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***: «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

Сіренко Юрій Сергійович

# **Тема роботи:**

Робота з файлами у мові програмування C++

# **Мета роботи:**

Ознайомитися з основними поняттями та операціями, пов’язаними з файлами у мові програмування C++, та навчитися створювати та використовувати власні бібліотеки для роботи з файлами

# **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 1 - [C++ Files](https://www.w3schools.com/cpp/cpp_files.asp).
    - Стаття 2 - [How to append text to a text file in c++](https://stackoverflow.com/questions/2393345/how-to-append-text-to-a-text-file-in-c)
    - Стаття 3 - <https://icpc.upc.edu.cn/CLibrary/iostream/streampos.html>
    - Курс: Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Як працювати з файлами у C++ за допомогою класів std::fstream, std::ifstream, std::ofstream. Методи для відкриття, закриття, читання, запису, переміщення курсору та контролю стану файлів.
    - Різні формати та режими файлів у C++, такі як текстові, бінарні, додавання, перезапис, читання тощо. Параметри для вказання типу та способу доступу до файлів при відкритті.
    - Як обрати найкращий формат та режим файлу для своєї задачі. Як опрацьовувати різні типи даних у файлах, такі як символи, рядки, числа, структури тощо.
    - Як перевіряти, чи файл відкритий, чи немає помилок, чи досягнуто кінця файлу тощо. Функції для перевірки стану файлу, такі як is\_open(), good(), eof(), fail().
    - Бібліотека fstream, яка надає класи та функції для роботи з файлами у C++. Огляд основних можливостей та прикладів використання бібліотеки fstream.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 07.12.2023
* Тема №2: Системи числення.
  + Джерела Інформації:
    - Відео 1 -[Системи численн | Двійкова арифметика](https://www.youtube.com/watch?v=ai65R0WGq2Q&ab_channel=homeofourelectronics)
    - Стаття 1 - [Системи числення](https://ru.stackoverflow.com/questions/1466656/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F-%D0%BD%D0%B0-%D1%81)
    - 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: 04.12.2023
  + Звершення опрацювання теми: 07.12.2023
* Тема №3: Бінарні Файли
  + Джерела Інформації:
    - Стаття 1 - [Binary file: write/read (c++)](https://ru.stackoverflow.com/questions/420525/%D0%91%D0%B8%D0%BD%D0%B0%D1%80%D0%BD%D1%8B%D0%B9-%D1%84%D0%B0%D0%B9%D0%BB-%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C-%D1%87%D1%82%D0%B5%D0%BD%D0%B8%D0%B5-%D1%81)
    - Стаття 2 - [Робота з бінарним файлом - C/C++](https://replace.org.ua/topic/7901/)
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Відкриття та Закриття Файлу. Читання та Запис Даних. Перевірка Успішності Операцій. Позиціонування в Файлі.
    - 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: 04.12.2023
  + Звершення опрацювання теми: 07.12.2023
* Тема №4 Створення й використання бібліотек.
  + Джерела Інформації:
    - Відео 1 - [Write Better Code! | How to Create Shared Libraries in C/C++](https://www.youtube.com/watch?v=Slfwk28vhws&ab_channel=LowLevelLearning).
    - Стаття 1 - [How do I create a library?](https://stackoverflow.com/questions/16693273/how-do-i-create-a-library).
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Статичні, динамічні та заголовкові бібліотеки
    - Розглянуто можливість створення влсних бібліотек за допомогою компілятора, IDE, або інших інструментів
  + Статус: Ознайомлений
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 07.12.2023

# **Виконання роботи:**

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

Завдання №1 Опрацювання теорії

* Деталі завдання: опрацювання вище перечисленої теорії. Розбір статей, перегляд відео та імплементація теорії на практиці. Вивчення найосновніших тем.

Завдання №2 VNS Lab №6

* Варіант завдання – 1

Надрукувати найдовше й найкоротше слово в цьому рядку

* Деталі завдання

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

Слова одне від одного відокремлюються одним або декількома пробілами.

Наприкінці тексту ставиться крапка.

* Важливі деталі для врахування в імплементації програми

Текст містить не більше 255 символів.

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

рядка у відповідності зі своїм варіантом.

Завдання №3 VNS Lab №8

* Варіант завдання – 1

Структура "Абітурієнт":

- прізвище, ім'я, по батькові;

- рік народження;

- оцінки вступних іспитів (3);

- середній бал атестата.

Знищити елемент із зазначеним номером, додати елемент після елемента із

зазначеним прізвищем.

* 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.

* Важливі деталі для врахування в імплементації програми

Використання вище наведеної теорії для імплементації програми.

Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

Завдання №4 VNS Lab №9

* Варіант завдання - 1
* Деталі завдання

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

1) Скопіювати у файл F2 тільки парні рядки з F1.

2) Підрахувати розмір файлів F1 й F2 (у байтах).Інформацію

* Важливі деталі для врахування в імплементації програми

Завдання №5 Class Practice Files 1

* 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

- 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.

