Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 6, 8, 9**

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

***з розділу***: «Файли. Системи числення. Бінарні файли. Символи і рядкові змінні та текстові файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення та використання бібліотек»

***Виконав:***

студент групи ШІ-12

Пастухов Олександр Сергійович

# **Тема роботи:**

Ознайомлення та робота з символьними та рядковими змінними в мові програмування C++. Принцип роботи з бінарними та текстовими файлами. Ознайомлення з типовими алгоритмами роботи з файлами та бібліотеками.

# **Мета роботи:**

Навчитися працювати з символьними та рядковими змінними в мові C++. Вивчити основні алгоритми роботи з файлами та бібліотеками. Застосувати набуті знання на практиці.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Текстові та бінарні файли.
* Тема №2: Основні методи роботи з файлами в C++.
* Тема №3: Стандартна бібліотека в C++.
* Тема №4: Робота з символьними та рядковими змінними.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Текстові та бінарні файли.
  + Джерела Інформації
  + Стаття: <https://kovelpost.com/questions/494>
    - Опрацьовано теоретичний матеріал про основні відмінності між текстовими та бінарними файлами, їх особливості
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.12.2023
  + Звершення опрацювання теми: 22.12.2023
* Тема №2: Основні методи роботи з файлами в C++.
  + Джерела Інформації:

Стаття: <http://www.kytok.org.ua/post/fajly-u-cplusplus-fstream>

* + Що опрацьовано:
    - Вивчено теоретичні основи роботи з файлами та потоками в мові програмування С++
    - Опрацьовано приклади практичного застосування файлів в програмному коді мовою C++
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.12.2023
  + Звершення опрацювання теми: 22.12.2023
* Тема №3: Стандартна бібліотека C++.
  + Джерела Інформації:
    - Стаття: <https://acode.com.ua/urok-204-standartna-biblioteka-shabloniv-stl/>
    - Стаття: <http://om.univ.kiev.ua/users_upload/15/upload/file/pr_lecture_19.pdf>
  + Що опрацьовано:
    - Опрацьовано поняття стандартної бібліотеки мови C++ та її використання
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.12.2023
  + Звершення опрацювання теми: 22.12.2023
* Тема №4: Робота з символьними та рядковими змінними.
  + Джерела Інформації:
    - Стаття: <https://www-simplilearn-com.translate.goog/tutorials/cpp-tutorial/cpp-string?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=uk&_x_tr_pto=sc>
  + Що опрацьовано:
    - Опрацьовано та вивчено основні методи роботи з символьними змінними та рядками
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.12.2023
  + Звершення опрацювання теми: 22.12.2023

# **Виконання роботи:**

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

Завдання №2 VNS Lab 8

* Варіант завдання: 19

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

* Важливі деталі для врахування в імплементації програми:

Дії над файлами та структурами мають виконуватися через функції

Завдання №3 VNS Lab 9

* Варіант завдання: 19
* 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 рядки, починаючи з 4.

2) Підрахувати кількість символів в останньому слові F2.

Завдання №4 Algotester Lab 4

* Варіант завдання: 3
* Деталі завдання: <https://algotester.com/en/ContestProblem/DisplayWithEditor/134641>
* Важливі деталі для врахування в імплементації програми

Завдання виконано без використання засобів STL, а натомість описано власні функції, які є аналогами функцій стандартної бібліотеки

Завдання №5 Algotester Lab 6

* Варіант завдання: 3
* Деталі завдання: <https://algotester.com/en/ContestProblem/DisplayWithEditor/134743>
* Важливі деталі для врахування в імплементації програми

Клітини не можуть містити цифру більшу за розмір поля  
Завдання №6 Class Practice Work Task 1

* Варіант завдання: немає
* Деталі завдання: Запис текстової стрічки у файл із заданим ім’ям

Реалізувати функцію створення файла і запису в нього даних:

enum FileOpResult { Success, Failure, … };

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

Умови задачі:

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

- 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.

