Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт 5**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Епік 5: Файли. Системи числення. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек»

***Виконав:***

студент групи ШІ-13

Юнко Дмитро Богданович

# **Тема роботи:**

# Робота із простими структурами даних та алгоритмами обробки. Робота з одновимірними та двовимірними масивами.

# **Мета роботи:**

# Мета полягає в освоєнні простих структур даних та алгритмів обробки масивів з використанням відповідних функцій. Навчитися передавати Символи і Рядкові Змінні та Текстові Файли

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

- Тема №1: Файли

- Тема №2: Системи числення

- Тема №3: Бінарні Файли

- Тема №4: Створення й використання бібліотек

1. Індивідуальний план опрацювання теорії:

Тема №1: Файли

* + Джерела Інформації
    - Стаття: [File Handling through C++ Classes - GeeksforGeeks](https://www.geeksforgeeks.org/file-handling-c-classes/)
    - Курс: Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Класи і об’єкти для роботи з файлами, такі як std::fstream, std::ifstream, std::ofstream, std::filebuf. Як відкривати, закривати, читати, писати, переміщатися та перевіряти стан файлів.
    - Формати і режими файлів (in, out, app, binary, trunc), такі як текстові, бінарні, додавання, перезапис, читання тощо. Різні способи доступу до даних у файлі
    - Як вибрати відповідний формат та режим файлу. Як працювати з різними типами даних у файлах.
    - Cпособи перевірки стану файлу, такі як функції is\_open(), good(), eof(), fail().
    - Бібліотека fstream, яка дозволяє відкривати, закривати, читати, писати, переміщатися та перевіряти стан файлів
  + Статус: Ознайомлений
  + Початок опрацювання теми: 10.12.2023
  + Звершення опрацювання теми: 10.12.2023
* Тема №2: Системи числення
  + Джерела Інформації:
    - Стаття: [Introduction to Number System in C++ - Dot Net Tutorials](https://dotnettutorials.net/lesson/introduction-to-number-system/)
    - Стаття: [5.5 — Numeral systems (decimal, binary, hexadecimal, and octal) – Learn C++ (learncpp.com)](https://www.learncpp.com/cpp-tutorial/numeral-systems-decimal-binary-hexadecimal-and-octal/)
    - Курс: Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - визначення та приклади різних систем числення, таких як десяткова, двійкова, вісімкова та шістнадцяткова
    - правила та методи перетворення чисел між різними системами числення, такі як ділення з остачею, множення на основу, групування бітів або використання стандартних функцій бібліотеки c++
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.12.2023
  + Звершення опрацювання теми: 01.12.2023
* Тема №3: Бінарні Файли.
  + Джерела Інформації:
    - Стаття: [Reading and writing binary file in C++ - CodeSpeedy](https://www.codespeedy.com/reading-and-writing-binary-file-in-cpp/)
    - Стаття: [An Introduction to Object Serialization in C++ | CodeGuru](https://www.codeguru.com/cplusplus/an-introduction-to-object-serialization-in-c/)
    - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Формати бінарних файлів: які переваги та недоліки бінарних файлів порівняно з текстовими, які стандарти та конвенції існують для бінарних файлів, як читати та записувати різні типи даних у бінарних файлах, як враховувати розмір та порядок байтів
    - Серіалізація та десеріалізація об’єктів у C++: як зберігати та відновлювати стан об’єктів у бінарних файлах, як використовувати fstream для цього, які проблеми та обмеження можуть виникнути при серіалізації
    - перетворення типів покажчиків при читанні та записуванні даних, оскільки read та write працюють з char\* покажчиками.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.12.2023
  + Звершення опрацювання теми: 01.12.2023

Тема №4: Створення й використання бібліотек

* + Джерела Інформації:
    - Курс: [Walkthrough: Create and use a static library (C++) | Microsoft Learn](https://learn.microsoft.com/en-us/cpp/build/walkthrough-creating-and-using-a-static-library-cpp?view=msvc-170)
    - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Що таке бібліотека в c++ і які її види (статичні, динамічні, заголовкові)
    - Як створити власну бібліотеку в c++ (за допомогою компілятора, IDE, або інших інструментів
  + Статус: Ознайомлений
  + Початок опрацювання теми: 29.11.2023
  + Звершення опрацювання теми: 29.11.2023

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

**Завдання №1 Опрацювання теорії**

* Деталі завдання: опрацювання різноманітних матеріалів, аналіз та дослідження відео, статей, книг на задані теми. Організація та структурування отриманих даних для можливості ефективніше засвоювати отримані знання на практиці. Вивчення найважливіших моментів.

**Завдання №2 Проектування за допомогою Draw.io(складання блок-схем)**

* Деталі завдання:

Побудова блок-схем до завдань 3-9.

