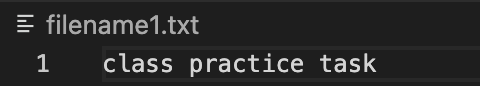
}

return 0;

}

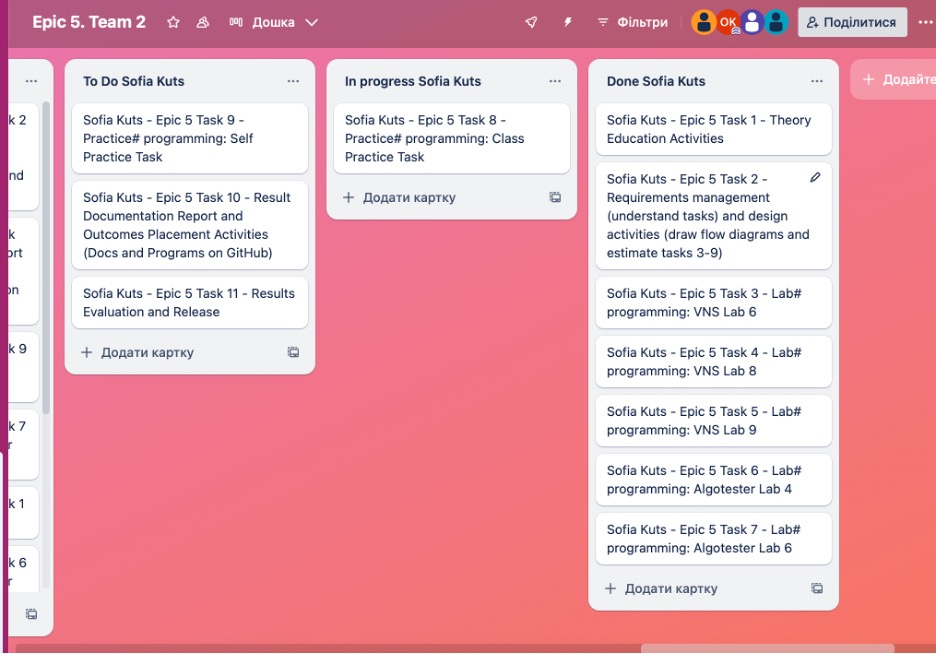
Результат:



 час виконання 1,5 год

Кооперація з командою :

скрін дошки в Trello

****

Скрін з онлайн зустрічі

