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 та Рядкові Змінні. Текстові файли. Стандартна бібліотека. Створення й використання бібліотек.»

***Виконав:***

студент групи ШІ-14

Іжелюк Назарій Миколайович

# **Тема роботи:**

Робота з файлами та структурами в C++: створення, копіювання та опрацювання даних.

# **Мета роботи:**

Метою цієї лабораторної роботи є ознайомлення з основними принципами роботи з файлами та структурами в мові програмування C++. Під час виконання лабораторної роботи студенти будуть вивчати процес створення, читання, запису та копіювання інформації із файлів, а також роботу із складними типами даних, такими як структури.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Робота з файлами.
* Тема №2: Стандартна бібліотека для роботи з файлами.
* Тема №3: Динамічні структури даних.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Робота з файлами.
  + Джерела Інформації
    - Відео: <http://surl.li/oefpd>
    - Стаття: <http://cpp.dp.ua/vykorystannya-fajliv/>
  + Що опрацьовано:
    - Основи роботи із файлами.
    - Ввід та вивід інформації в файл.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 13.11.2023
  + Звершення опрацювання теми: 20.11.2023
* Тема №2: Стандартна бібліотека для роботи з файлами.
  + Джерела Інформації:
    - Стаття: <https://purecodecpp.com/uk/archives/2751>
  + Що опрацьовано:
    - Розглянуто функції стандартної бібліотеки fstream.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 13.11.2023
  + Звершення опрацювання теми: 20.11.2023
* Тема №3: Динамічні структури даних.
  + Джерела Інформації:
    - Стаття:<https://allref.com.ua/uk/skachaty/Dinamichni_strukturi_danih_%28S++%29>
    - Лекції.
    - Практичні заняття.
  + Що опрацьовано:
    - Лінійний однонаправлений список.
    - Робота із двонаправленими списками.
    - Структури та рядки.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 13.11.2023
  + Звершення опрацювання теми: 20.11.2023

# **Виконання роботи:**

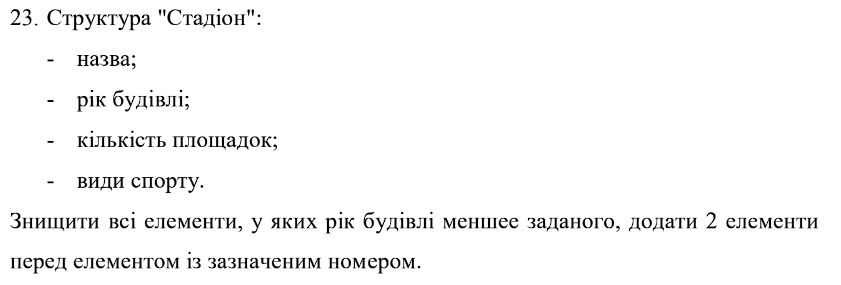
## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS Lab 6

* **Варіант:** 23
* **Деталі завдання:** Для рядка знайти довжину найкоротшого слова.
* **Важливі деталі:** Використати функцію getline().

Завдання №2 VNS Lab 8

* **Варіант:** 23
* **Деталі завдання:**

****

* **Важливі деталі:** Інформація із структури повинна записуватись в файл.

Завдання №3 VNS Lab 9

* **Варіант:** 23
* **Деталі завдання:** Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію
* **Важливі деталі:** 1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких міститься два однакових слова.

2) Визначити номер слова, у якому найбільше букв «А».

Завдання №4 Algotester Lab 4

* **Варіант:** 1
* **Деталі завдання:**

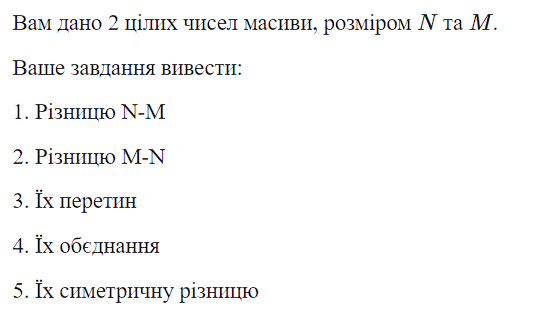
****

Рисунок 1 Завдання Algotester Lab 4

* **Важливі деталі:** Ввід та вивід повинен співпадати умові.

