Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**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-14

Шелеп Андрій Іванович

# **Тема роботи:**

Бінарні та текстові файли. Робота з файлами у С++. Символи та рядкові змінні. Зчитування з файлу та запис у файл. Створення й використання бібліотек

# **Мета роботи:**

Ознайомитися з файлами та з деталями роботи з ними у С++. Опрацювати зчитування та запис у файл. Ознайомитися з символами та рядковими змінними

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Файли. Бінарні та текстові файли. Використання файлів. Бібліотеки.
* Тема №2: Робота з файлами у С++.
* Тема №3: Символи та рядкові змінні.
* Тема №4: Зчитування та запис у файл.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Файли. Бінарні та текстові файли. Використання файлів.
  + Джерела Інформації:
    - Стаття.

<https://www.youtube.com/watch?v=m-WJikuZGuU&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://acode.com.ua/urok-24-zagolovkovi-fajly/>

<https://acode.com.ua/statychni-i-dynamichni-biblioteky/>

* + Що опрацьовано:
    - Заголовкові файли, їх створення та підключення у файл cpp
    - Бінарні та текстові файли
    - Стандартна бібліотека С++
    - Статичні та динамічні бібліотеки
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 04.12.2023
* Тема №2: Робота з файлами у С++.
  + Джерела Інформації:
    - Стаття.

<https://www.bestprog.net/uk/2020/05/14/c-examples-of-working-with-text-files-modification-of-files-sort-data-in-files-convert-file-data-to-list-ua/#q02>

* + Що опрацьовано:
    - Робота з файлами за допомогою fstream
    - Відкриття файлів у різних режимах
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 04.12.2023
* Тема №3: Символи та рядкові змінні.
  + Джерела Інформації:
    - Стаття.

<http://cpp.dp.ua/osnovni-funktsiyi-obrobky-symvolnyh-typiv/>

<https://acode.com.ua/urok-82-ryadky-c-style/>

* + Що опрацьовано:
    - Рядки та символи
    - Основні функції обробки символьних типів та рядків
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 04.12.2023
* Тема №4: Зчитування та запис у файл.
  + Джерела Інформації:
    - Стаття.

<https://www.bestprog.net/uk/2019/09/11/examples-of-using-c-tools-for-working-with-files-ua/>

* + Що опрацьовано:
    - Зчитування та запис рядків, структур, функція write() і read()
    - функція getline
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 04.12.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 6 - Task 1

* Варіант 14
* Деталі завдання:

Перетворити рядок таким чином, щоб у ньому залишилися тільки слова, що

містять букви й цифри, інші слова знищити.

* Важливі деталі для врахування в імплементації програми

Використовував функцію strtok для розбиття рядка на токени та функцію strcat для додавання до кінця одного рядка іншого.

Завдання №2 VNS Lab 8 - Task 1

* Варіант 14
* 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.

Структура "Стадіон":

- назва;

- адреса;

- місткість;

- види спорту.

Знищити елемент із заданою назвою, додати 2 елементи після елемента із

зазначеним номером.

* Важливі деталі для врахування в імплементації програми

Використання функцій блокового вводу/виводу read/write, для запису і зчитування структури.

Завдання №3 VNS Lab 9 - Task 1

* Варіант 14
* Деталі завдання

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

інформацію

1) Скопіювати з файлу F1 у файл F2 всі рядки, що не містять букву «А» і

розташовані між рядками з номерами N1 й N2.

2) Визначити номер того рядка, у якому найбільше голосних букв, файлу F2.

* Важливі деталі для врахування в імплементації програми

Використовував функцію find() для пошуку букви “A” у рядку.

Завдання №4 Algotester Lab 4

* Варіант 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].

* Важливі деталі для врахування в імплементації програми

Спочатку розв’язав без засобів stl, використовуючи двовимірний масив, а потім розв’язав іншим способ із використанням засобів stl;

Завдання №5 Algotester Lab 6

* Варіант 2
* Деталі завдання

У вас є шахова дошка розміром 8×8та 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 s - 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

Наявність 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

Використовую двовимірний масив символів та створюю функції типу bool для перевірки кожної фігури

Завдання №6 Class Practice Work

* Без варіанту
* Деталі завдання

1)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.

2)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

В першому завданні відкриваю файл зі знищенням попереднього вмісту та запису нового. В одному із завдань перевіряю існування файлу за введеною назвою

Завдання №7 Self Practice Work

* Задача з алготестеру - 0081 Допоможе чи заб’є
* Деталі завдання

До Тойлет-мена, відомого вам білоруського супергероя, дуже часто звертаються по допомогу різні люди. Проте, очевидно, усім мужній гігант допомогти не зможе — не вистачить часу. Та й не дуже хоче.

Саме тому, коли до нього приходить певне SMS-повідомлення з проханням про допомогу, він погодиться допомогти тоді й лише тоді, коли в цьому повідомленні знайдеться хоча б k входжень рядка TOILET, які не перетинаються.

За заданим повідомленням s, яке складається з великих латинських символів, виведіть YES, якщо Тойлет-мен погодиться допомогти людині, яка написала це повідомлення. У протилежному разі виведіть NO.

* Важливі деталі для врахування в імплементації програми

Завдання №8 Self Practice Work

* Задача з алготестеру - 0541 Дзідзьо і його пісня
* Деталі завдання

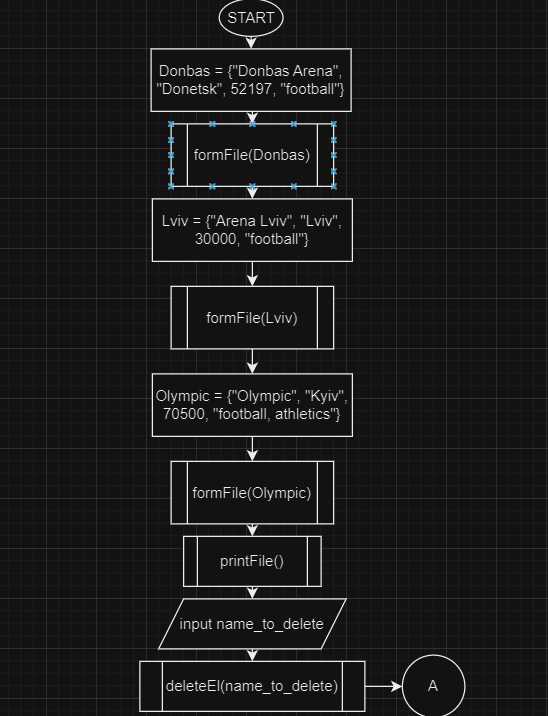
Одного разу, переглядаючи текст пісні, Дзідзьо видалив один символ з неї і помітив, що пісня стала зовсім іншою. Допитливому Дзідзю стало цікаво, скільки різних пісень може утворитись із заданої, якщо видалити з неї тільки один символ. Але Дзідзьо не має часу для того, щоб це рахувати.