* Важливі деталі для врахування в імплементації програми

Використання функції write\_to\_file() та вивід успішного/неуспішного статусу виконання програми

Завдання №7 Class Practice Work Task 2

* Варіант завдання: немає
* Деталі завдання: Копіювання вмісту файла у інший файл

Реалізувати функцію створення файла і запису в нього даних:

enum FileOpResult { Success, Failure, … };

FileOpResult copy\_file(char \*file\_from, char \*file\_to);

Умови задачі:

- 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.

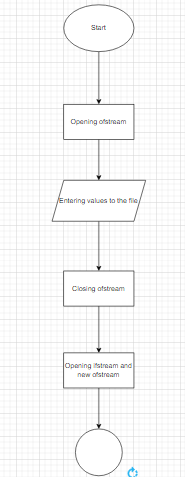
* Важливі деталі для врахування в імплементації програми

Використання функції copy\_file() та вивід успішного/неуспішного статусу виконання програми

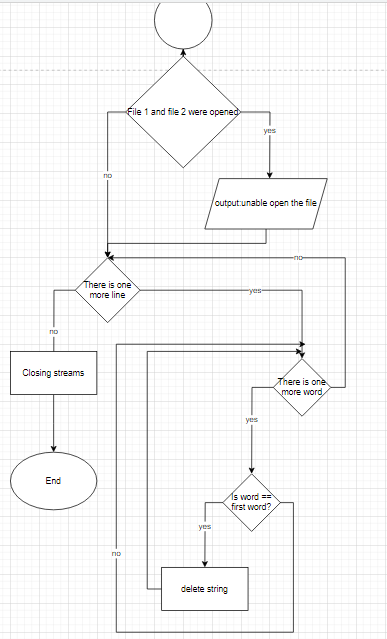
Завдання №8 Self Practice Work

* Варіант завдання: немає

Деталі завдання:Створити програму, яка дозволяє користувачу вводити ім'я та вік, записує ці дані у файл, а потім зчитує дані з файлу та виводить їх на екран.



Малюнок №1,завдання vns - 9



Малюнок №2,завдання vns - 9

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

Програма №1 VNS Lab 8

* Планований час на реалізацію: 80 хвилин

Програма №2 VNS Lab 9

* Планований час на реалізацію: 60 хвилин

Програма №3 Algotester Lab 4

* Планований час на реалізацію: 120 хвилин

Програма №4 Algotester Lab 4.2

* Планований час на реалізацію: 40 хвилин

Програма №5 Algotester Lab 6

* Планований час на реалізацію: 120 хвилин

Програма №6 Class Practice Work Task 1

* Планований час на реалізацію: 40 хвилин

Програма №7 Class Practice Work Task 2

* Планований час на реалізацію: 40 хвилин

Програма №8 Self Practice Work

* Планований час на реалізацію: 40 хвилин

## **3. Конфігурація середовища до виконання завдань:**

Для виконання поставлених завдань додаткова конфігурація середовища не є необхідною

## **4. Код програм з посиланням на зовнішні ресурси:**

Завдання №1 VNS Lab 8

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1260>

#include <iostream>

#include <fstream>

#include <string>

#include <cstring>

struct Movie {

    char title[100];

    char director[100];

    char country[30];

    double revenue;

};

void writeMoviesToFile(char\* filename, int n);

void readMoviesFromFile(char\* filename, int n);

void deleteLastTwoMovies(char\* filename);

void addMovieAfterTitle(char\* filename, char\* targetTitle, Movie& newMovie);

int main() {

using namespace std;

    int n;

    cout << "How many movies do you want to enter? ";

    cin >> n;

cin.ignore();

    writeMoviesToFile("movies.txt", n);

readMoviesFromFile("movies.txt", n);

    deleteLastTwoMovies("movies.txt");

readMoviesFromFile("movies.txt", n);