#### Завдання №3 VNS Lab 6 - Task 1

* Варіант завдання: **20**
* Деталі завдання: 20: знайти довжину найдовшого слова в цьому рядку.

#### Завдання №4 VNS Lab 8 - Task 1

* Варіант завдання: **20**

Деталі завдання:

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.

#### Завдання №5 VNS Lab 9 - Task 1

Варіант завдання: 1

* Деталі завдання: :

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.

#### Завдання №6 Alfotester Lab 4 using STL

* Варіант завдання: 1
* Деталі завдання:

Дано 2 цiлих чисел масиви, розмiром N та M.

Ваше завдання вивести:

1. Рiзницю N-M

2. Рiзницю M-N

3. Їх перетин

4. Їх обєднання

5. Їх симетричну рiзницю

Важливі деталі: використання засобів STL std::set\_intersection, std::set\_symmetric\_difference, std::set\_difference, std::set\_union

#### Завдання №7 Alfotester Lab 4

* Варіант завдання: **1**
* Деталі завдання: -

#### Завдання №8 Alfotester Lab 6

* Деталі завдання:

Дано N слiв та число K.

Завдання перечислити букви в словах, якi зустрiчаються в текстi бiльше-рiвне нiж K разiв (саме слово, не буква!). Великi та маленькi букви вважаються однаковими, виводити необхiдно малi, посортованi вiд

останьої до першої у алфавiтi. Букву потрiбно виводити лише один раз.

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

#### Завдання №9 Class Practice 1

* Деталі завдання:

- створити файл із заданим ім’ям; якщо файл існує – перезаписати його вміст

-      написати код стійкий до різних варіантів вхідних параметрів

-       name – ім’я, може не включати шлях

-       записати у файл вміст стрічки content, прочитати content із стандартного вводу

-       повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.

#### Завдання №10 Class Practice 2

* Деталі завдання:

копіювати вміст файла з ім’ям file\_from у файл з ім’ям file\_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів

      file\_from, file\_to – можуть бути повним або відносним шляхом

      повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

#### Завдання №11 Self Practice

* Деталі завдання:  реалізація програми, яка розбиватиме текстовий файл на чанки.

#### Завдання №12 Self Practice

* Деталі завдання:  покращення коду практичної з 3 епіку, з використанням свіжо здобутих знаннь про роботу з файлами