* Важливі деталі для врахування в імплементації програми

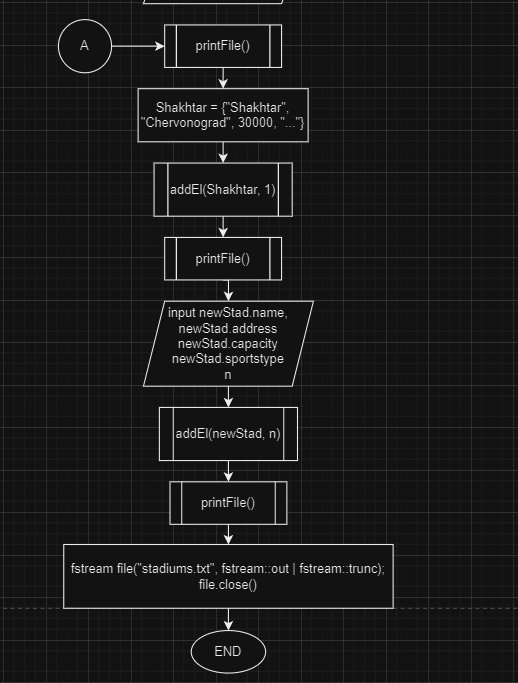
## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма №3 VNS Lab 8 - Task 1

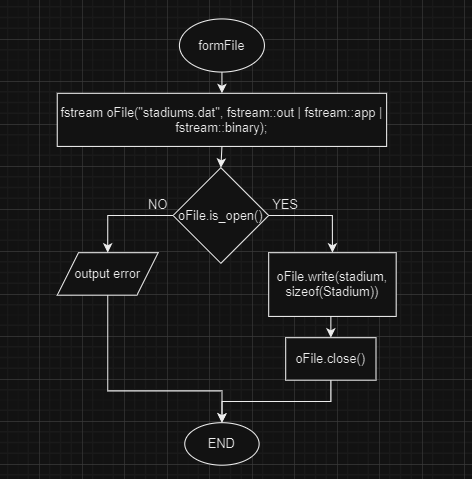
* Блок-схема:



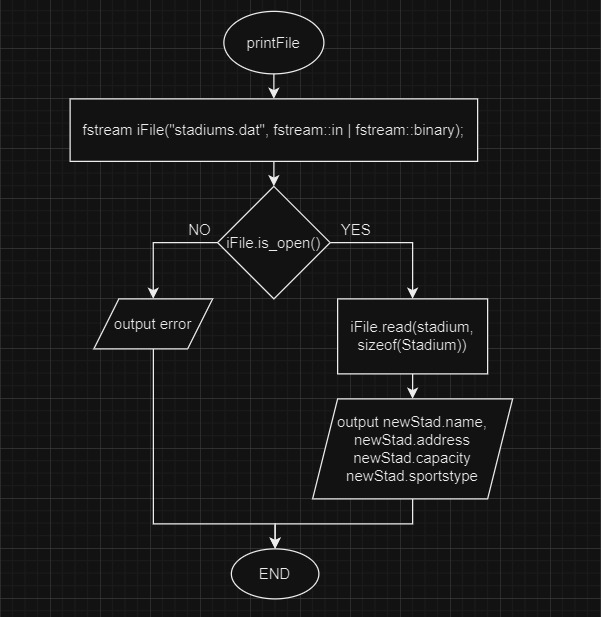
*Зображення №1. Блок-схема до програми №3 VNS Lab 8 Task 1*



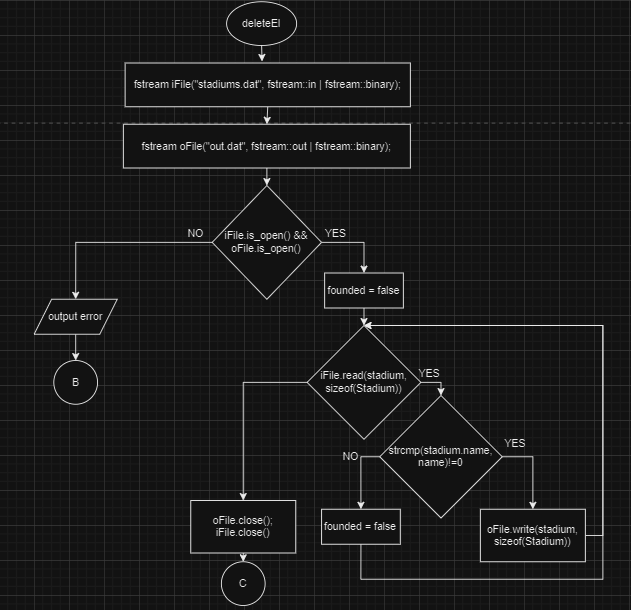
*Зображення №2. Блок-схема до програми №3 VNS Lab 8 Task 1*



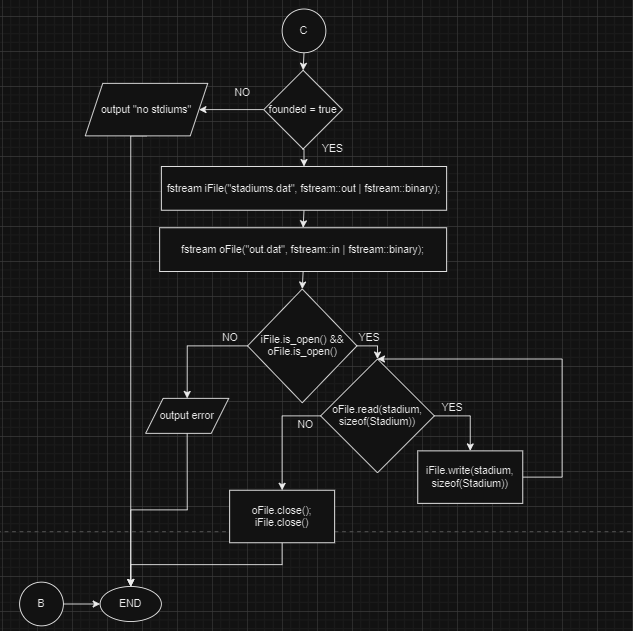
*Зображення №3. Блок-схема до програми №3 VNS Lab 8 Task 1*



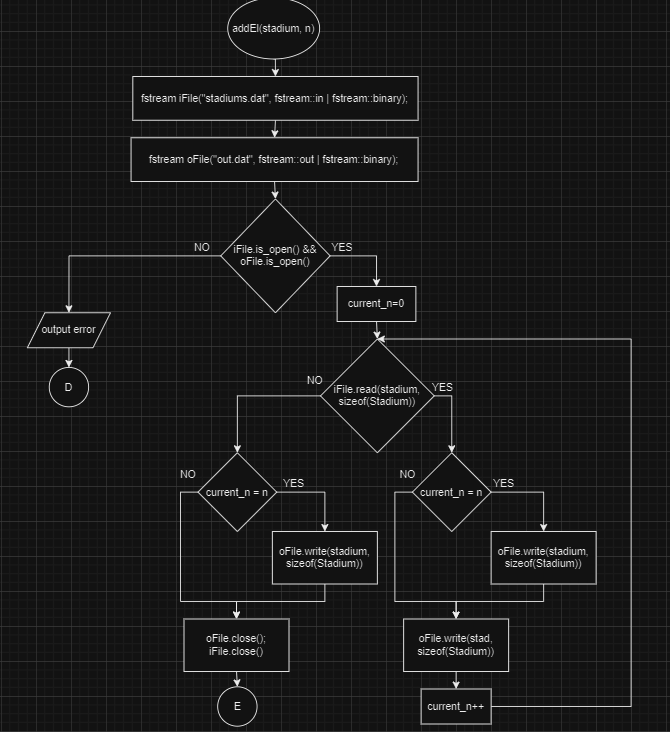
*Зображення №4. Блок-схема до програми №3 VNS Lab 8 Task 1*



