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 й використання бібліотек\_\_\_\_\_\_\_\_\_\_»

***Виконав:***

студент групи ШІ-13

Кібиш Тарас Юрійович

# **Тема роботи:**

Здобування нових навичок в c++. Знайомство з новими засобами STL. Знайомсто з структурами. Знайомство з роботою з рядками і файлами.

# **Мета роботи:**

Здобути професійні навички в роботі з C++. Навчитись використовувати нові знання на практиці.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Системи числення
* Тема №2: Структури і їхній синтаксис
* Тема №3: Засоби STL
* Тема №4: Ввід і вивід рядків
* Тема №5: Робота з файлами

1. Індивідуальний план опрацювання теорії:

* Тема №1: Системи числення
  + Джерела Інформації
    - <https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F>
    - Відео.
    - Стаття.
    - Курс.
  + Що опрацьовано:
    - Отримана інформація про різні види систем числення
  + Статус: Ознайомлений
  + Початок опрацювання теми: 05.12.2023
  + Звершення опрацювання теми: 08.12.2023
* Тема №2: Структури і їхній синтаксис
  + Джерела Інформації:
    - <https://www.youtube.com/watch?v=999IE-6b7_s&t=1875s&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>
    - Відео.
    - Стаття.
    - Курс.
  + Що опрацьовано:
    - Отримано базові поннятя, що таке структури
    - Коментар 2
  + Статус: Ознайомлений
  + Початок опрацювання теми: 03.12.2023
  + Звершення опрацювання теми: 05.12.2023
* Тема №3: Засоби STL
  + Джерела Інформації:
    - <https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%BE%D0%B2>
    - <https://ravesli.com/urok-199-algoritmy-stl/>
    - Курс.
  + Що опрацьовано:
    - Отримана інформація про роботу з STL
    - Коментар 2
  + Статус: Ознайомлений
  + Початок опрацювання теми: 05.12.2023
  + Звершення опрацювання теми: 08.12.2023
* Тема №4: Ввід і вивід рядків
  + Джерела Інформації:
    - <https://cplusplus.com/reference/string/string/getline/>
    - Відео.
    - Стаття.
    - Курс.
  + Що опрацьовано:
    - Ознайомлений з роботою getline

Статус: Ознайомлений

* + Початок опрацювання теми: 03.12.2023
  + Звершення опрацювання теми: 07.12.2023
* Тема №5 Робота з файлами
  + Джерела Інформації:
    - <https://www.youtube.com/watch?v=FeNqHytI0fA&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>
    - <https://www.youtube.com/watch?v=SSNJ7alki-E&t=2112s&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD%E2%A6%81%D0%A3%D1%80%D0%BE%D0%BA%D0%B8%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>
  + Що опрацьовано:
    - Вивчено як працювати з файлами
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.12.2023
  + Звершення опрацювання теми: 08.12.2023

# **Виконання роботи:**

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

Завдання №1: Lab# programming: VNS Lab 6

* Варіант завдання 12
* Деталі завдання
* Перетворити рядок таким чином, щоб букви кожного слова в ньому були
* відсортовані за зростанням.
* Важливі деталі для врахування в імплементації програми

Завдання №2: Lab# programming: VNS Lab 8

* Варіант завдання
* Деталі завдання
* Структура "Музичний диск":
* - назва;
* - автор;
* - тривалість;
* - ціна.
* Знищити перший елемент із заданою тривалістю, додати 2 елементи після
* елемента із заданим номером.
* Важливі деталі для врахування в імплементації програми
* Завдання №3: Lab# programming: VNS Lab 9
* Варіант завдання
* 1) Скопіювати з файлу F1 у файл F2 всі рядки, крім того рядка, у якій
* найбільше голосних букв.
* 2) Надрукувати номер цього рядка.
* Деталі завдання
* Важливі деталі для врахування в імплементації програми

Завдання №4: Lab# programming: Algotester Lab 4

* Варіант завдання 2
* Деталі завдання
* 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.
* Важливі деталі для врахування в імплементації програми

Завдання №5 Lab# programming: Algotester Lab 6

* Варіант завдання 2
* Деталі завдання

У вас є шахова дошка розміром 8×88×8 та дуже багато фігур.