Завдання №5 Algotester Lab 6

* **Варіант: 1**
* **Деталі завдання:**

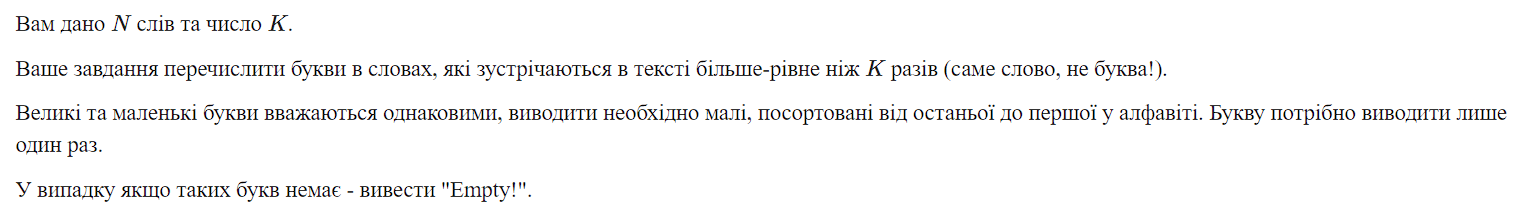


Рисунок 2 Завдання Algotester Lab 6

* **Важливі деталі:** Ввід та вивід повинен співпадати умові.

Завдання №6 Class Practice

* **Варіант: -**
* **Деталі завдання:**

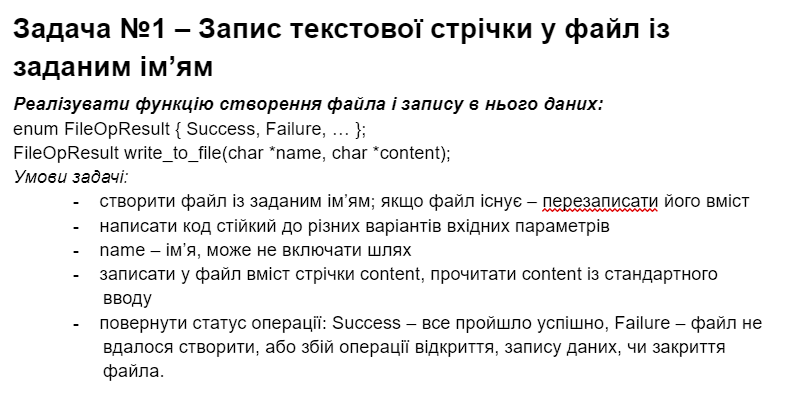


Рисунок 3 Завдання Class Practice(1)

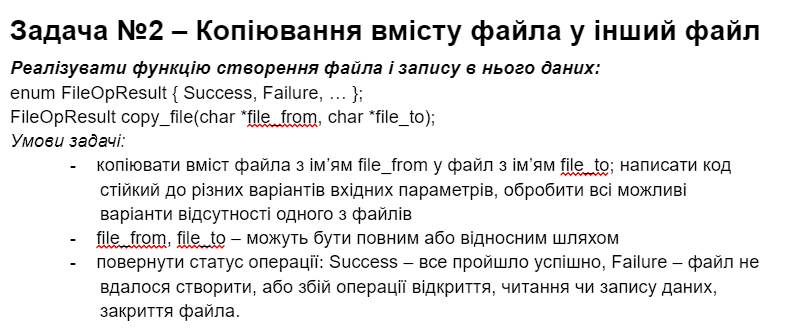


Рисунок 4 Завдання Class Practice(2)

* **Важливі деталі:** Використати enum FileOpResult { Success, Failure }; для передачі поточного стану виконання.

Завдання №7 Self Practice Algoteater Lab 4

* **Варіант:** 2
* **Деталі завдання:**

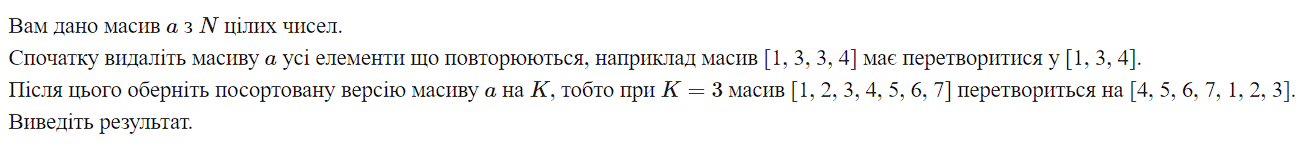


Рисунок 5 Завдання Sеlf Practice

* **Важливі деталі:** Ввід та вивід повинен співпадати умові.