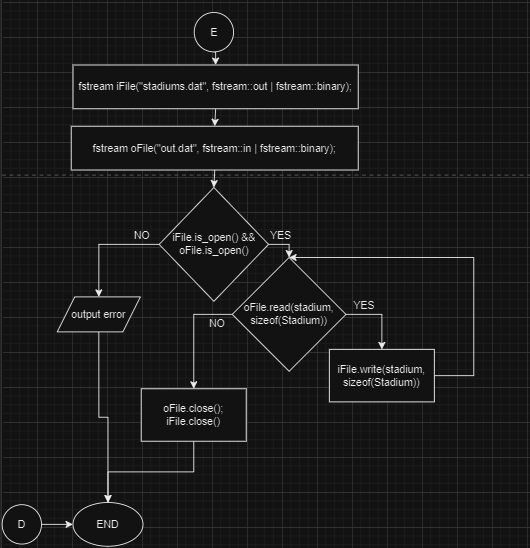
*Зображення №5. Блок-схема до програми №3 VNS Lab 8 Task 1*



*Зображення №6. Блок-схема до програми №3 VNS Lab 8 Task 1*



*Зображення №7. Блок-схема до програми №3 VNS Lab 8 Task 1*



*Зображення №1. Блок-схема до програми №3 VNS Lab 8 Task 1*

* Планований час на реалізацію: 2 год
* Важливі деталі для врахування в імплементації

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

Використано налаштування з попередніх лабораторних робіт.

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

Посилання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/691>

Завдання №1 VNS Lab 6 - Task 1

#include <iostream>

#include <cstring>

using *namespace* std;

*bool* correctWord(const *char* \* *word*, *int* *size*);

*int* main(){

*char* line[256];

cout << "Enter a string:" << endl;

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

*char* \* word;

word = strtok(line, " ");

line[0] = '\0';

while(word != NULL){

if (correctWord(word, strlen(word))){

strcat(line, word);

strcat(line, " ");

}

word = strtok(NULL, " ");

}

cout << line;

return 0;

}

*bool* correctWord(const *char* \* *word*, *int* *size*){

*bool* cond\_digit = false, cond\_alpha = false;

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

if(isdigit(*word*[i])){

cond\_digit = true;

} else if(isalpha(*word*[i])){

cond\_alpha = true;

} else{

return false;

}

}

return cond\_alpha && cond\_digit;

}

*Блок №1. Код до програми №1 VNS Lab 6 - Task 1*

Завдання №2 VNS Lab 8 - Task 1

#include <iostream>

#include <fstream>

#include <cstring>

using *namespace* std;

*struct* *Stadium*{

*char* name[30];

*char* address[30];

*int* capacity;

*char* sportstypes[40];

};

*void* formFile(*Stadium*& *stadium*);

*void* printFile();

*void* deleteEl(const *char*\* *name*);

*void* addEl(*Stadium*& *stadium*, *int* *number*);

*int* main(){

*Stadium* Donbas;

Donbas = {"Donbas Arena", "Donetsk", 52187, "football"};

formFile(Donbas);

*Stadium* Lviv;

Lviv = {"Arena Lviv", "Lviv", 30000, "karting, football"};

formFile(Lviv);

*Stadium* Olympic;

Olympic = {"Olympic", "Kyiv", 24255, "football, athletics"};

formFile(Olympic);

printFile();

cout << "enter the name of stadium, ehich you wnat to delete" << endl;

*char* name\_to\_del[30];

//cin.ignore();

cin.getline(name\_to\_del, sizeof(name\_to\_del));

deleteEl(name\_to\_del);

cout << "stadiums after deleting" << endl;

printFile();

*Stadium* Shakhtar;

Shakhtar = {"Shakhtar Chervonograd", "Chervonograd", 2425, "football, athletics, tennis"};

addEl(Shakhtar, 1);

printFile();

*Stadium* newStad;

cout << "enter the info about new stadium" << endl;

cin.getline(newStad.name, sizeof(newStad.name));

cin.getline(newStad.address, sizeof(newStad.address));

cin >> newStad.capacity;

cin.ignore();

cin.getline(newStad.sportstypes, sizeof(newStad.sportstypes));

cout << "enter the after position" << endl;

*int* pos;

cin >> pos;

addEl(newStad, pos);

printFile();

*fstream* file("stadiums.dat", *fstream*::out | *fstream*::trunc | *fstream*::binary);

file.close();

return 0;

}

*void* formFile(*Stadium*& *stadium*){

*fstream* oFile("stadiums.dat", *fstream*::out | *fstream*::app | *fstream*::binary);

if(oFile.is\_open()){

oFile.write((*char*\*)&*stadium*, sizeof(*Stadium*));

oFile.close();

}else{

cout << "error opening file" << endl;

return;

}

}

*void* printFile(){

*fstream* iFile("stadiums.dat", *fstream*::in | *fstream*::binary);

if(iFile.is\_open()){

*Stadium* stadium;

cout << " Stadiums" << endl;

while(iFile.read((*char*\*)&stadium, sizeof(*Stadium*))){

cout << "Name: " << stadium.name << endl;

cout << "Address: " << stadium.address << endl;

cout << "Capacity: " << stadium.capacity << endl;

cout << "Sports types: " << stadium.sportstypes << endl;

cout << "\*\*\*\*\*\*\*\*\*\*" << endl;

}

iFile.close();

}else{

cout << "error opening file" << endl;

return;

}

}

*void* deleteEl(const *char*\* *name*){

*fstream* iFile("stadiums.dat", *fstream*::binary | *fstream*::in);

*fstream* oFile("out.dat", *fstream*::binary | *fstream*::out);

if(iFile.is\_open() && oFile.is\_open()){

*Stadium* stadium;

*bool* founded = false;

while (iFile.read((*char*\*)&stadium, sizeof(*Stadium*))){

if(strcmp(stadium.name, *name*)!=0){

oFile.write((*char*\*)&stadium, sizeof(*Stadium*));

} else{

founded = true;

}

}

iFile.close();

oFile.close();

if(founded){

*fstream* iFile("stadiums.dat", *fstream*::binary | *fstream*::out);

*fstream* oFile("out.dat", *fstream*::binary | *fstream*::in);

if(oFile.is\_open() && iFile.is\_open()){

while(oFile.read((*char*\*)&stadium, sizeof(*Stadium*))){

iFile.write((*char*\*)&stadium, sizeof(*Stadium*));

}

oFile.close();

iFile.close();

}else{

cout << "error2 opening file" << endl;

return;

}

}else{

cout << "there is no stadiums with this name" << endl;

return;

}

}else{

cout << "error opening files" << endl;

return;

}

return;

}

*void* addEl(*Stadium*& *stadium*, *int* *number*){

*fstream* iFile("stadiums.dat", *fstream*::binary | *fstream*::in);

*fstream* oFile("out.dat", *fstream*::binary | *fstream*::out);

if(iFile.is\_open() && oFile.is\_open()){

*int* current\_number = 0;

*Stadium* stad;

while(iFile.read((*char*\*)&stad, sizeof(*Stadium*))){

if(current\_number==*number*){

oFile.write((*char*\*)&*stadium*, sizeof(*Stadium*));

}

oFile.write((*char*\*)&stad, sizeof(*Stadium*));

current\_number++;

}

if(current\_number==*number*){

oFile.write((*char*\*)&*stadium*, sizeof(*Stadium*));

}

iFile.close();

oFile.close();

*fstream* iFile("stadiums.dat", *fstream*::binary | *fstream*::out);