Кожна клітинка може мати таке значення:

* Пуста клітинка O
* Пішак P
* Тура R
* Кінь N
* Слон B
* Король K
* Королева Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1).

Далі йдуть Q запитів з координатами клітинки {x,y}. На кожен запит ви маєте вивести стрічку si - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз).

У випадку, якщо на клітинці стоїть якась фігура - виведіть символ X.

У випадку, якщо клітинку не атакують - виведіть O.

Наявніть фігури у певній клітинці не блокує атаку для іншої фігури. Тобто якщо між турою та клітинкою стоїть інша фігура - вважається що тура атакує цю клітинку.

* Важливі деталі для врахування в імплементації програми
* |rowi|=N
* rowi∈{O,P,R,N,B,K,Q}
* 1≤Q≤64
* 1≤x,y≤8

Завдання №:6 Practice# programming: Class Practice Task

* Варіант завдання
* Деталі завдання
* ***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
* -       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.
* ***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

Завдання №7: Practice# programming:  Self Practice Task

* Варіант завдання
* Деталі завдання
* Створити програму, яка обчислює мінімальну кількість мін для заданого поля
* Важливі деталі для врахування в імплементації програми

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

Програма №6: Lab# programming: Algotester Lab 6

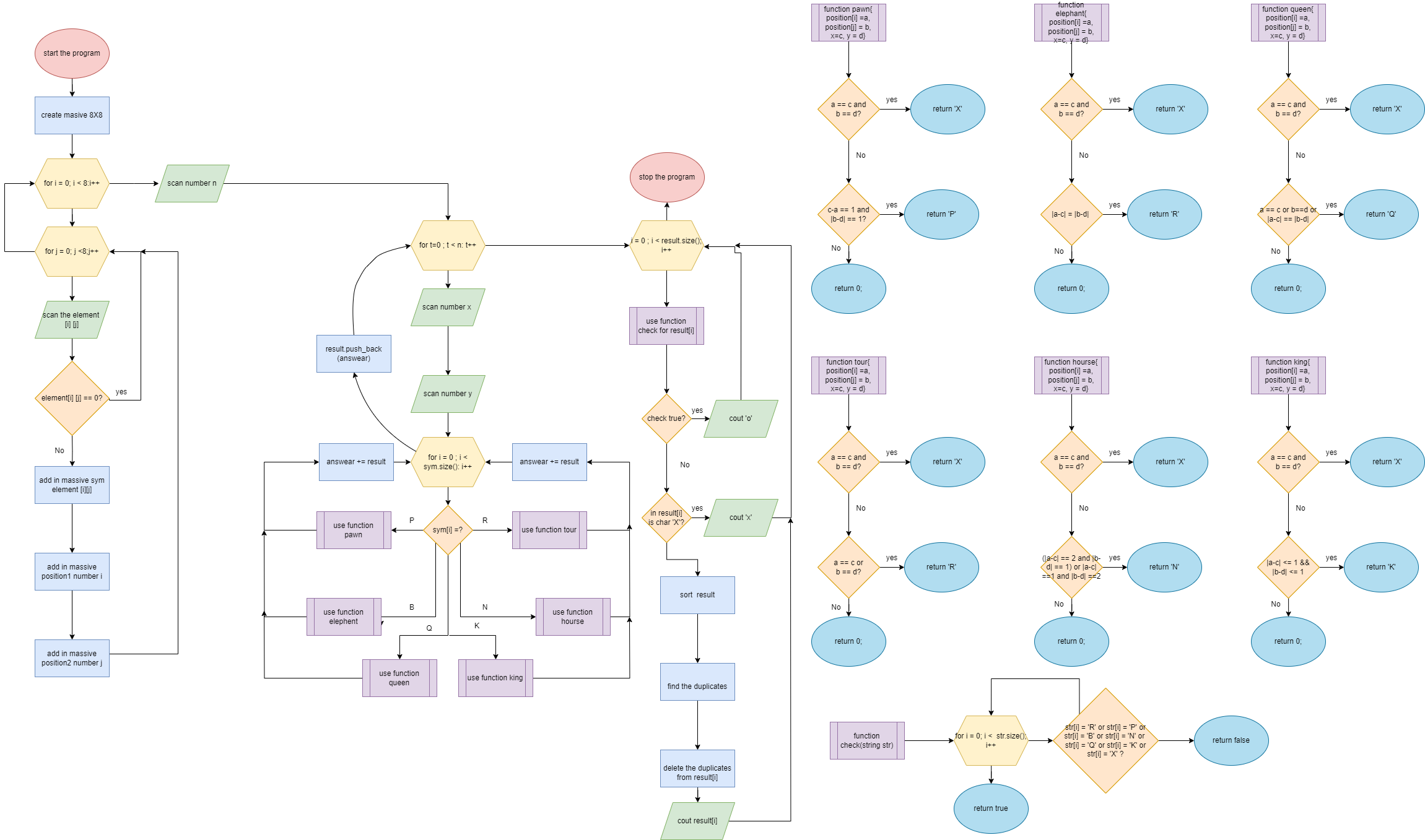
* Блок-схема

Рисунок 1:Lab# programming: Algotester Lab 6

* Планований час на реалізацію
* Важливі деталі для врахування в імплементації

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

Завдання №\_\_ Деталі по конфігурації середовища + скріншоти з підписами до скріншотів.

Підпис та № до блоку з скріншотами до конфігурації

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

Завдання №1 Lab# programming: VNS Lab 6

#include <iostream>

#include <algorithm>

#include <string>

std::string sortLettersInWords(const std::string& *input*) {

    std::string result;

    std::string word;

    for (*char* ch : *input*) {

        if (ch == ' ') {

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

            result += word + ' ';

            word = "";

        } else {

