Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

**про виконання лабораторних та практичних робіт блоку № 5**

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

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

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт №5

***Виконав:***

студент групи ШІ-13

Кузьо Іван Сергіович

# **Тема роботи:**

# Бібліотека fstream та файли, стрічки

# **Мета роботи:**

Task 1 - Theory Education Activities

Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-9)

Task 3 - Lab# programming: VNS Lab 6

Task 4 - Lab# programming: VNS Lab 8

Task 5 - Lab# programming: VNS Lab 9

Task 6 - Lab# programming: Algotester Lab 4

Task 7 - Lab# programming: Algotester Lab 6

Task 8 - Practice# programming: Class Practice Task

Task 9  - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

Task 10 - Results Evaluation and Release

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1:бібліотека fstream та файли
* Тема №2: стрічки.

1. Індивідуальний план опрацювання теорії:

* Тема №1: бібліотека fstream.
  + Джерела Інформації

Чат Гпт, лекції Олександра Пшеничного, методички з лаборатоних,практика написання коду, C++ reference, власна практика роботи з файлами.

Що опрацьовано:

Створення файлів для запису (ofstream) та зчитування з файлу (ifstream).Перевірка їх на успіше відкриття через is\_open та закриття close() . Робота з бінарними файлами ios та звичайними.

Копіювання файлів та виделення з них інформації.

* + Статус: Ознайомлений
  + Початок опрацювання теми:4/12
  + Звершення опрацювання теми: 10/12
* Тема №2 Стрічки.
  + Джерела Інформації

Чат Гпт, лекції Олександра Пшеничного, методички з лаборатоних,практика написання коду, C++ reference, пояснення на практичних Миколи Фаріона,власна практика роботи з string.

Що опрацьовано: вивчив деякі функкії з бібліотеки tring:(tolower(),toupper(),erase(),insert(),compare()clear(). Порівнював cin та getline для змінних типу string. дізнався, що робить cin.ignore().

* + Статус: Ознайомлений
  + Початок опрацювання теми: 4/12
  + Звершення опрацювання теми: 10/12

# **Виконання роботи:**

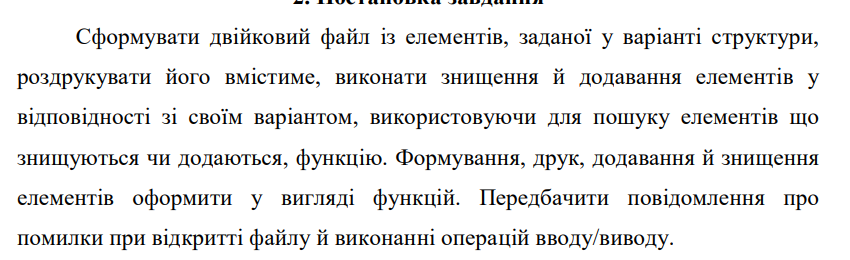
## **1. Опрацювання завдання та вимог до програм та середовища:**