*fstream* oFile("out.dat", *fstream*::binary | *fstream*::in);

if(oFile.is\_open() && iFile.is\_open()){

while(oFile.read((*char*\*)&*stadium*, sizeof(*Stadium*))){

iFile.write((*char*\*)&*stadium*, sizeof(*Stadium*));

}

oFile.close();

iFile.close();

}else{

cout << "error2 opening file" << endl;

return;

}

}else{

cout << "error opening files" << endl;

return;

}

}

*Блок №2. Код до програми №2 VNS Lab 8 - Task 1*

Завдання №3 VNS Lab 9 - Task 1

#include <iostream>

#include <string>

#include <fstream>

using *namespace* std;

*void* form(){

*fstream* f("file1.txt", *fstream*::out);

*string* line;

for(*int* i = 0; i < 10; i++){

getline(cin, line);

f << line << endl;

}

f.close();

}

*void* printFile(const *char*\* *name*){

*fstream* f(*name*, *fstream*::in);

if(f.is\_open()){

*string* line;

while (!f.eof()){

line = "";

getline(f, line);

cout << line << endl;

}

f.close();

} else{

cout << "file does not exist" << endl;

return;

}

};

*void* copyLine(*int* *n1*, *int* *n2*){

*fstream* f1("file1.txt", *fstream*::in);

*fstream* f2("file2.txt", *fstream*::out | *fstream*::app);

if(f1.is\_open() && f2.is\_open()){

*string* line;

*int* index=0;

while(getline(f1, line)){

++index;

if(index >= *n1* && index <= *n2*){

if((line.find('A')==*string*::npos)&&(line.find('a')==*string*::npos)){

f2 << line << endl;

}

}

}

f1.close();

f2.close();

} else{

cout << "error" << endl;

return;

}

};

*void* golos(){

*fstream* f("file2.txt", *fstream*::in);

if(f.is\_open()){

*string* line;

*int* count, index = 1;

*int* max\_count = 0, max\_index = 1;

while (!f.eof()){

count = 0;

line = "";

getline(f, line);

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

line[i] = tolower(line[i]);

if(line[i] == 'a' || line[i] == 'e' || line[i] == 'u' || line[i] == 'i' || line[i] == 'o'){

count++;

}

}

if(count >= max\_count){

max\_count = count;

max\_index = index;

}

index++;

}

cout << max\_index << endl;

f.close();

} else{

cout << "file does not exist" << endl;

return;

}

};

*int* main(){

form();

cout << endl << "created file:" << endl;

printFile("file1.txt");

*int* N1, N2;

cout << "enter the number of lines:" << endl;

cin >> N1 >> N2;

copyLine(N1, N2);

printFile("file2.txt");

golos();

*fstream* f2("file2.txt", *fstream*::out | *fstream*::trunc);

f2.close();

return 0;

}

*Блок №3. Код до програми №3 VNS Lab 9 Task 1*

Завдання №4 Algotester Lab 4

way 1

#include <iostream>

using *namespace* std;

*int* main(){

*int* N, K;

cin >> N >> K;

*int* \*a = new *int* [N];

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

cin >> a[i];

}

*int* new\_N = N;

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

for(*int* j = i+1; j < N; j++){

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

new\_N -= 1;

*int* \*arr = new *int* [new\_N];

for(*int* k = 0, ind=0; k < new\_N; k++, ind++){

if(ind == i){

k--;

} else{

arr[k]=a[ind];

}

}

delete[] a;

a = arr;

N = new\_N;

i--;

j--;

}

}

}

for(*int* i = 0; i < N-1; i++){

for(*int* j = 0; j < N-i-1; j++){

if(a[j]>a[j+1]){

*int* temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

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

cerr << a[i] << " ";

}

cerr << endl;

*int* \*new\_a = new *int* [N];

if(K>N){

K%=N;

}

for(*int* i = 0, j = K; j < N; i++, j++){

new\_a[i]=a[j];

}

for(*int* i = N-K, j = 0; i < N; i++, j++){

new\_a[i]=a[j];

}

cout << N << endl;

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

cout << new\_a[i] << " ";

}

cout << endl;

delete[] a;

delete[] new\_a;

return 0;

}

*Блок №4. Код до програми №4 Algotester Lab 4*

way 2

#include <iostream>

#include <algorithm>

#include <vector>

using *namespace* std;

*int* main(){

*int* N, K;

cin >> N >> K;

*vector*<*int*>a (N);

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

cin >> a[i];

}

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

*auto* new\_a = unique(a.begin(),a.end());

a.erase(new\_a, a.end());

if(K>a.size())

K%=a.size();

rotate(a.begin(), a.begin()+K, a.end());

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

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

cout << a[i] << " ";

}

return 0;

}

*Блок №5. Код до програми №4 Algotester Lab 4*

Завдання №5 Algotester Lab 6

#include <iostream>

#include <string>

#include <algorithm>

using *namespace* std;

*bool* cond\_Q(*int* *x*, *int* *y*, *char* *board*[8][8]);

*bool* cond\_N(*int* *x*, *int* *y*, *char* *board*[8][8]);

*bool* cond\_R(*int* *x*, *int* *y*, *char* *board*[8][8]);

*bool* cond\_B(*int* *x*, *int* *y*, *char* *board*[8][8]);

*bool* cond\_P(*int* *x*, *int* *y*, *char* *board*[8][8]);

*bool* cond\_K(*int* *x*, *int* *y*, *char* *board*[8][8]);

*int* main(){

*char* board[8][8];

for(*int* i = 0; i < 8; i++){

for(*int* j = 0; j < 8; j++){

cin >> board[i][j];

}

}

*int* req, x, y;

cin >> req;

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

cin >> x >> y;

x--;

y--;

if(board[x][y]!='O'){

cout << "X" << endl;

} else{

*string* figures="";

if(cond\_Q(x, y, board)){

figures += "Q";

}

if(cond\_N(x, y, board)){

figures+="N";

}

if(cond\_R(x, y, board)){

figures+="R";

}

if(cond\_B(x, y, board)){

figures+="B";

}

if(cond\_P(x, y, board)){

figures+="P";

}

if(cond\_K(x, y, board)){

figures+="K";

}

if(!cond\_Q(x, y, board) && !cond\_N(x, y, board) && !cond\_R(x, y, board) && !cond\_B(x, y, board) && !cond\_P(x, y, board) && !cond\_K(x, y, board))

figures+="O";

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

cout << figures << endl;

}

}

return 0;

}

*bool* cond\_Q(*int* *x*, *int* *y*, *char* *board*[8][8]){

//по діагоналях

for(*int* i = *x*+1, j = *y*+1; i < 8 && j<8; i++, j++){

if(*board*[i][j]=='Q')

return true;

}

for(*int* i = *x*-1, j = *y*-1; i>= 0 && j>=0; i--, j--){

if(*board*[i][j]=='Q')

return true;

}

for(*int* i = *x*-1, j = *y*+1; i >= 0 && j<8; i--, j++){

if(*board*[i][j]=='Q')

return true;

}

for(*int* i = *x*+1, j = *y*-1; i < 8 && j>=0; i++, j--){

if(*board*[i][j]=='Q')

return true;

}

//по вертикалі

for(*int* i = 0, j = *y*; i < 8; i++){

if(*board*[i][j]=='Q')

return true;

}

//по горизонталі

for(*int* i = *x*, j = 0; j<8; j++){

if(*board*[i][j]=='Q')

return true;

}

return false;