## **2. Дизайн та планована оцінка часу виконання завдань:**

#### Програма № 3 VNS Lab 6 - Task 1

* Блок-схема
* Планований час на реалізацію: 15хв

використання бібліотеки cmath

#### Програма №4 VNS Lab 8 - Task 1

* Планований час на реалізацію: 80хв

#### Програма №5 VNS Lab 9 - Task 1

* Планований час на реалізацію 30хв

#### Програма №6 Alfotester Lab 4 using STL

* Планований час на реалізацію 45 хв

#### Програма №7 Alfotester Lab 4

* Планований час на реалізацію 100 хв

#### Програма №8 Alfotester Lab 6

* Планований час на реалізацію 80 хв

#### Програма №9 Class Practice 1

* Планований час на реалізацію 10 хв

#### Програма №10 Class Practice 2

* Планований час на реалізацію 10 хв

#### Програма №11 Self Practice 1

* Планований час на реалізацію 15 хв

#### Програма №12 Self Practice | Library

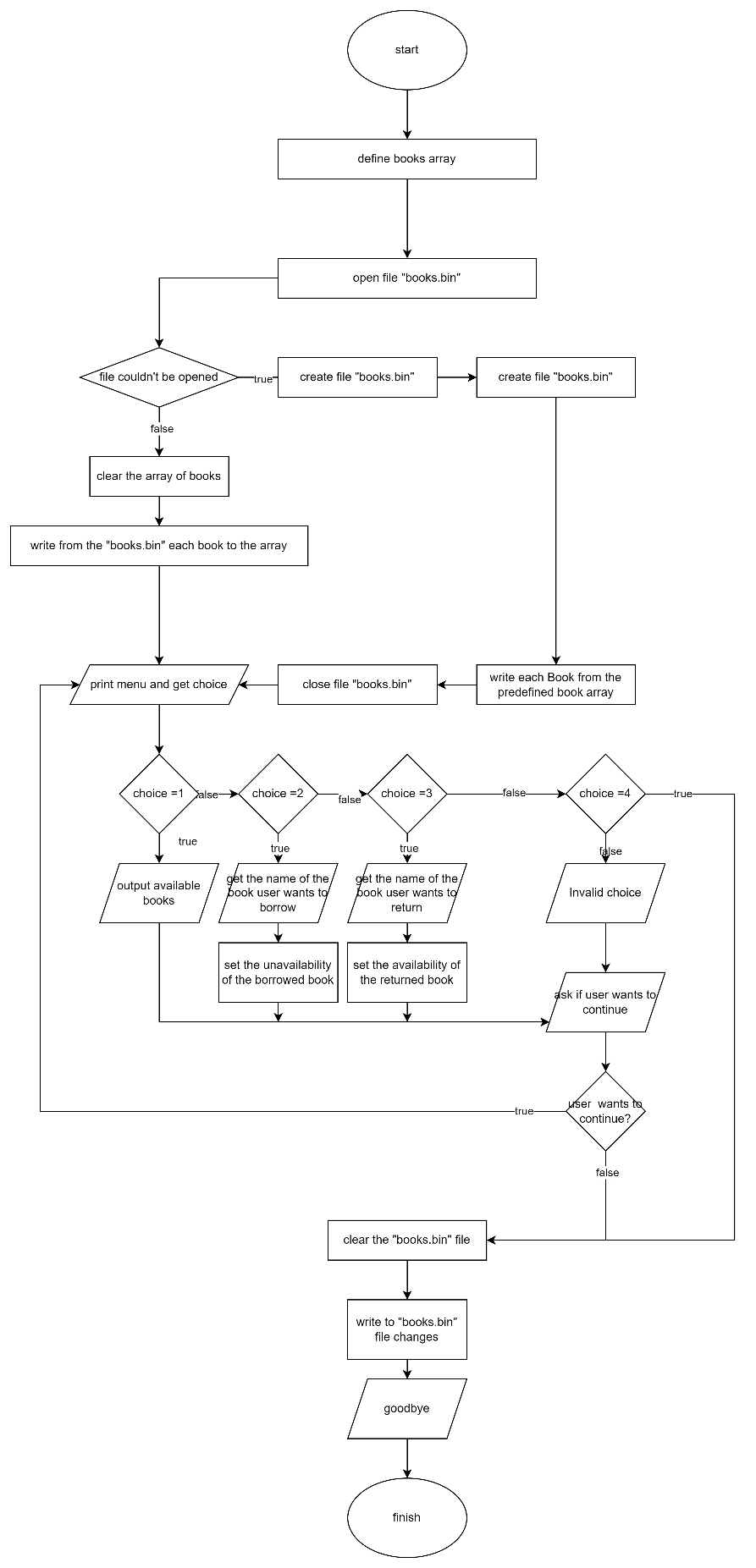


Рисунок 1 Self Practice

* Планований час на реалізацію: 60хв

## **4. Код програм з посиланням на зовнішні ресурси:** [https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/230/files#diff-097297c805c8250774272aa6616b4bbbdd5bd20fde5f2b0cf8b246d98169163d](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/230/files%23diff-097297c805c8250774272aa6616b4bbbdd5bd20fde5f2b0cf8b246d98169163d)

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

int main()

{

    string s;

    getline(cin, s);

    istringstream iss(s);

    string word;

    int max\_length = 0;

    while (iss >> word)

    {

        if (word.length() > max\_length)

        {

            max\_length = word.length();

        }

    }

    cout << "Max length: " << max\_length << endl;

    return 0;

}

###### Код до програми № 3 VNS Lab 6 - Task 1

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

struct Country

{

    string name;

    string language;

    string currency;

    double exchangeRate;

};

void writeStringToFile(ofstream &file, const string &str)

{

    size\_t size = str.size();

    file.write((char \*)&size, sizeof(size));

    file.write(str.c\_str(), size);

}

void writeStringToFile(ofstream &file, const double &db)

{

    string str = to\_string(db);

    size\_t size = sizeof(str);

    file.write((char \*)&size, sizeof(size));

    file.write(str.c\_str(), size);

}

string readStringFromFile(ifstream &file)

{

    size\_t size;

    file.read((char \*)&size, sizeof(size));

    string str(size, ' ');

    file.read(&str[0], size);

    return str;

}

void writeToFile(const vector<Country> &countries, const string &filename)

{

    cout << "writing countries file..." << endl;

    ofstream file(filename, ios::binary);

    if (!file)

    {

        cout << "Error writing file" << endl;

        return;

    }

    for (const Country &country : countries)

    {

        writeStringToFile(file, country.name);

        writeStringToFile(file, country.language);

        writeStringToFile(file, country.currency);

        writeStringToFile(file, country.exchangeRate);

    }

    file.close();

}

vector<Country> readFromFile(const string &filename)

{

    cout << "reading countries file..." << endl;

    ifstream file(filename, ios::binary);

    if (!file)

    {

        cout << "Error opening file" << endl;

        return {};

    }

    vector<Country> countries;

    while (file.peek() != EOF)

    {

        Country country;

        country.name = readStringFromFile(file);

        country.language = readStringFromFile(file);

        country.currency = readStringFromFile(file);

        country.exchangeRate = stod(readStringFromFile(file));

        countries.push\_back(country);