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

Завдання №5 Algotester Lab 6

* Блок-схема:

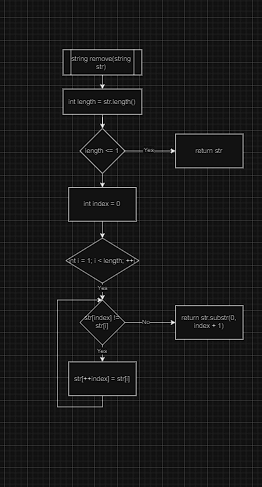


Рисунок 6 Блоксхема функції до завдання Algotester Lab 6

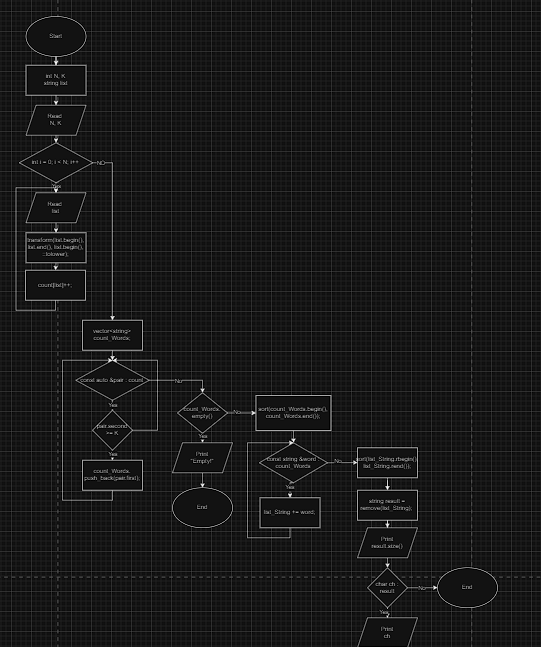


Рисунок 7 Основна програма Algotester Lab 6

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

Конфігурацію середовища не змінював.

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

Завдання №1 VNS Lab 6

#include <iostream>

#include <sstream>

#include <string>

using namespace std;

string FindMinWord(string input) {

istringstream iss(input);

string word;

string min\_word;

min\_word = string(input.length(), ' ');

while (iss >> word) {

if (word.length() < min\_word.length()) {

min\_word = word;

}

}

return min\_word;

}

int main() {

string input;

cout << "Введіть рядок: ";

getline(cin, input);

cout << "Найкоротше слово: " << FindMinWord(input);

return 0;

}

Завдання №2 VNS Lab 8

#include<iostream>

#include<string>

#include<vector>

#include<fstream>

using namespace std;

struct stadium

{

string name;

int year;

int number\_of\_platforms;

vector<string> kinds\_of\_sport;

};

void createFile() {

ofstream file("C:/Users/Lenovo Gaming/Desktop/file.txt");

//Стадіон 1

stadium stadium1;

stadium1.name = "Олімпійський";

stadium1.year = 1923;

stadium1.number\_of\_platforms = 5;

stadium1.kinds\_of\_sport.push\_back("Футбол");

stadium1.kinds\_of\_sport.push\_back("Теніс");

stadium1.kinds\_of\_sport.push\_back("Легка атлетика");

//Стадіон 2

stadium stadium2;

stadium2.name = "Донбас Арена";

stadium2.year = 2009;

stadium2.number\_of\_platforms = 3;

stadium2.kinds\_of\_sport.push\_back("Футбол");

stadium2.kinds\_of\_sport.push\_back("Легка атлетика");

//Стадіон 3

stadium stadium3;

stadium3.name = "Арена Львів";

stadium3.year = 2010;

stadium3.number\_of\_platforms = 3;

stadium3.kinds\_of\_sport.push\_back("Футбол");

//Стадіон 4

stadium stadium4;

stadium4.name = "ЦСК Київ";

stadium4.year = 1965;

stadium4.number\_of\_platforms = 1;

stadium4.kinds\_of\_sport.push\_back("Футбол");

stadium4.kinds\_of\_sport.push\_back("Легка атлетика");