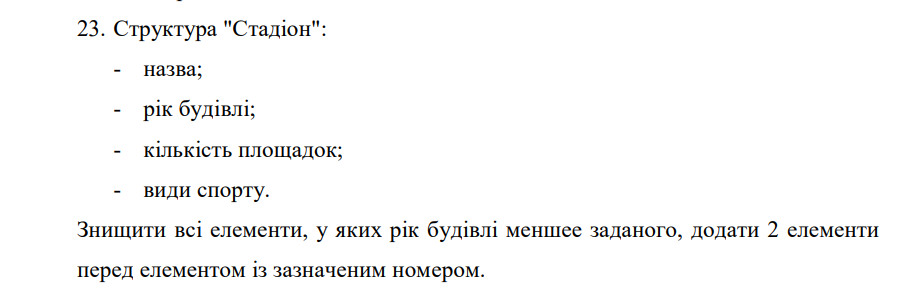
Lab# programming: VNS Lab 6

* Варіант 23
* 

I Lab# programming: VNS Lab 6

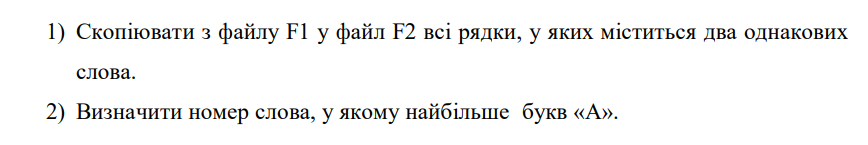
Lab# programming: VNS Lab 8

* Варіант 23
* 



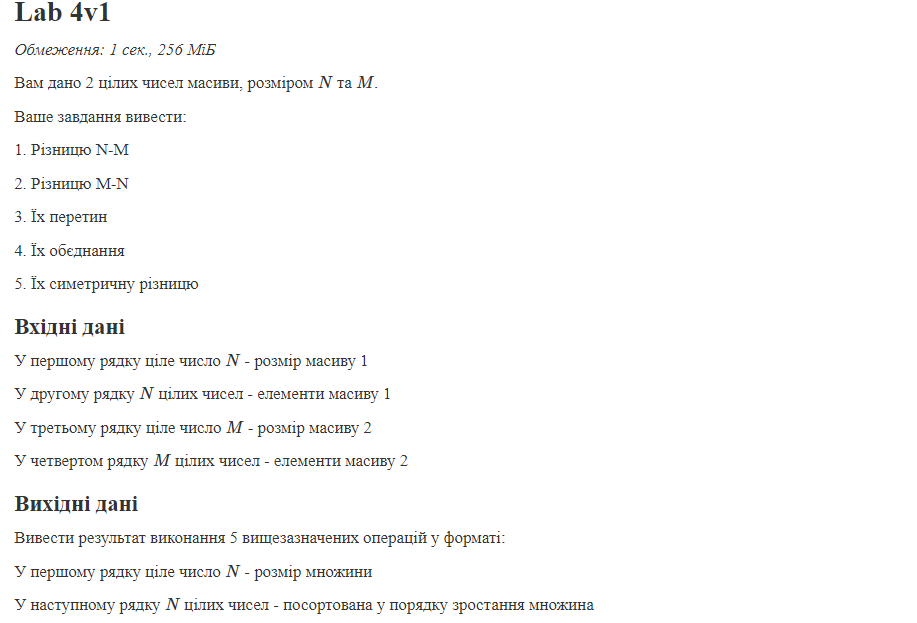
II Lab# programming: VNS Lab 8

Lab# programming: VNS Lab 9

* Варіант 23
* 

II Lab# programming: VNS Lab 9

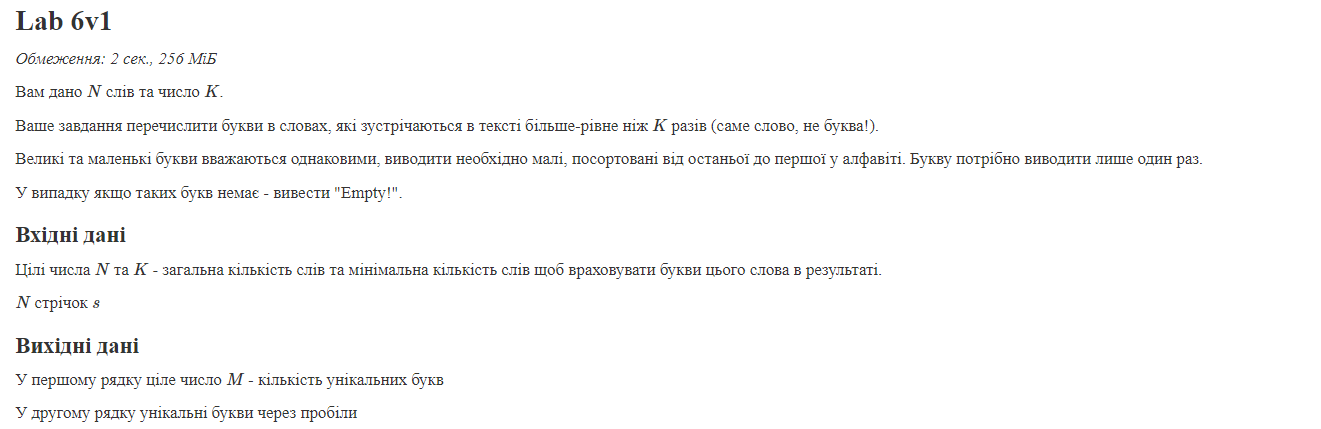
Lab# programming: algotester Lab 4task 1