    }

    file.close();

    return countries;

}

void printCountries(const vector<Country> &countries)

{

    for (const Country &country : countries)

    {

        cout << "Name: " << country.name << " Language: " << country.language

             << " Currency: " << country.currency << " Exchange rate: " << country.exchangeRate << endl;

    }

}

void removeCountry(vector<Country> &countries, const string &name)

{

    cout << "removing country " << name << endl;

    countries.erase(remove\_if(countries.begin(), countries.end(), [name](const Country &country){ return country.name.compare(name) == 0; }),countries.end());

}

void addCountry(vector<Country> &countries, const Country &country)

{

    cout << "adding a country " << country.name << endl;

    countries.push\_back(country);

}

int main()

{

    vector<Country> countries = {

        {"Ukraine", "Ukrainian", "Hryvnia", 39.676767},

        {"USA", "English", "Dollar", 1.0},

    };

    writeToFile(countries, "countries.bin");

    countries = readFromFile("countries.bin");

    printCountries(countries);

    removeCountry(countries, "USA");

    addCountry(countries, {"France", "French", "Euro", 1.0});

addCountry(countries, {"Spain", "Spanish", "Euro", 1.0});

    writeToFile(countries, "countries.bin");

    countries = readFromFile("countries.bin");

    printCountries(countries);

    return 0;

}

###### Код до програми № 4 VNS Lab 8 - Task 1 [https://github.com/artificial-intelligence department/ai\_programming\_playground/pull/494](https://github.com/artificial-intelligence%20department/ai_programming_playground/pull/494)

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

#include <vector>

#include <unordered\_set>

#include <algorithm>

using namespace std;

bool hasDuplicate(string line)

{

    istringstream iss(line);

    vector<std::string> words;

    string word;

    while (iss >> word)

    {

        if (find(words.begin(), words.end(), word) != words.end())

            return true;

        words.push\_back(word);

    }

    return false;

}

int getVowAm(string line)

{

    int count = 0;

    for (char c : line)

    {

        c = tolower(c);

        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u')

            count++;

    }

    return count;

}

int main()

{

    string myText;

    ofstream f1W("F1.txt");

    if (f1W.is\_open())

    {

        for (size\_t i = 0; i < 10; i++)

        {

            string text;

            getline(cin, text);

            f1W << text << "\n";

        }

        f1W.close();

    }

    vector<string> lines;

    string line;

    string lastLn;

    ifstream f1R("F1.txt");

    ofstream f2("F2.txt");

    if (f1R.is\_open())

    {

        while (getline(f1R, line))

        {

            if (hasDuplicate(line))

            {

                f2 << line << endl;

                lastLn = line;

            }

        }

    }

    cout << getVowAm(lastLn);

    return 0;

}

###### Код до програми № 5 VNS Lab 9 - Task 1

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

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

{

    int N{0};

    int M{0};

    cin >> N;

    if (!(1 <= N && N <= 100))

    {

        return 0;

    }

    vector<int> nAr;

    vector<int> mAr;

    vector<int> secDiff;

    vector<int> firstDiff;

    vector<int> inter;

    vector<int> unite;

    vector<int> symm;

    for (size\_t i = 0; i < N; i++)

    {

        int ni;

        cin >> ni;

        nAr.push\_back(ni);

    }

    cin >> M;

    if (!(1 <= M && M <= 100))

    {

        return 0;

    }

    for (size\_t i = 0; i < M; i++)

    {

        int mi;

        cin >> mi;

        mAr.push\_back(mi);

    }

    sort(nAr.begin(), nAr.end());

    sort(mAr.begin(), mAr.end());

    set\_difference(nAr.begin(), nAr.end(), mAr.begin(), mAr.end(), back\_inserter(firstDiff));

    set\_difference(mAr.begin(), mAr.end(), nAr.begin(), nAr.end(), back\_inserter(secDiff));

    set\_intersection(nAr.begin(), nAr.end(), mAr.begin(), mAr.end(), back\_inserter(inter));

    set\_union(nAr.begin(), nAr.end(), mAr.begin(), mAr.end(), back\_inserter(unite));

    set\_symmetric\_difference(nAr.begin(), nAr.end(), mAr.begin(), mAr.end(), back\_inserter(symm));

    cout << firstDiff.size() << endl;

    ;

    for (int x : firstDiff)

    {

        cout << x << " ";

    }

    cout << endl;

    cout << endl;

    cout << secDiff.size() << endl;

    ;

    for (int x : secDiff)

    {

        cout << x << " ";

    }

    cout << endl;

    cout << endl;

    cout << inter.size() << endl;

    for (int x : inter)

    {

        cout << x << " ";

    }

    cout << endl;

    cout << endl;

    cout << unite.size() << endl;

    for (int x : unite)