    Movie newMovie;

    cout << "Enter details for the movie to add after: " << endl;

    cout << "Title: ";

cin.getline(newMovie.title, sizeof(newMovie.title));

    cout << "Director: ";

cin.getline(newMovie.director, sizeof(newMovie.director));

    cout << "Country: ";

cin.getline(newMovie.country, sizeof(newMovie.country));

    cout << "Revenue: $";

cin >> newMovie.revenue;

cin.ignore();

    addMovieAfterTitle("movies.txt", "Target Movie Title", newMovie);

readMoviesFromFile("movies.txt", n);

    return 0;

}

void writeMoviesToFile(char\* filename, int n) {

Movie\* movies = new Movie[n];

    for (int i = 0; i < n; i++) {

        cout << "Enter title: ";

        cin.getline(movies[i].title, sizeof(movies[i].title));

        cout << "Enter director: ";

        cin.getline(movies[i].director, sizeof(movies[i].director));

        cout << "Enter country: ";

        cin.getline(movies[i].country, sizeof(movies[i].country));

        cout << "Enter revenue: $";

        cin >> movies[i].revenue;

        cin.ignore();

}

ofstream outFile(filename, ios::out | ios::binary);

    if (!outFile.is\_open()) {

        cerr << "Failure to open the file for writing." << endl;

        delete[] movies;

        return;

}

outFile.write(reinterpret\_cast<char\*>(movies), n \* sizeof(Movie));

    outFile.close();

    delete[] movies;

}

void readMoviesFromFile(char\* filename, int n) {

Movie\* movies = new Movie[n];

ifstream inFile(filename, ios::in | ios::binary);

    if (!inFile.is\_open()) {

        cerr << "Failure to open the file for reading." << endl;

        delete[] movies;

        return;

}

inFile.read(reinterpret\_cast<char\*>(movies), n \* sizeof(Movie));

    for (int i = 0; i < n; i++) {

        cout << "Title: " << movies[i].title << endl;

        cout << "Director: " << movies[i].director << endl;

        cout << "Country: " << movies[i].country << endl;

        cout << "Revenue: $" << movies[i].revenue << endl;

        cout << "------------------------" << endl;

}

    inFile.close();

    delete[] movies;

}

void deleteLastTwoMovies(char\* filename) {

    ifstream inFile(filename, ios::in | ios::binary);

ofstream tempFile("temp.bin", ios::out | ios::binary);

    if (!inFile.is\_open() || !tempFile.is\_open()) {

        cerr << "Failure to open the file for reading/writing." << endl;

        return;

}

    int totalMovies = 0;

Movie currentMovie;

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

        totalMovies++;

}

    inFile.clear();

inFile.seekg(0, ios::beg);

    for (int i = 0; i < totalMovies - 2; i++) {

        inFile.read(reinterpret\_cast<char\*>(&currentMovie), sizeof(Movie));

        tempFile.write(reinterpret\_cast<char\*>(&currentMovie), sizeof(Movie));

}

    inFile.close();

tempFile.close();

    remove(filename);

    rename("temp.bin", filename);

}

void addMovieAfterTitle(char\* filename, char\* targetTitle, Movie& newMovie) {

    ifstream inFile(filename, ios::in | ios::binary);

ofstream tempFile("temp.bin", ios::out | ios::binary);

    if (!inFile.is\_open() || !tempFile.is\_open()) {

        cerr << "Failure to open the file for reading/writing." << endl;

        return;

}

    Movie currentMovie;

bool targetFound = false;

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

        tempFile.write(reinterpret\_cast<char\*>(&currentMovie), sizeof(Movie));

        if (strcmp(currentMovie.title, targetTitle) == 0) {

            tempFile.write(reinterpret\_cast<char\*>(&newMovie), sizeof(Movie));

            targetFound = true;

        }

}

    inFile.close();

tempFile.close();

    if (!targetFound) {

        cerr << "Target movie not found." << endl;

        remove("temp.bin");

        return;

}

    remove(filename);

    rename("temp.bin", filename);