//Вивід стадіону1

file << stadium1.name << endl;

file << stadium1.year << endl;

file << stadium1.number\_of\_platforms << endl;

for(int i = 0; i < stadium1.kinds\_of\_sport.size(); i++){

file << stadium1.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону2

file << stadium2.name << endl;

file << stadium2.year << endl;

file << stadium2.number\_of\_platforms << endl;

for(int i = 0; i < stadium2.kinds\_of\_sport.size(); i++){

file << stadium2.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону3

file << stadium3.name << endl;

file << stadium3.year << endl;

file << stadium3.number\_of\_platforms << endl;

for(int i = 0; i < stadium3.kinds\_of\_sport.size(); i++){

file << stadium3.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону4

file << stadium4.name << endl;

file << stadium4.year << endl;

file << stadium4.number\_of\_platforms << endl;

for(int i = 0; i < stadium4.kinds\_of\_sport.size(); i++){

file << stadium4.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

file.close();

}

void delete\_information(int year){

//Стадіон 1

ofstream file("C:/Users/Lenovo Gaming/Desktop/file1.txt");

stadium stadium1;

stadium1.name = "Олімпійський";

stadium1.year = 1923;

stadium1.number\_of\_platforms = 5;

stadium1.kinds\_of\_sport.push\_back("Футбол");

stadium1.kinds\_of\_sport.push\_back("Теніс");

stadium1.kinds\_of\_sport.push\_back("Легка атлетика");

//Стадіон 2

stadium stadium2;

stadium2.name = "Донбас Арена";

stadium2.year = 2009;

stadium2.number\_of\_platforms = 3;

stadium2.kinds\_of\_sport.push\_back("Футбол");

stadium2.kinds\_of\_sport.push\_back("Легка атлетика");

//Стадіон 3

stadium stadium3;

stadium3.name = "Арена Львів";

stadium3.year = 2010;

stadium3.number\_of\_platforms = 3;

stadium3.kinds\_of\_sport.push\_back("Футбол");

//Стадіон 4

stadium stadium4;

stadium4.name = "ЦСК Київ";

stadium4.year = 1965;

stadium4.number\_of\_platforms = 1;

stadium4.kinds\_of\_sport.push\_back("Футбол");

stadium4.kinds\_of\_sport.push\_back("Легка атлетика");

if(stadium1.year < year){

//Вивід стадіону2

file << stadium2.name << endl;

file << stadium2.year << endl;

file << stadium2.number\_of\_platforms << endl;

for(int i = 0; i < stadium2.kinds\_of\_sport.size(); i++){

file << stadium2.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону3

file << stadium3.name << endl;

file << stadium3.year << endl;

file << stadium3.number\_of\_platforms << endl;

for(int i = 0; i < stadium3.kinds\_of\_sport.size(); i++){

file << stadium3.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону4

file << stadium4.name << endl;

file << stadium4.year << endl;

file << stadium4.number\_of\_platforms << endl;

for(int i = 0; i < stadium4.kinds\_of\_sport.size(); i++){

file << stadium4.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

}else if(stadium2.year < year){

//Вивід стадіону1

file << stadium1.name << endl;

file << stadium1.year << endl;

file << stadium1.number\_of\_platforms << endl;

for(int i = 0; i < stadium1.kinds\_of\_sport.size(); i++){

file << stadium1.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону3

file << stadium3.name << endl;

file << stadium3.year << endl;

file << stadium3.number\_of\_platforms << endl;

for(int i = 0; i < stadium3.kinds\_of\_sport.size(); i++){

file << stadium3.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону4

file << stadium4.name << endl;

file << stadium4.year << endl;

file << stadium4.number\_of\_platforms << endl;

for(int i = 0; i < stadium4.kinds\_of\_sport.size(); i++){

file << stadium4.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

}else if(stadium3.year < year){

//Вивід стадіону1

file << stadium1.name << endl;

file << stadium1.year << endl;

file << stadium1.number\_of\_platforms << endl;

for(int i = 0; i < stadium1.kinds\_of\_sport.size(); i++){

file << stadium1.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону2

file << stadium2.name << endl;

file << stadium2.year << endl;

file << stadium2.number\_of\_platforms << endl;