            word += ch;

        }

    }

    // Сортувати букви у останньому слові і додати його до результату

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

    result += word;

    return result;

}

*int* main() {

    std::string input = "ACDF BOAB MAD312";

    std::string result = sortLettersInWords(input);

    std::cout << "Original: " << input << std::endl;

    std::cout << "Sorted  : " << result << std::endl;

    return 0;

}

Програма сортує заданий ряд

Завдання №2 Lab# programming: VNS Lab 8

#include <iostream>

#include <cstring>

#include <vector>

using *namespace* std;

*struct* Disk {

*char* title[50];

*char* author[50];

*int* duration;

*float* price;

};

*void* writeDisks(FILE\* *file*, const vector<Disk>& *disks*) {

    for (const Disk& disk : *disks*) {

        fwrite(&disk, sizeof(Disk), 1, *file*);

    }

}

*void* readAndPrintDisks(FILE\* *file*) {

    rewind(*file*);

    Disk disk;

    while (fread(&disk, sizeof(Disk), 1, *file*)) {

        cout << "Name of disk: " << disk.title << endl;

        cout << "Author of disk: " << disk.author << endl;

        cout << "Duration of disk: " << disk.duration << " minutes" << endl;

        cout << "Price of disk: $" << disk.price << endl;

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

    }

}

*void* removeByDuration(FILE\* *file*, *int* *trigger*);

*int* main() {

    FILE\* file;

    file = fopen("Disks.bin", "wb");

    if (!file) {

        perror("Error opening file for writing");

        return 1;

    }

    vector<Disk> musicDisks = {

        {"show must go on", "ACDC", 20, 195.5},

        {"christmas song", "chor Abob", 33, 300},

        {"Last love", "Dua Lipa", 40, 120.45}

    };

    writeDisks(file, musicDisks);

    fclose(file);

    cout << "Data has been written to Disks.bin" << endl;

    file = fopen("Disks.bin", "rb");

    if (!file) {

        perror("Error opening file for reading");

        return 1;

    }

    readAndPrintDisks(file);

    fclose(file);

    file = fopen("Disks.bin", "r+b");

    if (!file) {

        perror("Error opening file for reading and writing");

        return 1;