    {

        cout << x << " ";

    }

    cout << endl;

    cout << endl;

    cout << symm.size() << endl;

    for (int x : symm)

    {

        cout << x << " ";

    }

    cout << endl;

    cout << endl;

    return 0;

}

###### Код до програми № 6 Alfotester Lab 4 using STL

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

void displayVect(vector<int> &vect)

{

    cout << vect.size() << endl;

    for (int n : vect)

    {

        cout << n << " ";

    }

    cout << endl;

    cout << endl;

}

void displayDiff(vector<int> first, vector<int> second, vector<int> &diff)

{

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

    {

        for (size\_t j = 0; j < second.size(); j++)

        {

            if (first[i] == second[j])

            {

                first[i] = 0;

                second[j] = 0;

            }

        }

    }

    first.erase(remove(first.begin(), first.end(), 0), first.end());

    displayVect(first);

    for (int n : first)

    {

        diff.push\_back(n);

    }

}

void displayEqual(vector<int> first, vector<int> second)

{

    vector<int> same;

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

    {

        for (int j = 0; j < second.size(); j++)

        {

            if (first[i] == second[j])

            {

                same.push\_back(first[i]);

                first[i] = 0;

                second[j] = 0;

            }

        }

    }

    same.erase(remove(same.begin(), same.end(), 0), same.end());

    displayVect(same);

}

void displayUn(vector<int> first, vector<int> &second, vector<int> &secDiff)

{

    first.insert(first.end(), secDiff.begin(), secDiff.end());

    sort(first.begin(), first.end());

    displayVect(first);

}

void displaySymdiff(vector<int> &firstDiff, vector<int> &secDiff)

{

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

    {

        firstDiff.push\_back(secDiff[i]);

    }

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

    displayVect(firstDiff);

}

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

{

    int N{0};

    int M{0};

    cin >> N;

    if (!(1 <= N && N <= 100))

    {

        return 0;

    }

    vector<int> nAr;

    vector<int> mAr;

    vector<int> firstDiff;

    vector<int> secDiff;

    for (size\_t i = 0; i < N; i++)

    {

        int ni;

        cin >> ni;

        nAr.push\_back(ni);

    }

    cin >> M;

    if (!(1 <= M && M <= 100))

    {

        return 0;

    }

    for (size\_t i = 0; i < M; i++)

    {

        int mi;

        cin >> mi;

        mAr.push\_back(mi);

    }

    sort(nAr.begin(), nAr.end());

    sort(mAr.begin(), mAr.end());

    displayDiff(nAr, mAr, firstDiff);

    displayDiff(mAr, nAr, secDiff);

    displayEqual(nAr, mAr);

    displayUn(nAr, mAr, secDiff);

    displaySymdiff(firstDiff, secDiff);

    return 0;

}

###### Код до програми № 7 Alfotester Lab 4

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

#include <set>

using namespace std;

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

{

    int N{0};

    int K{0};

    vector<string> words;

    cin >> N;

    cin >> K;

    cin.ignore();

    for (size\_t i = 0; i < N; i++)

    {

        string word;

        getline(cin, word);

        transform(word.begin(), word.end(), word.begin(), ::tolower);

        words.push\_back(word);

    }

    sort(words.begin(), words.end());

    // for (string n:words)

    // {

    //     for(char m: n){

    //         cout<<m;

    //     }

    //     cout<<endl;

    // }

    string repChars;

    for (vector<string>::iterator i = words.begin(); i != words.end(); i++)

    {

        int counter{1};

        if (\*i == "")

        {

            continue;

        }

        for (vector<string>::iterator j = i + 1; j != words.end(); j++)

        {

            if (i != j && \*i == \*j)

            {

                counter++;

                \*j = "";

            }

            else if (i != j && \*j != \*(j - 1))

            {

                break;

            }

        }

        if (counter >= K)

        {

            repChars.append(\*i);

        }

    }

    sort(repChars.begin(), repChars.end(), greater<char>());

    vector<char> outpuChars;

    for (size\_t i = 0; i < repChars.length(); i++)

    {

        if (repChars[i] != repChars[i + 1])

        {

            outpuChars.push\_back(repChars[i]);

        }

    }

    if (repChars.size() == 0)

    {

        cout << "Empty!";

    }

    else

    {

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

        for (char &ch : outpuChars)

        {

            cout << ch << " ";

        }

    }

    return 0;

}

###### Код до програми № 8 Algotester Lab 6 – Task 1

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

string createAndWriteToFile(string name, string content)

{

    ofstream file(name);

    if (file.is\_open())

    {

        file << content;

        file.close();

        return "Success";

    }

    else

    {

        return "Failure";

    }

}

int main()