* Важливі деталі для врахування в імплементації програми

Завдання №6 Class Practice Files 2

* 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

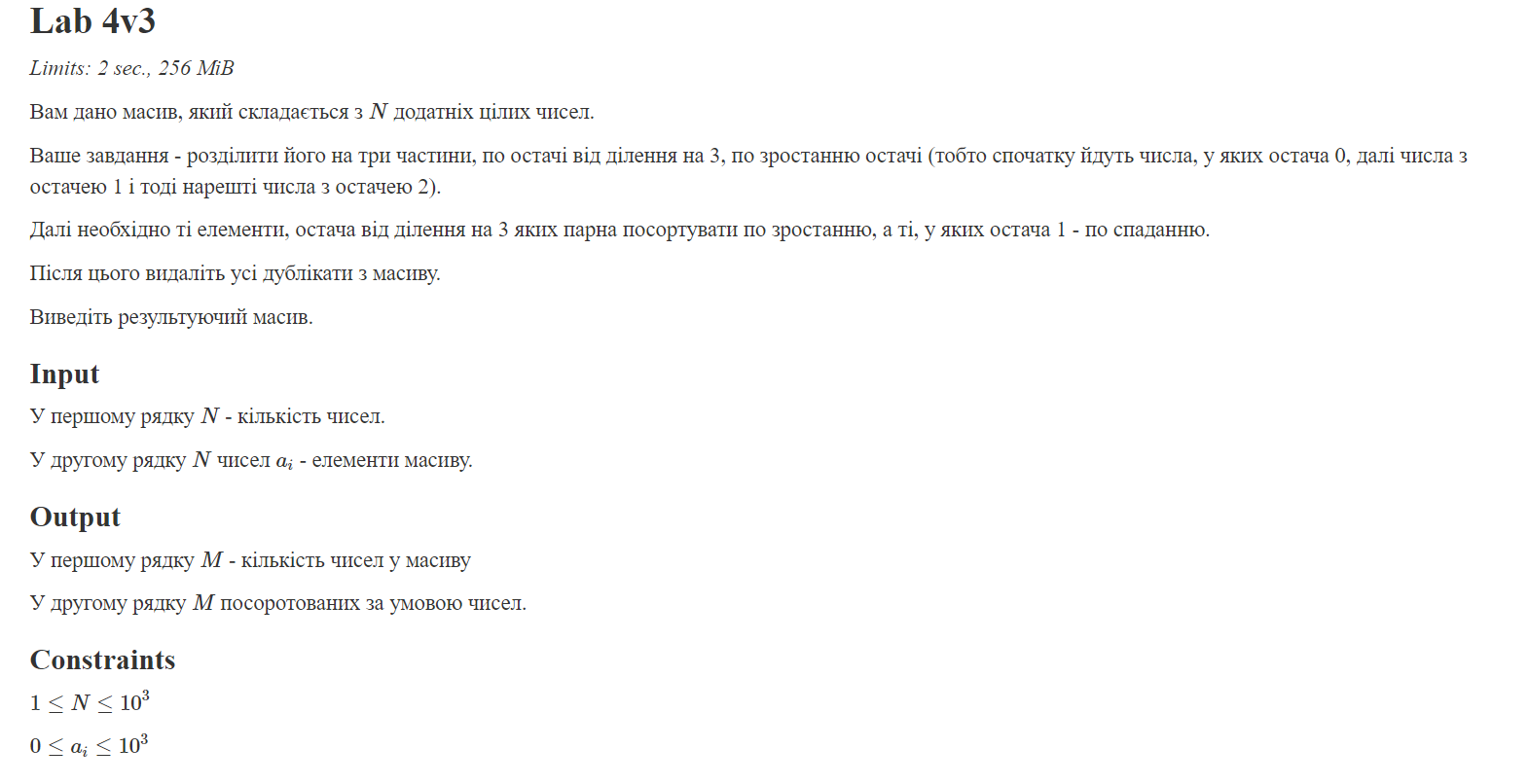
- file\_from, file\_to – можуть бути повним або відносним шляхом

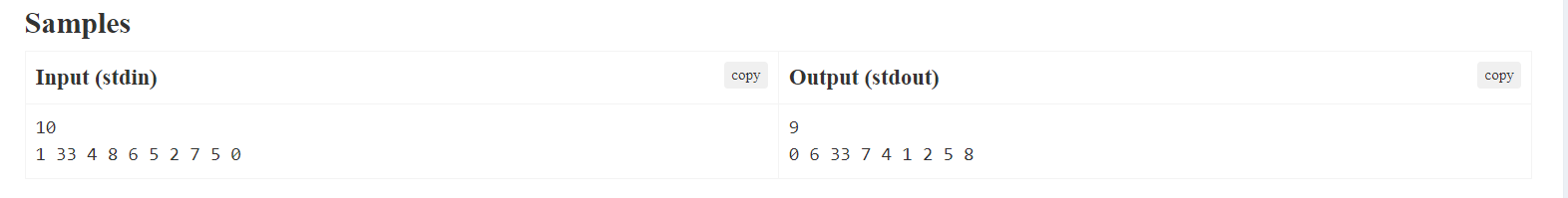
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