}

Завдання №2 VNS Lab 9

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

int main() {

using namespace std;

    ofstream f1("F1.txt");

    f1 << "apple orange banana" << endl;

    f1 << "apple peach kiwi" << endl;

    f1 << "grape mango pear" << endl;

    f1 << "cherry apple plum" << endl;

    f1 << "banana grape orange" << endl;

    f1 << "kiwi cherry mango" << endl;

    f1 << "pear peach apple" << endl;

    f1 << "plum mango kiwi" << endl;

    f1 << "orange banana cherry" << endl;

    f1 << "mango pear grape" << endl;

f1.close();

    ifstream f1Read("F1.txt");

ofstream f2("F2.txt");

    if(!f1Read.is\_open() || !f2.is\_open()) {

        cerr << "Unable to open the file." << endl;

        return 1;

}

    string firstWord;

getline(f1Read, firstWord);

    string line;

    while(getline(f1Read, line)) {

        istringstream iss(line);

        string word;

        bool containsFirstWord = false;

        while(iss >> word) {

            if (word == firstWord) {

                containsFirstWord = true;

                break;

            }

        }

        if(!containsFirstWord) {

            f2 << line << endl;

        }

}

    f1Read.close();

f2.close();

    return 0;

}

Завдання №3 Algotester Lab 4

#include <iostream>

#include <vector>

#include <algorithm>

bool comparator(int a, int b) {

    return (a > b);

}

int main() {

using namespace std;

    int N;

    cin >> N;

vector<int> rem0, rem1, rem2;

    for (int i = 0; i < N; i++) {

        int currentElem;

        cin >> currentElem;

        (currentElem % 3 == 0) ? rem0.push\_back(currentElem) : (currentElem % 3 == 1) ? rem1.push\_back(currentElem) : rem2.push\_back(currentElem);

}

    sort(rem0.begin(), rem0.end());

    sort(rem1.begin(), rem1.end(), comparator);

sort(rem2.begin(), rem2.end());

    vector<int> result;

    result.insert(result.end(), rem0.begin(), rem0.end());

    result.insert(result.end(), rem1.begin(), rem1.end());

result.insert(result.end(), rem2.begin(), rem2.end());

    result.erase(unique(result.begin(), result.end()), result.end());

//   unordered\_set<int> uniqSet(result.begin(), result.end());

//   result.assign(uniqSet.begin(), uniqSet.end());

    cout << result.size() << endl;

    for (int number : result) {

        cout << number << " ";

}

    return 0;

}

Завдання №4 Algotester Lab 4.2

#include <iostream>

#include <vector>

void customQuickSort(std::vector<int>& numbers, int left, int right) {

    if (left < right) {

        int pivot = numbers[left];

        int i = left + 1;

        int j = right;

        while (i <= j) {

            while (i <= j && numbers[i] <= pivot) {

                i++;

            }

            while (i <= j && numbers[j] > pivot) {

                j--;

            }

            if (i < j) {

                std::swap(numbers[i], numbers[j]);

            }

        }

        std::swap(numbers[left], numbers[j]);

        customQuickSort(numbers, left, j - 1);

        customQuickSort(numbers, j + 1, right);

    }

}

void customSort(std::vector<int>& numbers) {

    customQuickSort(numbers, 0, numbers.size() - 1);

}

void reverseSort(std::vector<int>& numbers) {

customQuickSort(numbers, 0, numbers.size() - 1);

    int i = 0;

int j = numbers.size() - 1;

    while (i < j) {

        std::swap(numbers[i], numbers[j]);

        i++;

        j--;

    }

}

void removeDuplicates(std::vector<int>& numbers) {

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

        for (int j = i + 1; j < numbers.size(); ++j) {

            if (numbers[i] == numbers[j]) {

                numbers.erase(numbers.begin() + j);

                --j;