* Варіант 1
* 

III Lab# programming: algotester Lab 4 task 1

Lab# programming: algotester Lab 6 task 1

* Варіант 1



IV Lab# programming: algotester Lab 6 task 1

Practice# programming: Class Practice Task



V Practice# programming: Class Practice Task

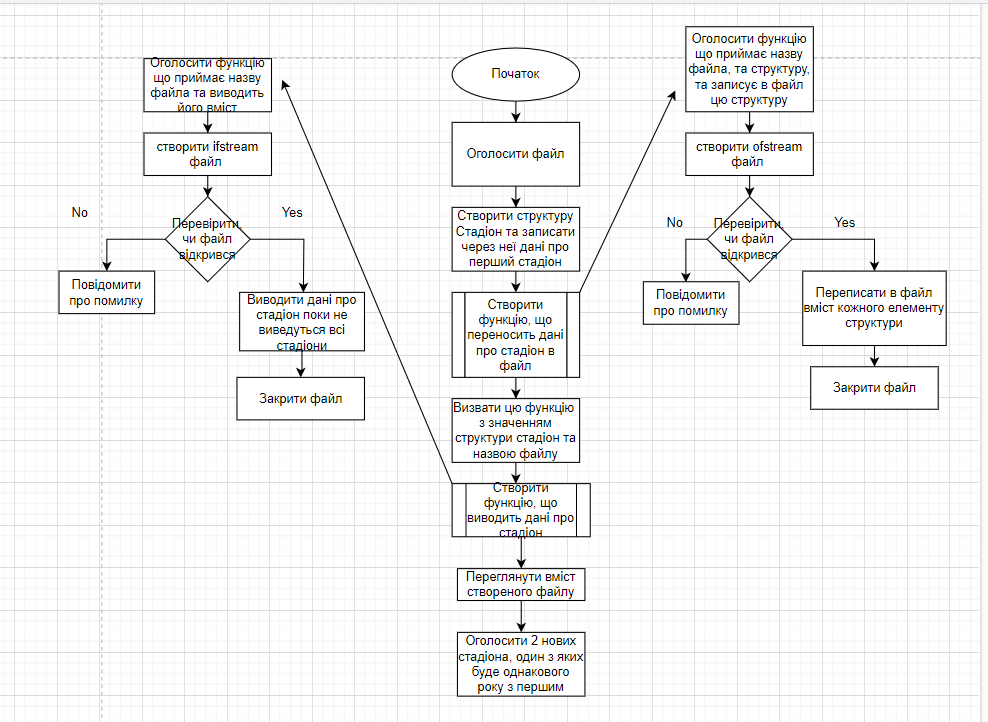
## **2. Дизайн та планована оцінка часу виконання завдань:**