* Важливі деталі для врахування в імплементації програми

Завдання №7 Algotester Lab №4

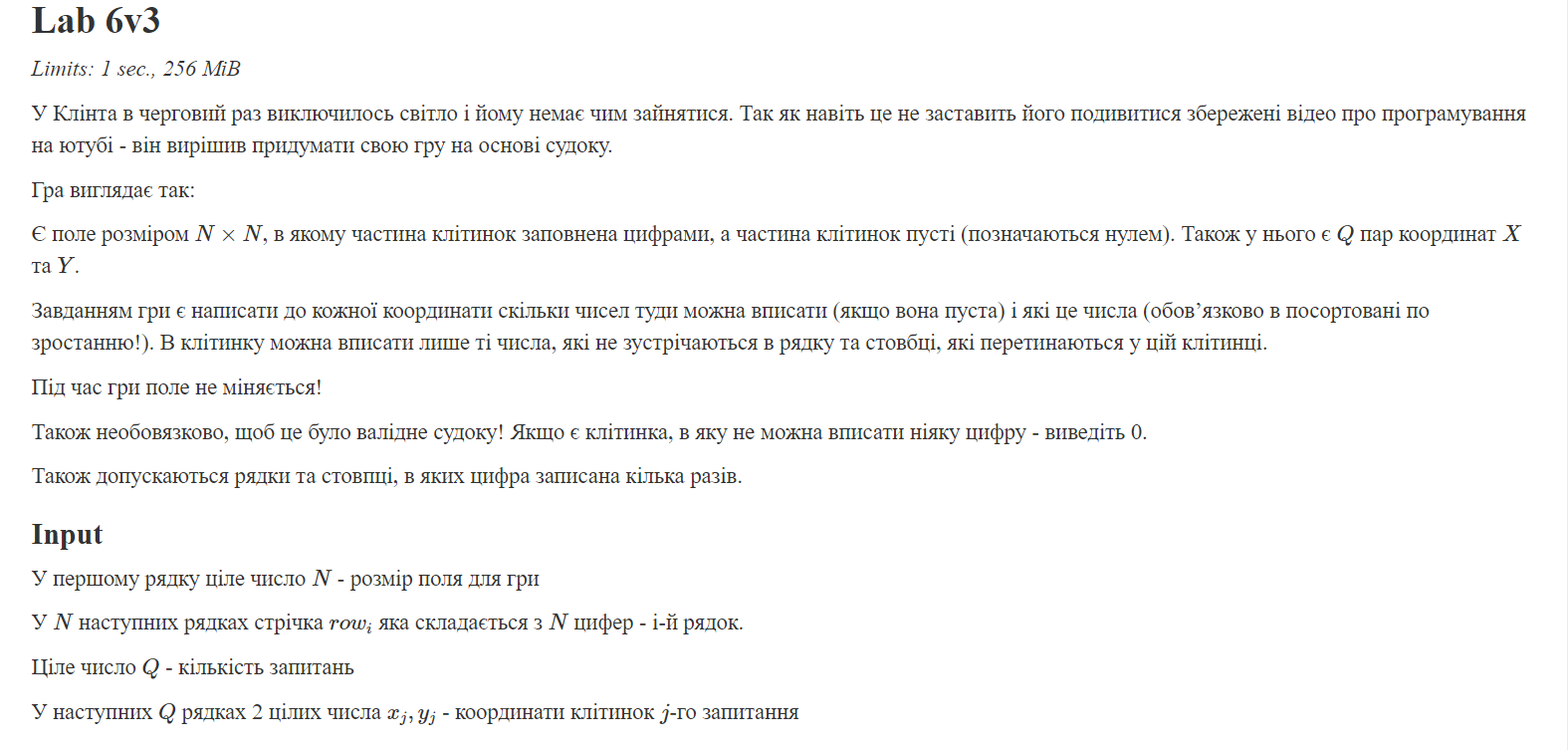
* Варіант завдання – 3

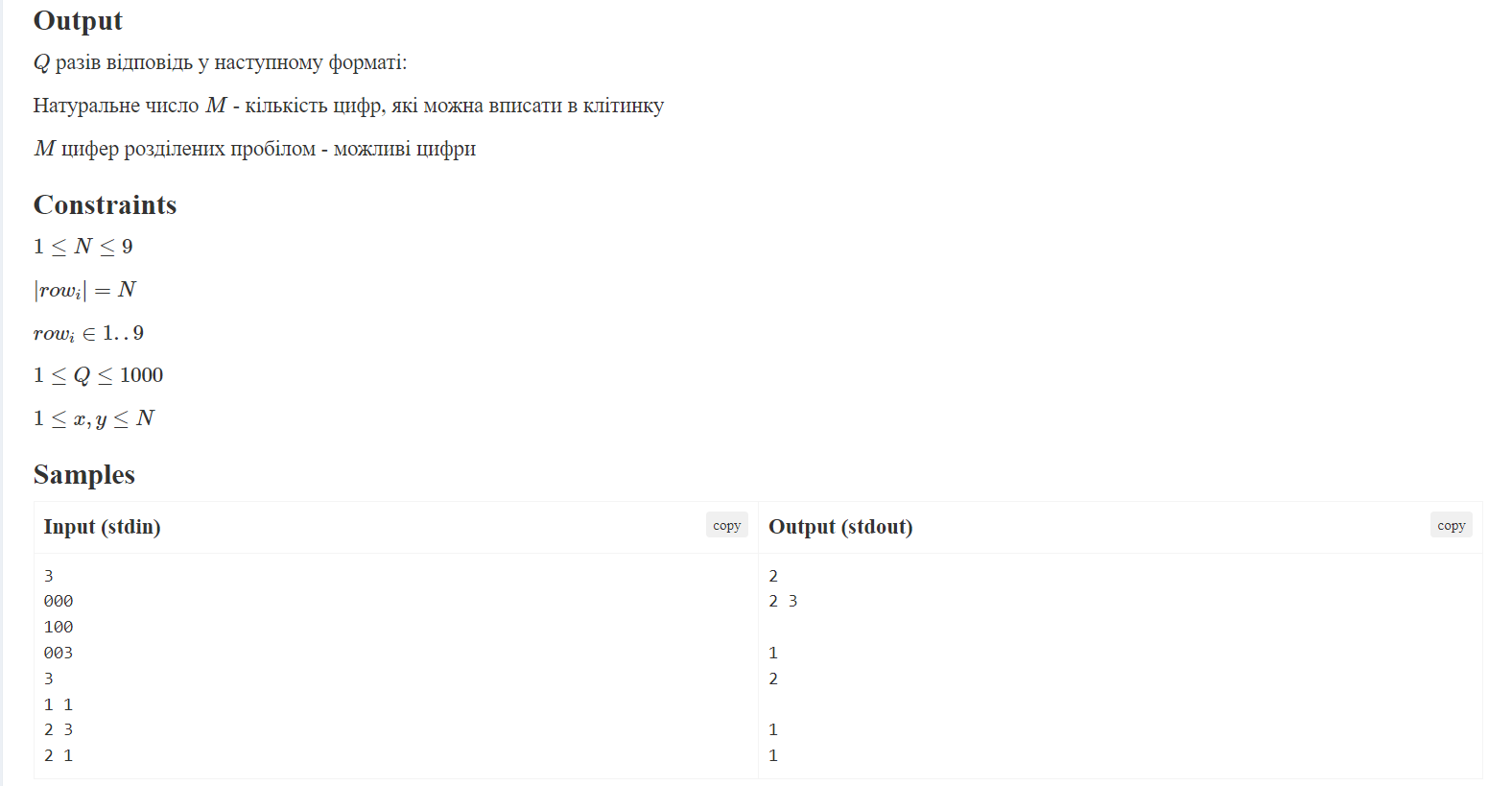




Завдання №8 Algotester Lab №6

* Варіант завдання – 3





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

Програма №1 - VNS Lab №6

* Блок-схема - відсутня
* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 2 години

Програма №2 - VNS Lab №8

* Блок-схема - відсутня
* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 1 доба

Програма №3 - VNS Lab №9

* Блок-схема - відсутня
* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 1 година

Програма №4 - Class Practice Files 1

* Блок-схема - відсутня
* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 40 хвилин