}

*bool* cond\_N(*int* *x*, *int* *y*, *char* *board*[8][8]){

if((*x*-1>=0 && *y*-2>=0 && *board*[*x*-1][*y*-2]=='N') ||

(*x*-2>=0 && *y*-1>=0 && *board*[*x*-2][*y*-1]=='N') ||

(*x*-2>=0 && *y*+1<8 && *board*[*x*-2][*y*+1]=='N') ||

(*x*-1>=0 && *y*+2<8 && *board*[*x*-1][*y*+2]=='N') ||

(*x*+1<8 && *y*+2<8 && *board*[*x*+1][*y*+2]=='N') ||

(*x*+2<8 && *y*+1<8 && *board*[*x*+2][*y*+1]=='N') ||

(*x*+2<8 && *y*-1>=0 && *board*[*x*+2][*y*-1]=='N') ||

(*x*+1<8 && *y*-2>=0 && *board*[*x*+1][*y*-2]=='N')){

return true;

}

return false;

};

*bool* cond\_R(*int* *x*, *int* *y*, *char* *board*[8][8]){

//по вертикалі

for(*int* i = 0, j = *y*; i < 8; i++){

if(*board*[i][j]=='R')

return true;

}

//по горизонталі

for(*int* i = *x*, j = 0; j<8; j++){

if(*board*[i][j]=='R')

return true;

}

return false;

}

*bool* cond\_B(*int* *x*, *int* *y*, *char* *board*[8][8]){

//по діагоналях

for(*int* i = *x*+1, j = *y*+1; i < 8 && j<8; i++, j++){

if(*board*[i][j]=='B')

return true;

}

for(*int* i = *x*-1, j = *y*-1; i>= 0 && j>=0; i--, j--){

if(*board*[i][j]=='B')

return true;

}

for(*int* i = *x*-1, j = *y*+1; i >= 0 && j<8; i--, j++){

if(*board*[i][j]=='B')

return true;

}

for(*int* i = *x*+1, j = *y*-1; i < 8 && j>=0; i++, j--){

if(*board*[i][j]=='B')

return true;

}

return false;

}

*bool* cond\_P(*int* *x*, *int* *y*, *char* *board*[8][8]){

if(*x*-1>=0 && *y*-1>=0 && *board*[*x*-1][*y*-1]=='P')

return true;

if(*x*-1>=0 && *y*+1<8 && *board*[*x*-1][*y*+1]=='P')

return true;

return false;

}

*bool* cond\_K(*int* *x*, *int* *y*, *char* *board*[8][8]){

if(*x*-1 >= 0 && *y*-1>=0 && *board*[*x*-1][*y*-1]=='K')

return true;

if(*x*-1 >= 0 && *board*[*x*-1][*y*]=='K')

return true;

if(*x*-1 >= 0 && *y*+1<8 && *board*[*x*-1][*y*+1]=='K')

return true;

if(*y*-1>=0 && *board*[*x*][*y*-1]=='K')

return true;

if(*y*+1<8 && *board*[*x*][*y*+1]=='K')

return true;

if(*x*+1<8 && *y*-1>=0 && *board*[*x*+1][*y*-1]=='K')

return true;

if(*x*+1<8 && *board*[*x*+1][*y*]=='K')

return true;

if(*x*+1<8 && *y*+1<8 && *board*[*x*+1][*y*+1]=='K')

return true;

return false;

}

*Блок №6. Код до програми №5 Algotester Lab 6*

Завдання №6 ClassPractice Work

Task 1

#include <iostream>

#include <fstream>

#include <string>

using *namespace* std;

*enum* *FileOpResult*{Success, Failure};

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

*int* main(){

*char* name[] = "file1.txt";

*char* content[100];

cout << "enter the content, which you want to write to file" << endl;

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

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

if(res == Success){

cout << "The operation was successful!" << endl;

*int* choice;

cout << "Do you want print the file?\n1 - yes\nother - no" << endl;

cin >> choice;

if(choice == 1){

*fstream* file(name, *fstream*::in);

if(file.is\_open()){

*string* line;

while(getline(file, line)){

cout << line << endl;

}

} else{

cout << "error" << endl;

}

}

} else{

cout << "The operation was failed!" << endl;

}

return 0;

}

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

*fstream* file(*name*, *fstream*::out);

if(file.is\_open()){

file << *content*;

file.close();

return Success;

} else{

cout << "error" << endl;

return Failure;

}

}

*Блок №7. Код до програми №6 Class Practice Work Task 1*

Task 2

#include <iostream>

#include <fstream>

#include <string>

using *namespace* std;

*enum* *FileOpResult*{Success, Failure, Notexist};

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

*int* main(){

*char* file\_from[20];

*char* file\_to[20];

cout << "enter the name of file from which you want to copy content:" << endl;

cin.getline(file\_from, sizeof(file\_from));

cout << "enter the name of file in which you want to copy content:" << endl;

cin.getline(file\_to, sizeof(file\_to));

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

if(res == Success){

cout << "the operation was successful" << endl;

*int* choice;

cout << "Do you want to see a new file?\n1 - yes\nother - no" << endl;

cin >> choice;

if(choice == 1){

*fstream* inFile(file\_to, *fstream*::in);

if(inFile.is\_open()){

*string* line;

while(getline(inFile, line)){

cout << line << endl;

}

} else{

cout << "error" << endl;

}

}

} else if(res == Notexist){

cout << file\_from << " does not exist" << endl;

} else{

cout << "the operation failed" << endl;

}

return 0;

}

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

*fstream* inFile(*file\_from*, *fstream*::in);

if(inFile.fail()){

return Notexist;

}

if(inFile.is\_open()){

*fstream* outFile(*file\_to*, *fstream*::out | *fstream*::app);

if(outFile.is\_open()){

*string* line;

while(getline(inFile, line)){

outFile << line << endl;

}

return Success;

} else {

return Failure;

}

} else{

return Failure;

}

}

*Блок №8. Код до програми №6 Class Practice Work Task 2*

Завдання №7 Self Practice Work

//задача на алготестері - 0081 Допоможе чи заб'є?

#include <iostream>

#include <string>

using *namespace* std;

*int* main(){

*string* str;

*int* number;

cin >> number;

cin >> str;

*int* current\_n=0;

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

if(str[i]=='T' && i+5<str.size()){

if(str[i+1]=='O' && str[i+2]=='I' && str[i+3]=='L' && str[i+4]=='E' && str[i+5]=='T'){

current\_n++;

i+=5;

}

}

}

if(current\_n>=number){

cout << "YES" << endl;

} else{

cout << "NO" << endl;

}

return 0;

}

*Блок №9. Код до програми №7 Self Practice Work*

Завдання №8 Self Practice Work

//Задача на алготестері - 0541 Дзідзьо і його пісня

#include <iostream>

#include <string>

using *namespace* std;

*int* main(){

*string* str;

cin >> str;

*string* new\_str;

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

if(str[i]!=str[i+1]){

new\_str+=str[i];

} else{

while(str[i]==str[i+1]){

i++;

}

i--;

}

}

cout << new\_str.size() << endl;

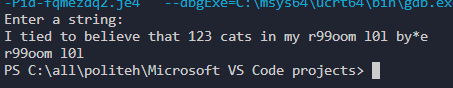
return 0;