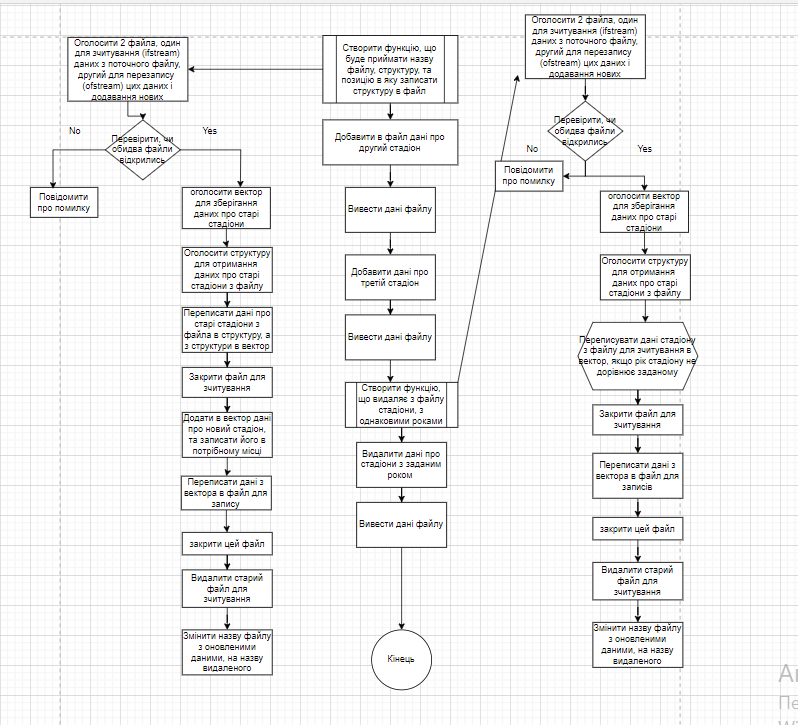
Lab# programming: VNS Lab 6

* Планований час - 30 хв,реальний час - 30 хвилин

Lab# programming: VNS Lab 8

* Планований час -2 години, реальний час 3 години





Lab# programming: VNS Lab 9

* Планований час -2 години, реальний час 1.5 години

Lab# programming: algotester Lab 4 task 1

Планований час - 30 хвилин,реальний час -2 години

Lab# programming: algotester Lab 6 task 1

Планований час - 1 година,реальний час - 1 година

Practice# programming: Class Practice Task

Блок-схема

* Планований час -2 години, реальний час -30 хвилин

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

Lab# programming: VNS Lab 6

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic__5_practice_and_labs_kuzo_ivan/ai_13/ivan_kuzo/Epic_5_ivan_kuzo/vns_lab_6_task_1_kuzo_ivan.cpp>

#include <stdio.h>

#include<string.h>

#define n 266

int smollest\_word\_size(char words[]){

int min=265;

int current\_size=0;

for (int i=0;words[i]!='\0';i++){

if(words[i] != ' ' && words[i]!=',' && words[i]!='.')

current\_size++;

else{

if(current\_size<min)

min=current\_size;

current\_size=0;

}

}

return min;

}

int main()

{

char str[n];

printf("Print a list of words with dot after last: ");

fgets(str,n,stdin);

int max\_size=smollest\_word\_size(str);

printf("max size of word is: %d",max\_size);

return 0;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab# programming: VNS Lab 8

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic__5_practice_and_labs_kuzo_ivan/ai_13/ivan_kuzo/Epic_5_ivan_kuzo/vns_lab_8_task_1_kuzo_ivan.cpp>

#include <iostream>

#include <vector>

#include <fstream>

#include <string>

using namespace std;

struct Stadium {

string name;

int year;

int site;

string kind\_of\_sports;

};

void write\_data\_to\_file(const string& filename, const Stadium& stadium) {

ofstream file(filename, ios::app);

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

cerr << "Trouble with opening" << endl;

return;

}

file << stadium.name << " " << stadium.year << " " << stadium.site << " " << stadium.kind\_of\_sports << endl;

file.close();

}

void watch\_file(const string& filename) {

ifstream file(filename, ios::binary);

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

cerr << "Trouble with opening" << endl;

return;

}

Stadium stadium;

while (file >> stadium.name >> stadium.year >> stadium.site >> stadium.kind\_of\_sports) {

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

cout << "Year of stadium is: " << stadium.year << endl;

cout << "Num of stadiums sites is: " << stadium.site << endl;

cout << "There are such sports: " << stadium.kind\_of\_sports << endl;

cout << endl;

}

file.close();

}

void add\_stadium\_define\_position(const string& old\_fileName, int position, const Stadium& stadium) {

ifstream old\_file(old\_fileName);

ofstream new\_file("new\_fileName");

if (!new\_file.is\_open() || !old\_file.is\_open()) {

cerr << "Trouble with opening" << endl;

return ;

}

vector<Stadium> stadiums;

Stadium current\_stadium;

while (old\_file >> current\_stadium.name >> current\_stadium.year >> current\_stadium.site >> current\_stadium.kind\_of\_sports) {

stadiums.push\_back(current\_stadium);

}

old\_file.close();

stadiums.insert(stadiums.begin() + position, stadium);

for (const auto& stadium : stadiums) {

new\_file << stadium.name << " " << stadium.year << " " << stadium.site << " " << stadium.kind\_of\_sports << endl;

}

new\_file.close();

remove(old\_fileName.c\_str());

rename("new\_fileName", old\_fileName.c\_str());

}

void delete\_stadium(const string& old\_fileName, int delete\_year) {

ifstream old\_file(old\_fileName);

ofstream new\_file("new\_fileName");

if (!new\_file.is\_open() || !old\_file.is\_open()) {

cerr << "Trouble with opening" << endl;

return ;

