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

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

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт №5

***Виконав:***

студент групи ШІ-12

Пушак Владислав Русланович

# **Тема роботи:**

Файли. Системи числення. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек

# **Мета роботи:**

Дослідження та засвоєння концепцій та методів роботи з файлами, системами числення, бінарними файлами, символами і рядковими змінними, а також текстовими файлами в програмуванні.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Бінарні та Текстові Файли в C++
* Тема №2: Символи і Рядкові Змінні в С++

1. Індивідуальний план опрацювання теорії:

* Тема №1: Тема №1: Бінарні та Текстові Файли в C++
  + Джерела Інформації
    - <https://www.youtube.com/watch?v=CBnB2fvfu_I&t=922s>
    - Що опрацьовано:
    - Ознайомився з бібліотекою fstream, та навчився з нею працювати
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.12.2023
  + Звершення опрацювання теми: 17.12.2023
* Тема №2: Символи і Рядкові Змінні в С++
  + Джерела Інформації
    - Попередній досвід та лекції
  + Що опрацьовано:
    - Ознайомився зі змінними char і string та методам роботи з ними, навчився працювати з getline()
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.12.2023
  + Звершення опрацювання теми: 17.12.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 6 - Task 1

* Варіант завдання 11
* Перетворити рядок таким чином, щоб всі слова в ньому були надруковані

навпаки.

Завдання №2 VNS Lab 8 – Task 1

* Варіант завдання 11

Структура "Відеокасета":

- назва фільму;

- режисер;

- тривалість;

- ціна.

Знищити всі елементи із ціною вищою заданої, додати 3 елементи в кінець

файлу.

Завдання №3 VNS Lab 9 - Task 1-N

* Варіант завдання 11
* 1) Скопіювати з файлу F1 у файл F2 всі рядки, крім того рядка, що містить
* найкоротше слово.
* 2) Надрукувати номер цього рядка.

Завдання №4 Algotester Lab 4

* Варіант завдання 1
* Вам дано 2 цілих чисел масиви, розміром
* Ваше завдання вивести:
* 1. Різницю N-M
* 2. Різницю M-N
* 3. Їх перетин
* 4. Їх обєднання
* 5. Їх симетричну різницю

Завдання №5 Algotester Lab 6

* Варіант завдання 1
* Перечислити букви в словах, які зустрічаються в тексті більше-рівне ніж K разів

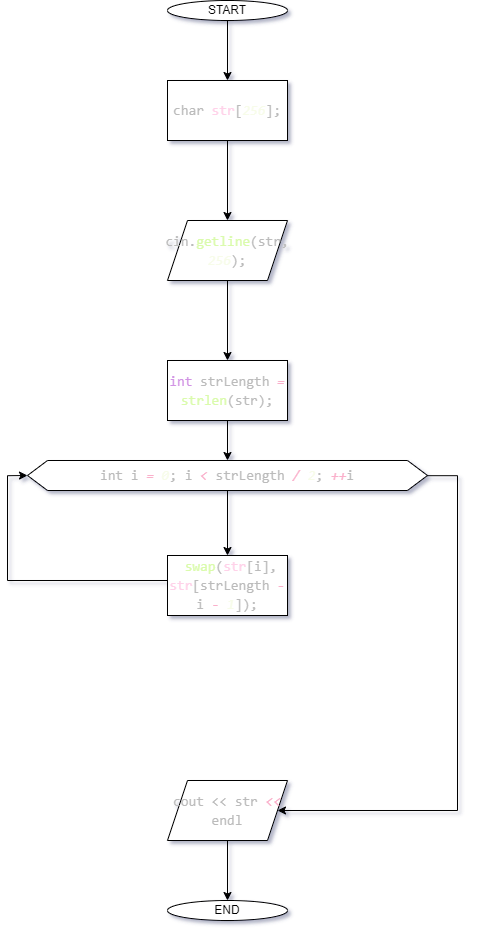
Завдання №6 Class Practice Work - Task 1-N

* Робота з текстовими файлами

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

Програма №1 VNS Lab 6 - Task 1-N

* Блок-схема



* Планований час на реалізацію 1 год

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

Завдання №1 VNS Lab 6 - Task 1-N

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1098/files#diff-f28cebf8dd030bb926596505966de30665dfb1e09db9139a0488d7945d356617>

*#include* <iostream>

*#include* <cstring>

*using* *namespace* std;

int main() {

    char str[*256*];

    cin.getline(str, *256*);

    int strLength *=* strlen(str);