Програма №5 - Class Practice Files 2

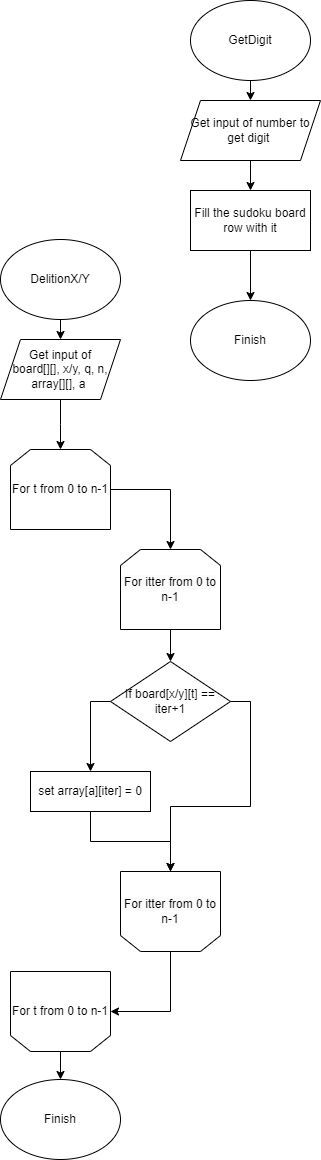
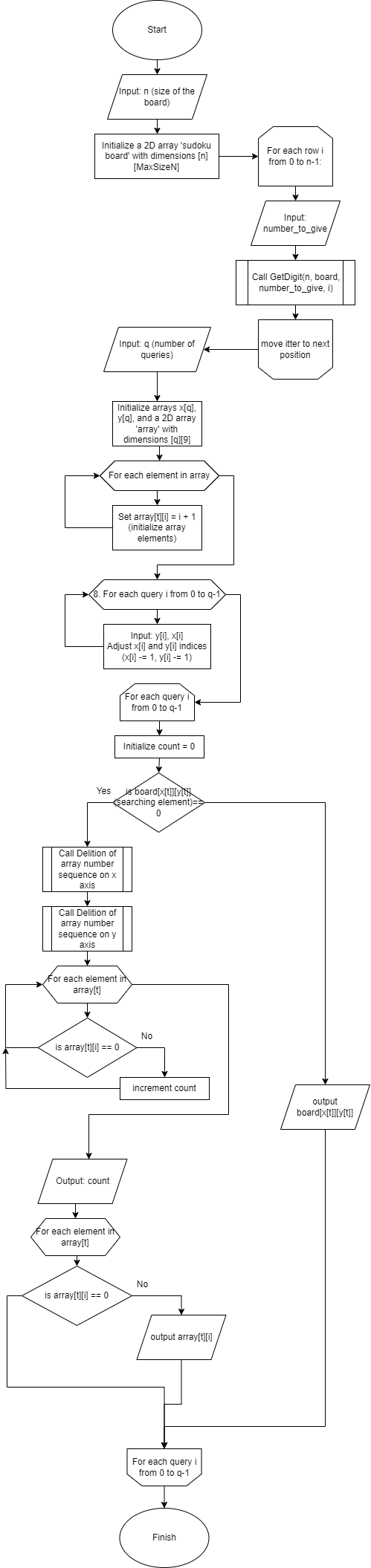
* Блок-схема - відсутня
* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 20 хвилин

Програма №6 - Algotester Lab №4

* Блок-схема - відсутня
* Планований час на реалізацію – 3 години
* Час затрачений на виконання завдання – 3 години

Програма №7 - Algotester Lab №6

* Блок-схема –



* Планований час на реалізацію – 3 години
* Час затрачений на виконання завдання – 5 годин

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

Завдання №1 - VNS Lab №6

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/730/files#diff-f54f95f8dcbb372efb2c0bf688bcd8a90658b589f93f5e99120c47fe0cae215b>

#include <iostream>

#include <cstring>

using namespace std;

int main()

{

    char str[256];

    cout << "Enter string: ";

    gets(str);

    char \*word = strtok(str, " ");

    cout<<word<<"\n";

    char \*minWord = word, \*maxWord = word;

    while(word != NULL) {

        if(strlen(word) < strlen(minWord)) {

            minWord = word;

        }

        if(strlen(word) > strlen(maxWord)) {

            maxWord = word;

        }

        word = strtok(NULL, " ");

    }

    cout << "Shortest word: " << minWord << endl;

    cout << "Longest word: " << maxWord << endl;

    return 0;

}

*Код програми №1 - VNS Lab №6*

Завдання №2 - VNS Lab №8

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/730/files#diff-4253a2f70c808d8bceee3b63ee8b42b2b5f10b1c81285afa6ee0153d34679cc9>

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

struct Student

{

    string surname;

    string name;

    string patronymic;

    int birthYear;

    int entranceExamScores[3];

    double averageScore;

};

double averageRate(const int \*arr)

{

    double s=0;

    for(int i = 0;i<3;i++)

    {

        s+=arr[i];

    }

    return s/3;

}

void writeStrFile(ofstream &inFile, const string &str)

{

    size\_t size = str.size();

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

    inFile.write(str.c\_str(), size);

}

void writeStrFile(ofstream &inFile, const int &i)

{

    string str = to\_string(i);

    size\_t size = sizeof(str);

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

    inFile.write(str.c\_str(), size);

}

void writeStrFile(ofstream &inFile, const double &db)

{

    string str = to\_string(db);

    size\_t size = sizeof(str);

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

    inFile.write(str.c\_str(), size);

}

string readFile(ifstream &outFile)

{

    size\_t size;

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

    string str(size,' ');

    outFile.read(&str[0], size);

    return str;

}

void writeFile(const vector <Student> &students, const string &filename)

{

    ofstream inFile(filename, ios::binary);

    if (!inFile)

    {

        cout << "Cannot open file.\n";

        return;

    }

    for (const Student &student : students)

    {

        writeStrFile(inFile, student.surname);

        writeStrFile(inFile, student.name);

        writeStrFile(inFile, student.patronymic);

        writeStrFile(inFile, student.birthYear);

        writeStrFile(inFile, student.averageScore);

    }

    inFile.close();

}

vector<Student> readFile(const string &filename)

{

    ifstream outFile(filename, ios::binary);

    if (!outFile)

    {

        cout << "Cannot open file.\n";

        return {};

    }

    vector<Student> students;

    while (outFile.peek() != EOF)

    {

        Student student;

        student.surname = readFile(outFile);

        student.name = readFile(outFile);

        student.patronymic = readFile(outFile);

        student.birthYear = stoi(readFile(outFile));

        student.averageScore = stod(readFile(outFile));

        students.push\_back(student);

    }

    outFile.close();

    return students;

}

void addElement(vector<Student> &students, const Student& newStudent, const std::string& surname)

{

    vector<Student> tempStudents;

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

    {

        tempStudents.push\_back(students[i]);

        if (students[i].surname == surname)

        {

            tempStudents.push\_back(newStudent);

        }

    }

    students = tempStudents;

}

void deleteElement( vector<Student> &students, int index)

{

    Student student;

    students.erase(students.begin() + index);

}

void outputStudents(const vector<Student> &students)

{

    for(const Student &student : students)

    {

        cout<<student.surname<< " " <<student.name<< " " <<student.patronymic<< " " <<student.birthYear<< " " <<student.averageScore<<"\n";;

    }

}

int main()

{

    string surnameToAdd;

    Student student;

    std::vector<Student> students = {

        {"Ivanov", "Ivan", "Ivanovich", 2000, {200, 181, 179},186},

        {"Petrov", "Petr", "Petrovich", 2001, {156, 135, 167},154.1304},

        {"Sidorov", "Sidor", "Sidorovich", 2002, {200, 195, 193},197}

    };

    std::string filename = "students.bin";

    cout<<"\nInitial students list\n";

    writeFile(students, "students.bin");

    outputStudents(students);

    students = readFile("students.bin");

    cout<<"\nAdding new student after certein surname\n";

    string surname = "Ivanov";