}

vector<Stadium> stadiums;

Stadium current\_stadium;

while (old\_file >> current\_stadium.name >> current\_stadium.year >> current\_stadium.site >> current\_stadium.kind\_of\_sports) {

if (current\_stadium.year != delete\_year) {

stadiums.push\_back(current\_stadium);

}

}

old\_file.close();

for (const auto& stadium : stadiums) {

new\_file << stadium.name << " " << stadium.year << " " << stadium.site << " " << stadium.kind\_of\_sports << endl;

}

new\_file.close();

remove(old\_fileName.c\_str());

rename("new\_fileName", old\_fileName.c\_str());

}

void clear\_file(const string& filename) {

ofstream file(filename, ios::trunc);

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

cerr << "Trouble with opening" << endl;

return ;

}

file.close();

}

int main() {

string filename = "stadiums.txt";

clear\_file(filename);

Stadium stadium1 = {"Stadium\_name1", 1998, 5, "Football"};

write\_data\_to\_file(filename, stadium1);

cout << "Content of file: " << endl;

watch\_file(filename);

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

Stadium stadium2 = {"Stadium\_name2", 2003, 10, "Baseball"};

Stadium stadium3 = {"Stadium\_name3", 1998, 15, "Basketball"};

add\_stadium\_define\_position(filename, 1, stadium2);

cout << "Content of file after adding stadium2: " << endl;

watch\_file(filename);

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

add\_stadium\_define\_position(filename, 1, stadium3);

cout << "Content of file after adding stadium3: " << endl;

watch\_file(filename);

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

delete\_stadium(filename, 1998);

cout << "Content of file after deleting by year: " << endl;

watch\_file(filename);

return 0;

}

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

Lab# programming: VNS Lab 9

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic__5_practice_and_labs_kuzo_ivan/ai_13/ivan_kuzo/Epic_5_ivan_kuzo/vns_lab_9_task_1_kuzo_ivan.cpp>

#include <iostream>

#include <fstream>

#include <vector>

#include <sstream>

using namespace std;

int numMaxA(const string& line){

int num\_count=0;

int num=0;

int currentA=0;

string word;

int max=0;

istringstream iss(line);

while(iss>>word){

num\_count++;

for(char letter : word){

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

currentA++;

}

}

if (currentA>max){

max=currentA;

num =num\_count;

}

}

if(num!=0)

return max;

else

return 0;

}

int numMaxA(vector<string>lines){

int max=0;

int index=0;

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

if(max<numMaxA(lines[i])){

max=numMaxA(lines[i]);

index=i+1;

}

}return index;

}

bool theSameWord(const string& line){

vector<string>words;

string word;

istringstream iss(line);

while(iss>>word)

words.push\_back(word);

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

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

if(words[i]==words[j] && i!=j)

return true;

}

}return false;

}

void watch(const string& filename){

ifstream file(filename);

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

cerr<<"Trouble with opening";

return;

}

cout<<"Content of file: "<<endl;

string line;

while(getline(file,line))

cout<<line<<endl;

file.close();

}

int main()

{

ofstream file("newFile.txt");

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

cerr<<"Trouble with opening";

return 1;

}

vector<string>lines;

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

cout<<i+1<<" line of file: ";

string line;

getline(cin,line);

lines.push\_back(line);

}

for (const auto& raws: lines)

file<<raws<<endl;

file.close();

watch("newFile.txt");

int a=numMaxA(lines);

cout<<"the most letters a in a line: "<<a<<endl;

ifstream file1("newFile.txt");

if(!file1.is\_open()){

cerr<<"Trouble with opening";

return 1;

}

ofstream file2("newFile2.txt");

if(!file2.is\_open()){

cerr<<"Trouble with opening";

return 1;

}

string copy;

while(getline(file1,copy)){

if(theSameWord(copy))

file2<<copy<<endl;

}

cout<<"with 2+ same words: "<<endl;

watch("newFile2.txt");

return 0;

}

Lab# programming: algotester Lab 4 task 1

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic__5_practice_and_labs_kuzo_ivan/ai_13/ivan_kuzo/Epic_5_ivan_kuzo/algotester_lab_4_task_1_kuzo_ivan.cpp>