}

*Блок №10. Код до програми №8 Self Practice Work*

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

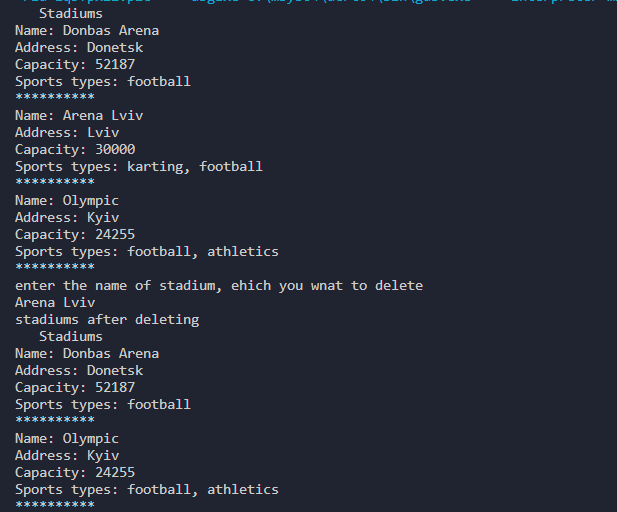
Завдання №1 VNS Lab 6 - Task 1



*Зображення №9. Результати №1 VNS Lab 6 Task 1*

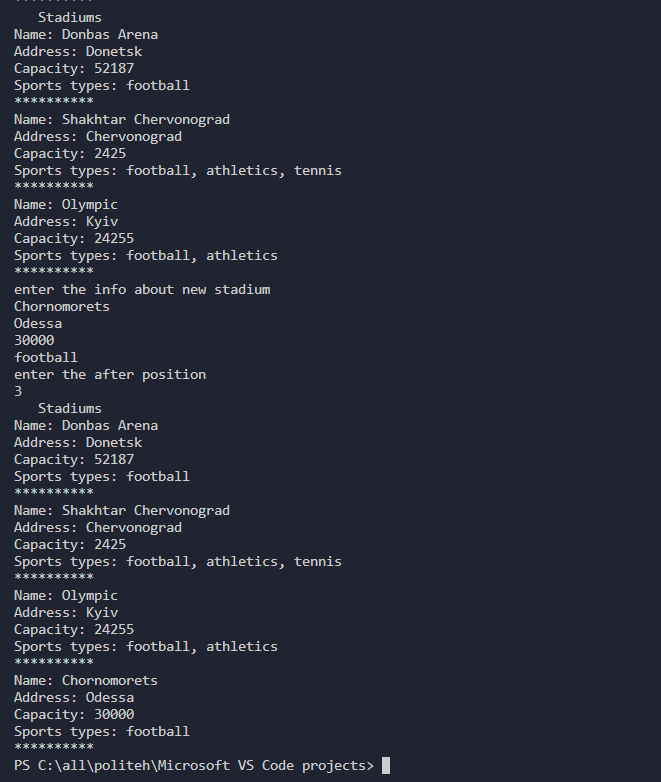
Час затрачений на виконання завдання: 1 год

Завдання №2 VNS Lab 8 - Task 1



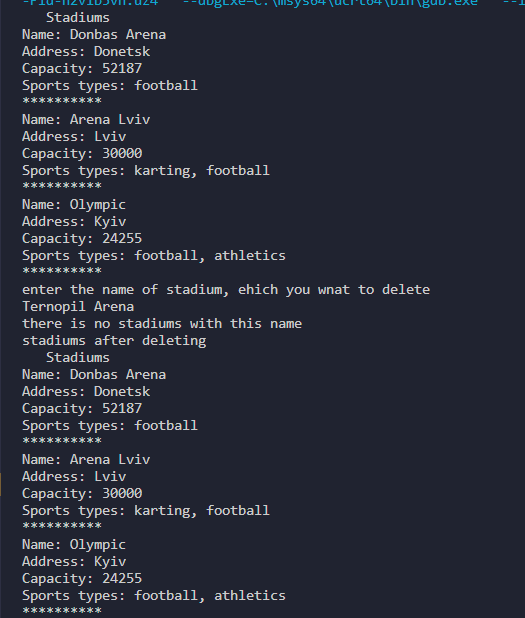
*Зображення №10. Результати №2 VNS Lab 8 Task 1*

Сформував файл з трьох стадіонів та роздрукував його. Далі ввів ім’я стадіону, який захотів видалити з файлу і вивів вмістиме файлу після видалення.



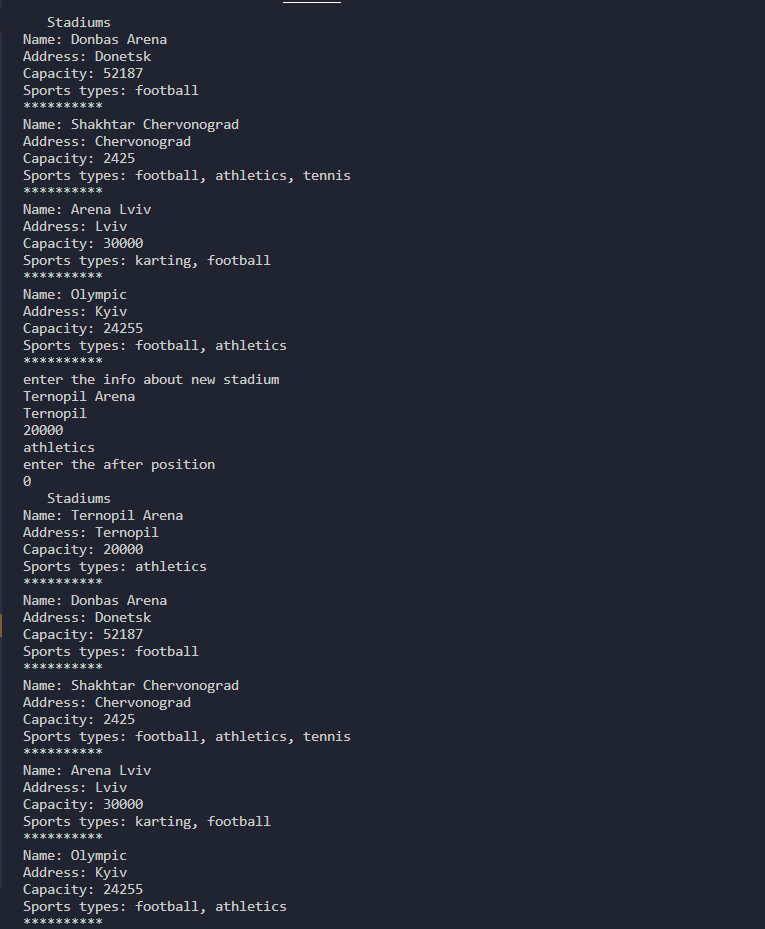
*Зображення №11. Результати №2 VNS Lab 8 Task 1*

Далі вставив готовий стадіон “Шахтар Червоноград”. Попросив користувача ввести інформацію про новий стадіон і номер стадіона, після якого він буде у файлі.



*Зображення №12. Результати №2 VNS Lab 8 Task 1*

Ввів назву стадіона, якого не існує, і програма вивела, що нема стадіону з таким ім’ям, тому програма нічого не видалила.