    Student newStudent = {"Sirenko","Yurii","Ser", 2006,{200, 200, 200},200};

    addElement(students, newStudent, surname);

    writeFile(students, "students.bin");

    outputStudents(students);

    cout<<"\nDeliting 2nd student\n";

    deleteElement(students, 2-1);

    writeFile(students, "students.bin");

    outputStudents(students);

    cout<<"\nAdding 2 new students after certein surname\n";

    surname = "Petrov";

    newStudent = {"Schur","Petro","Opanasovuch", 1978,{300, 300, 300},300};

    addElement(students, newStudent, surname);

    surname = "Sidorov";

    newStudent = {"Starostiv","Dmytro","Romanovuch", 2008,{187, 120, 130},152.23};

    addElement(students, newStudent, surname);

    writeFile(students, "students.bin");

    outputStudents(students);

    cout<<"\nDeliting 3rd student\n";

    deleteElement(students, 3-1);

    writeFile(students, "students.bin");

    outputStudents(students);

}

*Код програми №2 - VNS Lab №8*

Завдання №3 - VNS Lab №9

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/730/files#diff-6e507cdc5b049db6eb1c3daa3c30c75bb31c71e16f071e28227856be2fdd4d38>

#include <fstream>

#include <string>

#include <iostream>

using namespace std;

int main()

{

    ofstream outF1("F1.txt");

    for(int i = 0;i<20;i++)

    {

        outF1<<"Line"<<i<<"\n";

    }

    outF1.close();

    ifstream inFile("F1.txt");

    ofstream outF2("F2.txt");

    string line;

    for(int lineIt=1;getline(inFile, line);lineIt++)

    {

        if(lineIt % 2 == 0)

        {

            outF2<<line<<"\n";

        }

    }

    inFile.close();

    outF2.close();

    streampos fileSize;

    inFile.open("F1.txt");

    inFile.seekg(0, ios::end);

    fileSize = inFile.tellg();

    inFile.close();

    cout << "File size of F1: " << fileSize << " b\n";

    inFile.open("F2.txt");

    inFile.seekg(0, ios::end);

    fileSize = inFile.tellg();

    inFile.close();

    cout << "File size of F2: " << fileSize << " b\n";

}

*Код програми №3 - VNS Lab №9*

Завдання №4 - Class Practice Files 1

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/730/files#diff-6a4bf2eb2305e2d8fe6c04ae7bc3ab9f4e6c659b47232647670b4a744b7fbdf4>

#include <iostream>

#include <fstream>

using namespace std;

enum FileOpResult { Success, Failure,};

FileOpResult write\_to\_file(const string &name, const string &content)

{

    ofstream outFile(name, ios\_base::out);

    if(!outFile)

    {

        return Failure;

    }

    size\_t size = content.size();

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

    outFile.write(content.c\_str(), sizeof(content));

    return Success;

}

FileOpResult read\_from\_file(const string &name)

{

    ifstream inFile(name, ios::out);

    if(!inFile)

    {

        return Failure;

    }

    size\_t size;

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

    string str(size,' ');

    inFile.read(&str[0], size);

    cout<<"The current text file is: \n"<<str;

    return Success;

}

int main()

{

    string input;

    string action;

    string name;

    while(true)

    {

        cout<<"\nChoose action to do:\n"

        <<"1) write\n"

        <<"2) read\n"

        <<"3) end\n";

        cin>>action;

        cin.ignore();

        if(action == "write")

        {

            cout<<"Enter name of file to write: \n";

            cin>>name;

            cin.ignore();

            getline(cin, input);

            if(write\_to\_file(name, input)==1)

            {

                cout<<"\nFailure\n";

            }

            else

            {

                cout<<"\nSuccess\n";

            }

        }

        else if(action=="read")

        {

            cout<<"Enter name of file to read: \n";

            cin>>name;

            cin.ignore();

            if(read\_from\_file(name)==1)

            {

                cout<<"\nFailure\n";

            }

            else

            {

                cout<<"\nSuccess\n";

            }

        }

        else if(action=="end")

        {

            break;

        }

    }

}

*Код програми №4 – Class Practice 1*

Завдання №5 - Class Practice Files 2

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/730/files#diff-912ef33c301b183ed529b6c9f1aafe7ddc88e7697a67b3cf15cc165b961400b2>

#include <iostream>

#include <fstream>

using namespace std;

enum FileOpResult { Success, Failure,};

FileOpResult copy\_file(const string file\_from, const string file\_to)

{

    ifstream inFile(file\_from,ios::in);

    ofstream outFile(file\_to,ios::out);

    if(!inFile||!outFile)

    {

        return Failure;

    }

    string line;

    while(getline(inFile,line))

    {

        outFile<<line;

    }

    inFile.close();

    outFile.close();

    return Success;

}

int main()

{

    string file\_from;

    string file\_to;

    cout<<"Enter the file\_from :\n";

    cin>>file\_from;

    cout<<"Enter the file\_to :\n";

    cin>>file\_to;

    if(copy\_file(file\_from,file\_to)==1)

    {

        cout<<"Failure";

    }

    else

    {

        cout<<"Success";

    }

}

*Код програми №5 – Class Practice 2*

Завдання №6 - Algotester Lab №4

Посилання на файл програми у пул-запиті GitHub

[https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/730/files#diff-2513fc272d8c5a38c0e035e2211c20f21fa726901159973e14f88058c91a1d4d](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/730/files%23diff-2513fc272d8c5a38c0e035e2211c20f21fa726901159973e14f88058c91a1d4d)

#include <iostream>

#include <vector>

using namespace std;

void swap(int arr[], int a, int b)

{

    int t = arr[a];

    arr[a]=arr[b];

    arr[b]=t;

}

void sort2(int arr[],int n)

{

    for(int i=1, j = 0; i<n; i++)

    {

        if(arr[j]>arr[i])

        {

            swap(arr,i,j);

        }

        if(i==n-1)

        {

            i=j;

            j++;

        }

    }

}

void sort1(int arr[],int n)

{

    for(int i=1, j = 0; i<n; i++)

    {

        if(arr[j]<arr[i])

        {

            swap(arr,i,j);

        }

        if(i==n-1)

        {

            i=j;

            j++;

        }

    }

}

int main()

{

    int n,m;

    cin>>n;

    int num\_arr[n];

    vector<int> arr0,arr1,arr2;

    bool possibility;

    for(int i = 0;i<n;i++)

    {

        cin>>num\_arr[i];

    }

    for(int i = 0;i<n;i++)

    {

        possibility=0;

        for(int t = i+1;t<n;t++)

        {

            if(num\_arr[i]==num\_arr[t])

            {

                possibility=1;

                break;

            }

        }

        if(num\_arr[i] % 3 == 0 && possibility ==0)

        {

            arr0.push\_back(num\_arr[i]);