    }

*int* trigger = 33;

    removeByDuration(file, trigger);

     fclose(file);

file = fopen("Disks.bin", "r+b");

    cout << "\_\_\_\_\_\_\_\_\_\_\_New list of disk\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

    readAndPrintDisks(file);

    fclose(file);

    return 0;

}

*void* removeByDuration(FILE\* *file*, *int* *trigger*) {

    // Go to the end of the file

    fseek(*file*, 0, SEEK\_END);

*long* numRecords = ftell(*file*) / sizeof(Disk);

    rewind(*file*);

    std::vector<Disk> result;

    for (*long* i = 0; i < numRecords; ++i) {

        Disk currentDisk;

        fread(&currentDisk, sizeof(Disk), 1, *file*);

        if (currentDisk.duration == *trigger*) {

            Disk disk1 = {"Lost on you", "LP", 36, 100.34};

            Disk disk2 = {"Kadilak", "Dzidzio", 36, 100.34};

            result.push\_back(disk1);

            result.push\_back(disk2);

        }

        else{

             result.push\_back(currentDisk);

        }

    }

    fclose(*file*);

*file* = fopen("Disks.bin", "wb");

    for (const Disk& disk : result) {

        fwrite(&disk, sizeof(Disk), 1, *file*);

    }

    fclose(*file*);

}

Програма видалила елемент з заданою тривалістю і додала на його місце 2 нових елемента

Завдання №3 Lab# programming: VNS Lab 9

#include <iostream>

#include <fstream>

#include <string>

#include <cctype>

*int* countVowels(const std::string& *str*) {

*int* vowelCount = 0;

    for (*char* ch : *str*) {

        // Перевіряємо, чи символ є голосним

        if (std::isalpha(ch) && (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

                                 ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')) {

            vowelCount++;

        }

    }

    return vowelCount;

}

*void* createFirstFile(std::string *str*){

    std::ofstream outputFile("F1.txt", std::ios::trunc);

     if (outputFile.is\_open()) {

        outputFile << "Bright stars hung in the dark sky, resembling diamonds on a black velvet blanket.\n";

        outputFile << "The aroma of freshly brewed coffee pierces the morning mist, filling the kitchen with warmth and comfort.\n";

        outputFile << "The laughter of children echoes in the backyard, spreading joy and merriment.\n";

        outputFile << "The babbling brook softly whispers among the rocks, creating a natural symphony of forest life.\n";

        outputFile << "The vibrant colors of spring flowers blossom in the park, awakening nature from its winter slumber.\n";

        outputFile << "The sun sets on the horizon, painting the sky in shades of pink and orange.\n";

        outputFile << "A gentle breeze embraces the face, carrying the scent of the sea and distant adventures.\n";

        outputFile << "Ancient streets in the city center glisten with reflections from evening lanterns.\n";

        outputFile << "The soft purple light of street café lanterns creates an atmosphere of warmth and romance.\n";

        outputFile << "Fresh fruits sit on the table in the room, enticing appetite and promising a flavorful day.\n";

        // Закрити файл

        outputFile.close();

}

}

*int* main() {

    std::string name = "F1.txt";

    createFirstFile(name);

    std::ifstream inputFile("F1.txt"); // Відкриваємо вхідний файл для читання

    std::ofstream outputFile("F2.txt",std::ios::out | std::ios::trunc); // Відкриваємо вихідний файл для запису

    if (!inputFile.is\_open() || !outputFile.is\_open()) {

        std::cerr << "Error!" << std::endl;

        return 1;

    }

    std::string line;

*int* lineCount = 0;

*int* maxVowelCount = 0;

*int* lineWithMaxVowels = 0;

    // Читаємо рядки з файлу та знаходимо рядок з найбільше голосними

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

        lineCount++;

*int* currentVowelCount = countVowels(line);

        // Порівнюємо з поточним найбільшим числом голосних символів

        if (currentVowelCount > maxVowelCount) {

            maxVowelCount = currentVowelCount;

            lineWithMaxVowels = lineCount;