#include<iostream>

#include<algorithm>

#include<vector>

using namespace std;

void difference(vector<int>& N, vector<int>& M) {

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

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

vector<int> differenceNM;

set\_difference(N.begin(), N.end(), M.begin(), M.end(), back\_inserter(differenceNM));

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

for (const auto& b : differenceNM)

cout << b << " ";

cout << endl;

}

void union\_arr(vector<int>& N, vector<int>& M) {

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

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

vector<int> unionNM;

set\_union(N.begin(), N.end(), M.begin(), M.end(), back\_inserter(unionNM));

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

for (const auto& b : unionNM)

cout << b << " ";

cout << endl;

}

void intersection\_arr(vector<int>& N, vector<int>& M) {

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

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

vector<int> intersectionNM;

set\_intersection(N.begin(), N.end(), M.begin(), M.end(), back\_inserter(intersectionNM));

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

for (const auto& b : intersectionNM)

cout << b << " ";

cout << endl;

}

void symmetrical\_diff(vector<int>& N, vector<int>& M) {

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

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

vector<int> intersectionNM;

set\_intersection(N.begin(), N.end(), M.begin(), M.end(), back\_inserter(intersectionNM));

vector<int> differ;

set\_symmetric\_difference(N.begin(), N.end(), M.begin(), M.end(), back\_inserter(differ));

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

for (const auto& d : differ)

cout << d << " ";

cout << endl;

}

int main() {

vector<int> N\_arr;

vector<int> M\_arr;

int N = 0;

cin >> N;

int ni = 0;

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

cin >> ni;

N\_arr.push\_back(ni);

}

int M = 0;

cin >> M;

int mi = 0;

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

cin >> mi;

M\_arr.push\_back(mi);

}

difference(N\_arr, M\_arr);

cout << endl;

difference(M\_arr, N\_arr);

cout << endl;

intersection\_arr(N\_arr, M\_arr);

cout << endl;

union\_arr(N\_arr, M\_arr);

cout << endl;

symmetrical\_diff(N\_arr, M\_arr);

return 0;

}

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

Lab# programming: algotester Lab 6 task 1

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic__5_practice_and_labs_kuzo_ivan/ai_13/ivan_kuzo/Epic_5_ivan_kuzo/algotester_lab_6_task_1_kuzo_ivan.cpp>

#include <iostream>

#include <algorithm>

#include <vector>

#include <unordered\_map>

#include <unordered\_set>

using namespace std;

int main() {

vector<string> words;

unordered\_map<string, int> wordCount;

unordered\_set<char> uniqueLetters;

int N, K;

cin >> N >> K;

string word;

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

cin >> word;

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

words.push\_back(word);

wordCount[word]++;

}

for (const auto& entry : wordCount) {

if (entry.second >= K) {

for (char ch : entry.first) {

uniqueLetters.insert(ch);

}

}

}

if (!uniqueLetters.empty()) {

vector<char> sortedLetters(uniqueLetters.begin(), uniqueLetters.end());

sort(sortedLetters.begin(), sortedLetters.end(), greater<char>());

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

for (char ch : sortedLetters) {

cout << ch << " ";

}

} else {

cout << "Empty!";

}

return 0;

}

Practice# programming: Class Practice Task

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic__5_practice_and_labs_kuzo_ivan/ai_13/ivan_kuzo/Epic_5_ivan_kuzo/practice_work_task_kuzo_ivan.cpp>