        }

        else if(num\_arr[i] % 3 == 1 && possibility ==0)

        {

            arr1.push\_back(num\_arr[i]);

        }

        else if(num\_arr[i] % 3 == 2 && possibility ==0)

        {

            arr2.push\_back(num\_arr[i]);

        }

    }

    int sorted0[arr0.size()],sorted1[arr1.size()],sorted2[arr2.size()];

    for(int i = 0;i<arr0.size();i++)

    {

        sorted0[i]=arr0[i];

    }

    for(int i = 0;i<arr1.size();i++)

    {

        sorted1[i]=arr1[i];

    }

    for(int i = 0;i<arr2.size();i++)

    {

        sorted2[i]=arr2[i];

    }

    sort2(sorted0,arr0.size());

    sort1(sorted1,arr1.size());

    sort2(sorted2,arr2.size());

    cout<<arr0.size()+arr1.size()+arr2.size()<<"\n";

    for(int i = 0;i<arr0.size();i++)

    {

        cout<<sorted0[i]<<" ";

    }

    for(int i = 0;i<arr1.size();i++)

    {

        cout<<sorted1[i]<<" ";

    }

    for(int i = 0;i<arr2.size();i++)

    {

        cout<<sorted2[i]<<" ";

    }

}

*Код програми №6 – Algotester Lab №4*

Завдання №7 - Algotester Lab №6

Посилання на файл програми у пул-запиті GitHub

[https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/730/files#diff-6c85797cba20a09c780ae41e30165b2fa347451d6642f0537fe9aa1184daeae8](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/730/files%23diff-6c85797cba20a09c780ae41e30165b2fa347451d6642f0537fe9aa1184daeae8)

#include <iostream>

#include <cmath>

using namespace std;

const int MaxSizeN = 9;

void GetDigit(int &n,int board[][MaxSizeN],int &given\_number,int &row)

{

    for(int digit\_counter=10,i=0; given\_number!=0 || i<n ;i++)

    {

        board[n-i-1][row] =(given\_number % digit\_counter);

        given\_number -=(given\_number % digit\_counter);

        given\_number /= 10;

    }

}

void Deletionx(int board[][MaxSizeN],int x,int q,int n,int array[][9],int a)

{

    for(int t = 0; t<n;t++)

    {

        for(int iter = 0; iter<n;iter++)

        {

            if(board[x][t]==iter+1)

            {

                array[a][iter]=0;

            }

        }

    }

}

void Deletiony(int board[][MaxSizeN],int y,int q,int n,int array[][9],int a)

{

    for(int t = 0; t<n;t++)

    {

        for(int iter = 0; iter<n;iter++)

        {

            if(board[t][y]==iter+1)

            {

                array[a][iter]=0;

            }

        }

    }

}

int main()

{

    int n;

    cin>>n;

    int board[n][MaxSizeN];

    for(int i=0,number\_to\_give;i<n;i++)

    {

        cin>>number\_to\_give;

        GetDigit(n,board,number\_to\_give,i);

    }

    int q;

    cin>>q;

    int x[q],y[q];

    int array[q][9];

    for(int t = 0;t<q;t++)

    {

        for(int i = 0;i<9;i++)

        {

            array[t][i]=i+1;

        }

    }

    for(int i = 0;i<q;i++)

    {

        cin>>y[i]>>x[i];

        x[i]-=1;

        y[i]-=1;

    }

    for(int t = 0;t<q;t++)

    {

        int count=0;

        if(board[x[t]][y[t]]==0)

        {

            Deletionx(board,x[t],q,n,array,t);

            Deletiony(board,y[t],q,n,array,t);

            for(int i = 0;i<n;i++)

            {

                if(array[t][i]!=0)

                {

                    count++;

                }

            }

            cout<<count;

            cout<<"\n";

            for(int i = 0;i<n;i++)

            {

                if(array[t][i]!=0)

                {

                    cout<<array[t][i]<<" ";

                }

            }

            cout<<"\n";

        }

        else

        {

            cout<<"1\n"<<board[x[t]][y[t]];

        }

        cout<<"\n";

    }

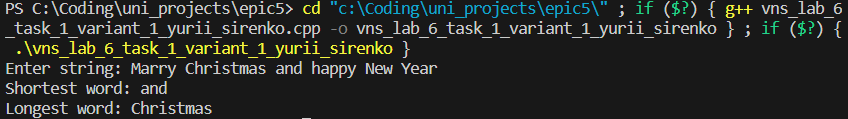
}

*Код програми №6 – Algotester Lab №6*

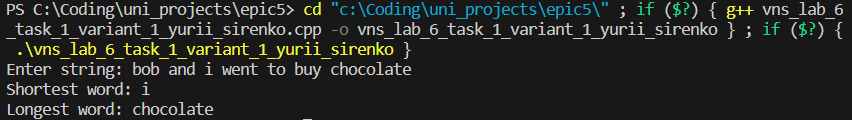
## **4. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 - VNS Lab №6

Деталі по виконанню і тестуванню програми



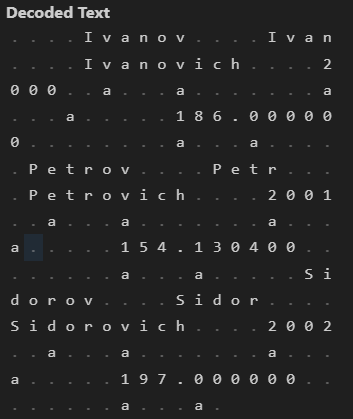
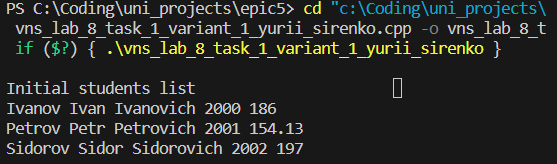
VNS Lab №6 test 1



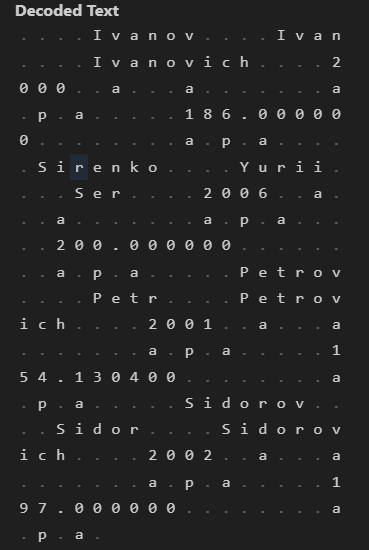
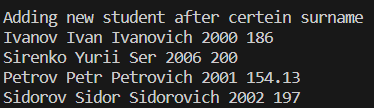
VNS Lab №6 test 2