*for* (int i *=* *0*; i *<* strLength */* *2*; *++*i) {

        swap(str[i], str[strLength *-* i *-* *1*]);

    }

    c;

*return* *0*;

}

Завдання №2 VNS Lab 8 - Task 1-N

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1098/files#diff-dee9cd538b802481010675feb5d5c05689a0537834191b54775f5f5397eb74ff>

*#include* <stdio.h>

*typedef* struct {

    char title[*50*];

    char director[*50*];

    int duration;

    float price;

} VideoCassette;

VideoCassette newCassettesToAdd[] *=* {

    {"The Gentlemen", "Guy Ritchie", *113*, *120.0*},

    {"John Wick", "Chad Stahelski", *101*, *110.0*},

    {"Men in Black", "Barry Sonnenfeld", *98*, *135.0*}

};

VideoCassette cassettes[] *=* {

    {"Django Unchained", "Quentin Tarantino", *165*, *150.0*},

    {"Inglourious Basterds", "Quentin Tarantino", *153*, *140.0*},

    {"TENET", "Christopher Nolan", *150*, *160.0*}

};

void printContents(const char \*filename) {

    FILE *\**file;

    VideoCassette tapes[*10*];

    int numTapes *=* *0*;

*if* ((file *=* fopen(filename, "rb")) *==* NULL) {

        perror("\nError opening file");

*return*;

    }

*while* (fread(*&*tapes[numTapes], *sizeof*(VideoCassette), *1*, file) *==* *1*) {

        numTapes*++*;

    }

*for* (int i *=* *0*; i *<* numTapes; i*++*) {

        printf("Title: %s\n", tapes[i].title);

        printf("Director: %s\n", tapes[i].director);

        printf("Duration: %d\n", tapes[i].duration);

        printf("Price: %.2f\n", tapes[i].price);

        printf("\n");

    }

    fclose(file);

}

void removeElements(const char \*filename, float maxPrice) {

    FILE *\**file;

    VideoCassette tapes[*10*];

    int numTapes *=* *0*;

*if* ((file *=* fopen(filename, "r+b")) *==* NULL) {

        perror("\nError opening file");

*return*;

    }

*while* (fread(*&*tapes[numTapes], *sizeof*(VideoCassette), *1*, file) *==* *1*) {

*if* (tapes[numTapes].price *<=* maxPrice) {

            numTapes*++*;

        }

    }

    rewind(file);

    fwrite(tapes, *sizeof*(VideoCassette), numTapes, file);

    fclose(file);

}

void addElements(const char \*filename, VideoCassette \*addedTapes, int numNewTapes) {

    FILE *\**file;

*if* ((file *=* fopen(filename, "ab")) *==* NULL) {

        perror("\nError opening file");

*return*;

    }

    fwrite(addedTapes, *sizeof*(VideoCassette), numNewTapes, file);

    fclose(file);

}

int main() {

    FILE *\**file;

*if* ((file *=* fopen("videotapes.bin", "wb")) *==* NULL) {

        perror("\nError opening file");

*return* *1*;

    }

    VideoCassette tapes[] *=* {

        {"Django Unchained", "Quentin Tarantino", *165*, *150.0*},

        {"Inglourious Basterds", "Quentin Tarantino", *153*, *140.0*},

        {"TENET", "Christopher Nolan", *150*, *160.0*}

    };

    fwrite(tapes, *sizeof*(VideoCassette), *sizeof*(tapes) */* *sizeof*(VideoCassette), file);

    fclose(file);

    printf("Contents before removal:\n");

    printContents("videotapes.bin");

    float maxPrice *=* *145.0*;

    removeElements("videotapes.bin", maxPrice);

    printf("\nContents after removal price higher than %.2f:\n", maxPrice);

    printContents("videotapes.bin");

    VideoCassette addedTapes[] *=* {

        {"The Gentlemen", "Guy Ritchie", *113*, *120.0*},

        {"John Wick", "Chad Stahelski", *101*, *110.0*},

        {"Men in Black", "Barry Sonnenfeld", *98*, *135.0*}

    };

    addElements("videotapes.bin", addedTapes, *sizeof*(addedTapes) */* *sizeof*(VideoCassette));

    printf("\nContents after addition:\n");

    printContents("videotapes.bin");

    fclose(file);

*return* *0*;

}

Завдання №3 VNS Lab 9 - Task 1-N  
<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1098/files#diff-adec8710e19438e621ca6f68e728c387b2a91db1de96c36c74d5e2a25df4298d>