        }

    }

    // Повторно відкриваємо вхідний файл для повторного читання

    inputFile.clear();

    inputFile.seekg(0);

    // Читаємо рядки з файлу та записуємо їх у вихідний файл, крім рядка з найбільше голосними

    lineCount = 0;

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

        lineCount++;

        if (lineCount != lineWithMaxVowels) {

            outputFile << line << std::endl;

        }

    }

    // Закриваємо файли

    inputFile.close();

    outputFile.close();

    std::cout << "Видалено рядок № " << lineWithMaxVowels

              << " з " << maxVowelCount << " голосними буквами." << std::endl;

    return 0;

}

Програма видалила рядок з найбільшою кількістю голосних букв

Завдання №4 Lab# programming: Algotester Lab 4

#include <iostream>

#include <vector>

#include <algorithm>

*int* main(){

using *namespace* std;

vector<*int*> massive;

*int* n, k, b;

cin >> n >> k;

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

    cin >> b;

 massive.push\_back(b);

}

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

*auto* it = unique(massive.begin(), massive.end());

massive.erase(it, massive.end());

rotate(massive.begin(), massive.begin() + k % massive.size(), massive.end());

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

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

    cout << massive[j] << " ";

}

    return 0;

}

Програма посортувала всі елементи масиву, видалила дублікати, і посунула його на k елементів

Завдання №5:Lab# programming: Algotester Lab 6

#include <iostream>

#include <cmath>

#include <vector>

#include <string>

#include <algorithm>

*char* pawn(*int* *a*, *int* *b*, *int* *c*, *int* *d*){

if( *a* == *c* && *b* == *d*)

return 'X';

else if(*c*-*a* == 1 && abs(*b*-*d*) == 1)

return 'P';

else{

    return 0;

}

}

*char* tour(*int* *a*, *int* *b*, *int* *c*,*int* *d*){

if( *a* == *c* && *b* == *d*)

return 'X';

else if(*a* == *c* || *b* == *d*)

return 'R';

else{

    return 0;

}

}

*char* elephent(*int* *a*, *int* *b*, *int* *c*, *int* *d*){

if( *a* == *c* && *b* == *d*)

return 'X';

else if(abs(*a*-*c*) == abs(*b*-*d*))

return 'B';

else{

    return 0;

}

}

*char* hourse(*int* *a*, *int* *b*, *int* *c*,*int* *d*){

    if( *a* == *c* && *b* == *d*)

    return 'X';

    else if(abs(*a*-*c*) == 2 && abs(*d*-*b*) == 1 || abs(*a*-*c*) == 1 && abs(*d*-*b*) == 2)

    return 'N';

else{

    return 0;

}

}

*char* queen(*int* *a*, *int* *b*, *int* *c*, *int* *d*){

if(*a* == *c* && *b* == *d*)

return 'X';

else if(*a* == *c* || *b* == *d* || abs(*a*-*c*) == abs(*b*-*d*))

return 'Q';

else{

    return 0;

}

}

*char* king(*int* *a*, *int* *b*, *int* *c*, *int* *d*){

if(*a* == *c* && *b* == *d*)

return 'X';

else if( abs(*a*-*c*) <= 1 && abs(*b*-*d*) <= 1)

return 'K';

else{

    return 0;

}

}

*bool* check(const std::string &*str*){

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

if((*str*)[i] == 'R' || (*str*)[i] == 'P'|| (*str*)[i] == 'R'|| (*str*)[i] == 'B'|| (*str*)[i] == 'N'|| (*str*)[i] == 'Q'|| (*str*)[i] == 'K' || (*str*)[i] == 'X'){

return false;

}

}

    return true;

}

*int* main(){

    using *namespace* std;

*char* board[8][8];

     vector<string> result;

    vector<*char*> sym;

    vector<*int*> position1;

    vector<*int*> position2;

    for(*int* i = 0; i < 8; i++){

        for(*int* j = 0; j < 8; j++){

        cin >> board[i][j];

        if(board[i][j] != 'O'){

        sym.push\_back(board[i][j]);

        position1.push\_back(i);

        position2.push\_back(j);