{

    string name;

    string content;

    cout << "Enter file name: ";

    getline(cin, name);

    cout << "Enter content: ";

    getline(cin, content);

    string status = createAndWriteToFile(name, content);

    cout << "Operation status: " << status << endl;

    return 0;

}

###### Код до програми № 9 Class Practice 1

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main() {

    const int chunk\_size = 200;

    char\* buffer = new char[chunk\_size];

    string myText;

  ifstream MyFile("F1.txt");

  int chunk\_count = 0;

 while (MyFile) {

    MyFile.read(buffer, chunk\_size);

    int bytesRead = MyFile.gcount();

    if (bytesRead > 0) {

      chunk\_count++;

      string outfile\_name = "output" + to\_string(chunk\_count) + ".txt";

      ofstream outfile(outfile\_name);

      if (!outfile) {

        cerr << "Error opening output file\n";

        return 2;

      }

      outfile.write(buffer, bytesRead);

      outfile.close();

    }

  }

MyFile.seekg(0);

  MyFile.close();

}

###### Код до програми № 11 Self Practice 1

#include <cstdio>

#include <iostream>

#include <vector>

#include <sstream>

#include <string>

#include <fstream>

using namespace std;

struct Book {

    string title;

    bool status;

};

string serialize(const Book& book) {

    stringstream ss;

    ss << book.title << '\n' << book.status;

    return ss.str();

}

Book deserialize(const string& str) {

    stringstream ss(str);

    Book book;

    getline(ss, book.title);

    ss >> book.status;

    return book;

}

void writeEachBookToFile(fstream &file, const vector<Book>& books)

{

    file.close();

    file.open("books.bin", ios::out | ios::binary | ios::trunc);

    for (const Book &book : books) {

            string str = serialize(book);

        size\_t len = str.size();

        file.write((char\*)&len, sizeof(size\_t));

        file.write(str.c\_str(), len);

        }

}

int main() {

    vector<Book> books = {{"Harry Potter", true}, {"The Hobbit", true}, {"Pride and Prejudice", true}, {"1984", true}};

    int selection;

    string titleName;

    char continueSelection;

    fstream file;

    file.open("books.bin", ios::in | ios::binary);

    if (!file.is\_open()) {

        file.open("books.bin", ios::out | ios::binary);

        writeEachBookToFile(file,books);

        file.close();

        file.open("books.bin", ios::in | ios::out | ios::binary);

    }else {

        books.clear();

        while (file.peek() != EOF ) {

            size\_t size;

            file.read((char\*)&size, sizeof(size\_t));

            string str(size, ' ');

            file.read(&str[0], size);

            Book book = deserialize(str);

            books.push\_back(book);

        }

    }

    do {

        cout << "Library Management\n";

        cout << "1. List all books\n";

        cout << "2. Borrow a book\n";

        cout << "3. Return a book\n";

        cout << "4. Exit\n";

        cout << "Enter your selection: ";

        cin >> selection;

        switch(selection) {

            case 1:

                for (int i = 0; i < books.size(); i++) {

                    cout << i + 1 << ". " << books[i].title << (books[i].status ? " (Available)" : " (Borrowed)") << "\n";

                }

                break;

            case 2:

                cin.ignore();

                cout << "Enter book name to borrow: ";

                getline(cin, titleName);

                for (const auto &book : books) {

                    if (book.title == titleName) {

                        int index = &book - &books[0];

                        if (books[index].status) {

                            books[index].status = false;

                            cout << "Book borrowed successfully!\n";

                        } else {

                            cout << "Book is already borrowed.\n";

                        }

                        break;

                    }

                }

                break;

            case 3:

                cin.ignore();

                cout << "Enter book name to return: ";

                getline(cin, titleName);

                for (const auto &book : books) {

                    if (book.title == titleName) {

                        int index = &book - &books[0];

                        if (!books[index].status) {

                            books[index].status = true;

                            cout << "Book returned successfully!\n";

                        } else {

                            cout << "Book wasn't borrowed.\n";

                        }

                        break;

                    }

                }

                break;

            case 4:

                writeEachBookToFile(file,books);

                cout << "Goodbye!";

                file.close();

                return 0;

            default:

                cout << "Invalid selection.\n";

        }

        do {

            cout << "Do you want to perform another operation? (Y/N): ";

            cin >> continueSelection;

        } while (continueSelection != 'Y' && continueSelection != 'N' && continueSelection != 'y' && continueSelection != 'n');

    } while (continueSelection == 'Y' || continueSelection == 'y');

    cout << "Goodbye!";

    writeEachBookToFile(file,books);

    file.close();

    return 0;

}

###### Код до програми № 12 Self Practice 2 | Library

## **5.Результати виконання завдань, тестування та фактично затрачений час:**

#### Завдання №3 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_05\output> & .\'vns\_06\_20.exe'

i am batman

Max length: 6

VNS Lab 6 - Task 1

#### Завдання №4 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_05\output> & .\'vns\_08\_20.exe'

writing countries file...

reading countries file...