Завдання №2 - VNS Lab №8

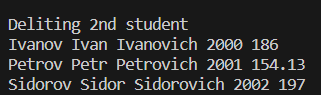
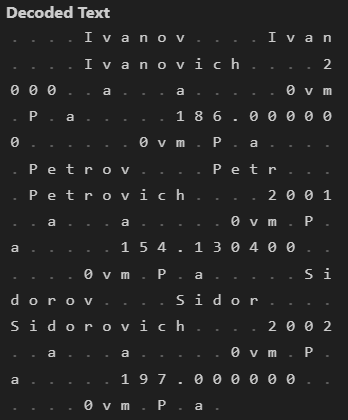
Деталі по виконанню і тестуванню програми



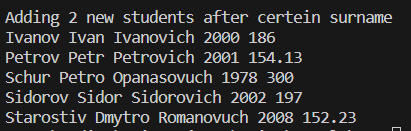
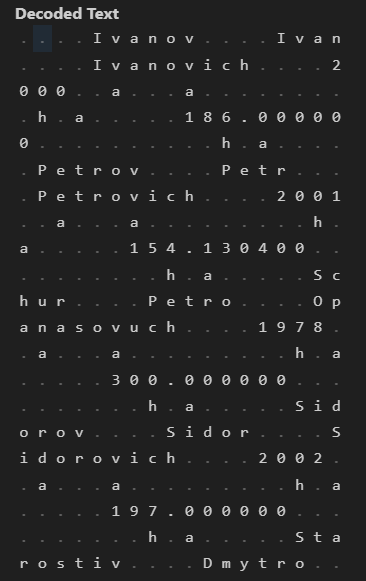
VNS Lab №8 test 1



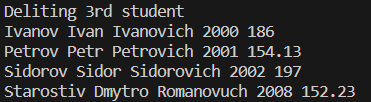
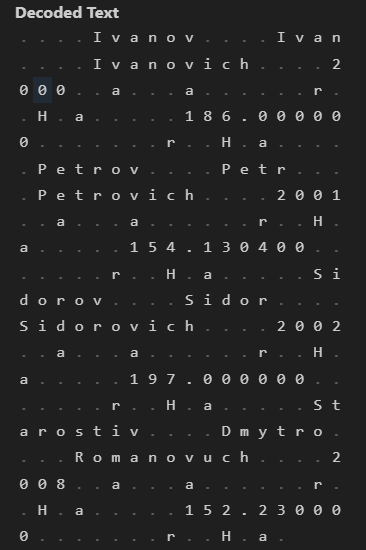
VNS Lab №8 test 2

VNS Lab №8 test 3

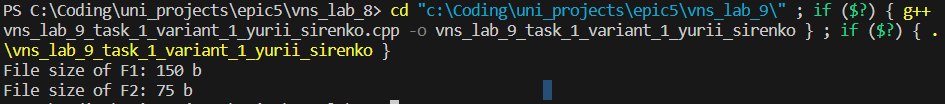
VNS Lab №8 test 4

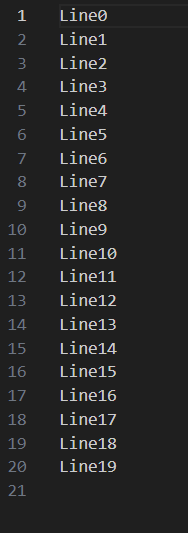
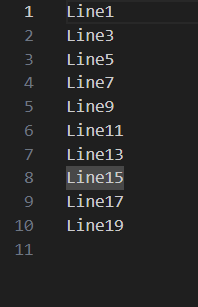
VNS Lab №8 test 5