## **#include<iostream>**

## **#include<fstream>**

## **#include<vector>**

## **using namespace std;**

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

## **FileOpResult write\_to\_file(const string& fileName,const string& content){**

## **ofstream file(fileName);**

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

## **cerr<<"Trouble with opening";**

## **return Failure;**

## **}**

## **file<<content;**

## **return Success;**

## **}**

## **int main()**

## **{**

## **string fileName;**

## **string content;**

## **cout<<"Enter the name of file: ";**

## **cin>>fileName;**

## **cin.ignore();**

## **cout<<"write something to file: ";**

## **getline(cin, content);**

## **if(write\_to\_file(fileName,content)==Success)**

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

## **else**

## **cout<<"Failure";**

## 

## **return 0;**

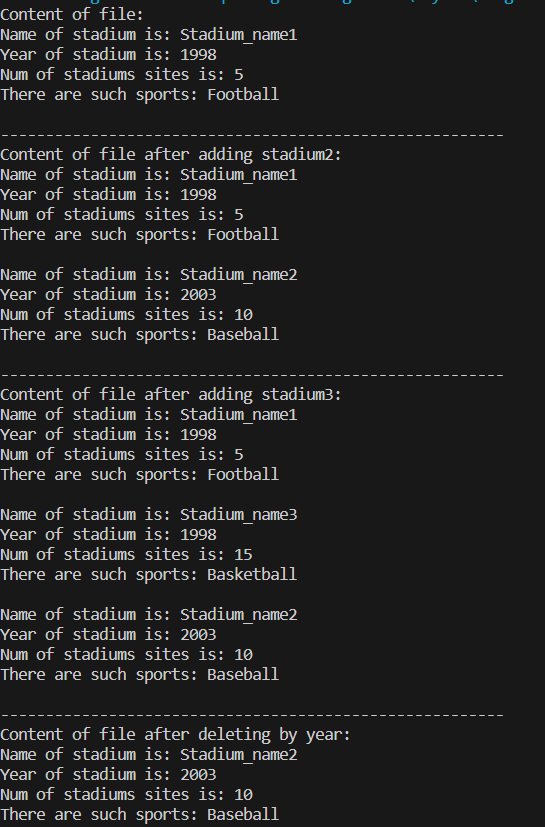
## **}**

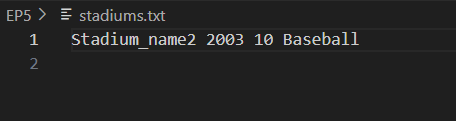
## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Lab# programming: VNS Lab 6

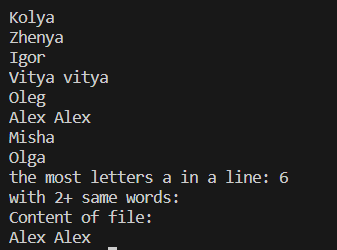


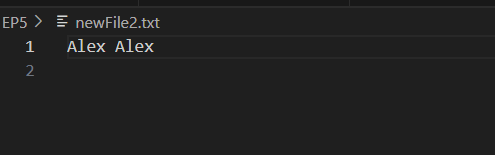
Lab# programming: VNS Lab 8



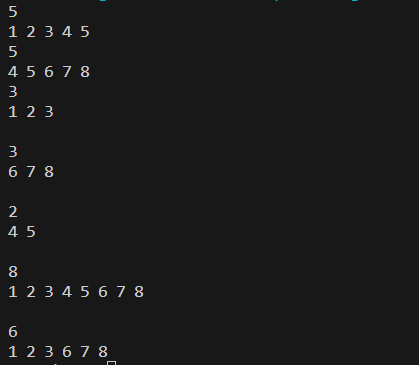


Lab# programming: VNS Lab 9

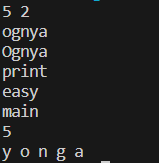




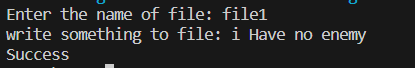
Lab# programming: algotester Lab 4 task 1

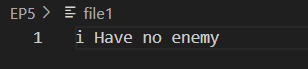


Lab# programming: algotester Lab 6 task 1



Practice# programming: Class Practice Task





# **Висновок:**

Детально ознайомився з свореннямфайлів для запису (ofstream) та зчитування з файлу (ifstream).Перевіряв на успіше відкриття через is\_open та закриття close() . Працював з бінарними файлами ios та звичайними.

Копіював файли та видаляв з них інформацію.

Вчив деякі функкії з бібліотеки String toupper(),erase(),insert(),compare()clear(). Порівнював cin та getline для змінних типу string. дізнався, що робить cin.ignore()tolower().

Навчився працювати з структурами,оголошувати їх, та надавати їм значення.

Поглибив знання в циклах, стикнувся на алготестері з проблемою ліміту часу, через що довелось вивчати час виконання кожного цикла. Також в ході навчився їх оптимізовувати.

Також розвинув навики тайммененджменту і зрозумів, що краще не відкладати все до дедлайну.