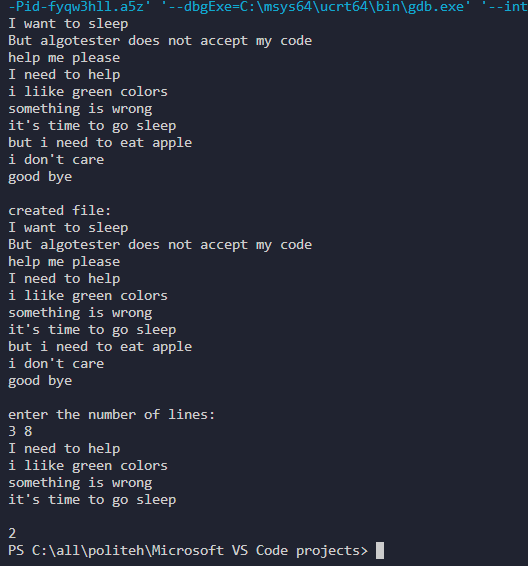


*Зображення №13. Результати №2 VNS Lab 8 Task 1*

Ввів інформацію про новий Стадіон та число 0 (номер елемента, після якого буде розташований новий). Отримали новий список стадіонів і новий стадіон на першому місці.

Час затрачений на виконання завдання: 4 год

Завдання №3 VNS Lab 9 - Task 1

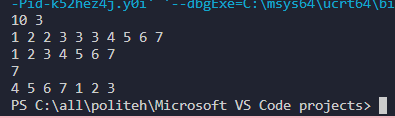


*Зображення №14. Результати №3 VNS Lab 9 Task 1*

Ввів текст з 10 ліній. Вибрав лінії, з якої та по яку копіювати. Рядок №3 програма не скопіювала, бо у слові “please” присутня літера “a”. Рядок №8 програма проігнорувала, бо у слові “apple” присутня літера “a”. Потім програма вивела число 2 - це рядок у новому файлі, де найбільше голосних.

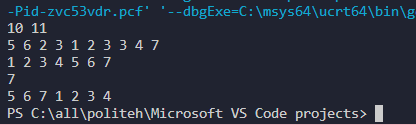
Час затрачений на виконання завдання: 1 год

Завдання №4 Algotester Lab 4



*Зображення №15. Результати №4 Algotester Lab 4*

Програма видалила дублікати та поставила 3 перших елементи в кінець

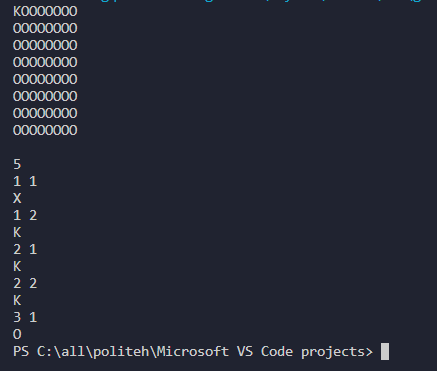


*Зображення №16. Результати №4 Algotester Lab 4*

Програма видалила дублікати, посортувала та 4 перших елемене переставила в кінець масив (бо 11%7=4)

Час затрачений на виконання завдання: 2,5 год

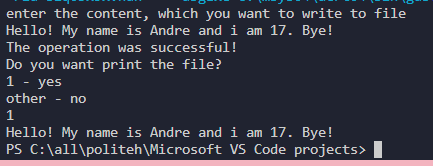
Завдання №5 Algotester Lab 6



*Зображення №16. Результати №5 Algotester Lab 6*

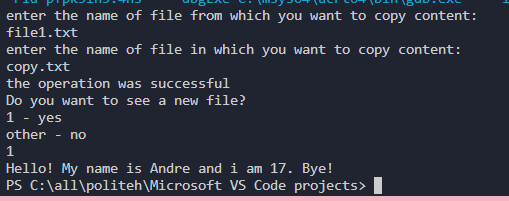
Час затрачений на виконання завдання: 4 год

Завдання №6 ClassPractice Work



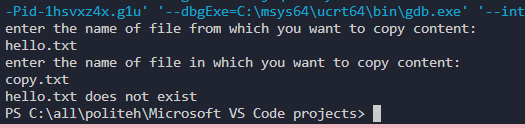
*Зображення №17. Результати №6 Class Practice Work*

Записав у файл текст та попросив вивести його



*Зображення №18. Результати №6 Class Practice Work*

Скопіював цей текст з файлу в інший файл, та попросив вивести вмістиме іншого файлу після копіювання.

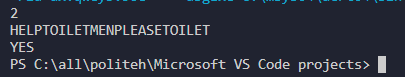


*Зображення №19. Результати №6 Class Practice Work*

Хотів скопіювати з файлу, якого не існує.

Час затрачений на виконання завдання: 1,5 год

Завдання №7 Self Practice Work

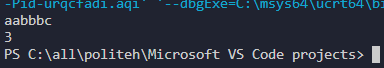


*Зображення №20. Результати №7 Self Practice Work*

“TOILET” у рядку зустрічається достатню кількість разі, тому виводить YES.

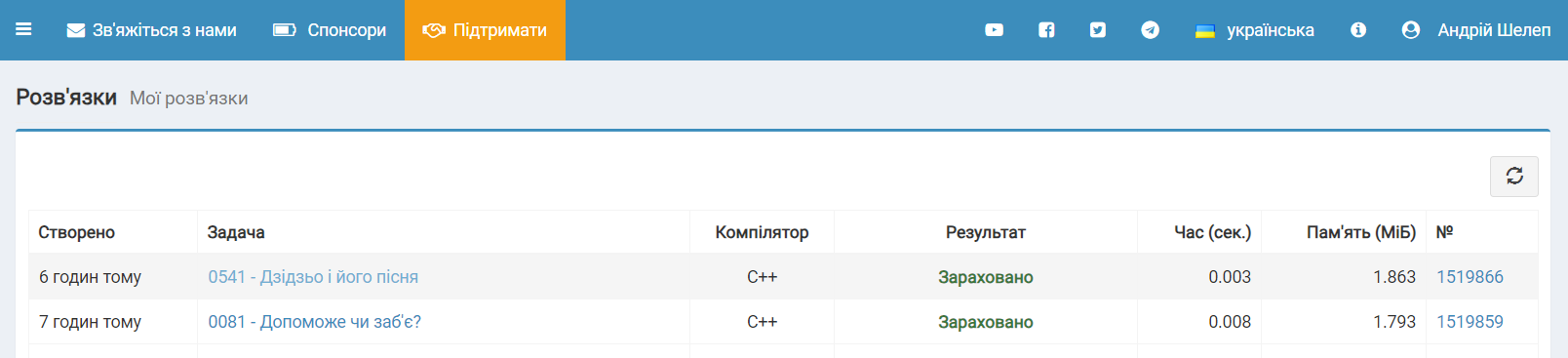
Час затрачений на виконання завдання: 25 хв

Завдання №8 Self Practice Work



*Зображення №21. Результати №8 Self Practice Work*

Якщо забрати букву “a”, то буде abbbc. Якщо забрати букву “b” то буде aabbc. Якщо забрати букву “c”, то буде aabbb.