        }

        }

    }

*int* n, x,y;

    cin >> n;

    for(*int* t = 0;t < n; t++){

    cin >> x >> y;

    string answear;

    answear.clear();

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

*char* resultChar = 0;  // Ініціалізуємо символ значенням 0 (нуль), яке не є дійсним символом.

    switch (sym[i]) {

        case 'P':

            resultChar = pawn(position1[i] + 1, position2[i] + 1, x, y);

            break;

        case 'R':

            resultChar = tour(position1[i] + 1, position2[i] + 1, x, y);

            break;

        case 'B':

            resultChar = elephent(position1[i] + 1, position2[i] + 1, x, y);

            break;

        case 'N':

            resultChar = hourse(position1[i] + 1, position2[i] + 1, x, y);

            break;

        case 'Q':

            resultChar = queen(position1[i] + 1, position2[i] + 1, x, y);

            break;

        case 'K':

            resultChar = king(position1[i] + 1, position2[i] + 1, x, y);

            break;

    }

    if (isalpha(resultChar)) {

        answear += resultChar;

    }

}

        result.push\_back(answear);

}

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

    if(check(result[i])){

    cout << 'O' << endl;

    if(i == result.size()-1)

    break;

    else{

    continue;

    }

    }

    size\_t position = result[i].find('X');

    if(position != std::string::npos ){

    cout << "X" << endl;

    }

    else{

    sort(result[i].begin(), result[i].end());

*auto* it = unique(result[i].begin(), result[i].end());

    result[i].erase(it, result[i].end());

    cout << result[i] << endl;

    }

    }

    return 0;

}

Програма показала всі фігури, які атакують задані клітинки

Завдання №6 Practice# programming: Class Practice Task

#include <iostream>

#include <fstream>

#include <algorithm>

#include <string>

*enum* FileOpResult {

    Success,

    Failure,

};

FileOpResult copy\_file(const std::string& *file\_from*, const std::string& *file\_to*);

FileOpResult create\_file(const std::string& *file*, const std::string& *content*);

*int* main() {

    using *namespace* std;

    string contest;

    cout << "Enter content for the file: ";

    getline(cin, contest);

    FileOpResult createResult = create\_file("from.txt", contest);

    if (createResult == Success) {

        cout << "File created successfully." << endl;

        FileOpResult copyResult = copy\_file("from.txt", "file\_to.txt");

        if (copyResult == Success) {

            cout << "File contents copied successfully." << endl;

        } else {

            cerr << "File contents copy failed." << endl;

        }

    } else {

        cerr << "File creation failed." << endl;

    }

    return 0;

}

FileOpResult create\_file(const std::string& *file*, const std::string& *content*) {

    std::ofstream outputFile(*file*, std::ios::trunc);

    if (outputFile.is\_open()) {

        outputFile << *content*;

        outputFile.close();

        return Success;

    } else {

        std::cerr << "Could not open file '" << *file* << "' for writing." << std::endl;

        return Failure;

    }

}

FileOpResult copy\_file(const std::string& *file\_from*, const std::string& *file\_to*) {

    using *namespace* std;

    std::ifstream from(*file\_from*);

    if (!from.is\_open()) {

        cerr << "Could not open source file." << endl;

        return Failure;

    }

    std::ofstream to(*file\_to*);

    using *namespace* std;

    if (!to.is\_open()) {

        cerr << "Could not open target file." << endl;

        return Failure;

    }

*char* c;

    while (from.get(c)) {

        to.put(c);

    }

    from.close();

    to.close();

    return Success;

}

Підпис та № до блоку з кодом програми

Завдання №7: Practice# programming:  Self Practice Task

#include <iostream>

*void* str1(*int* *m*){

    for(*int* j = 0; j < *m*;j++){

    if(j%2 == 1)

    std::cout << 'x';

    else{

    std::cout << '.';

        }

    }

std::cout << std::endl;

}

*void* str2(*int* *m*){

     for(*int* j = 0; j < *m*;j++){

        std::cout << '.';