Name: Ukraine Language: Ukrainian Currency: Hryvnia Exchange rate: 39.6768

Name: USA Language: English Currency: Dollar Exchange rate: 1

removing country USA

adding a country France

adding a country Spain

writing countries file...

reading countries file...

Name: Ukraine Language: Ukrainian Currency: Hryvnia Exchange rate: 39.6768

Name: France Language: French Currency: Euro Exchange rate: 1

Name: Spain Language: Spanish Currency: Euro Exchange rate:1

VNS Lab 8 - Task 1

#### Завдання №5 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_05\output> & .\'vns\_09\_20.exe'

Say my name my

my name is

dfsdfs

df

ds

fs

df

sd

fsd

f

3

VNS Lab 9 - Task 1

#### Завдання №6 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_05\output> & .\'alg\_04\_v\_01\_STL.exe'

5

1 2 3 4 5

5

4 5 6 7 8

3

1 2 3

3

6 7 8

2

4 5

8

1 2 3 4 5 6 7 8

6

1 2 3 6 7 8

Alfotester Lab 4 Task-1 using STL

#### Завдання №7 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_05\output> & .\'alg\_04\_v\_01.exe'

5

1 2 3 4 5

5

4 5 6 7 8

3

1 2 3

3

6 7 8

2

4 5

8

1 2 3 4 5 6 7 8

6

1 2 3 6 7 8

Alfotester Lab 4 Task - 1

#### Завдання №8 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_05\output> & .\'alg\_06\_v\_01.exe'

5 2

stugna

neptune

grim

oplot

Grim

4

r m i g

Algotester Lab 6 – Task 1

#### Завдання №9 Деталі по виконанню і тестуванню програми

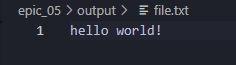


Рисунок 2 created file “file.txt”

PS D:\ED\University\C++\epic\_05\output> & .\'pactice1.exe'

Enter file name: file.txt

Enter content: hello world!

Operation status: Success

Class Practice 1

#### Завдання №10 Деталі по виконанню і тестуванню програми

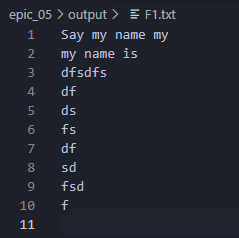


Рисунок 3 file being copied

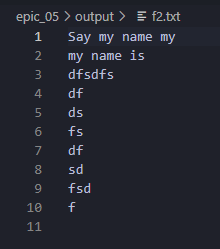


Рисунок 4 copy file

Enter source file name: F1.txt

Enter destination file name: f2.txt

Operation status: Success

Class Practice 2

#### Завдання №12 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_05\output> & .\'seld\_libr.exe'

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 1

1. Harry Potter (Borrowed)

2. The Hobbit (Available)

3. Pride and Prejudice (Available)

4. 1984 (Available)

Do you want to perform another operation? (Y/N): н

Do you want to perform another operation? (Y/N): y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 2

Enter book name to borrow: Harry Potter

Book is already borrowed.

Do you want to perform another operation? (Y/N): N

Goodbye!

PS D:\ED\University\C++\epic\_05\output> & .\'seld\_libr.exe'

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 1

1. Harry Potter (Borrowed)

2. The Hobbit (Available)

3. Pride and Prejudice (Available)

4. 1984 (Available)

Do you want to perform another operation? (Y/N): Y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 3

Enter book name to return: Harry Potter

Book returned successfully!

Do you want to perform another operation? (Y/N): N

Goodbye!

PS D:\ED\University\C++\epic\_05\output> & .\'seld\_libr.exe'

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 1

1. Harry Potter (Available)

2. The Hobbit (Available)

3. Pride and Prejudice (Available)

4. 1984 (Available)

Do you want to perform another operation? (Y/N):