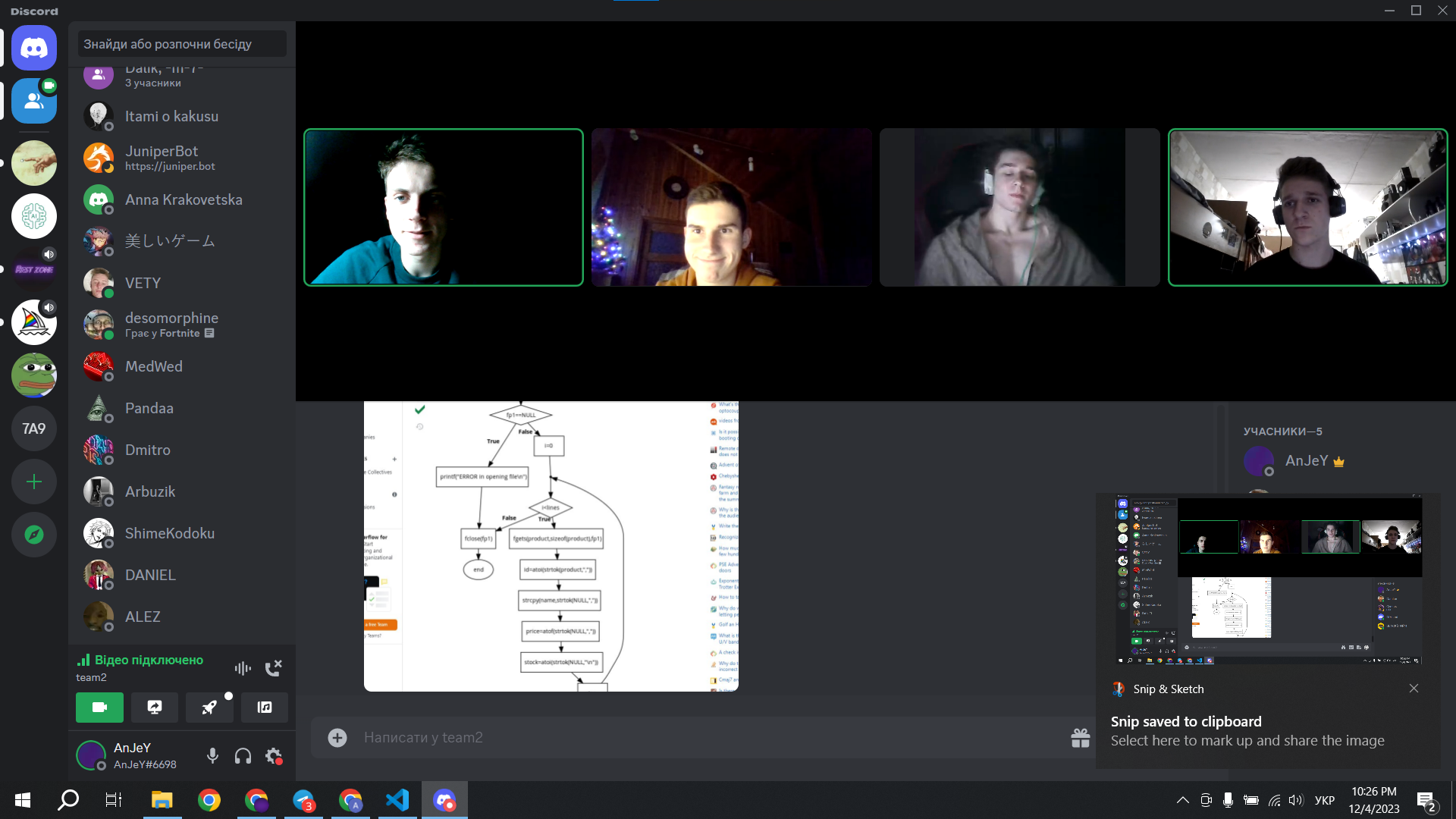
*Зображення №22. Результати №7-8 Self Practice Work*

Прийнято на алготестері

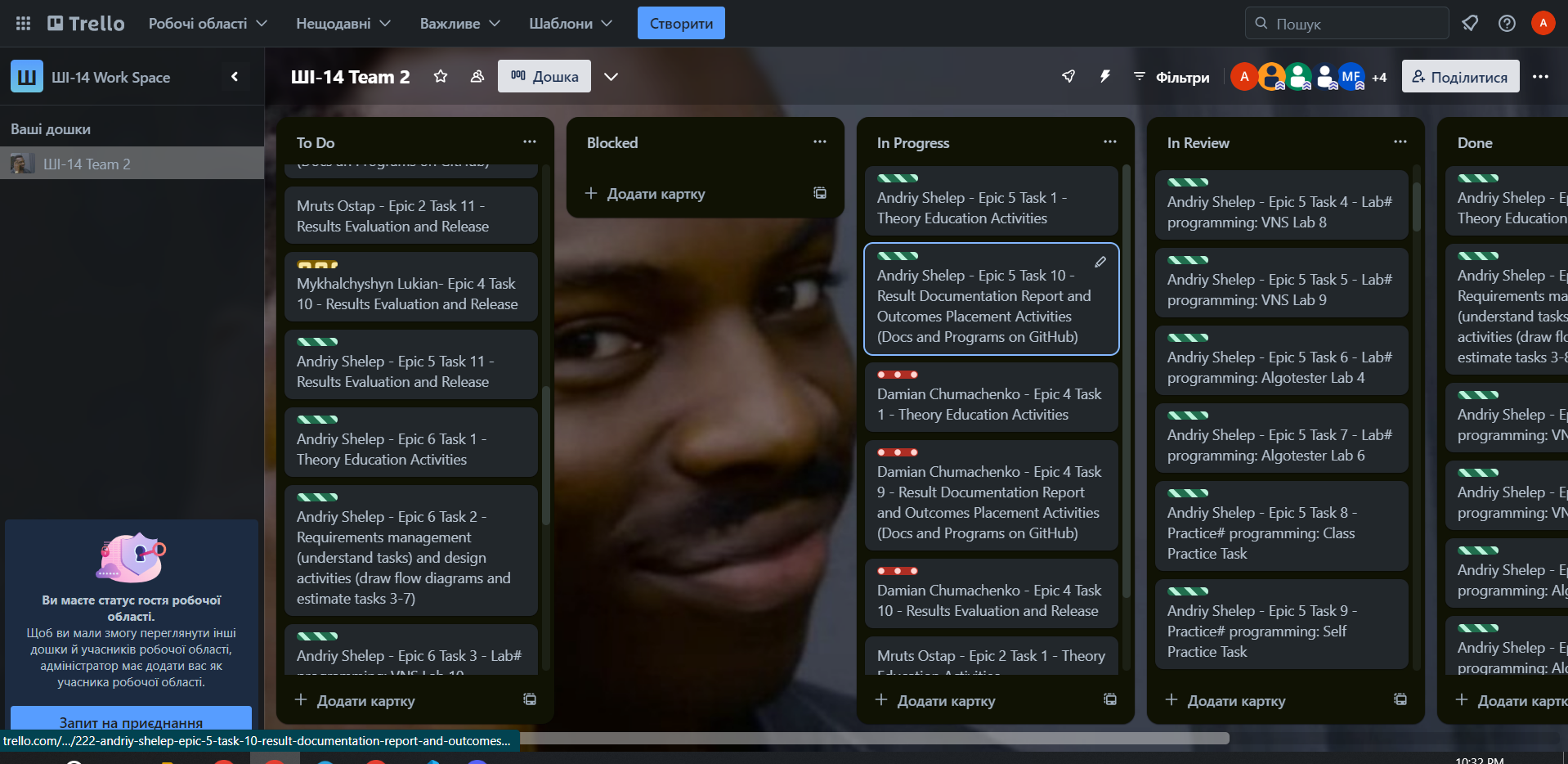
Час затрачений на виконання завдання: 20 хв

## **6. Кооперація з командою:**

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



*Зображення №23. Кооперація з командою*



*Зображення №24. Кооперація з командо**ю*

# **Висновки:**

Ознайомився з основними методами та деталями роботи з файлами, запису у файл і зчитування з нього у С++. Застосував ці знання, виконуючи завдання лабораторних робіт з ВНС та практичного завдання. Також опрацював рядкові змінні і символи використовував їх у завданнях з алготестера та у лабораторних з ВНС.