     }

     std::cout << std::endl;

}

*void* str3(*int* *m*){

    for(*int* j = 0; j < *m*;j++){

    if(j%2 != 1)

    std::cout << 'x';

    else{

    std::cout << '.';

        }

    }

std::cout << std::endl;

}

*int* main(){

    using *namespace* std;

*int* n, m;

    cin >> n >> m;

*int* field[n][m];

    for(*int* i = 0; i<n - 1;i++){

    if(i%4 == 0 || i == 0){

        str1(m);

    }

    else if(i%2 == 1){

        str2(m);

    }

    else if(i%2 == 0)

        str3(m);

    }

*int* check = n/2;

check = check%2;

if(n%2 == 0)

str3(m);

else if( check == 1 && n%2 == 1)

str3(m);

else if( check == 0 && n%2 == 1)

str1(m);

    return 0;

}

Програма вивела розстановку на полі

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

Завдання №1:Lab# programming: VNS Lab 6

Original: ACDF BOAB MAD312

Sorted : ACDF ABBO 123ADM

PS C:\Users\User>

Підпис та № до блоку з виконанням та тестуванням програми

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

Завдання №2: Lab# programming: VNS Lab 8

Data has been written to Disks.bin

Name of disk: show must go on

Author of disk: ACDC

Duration of disk: 20 minutes

Price of disk: $195.5

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

Name of disk: christmas song

Author of disk: chor Abob

Duration of disk: 33 minutes

Price of disk: $300

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

Name of disk: Last love

Author of disk: Dua Lipa

Duration of disk: 40 minutes

Price of disk: $120.45

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

\_\_\_\_\_\_\_\_\_\_\_New list of disk\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of disk: show must go on

Author of disk: ACDC

Duration of disk: 20 minutes

Price of disk: $195.5

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

Name of disk: Lost on you

Author of disk: LP

Duration of disk: 36 minutes

Price of disk: $100.34

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

Name of disk: Kadilak

Author of disk: Dzidzio

Duration of disk: 36 minutes

Price of disk: $100.34

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

Name of disk: Last love

Author of disk: Dua Lipa

Duration of disk: 40 minutes

Price of disk: $120.45

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

PS E:\coding>

Підпис та № до блоку з виконанням та тестуванням програми

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

Завдання №3:Lab# programming: VNS Lab 9

Видалено рядок № 2 з 28 голосними буквами.

PS E:\coding>

Підпис та № до блоку з виконанням та тестуванням програми

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

Завдання №4 : Lab# programming: Algotester Lab 4

10 3

1 2 2 3 3 3 4 5 6 7

7

4 5 6 7 1 2 3

PS E:\coding>

Підпис та № до блоку з виконанням та тестуванням програми

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

Завдання №5: Lab# programming: Algotester Lab 6

OOOOOOOO

OROOOOOO

OONOOOOO

OOOOPOOO

OOOOOOOO

OOOOOOOO

KOQOOOOO

OOOOOOOR

7

8 1

1 2

5 4

5 1

6 2

8 4

6 7

KR

NR

NP

Q

KQR

QR

O

Підпис та № до блоку з виконанням та тестуванням програми

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

Завдання №6 : Practice# programming: Class Practice Task

Enter content for the file: Hello world

File created successfully.

File contents copied successfully.

PS C:\Users\User>

Підпис та № до блоку з виконанням та тестуванням програми

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

Завдання №7: : Practice# programming: Self Practice Task

6 6

.x.x.x

......

x.x.x.

......

.x.x.x

x.x.x.