for(int i = 0; i < stadium2.kinds\_of\_sport.size(); i++){

file << stadium2.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону4

file << stadium4.name << endl;

file << stadium4.year << endl;

file << stadium4.number\_of\_platforms << endl;

for(int i = 0; i < stadium4.kinds\_of\_sport.size(); i++){

file << stadium4.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

}else if(stadium4.year < year){

//Вивід стадіону1

file << stadium1.name << endl;

file << stadium1.year << endl;

file << stadium1.number\_of\_platforms << endl;

for(int i = 0; i < stadium1.kinds\_of\_sport.size(); i++){

file << stadium1.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону2

file << stadium2.name << endl;

file << stadium2.year << endl;

file << stadium2.number\_of\_platforms << endl;

for(int i = 0; i < stadium2.kinds\_of\_sport.size(); i++){

file << stadium2.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

//Вивід стадіону3

file << stadium3.name << endl;

file << stadium3.year << endl;

file << stadium3.number\_of\_platforms << endl;

for(int i = 0; i < stadium3.kinds\_of\_sport.size(); i++){

file << stadium3.kinds\_of\_sport[i] << " ";

}

file << endl << endl;

}

}

int main(){

int year;

createFile();

cout << "Введіть рік: ";

cin >> year;

delete\_information(year);

return 0;

}

Завдання №3 VNS Lab 9

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

using namespace std;

void similar\_words(){

ifstream file1("C:/Users/Lenovo Gaming/Desktop/F1.txt");

ofstream file2("C:/Users/Lenovo Gaming/Desktop/F2.txt");

string line;

while (getline(file1, line)) {

istringstream iss(line);

string word;

iss >> word;

while (iss >> word) {

size\_t found = line.find(word);

if (found != string::npos && line.find(word, found + 1) != string::npos) {

file2 << line << endl;

break;

}

}

}

}

int find\_A() {

ifstream file1("C:/Users/Lenovo Gaming/Desktop/F1.txt");

string line;

int wordNumber = 0;

int max\_A\_count = 0;

int word\_with\_most\_A\_number = 0;

while (getline(file1, line)) {

istringstream iss(line);

string word;

wordNumber = 0;

while (iss >> word) {

int count = 0;

for (char letter : word) {

if (letter == 'A' || letter == 'a') {

count++;

}

}

wordNumber++;

if (count > max\_A\_count) {

max\_A\_count = count;

word\_with\_most\_A\_number = wordNumber;

}

}

}

return word\_with\_most\_A\_number;

}

int main(){

string text;

ofstream file1("C:/Users/Lenovo Gaming/Desktop/F1.txt");

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

getline(cin, text);

file1 << text;

file1 << endl;

text = "";

}

similar\_words();

int wordNumberWithMostA = find\_A();

cout << wordNumberWithMostA << endl;

return 0;

}

Завдання №4 Algotester Lab 4

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

int N, M;

cin >> N;

int arN[N];

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

cin >> arN[i];

}

cin >> M;

int arM[M];

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

cin >> arM[i];

}

sort(arN, arN + N);

sort(arM, arM + M);

// Знаходимо різницю між масивами

int differenceN[N];

auto itN = set\_difference(arN, arN + N, arM, arM + M, differenceN);

int diff\_Count\_N = itN - differenceN;

//різниця N-M

cout << diff\_Count\_N << endl;

for (int i = 0; i < diff\_Count\_N; ++i) {

cout << differenceN[i] << " ";

}

cout << endl;

//Знаходимо різницю між M-N

int differenceM[M];

auto itM = set\_difference(arM, arM + M, arN, arN + N, differenceM);

int diff\_Count\_M = itM - differenceM;

cout << diff\_Count\_M << endl;

for (int i = 0; i < diff\_Count\_M; i++){

cout << differenceM[i] << " ";

}

cout << endl;

//Перетин

int intersection[N + M];

auto it\_intersection = set\_intersection(arN, arN + N, arM, arM + M, intersection);

int inter\_Count = it\_intersection - intersection;

cout << inter\_Count << endl;

for (int i = 0; i < inter\_Count; i++){

cout << intersection[i] << " ";

}

cout << endl;