            }

        }

    }

}

int main() {

using namespace std;

    int N;

    cin >> N;

vector<int> rem0, rem1, rem2;

    for (int i = 0; i < N; i++) {

        int currentElem;

        cin >> currentElem;

        if (currentElem % 3 == 0) {

            rem0.push\_back(currentElem);

        } else if (currentElem % 3 == 1) {

            rem1.push\_back(currentElem);

        } else {

            rem2.push\_back(currentElem);

        }

}

    customSort(rem0);

    reverseSort(rem1);

customSort(rem2);

    vector<int> result = rem0;

    result.insert(result.end(), rem1.begin(), rem1.end());

result.insert(result.end(), rem2.begin(), rem2.end());

removeDuplicates(result);

    cout << result.size() << endl;

    for (int num : result) {

        cout << num << " ";

}

    return 0;

}

Завдання №5 Algotester Lab 6

#include <iostream>

#include <vector>

#include <algorithm>

std::vector<int> getAvailableValues(const std::vector<std::vector<int>>& board, int row, int col) {

    int boardSize = board.size();

std::vector<int> result;

    for (int i = 1; i <= boardSize; i++) {

        result.push\_back(i);

}

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

        auto it = find(result.begin(), result.end(), board[i][col]);

        if (it != result.end()) {

            result.erase(it);

        }

}

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

        auto it = find(result.begin(), result.end(), board[row][i]);

        if (it != result.end()) {

            result.erase(it);

        }

}

    return result;

}

int main() {

using namespace std;

    int boardSize;

    cin >> boardSize;

vector<vector<int>> board;

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

        vector<int> currentRow;

        string line;

        cin >> line;

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

            int currentNumber = line[j] - '0';

            currentRow.emplace\_back(currentNumber);

        }

        board.emplace\_back(currentRow);

}

    int coordinatesCount;

    cin >> coordinatesCount;

vector<pair<int, int>> coordinates;

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

        int x, y;

        cin >> x >> y;

        coordinates.emplace\_back(x - 1, y - 1);

    }

    for (auto coord : coordinates) {

        if (board[coord.first][coord.second] != 0) {

            cout << "1" << endl << board[coord.first][coord.second] << "\n\n";

        } else {

            vector<int> availableValues = getAvailableValues(board, coord.first, coord.second);

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

            for (int value : availableValues) {

                cout << value << " ";

            }

            cout << endl;

        }

}

    return 0;

}

Завдання №6 Class Practice Work Task 1

#include <iostream>

#include <fstream>

enum FileOpResult {success, failure};

FileOpResult writeToFile(char\* fileName,char\* content) {

std::ofstream file(fileName);

    if (!file.is\_open() || file.fail()) {

        return failure;

}

file << content;

    return success;

}

int main() {

using namespace std;

    int fileNameSize = 15;

int contentSize = 250;

    char fileName[fileNameSize];

char content[contentSize];

    cout << "Enter the name of the file: ";

cin.getline(fileName, sizeof(content));

    cout << "Enter your content: ";

    cin.ignore();

cin.getline(content, sizeof(content));

FileOpResult result = writeToFile(fileName, content);

    if (result == success) {

        cout << "Your file is successfully created!" << endl;

    } else {

        cout << "ERROR" << endl;

}

    return 0;

}

Завдання №7 Class Practice Work Task 2

#include <iostream>

#include <fstream>

using namespace std;

enum FileOpResult { Success, Failure };

FileOpResult copy\_file(char\* file\_from, char\* file\_to);

int main()

{

    char\* name\_to;

    char\* name\_from;

    cout << "Enter file to copy: ";

cin >> name\_from;

    cout << "Enter destination: ";

cin >> name\_to;

FileOpResult result = copy\_file(name\_from, name\_to);

    if (result == FileOpResult::Success)

    {

        cout << "Success" << endl;

    } else {

        cout << "Failure" << endl;

}

    return 0;

}

FileOpResult copy\_file(char\* file\_from, char\* file\_to)

{

    ifstream file1(file\_from);

    if (!file1.is\_open())