*#include* <stdio.h>

*#include* <stdlib.h>

*#include* <string.h>

*#define* MAXLENGTH *255*

int main() {

    FILE *\**file1, *\**file2;

    char buffer[MAXLENGTH];

    char minWord[MAXLENGTH];

    int minLine *=* *-1*;

    int curLine *=* *0*;

    file1 *=* fopen("file1.txt", "r");

*if* (file1 *==* NULL) {

        printf("File file1.txt does not exist. Creating new...\n");

        file1 *=* fopen("file1.txt", "w");

*if* (file1 *==* NULL) {

            perror("Failed to create file file1:");

*return* *1*;

        }

        fprintf(file1, "Sunny day.\n");

        fprintf(file1, "Green tree.\n");

        fprintf(file1, "Happy cat.\n");

        fprintf(file1, "Tasty coffee.\n");

        fclose(file1);

        printf("File file1.txt successfully created.\n");

*return* *0*;

    }

*while* (fgets(buffer, MAXLENGTH, file1) *!=* NULL) {

        curLine*++*;

        char *\**word *=* strtok(buffer, " \t\n");

*if* (word *!=* NULL *&&* (strlen(word) *<* strlen(minWord) *||* minLine *==* *-1*)) {

            strcpy(minWord, word);

            minLine *=* curLine;

        }

    }

    rewind(file1);

    file2 *=* fopen("file2.txt", "w");

*if* (file2 *==* NULL) {

        perror("Failed to open file file2");

        fclose(file1);

*return* *1*;

    }

    curLine *=* *0*;

*while* (fgets(buffer, MAXLENGTH, file1) *!=* NULL) {

        curLine*++*;

*if* (curLine *!=* minLine) {

            fputs(buffer, file2);

        }

    }

    printf("Row with the shortest word: %d\n", minLine);

    fclose(file1);

    fclose(file2);

*return* *0*;

}

Завдання №4 Algotester Lab 4

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1098/files#diff-9cab5b252f772fa93620385931e04c26a7bd007422f17edbcf37557d5625e7dc>

*#include* <iostream>

*#include* <algorithm>

*#include* <vector>

void printVector(const std::vector<int>& vec) {

    std::cout *<<* vec.size() *<<* std::endl;

*for* (const auto*&* el : vec) {

        std::cout *<<* el *<<* " ";

    }

    std::cout *<<* std::endl;

}

int main() {

*using* *namespace* std;

    int N, M;

    cin *>>* N;

    vector*<*int*>* arrayN(N);

*for* (auto*&* el : arrayN) {

        cin *>>* el;

    }

    cin *>>* M;

    vector*<*int*>* arrayM(M);

*for* (auto*&* el : arrayM) {

        cin *>>* el;

    }

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

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

    vector*<*int*>* differenceNM;

    set\_difference(arrayN.begin(), arrayN.end(), arrayM.begin(), arrayM.end(), back\_inserter(differenceNM));

    printVector(differenceNM);

    vector*<*int*>* differenceMN;

    set\_difference(arrayM.begin(), arrayM.end(), arrayN.begin(), arrayN.end(), back\_inserter(differenceMN));

    printVector(differenceMN);

    vector*<*int*>* intersection;

    set\_intersection(arrayN.begin(), arrayN.end(), arrayM.begin(), arrayM.end(), back\_inserter(intersection));

    printVector(intersection);

    vector*<*int*>* Union;

    set\_union(arrayN.begin(), arrayN.end(), arrayM.begin(), arrayM.end(), back\_inserter(Union));

    printVector(Union);

    vector*<*int*>* symmetricDifference;

    set\_symmetric\_difference(arrayN.begin(), arrayN.end(), arrayM.begin(), arrayM.end(), back\_inserter(symmetricDifference));

    printVector(symmetricDifference);

}

Завдання №5 Algotester Lab 6  
<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1098/files#diff-ce15f305d3c4a5d326241e0b4f2d252a7dd3303aac0f2e2f4d88eb96bf1abc22>

*#include* <iostream>

*#include* <vector>

*#include* <algorithm>

*#include* <map>

*#include* <set>

*using* *namespace* std;

void printVector(const vector<char>& vec) {

    cout *<<* vec.size() *<<* endl;

*for* (const auto*&* el : vec) {

        cout *<<* el *<<* " ";

    }

    cout *<<* endl;