//Обєднання

int unionn[N + M];

auto uni = set\_union(arN, arN + N, arM, arM + M, unionn);

int unionn\_Count = uni - unionn;

cout << unionn\_Count << endl;

for (int i = 0; i < unionn\_Count; ++i) {

cout << unionn[i] << " ";

}

cout << endl;

//Симетрична різниця

int symmetric\_diff[N + M];

auto sym\_diff = set\_symmetric\_difference(arN, arN + N, arM, arM + M, symmetric\_diff);

int sym\_diff\_Count = sym\_diff - symmetric\_diff;

cout << sym\_diff\_Count << endl;

for (int i = 0; i < sym\_diff\_Count; ++i) {

cout << symmetric\_diff[i] << " ";

}

cout << endl;

}

Завдання №5 Algotester Lab 6

#include <iostream>

#include <algorithm>

#include <string>

#include <vector>

#include <unordered\_map>

using namespace std;

string remove(string str) {

int length = str.length();

if (length <= 1) {

return str;

}

int index = 0;

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

if (str[index] != str[i]) {

str[++index] = str[i];

}

}

return str.substr(0, index + 1);

}

int main() {

int N, K;

cin >> N >> K;

unordered\_map<string, int> count;

string list;

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

cin >> list;

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

count[list]++;

}

vector<string> count\_Words;

for (const auto &pair : count) {

if (pair.second >= K) {

count\_Words.push\_back(pair.first);

}

}

if (count\_Words.empty()) {

cout << "Empty!";

return 0;

}

sort(count\_Words.begin(), count\_Words.end());

string list\_String;

for (const string &word : count\_Words) {

list\_String += word;

}

sort(list\_String.rbegin(), list\_String.rend());

string result = remove(list\_String);

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

for (char ch : result) {

cout << ch << " ";

}

return 0;

}

Завдання №6 Class Practice

#include<iostream>

#include<fstream>

#include<string>

using namespace std;

enum FileOpResult {

Success,

Failure

};

FileOpResult file\_create() {

ofstream file("C:/Users/Lenovo Gaming/Desktop/file\_from.txt");

string input;

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

return Failure;

}

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

getline(cin, input);

file << input << endl;

}

file.close();

return Success;

}

FileOpResult file\_copy() {

ifstream file\_from("C:/Users/Lenovo Gaming/Desktop/file\_from.txt");

ofstream file\_to("C:/Users/Lenovo Gaming/Desktop/file\_to.txt");

if (!file\_from.is\_open()) {

return Failure;

}

char text;

while (file\_from.get(text)) {

file\_to.put(text);

}

file\_from.close();

file\_to.close();

return Success;

}

int main() {

FileOpResult result1 = file\_create();

FileOpResult result2 = file\_copy();

if (result1 == Success && result2 == Success) {

cout << "Success ";

}

else {

cout << "Failure ";

}

return 0;

}

Завдання №7 Self Practice Algoteater Lab 4

#include<iostream>

#include<string>

#include<vector>

#include<algorithm>

using namespace std;

int main() {

int N, K;

cin >> N >> K;

vector<int> aray(N);

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

cin >> aray[i];

}

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

auto last = unique(aray.begin(), aray.end());

aray.erase(last, aray.end());

rotate(aray.begin(), aray.begin() + K % aray.size(), aray.end());

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

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

cout << aray[i] << " ";

}

return 0;

}

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

Завдання №1 Введено рядок і отримано найкоротше слово



Рисунок Тестування програми 1

**Час затрачений на виконання завдання:** 30хв.

Завдання №2 Створено 2 Файли в першому відображено всю інформацію про стадіони, а в другому додано інформацію в залежності від введеного року.