Self Practice | Library

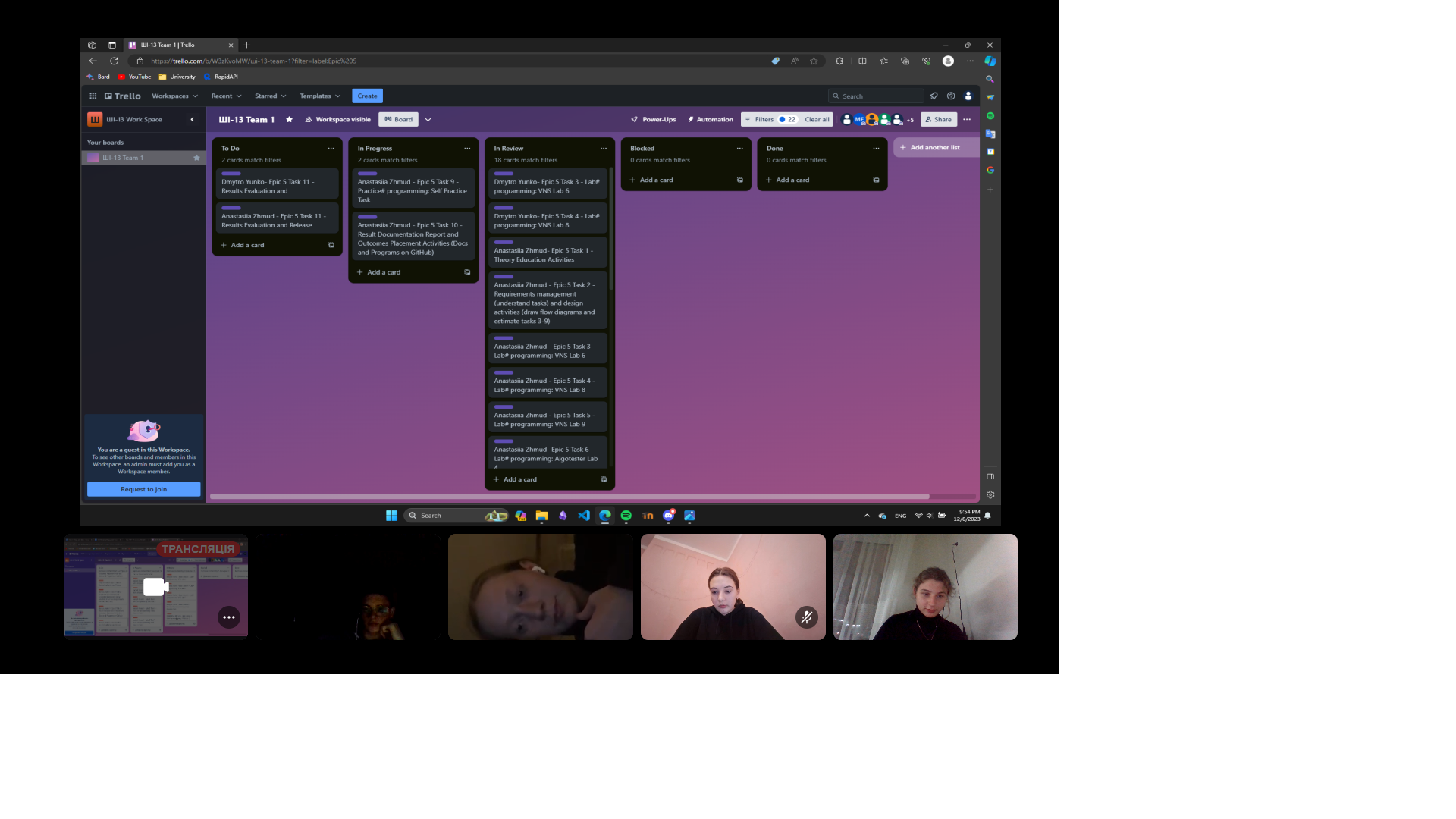


Рисунок 5 team meeting

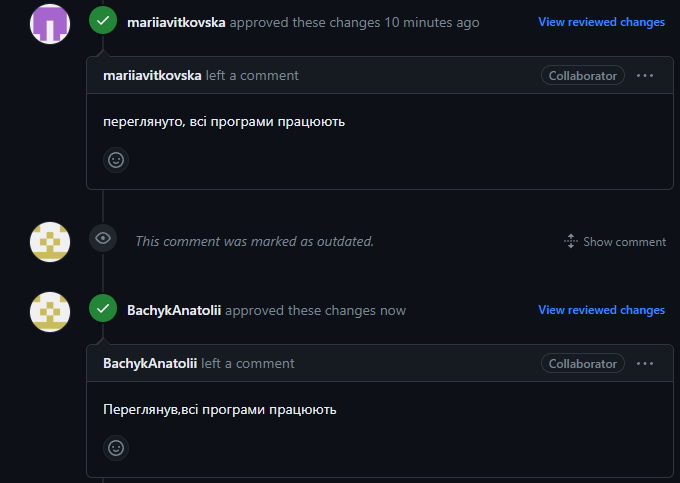


Рисунок 6 ПР звіт

Час затрачений на виконання завдання: 2дн

# **Висновки:**

У цій лабораторній робоиі я дослідив ключові аспекти роботи з файлами в C++, включаючи використання класів і об’єктів для роботи з файлами, формати і режими файлів, перевірку стану файлу, системи числення, запис та використання числових літералів, формати бінарних файлів, серіалізацію та десеріалізацію об’єктів у C++ та перетворення типів покажчиків.