}

int main() {

    int numWords, threshold;

    cin *>>* numWords *>>* threshold;

    map*<*string, int*>* wordFrequency;

    set*<*string*>* frequentWords;

*for* (int i *=* *0*; i *<* numWords; *++*i) {

        string word;

        cin *>>* word;

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

        wordFrequency[word]*++*;

    }

*for* (const auto*&* pair : wordFrequency) {

*if* (pair.second *>=* threshold) {

            frequentWords.insert(pair.first);

        }

    }

    vector*<*char*>* uniqueChars;

*if* (*!*frequentWords.empty()) {

*for* (const auto*&* word : frequentWords) {

*for* (const auto*&* ch : word) {

                uniqueChars.push\_back(ch);

            }

        }

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

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

        reverse(uniqueChars.begin(), uniqueChars.end());

        printVector(uniqueChars);

    } *else* {

        cout *<<* "Empty!" *<<* endl;

    }

*return* *0*;

}

Завдання №6 Class Practice Work - Task 1-N  
<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1098/files#diff-ecab7a7e7687d2bd6a98a87a4fbb24c6032b227ac569f5f8d7593f3d1efe6f71>

*#include* <iostream>

*#include* <fstream>

*#include* <string>

*using* *namespace* std;

enum Status { Ok, Error };

Status saveToFile(string filename, string data) {

    ofstream outFile(filename);

*if* (*!*outFile) {

*return* Status::Error;

    }

    outFile *<<* data;

    outFile.close();

*if* (*!*outFile) {

*return* Status::Error;

    }

*return* Status::Ok;

}

Status duplicateFile(string original, string copy) {

    ifstream inFile(original);

*if* (*!*inFile) {

*return* Status::Error;

    }

    ofstream outFile(copy);

*if* (*!*outFile) {

        inFile.close();

*return* Status::Error;

    }

    outFile *<<* inFile.rdbuf();

    inFile.close();

    outFile.close();

*if* (*!*inFile *||* *!*outFile) {

*return* Status::Error;

    }

*return* Status::Ok;

}

int main() {

    string file, text, orig, dup;

    cin *>>* file;

    cin.ignore();

    getline(cin, text);

*if* (saveToFile(file, text) *==* Status::Ok) {

        cout *<<* "Success" *<<* endl;

    } *else* {

        cout *<<* "Failure" *<<* endl;

    }

    cin *>>* orig;

    cin *>>* dup;

*if* (duplicateFile(orig, dup) *==* Status::Ok) {

         cout *<<* "Success" *<<* endl;

    } *else* {

        cout *<<* "Failure" *<<* endl;

    }

*return* *0*;

}

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

Завдання №1 VNS Lab 6 - Task 1-N



Результат 1

Час затрачений на виконання завдання 1 год

Завдання №2 VNS Lab 8 - Task 1-N

*Зображення, що містить текст, знімок екрана, меню

Автоматично згенерований опис*

Результат 2

Час затрачений на виконання завдання 3 год

Завдання №3 VNS Lab 9 - Task 1-N

Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис

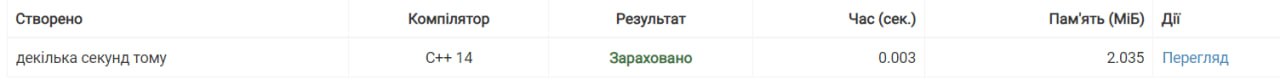
Результат 3

Час затрачений на виконання завдання 1.5 год

Завдання №4 Algotester Lab 4

Зображення, що містить знімок екрана, текст, Шрифт, дизайн

Автоматично згенерований опис

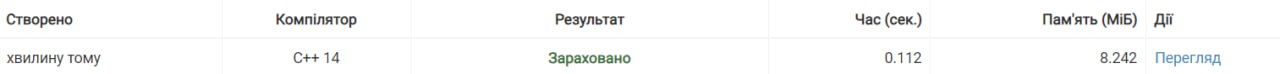
**

Результат 4

Час затрачений на виконання завдання 1 год

Завдання №5 Algotester Lab 6

*Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис*

Результат 5

Час затрачений на виконання завдання 2 год

# **Висновки:**

Я досліджував нові можливості мови програмування C++, зокрема роботу з текстовими і бінарними файлами. Узявся за вивчення методів роботи з типами даних char та string, і закріпив ці знання через важкі завдання на алготестері та роботу з відповідними лабораторними. Отримав досвід використання бібліотек fstream, set та роботи з рядками.