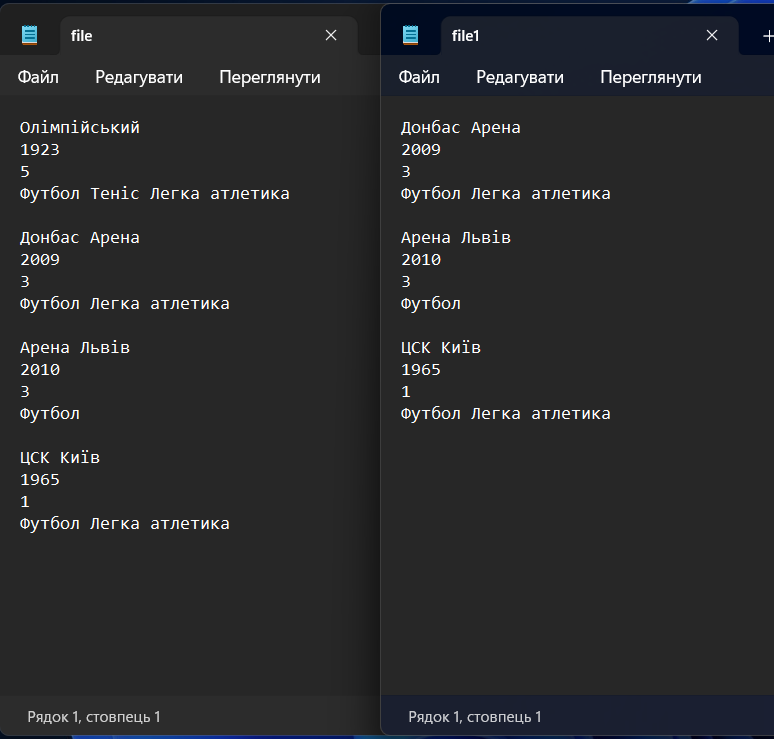


Рисунок Тестування програми 2

**Час затрачений на виконання завдання:** 2год

Завдання №3 В першому файлі відображено ввід інформації в програму, а в другому рядок в якому є однакові слова

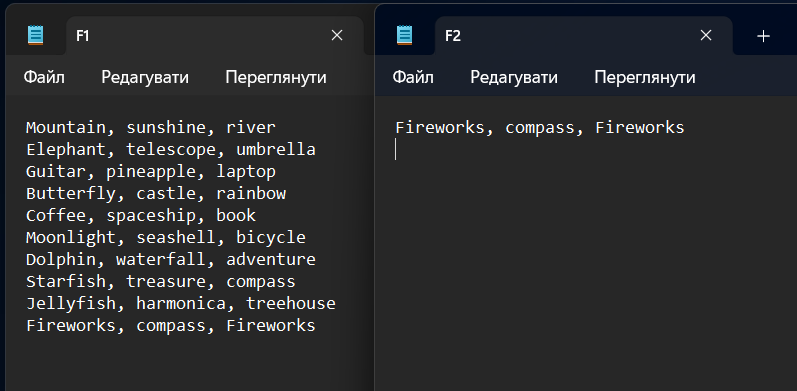


Рисунок Тестування програми 3

**Час затрачений на виконання завдання:** 1 год.

Завдання №4 Програму перевірено на плотформі Algotester.



Рисунок Тестування програми 4

**Час затрачений на виконання завдання:** 1 год.

Завдання №5 Програму перевірено на плотформі Algotester.



Рисунок Тестування програми 5

**Час затрачений на виконання завдання:** 1год

Завдання №6 Створено файл file\_from в якій записано певні дані, а в файл file\_to вони скопійовані.

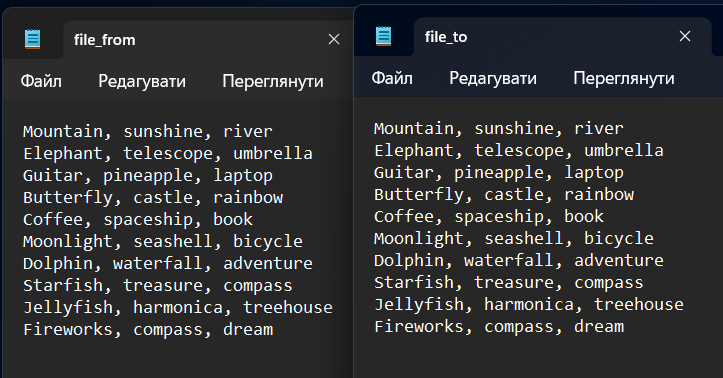


Рисунок Тестування програми 6

**Час затрачений на виконання завдання:** 1год.

Завдання №7 Програму перевірено на плотформі Algotester.



Рисунок Тестування програми 7

Час затрачений на виконання завдання: 30 хв.

# **Висновки:**

Під час лабораторних та практичних робіт №5 я освоїв низку важливих навичок. Навчився створювати файли, оновив свої знання про різні системи числення, дослідив методи запису та читання інформації з файлів та вивчив застосування різних бібліотек.