Підпис та № до блоку з виконанням та тестуванням програми

Час затрачений на виконання завдання

## **6. Кооперація з командою:**

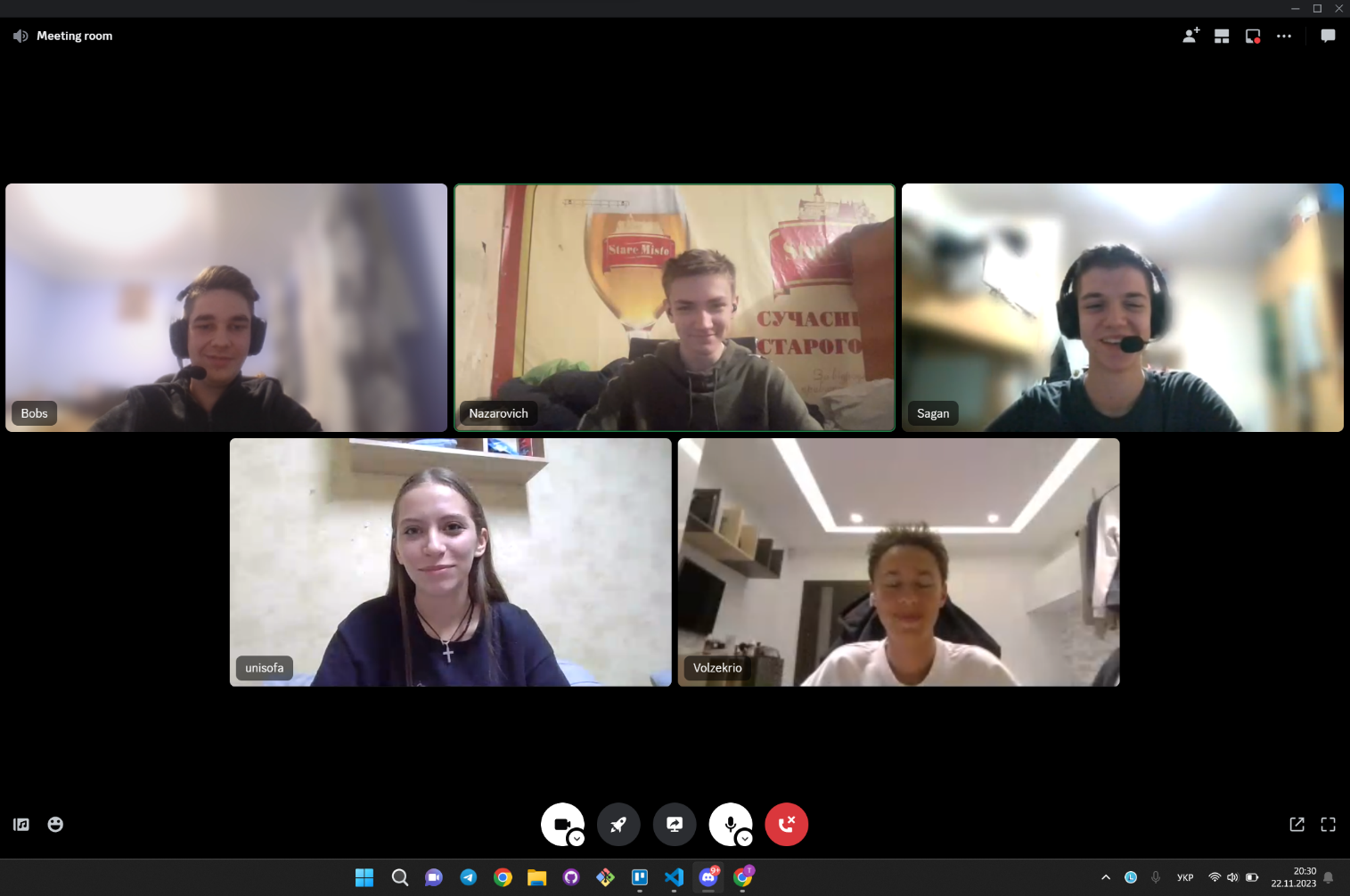
* Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло
* 

Рисунок 2: онлайн мітинг

* Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло
* Скрін з 3-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло (опційно)
* Скрін з 2-му коментарями від учасників команди на пул реквесті з Ревю Роботи

# **Висновки:**

Я навчився використовувати класи і рядки в моїх програмах. Також розширив свої знання про засоби STL. Навчився використовувати файли в роботі з с++. На основі вище зазначених тем виконав завдання, які виконують ті чи інші завдання.