    {

        return FileOpResult::Failure;

}

ofstream file2(file\_to);

    if (!file2.is\_open())

    {

        file1.close();

        return FileOpResult::Failure;

}

file2 << file1.rdbuf();

    if (file2.fail())

    {

        file1.close();

        file2.close();

        return FileOpResult::Failure;

    }

    return FileOpResult::Success;

}

Завдання №8 Self Practice Work

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct Person {

    string name;

    int age;

};

void writeToTextFile(const string& filename, const Person& person) {

ofstream file(filename);

    if (file.is\_open()) {

        file << person.name << " " << person.age;

        file.close();

        cout << "Дані успішно записано у файл." << endl;

    } else {

        cout << "Помилка відкриття файлу для запису." << endl;

    }

}

void readFromTextFile(const string& filename) {

ifstream file(filename);

    if (file.is\_open()) {

        Person person;

        file >> person.name >> person.age;

        cout << "Ім'я: " << person.name << endl <<"Вік: " << person.age << endl;

        file.close();

    } else {

        cout << "Помилка відкриття файлу для читання." << endl;

    }

}

int main() {

Person person;

    cout << "Введіть ваше ім'я: ";

getline(cin, person.name);

    cout << "Введіть ваш вік: ";

cin >> person.age;

writeToTextFile("person\_data.txt", person);

    cout << endl << "Дані, зчитані з файлу:" << endl;

readFromTextFile("person\_data.txt");

    return 0;

}

if (first\_file\_line.length() != second\_file\_line.length())

{

std::cout << "Files have lines with different length" << endl;

return 0;

}

}

std::cout << "All lines in files have the same length" << endl;

F1\_.close();

F2\_.close();

return 0;

}

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

Завдання №1 VNS Lab 8

How many movies do you want to enter? 2

Enter title: King-kong

Enter director: Piter

Enter country: USA

Enter revenue: $1000000

Enter title: Atlantic

Enter director: James

Enter country: Ukraine

Enter revenue: $100500

Title: King-kong

Director: Piter

Country: USA

Revenue: $1e+06

------------------------

Title: Atlantic

Director: James

Country: Ukraine

Revenue: $100500

------------------------

Enter details for the movie to add after:

Title: Aladin

Director: Lopes

Country: USA

Revenue: $70000

Час затрачений на виконання завдання: 150 хвилин

Завдання №2 VNS Lab 9

Час затрачений на виконання завдання: 80 хвилин

Завдання №3 Algotester Lab 4 , Завдання№4 Algotester Lab 4.2

10

1 33 4 8 6 5 2 7 5 0

9

0 6 33 7 4 1 2 5 8

Час затрачений на виконання завдання: 180 хвилин

Завдання №5 Algotester Lab 6

Inp:

3

000

100

003

3

1 1

2 3

2 1

Out:

2

2 3

1

2

1

1

*Рисунок 3. Результат зарахування програми №5 на Алготестері*

Час затрачений на виконання завдання: 250+ хвилин

Завдання №6 Class Practice Work Task 1

Enter the name of the file: test

Enter your content: i want to eat and sleep

Your file is successfully created!

Час затрачений на виконання завдання: 50 хвилин

Завдання №7 Class Practice Work Task 2

Enter file to copy: aaa.txt

Enter destination: bbb.txt

Failure

Час затрачений на виконання завдання: 120 хвилин

Завдання №8 Class Practice Work 8

Введіть ваше ім'я: Alex

Введіть ваш вік: 18

Дані успішно записано у файл.

Дані, зчитані з файлу:

Ім'я: Alex

Вік: 18

Час затрачений на виконання завдання: 60 хвилин

# **Висновки:**

Під час опрацювання теоретичного матеріалу та роботи над завданнями розділу я навчився працювати з файлами, рядками та символьними елементами в мові C++. Весь пройдений матеріал закріплено практично завдяки виконанні лабораторних та практичних робіт.