Завдання №3 - VNS Lab №9

Деталі по виконанню і тестуванню програми

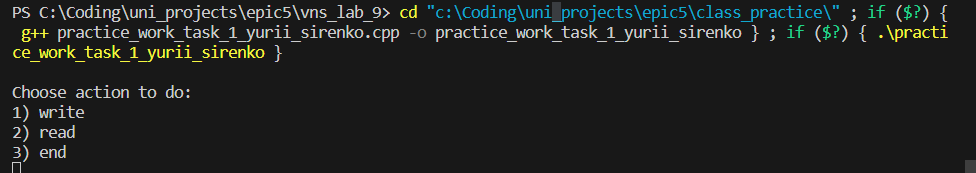


VNS Lab №9 test 1

 F1, F2

Завдання №4 - Class Practice Files 1

Деталі по виконанню і тестуванню програми

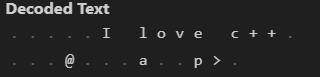


Class Practice Files 1 Main menu

\*Text editor is empty



Class Practice Files 1 Choosing to write text “I love c++” in text.txt



Class Practice Files 1 result 1

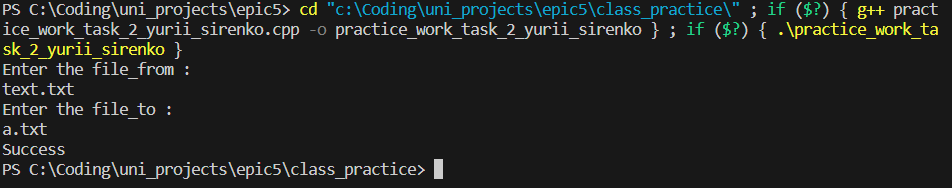


Class Practice Files 1 Choosing to read text in text.txt

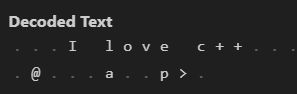
Завдання №5 - Class Practice Files 2

Деталі по виконанню і тестуванню програми

\*a.txt is empty

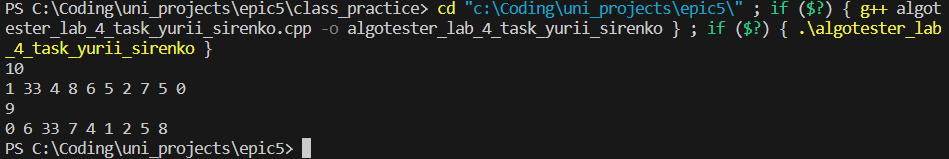


Class Practice Files 2 Copying from text.txt to a.txt



Class Practice Files 2 result in a.txt

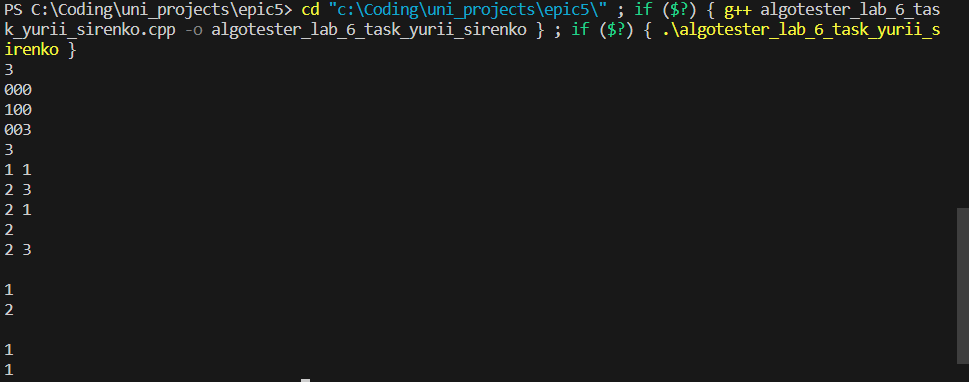
Завдання №6 - Algotester Lab №4

Деталі по виконанню і тестуванню програми 

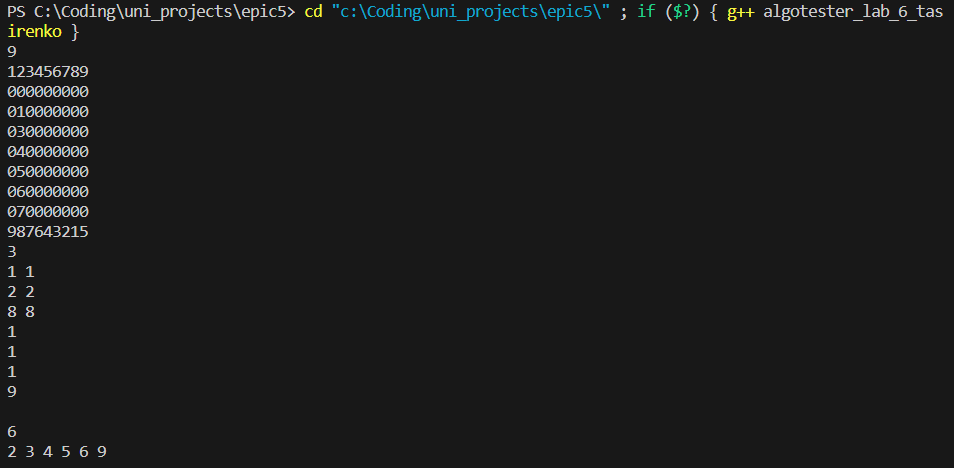
Algotester Lab №4 test 1

Завдання №7 - Algotester Lab №6

Деталі по виконанню і тестуванню програми

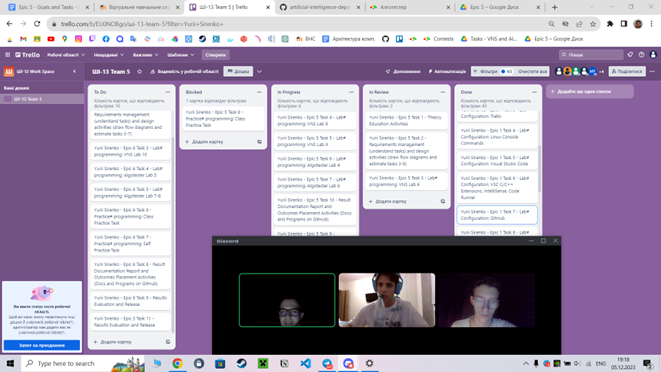
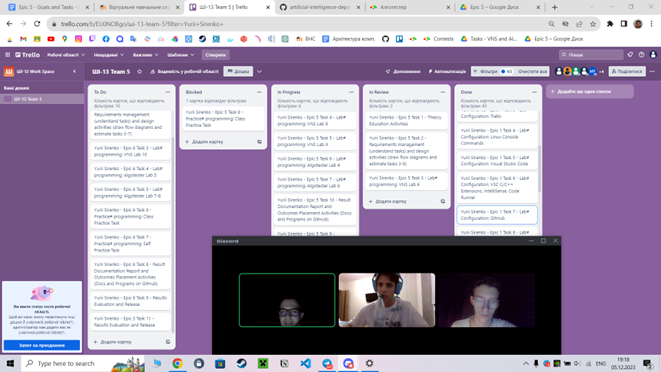


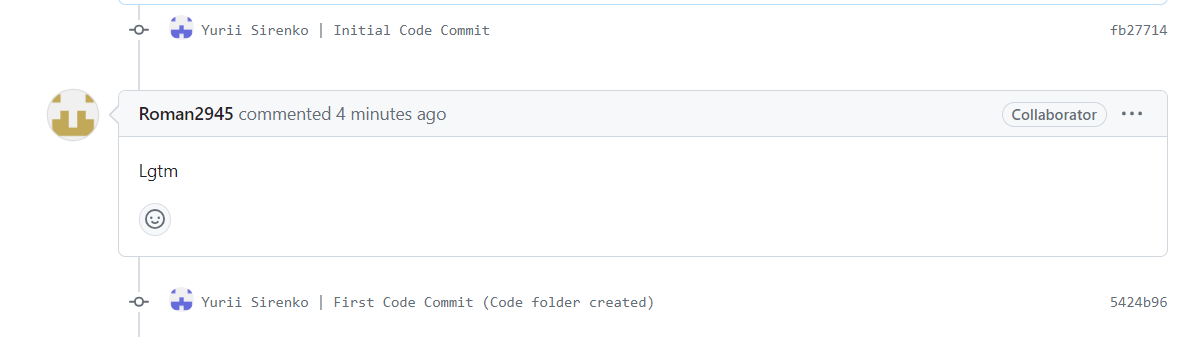
Algotester Lab №6 test 1



## 

## **5. Кооперація з командою:**

* Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло
* 
* Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло
* 
* Скрін з 2-му коментарями від учасників команди на пул реквесті з Ревю Роботи



# **Висновки:**

Деталі по результатам виконання робіт та висновки згідно тем та завдань

У результаті виконання лабораторної роботи, присвяченої роботі з файлами та використанню бібліотек у мові програмування C++, я здобув важливі навички та знання для ефективної роботи з файловою системою. Ознайомлення з основними концепціями, такими як робота з бінарними та текстовими файлами, системами числення, символами та рядками, дозволило мені глибше розібратися в механізмах обробки інформації в мові C++. Особливу увагу приділив створенню та використанню власних бібліотек, що відкрило нові можливості для оптимізації та узагальнення коду при подальшому програмуванні. Ця лабораторна робота була важливим етапом у моєму навчанні, дозволяючи поглибити розуміння принципів роботи з файлами та ефективного їх використання у програмах на C++.