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

Ласкавий Артем Олександрович

# **Тема роботи:**

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

# **Мета роботи:**

Ознайомитись з методами роботи з файлами, зокрема текстовими та бінарними. Ознайомитись з символами та рядковими змінними, системами числення, STL в С++.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Файли. Бінарні та текстові файли.
* Тема №2: Символи і рядкові змінні
* Тема №3: Стандартна бібліотека

1. Індивідуальний план опрацювання теорії:

* Тема №1: Файли. Бінарні та текстові файли.
  + Джерела Інформації.
    - Відео.
      * <https://youtu.be/rkJJzH-ymg8?si=Re0jBT47DOap8UXs>
      * <https://youtu.be/SJMxuXrP_Rk?si=5-fPzWqzoijMlkOL>
      * https://youtu.be/8jLOx1hD3\_o?si=56bJhdsRpmcvIQhx
    - Стаття.
      * https://cplusplus.com/reference/fstream/fstream/
      * https://en.cppreference.com/w/cpp/io
      * https://www.geeksforgeeks.org/c-stream-classes-structure/
  + Що опрацьовано:
    - Опрацьовано види файлів, їх особливості, поняття потоку та його види в С++.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.12.2023
  + Звершення опрацювання теми: 07.12.2023
* Тема №2: Символи і рядкові змінні
  + Джерела Інформації:
    - Відео.
      * https://youtu.be/FeHZHF0f2dw?si=X0Z6wY6ag5RHylMs
      * https://youtu.be/lTPT1cPfVmI?si=e2eBv2Y0vcXDh5fn
      * https://youtu.be/8jLOx1hD3\_o?si=56bJhdsRpmcvIQhx
  + Стаття.

https://www.geeksforgeeks.org/strings-in-cpp/

* + Що опрацьовано:
    - Опрацьовано поняття символів і рядкових зміних, різницю між рядковими змінними в С і С++, корисні методи рядкових змінних у С++.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.11.2023
  + Звершення опрацювання теми: 07.12.2023
* Тема №3: Стандартна бібліотека
  + Джерела Інформації:
    - Відео.
      * https://youtu.be/Yevgn0yu5mI?si=CQVHOohlXNiGqKnM
      * <https://youtu.be/1YpxRmsDuls?si=MtcFpT_4m3O5blgx>
      * https://youtu.be/Yevgn0yu5mI?si=HKHqrdq9Uz54U\_9I
  + Стаття.
    - <https://www.geeksforgeeks.org/set-in-cpp-stl/>
    - <https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/>
    - https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/
  + Що опрацьовано:
    - Опрацьовано складові частини стандартної бібліотеки – конейнери(вектори, списки, сети та мапи), алгоритми(сортування та пошуку), функції, ітератори.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 08.11.2023
  + Звершення опрацювання теми: 07.12.2023

# **Виконання роботи:**

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

Завдання №1 **VNS Lab 6 - Task 1-17**

* Варіант завдання 17
* 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. Текст містить не більше 255 символів.

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

рядка у відповідності зі своїм варіантом. Всі слова рядка, які починаються з букви, відсортувати за абеткою.

Завдання №2 **VNS Lab 8 - Task 1-17**

* Варіант завдання 17
* 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. Передбачити повідомлення про

помилки при відкритті файлу й виконанні операцій вводу/виводу.

Структура "Фільм":

- назва;

- режисер;

- рік випуску;

- вартість.

Знищити всі елементи, у яких вартість перевищує задану, додати елемент на

початок файлу.

Завдання №3 **VNS Lab 9 - Task 1-17**

* Варіант завдання 1
* Деталі завдання:

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

інформацію

Виконати завдання. 1) Скопіювати з файлу F1 у файл F2 всі рядки, що починаються на букву «А»

розташовані між рядками з номерами N1 й N2, а потім всі рядки від N2+3 і до

останнього.

2) Визначити кількість слів в останньому рядку файлу F2.

Завдання №4 **Algotester Lab 4**

Варіант завданння 2

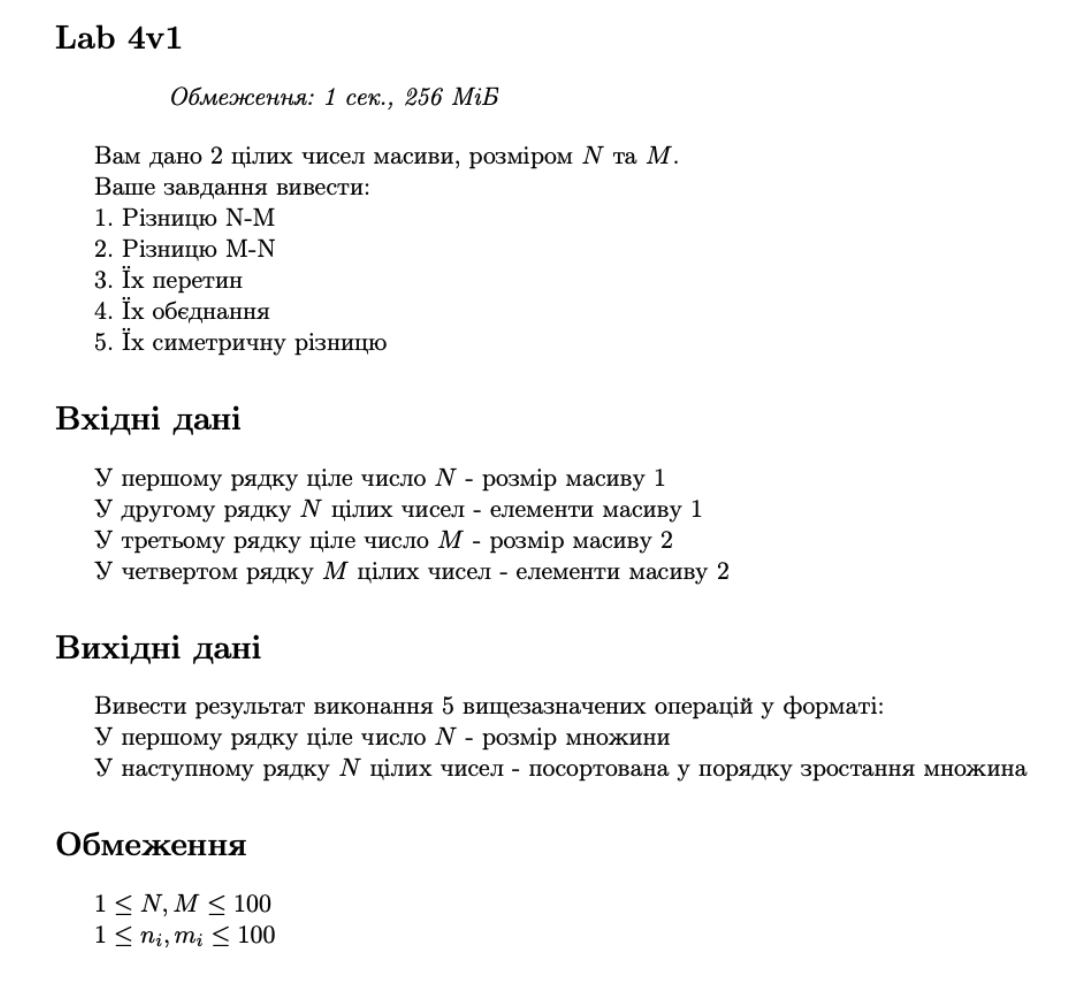
* Деталі завдання
* 

Figure Умова до завдання №4

Завдання №5 **Algotester Lab 6**

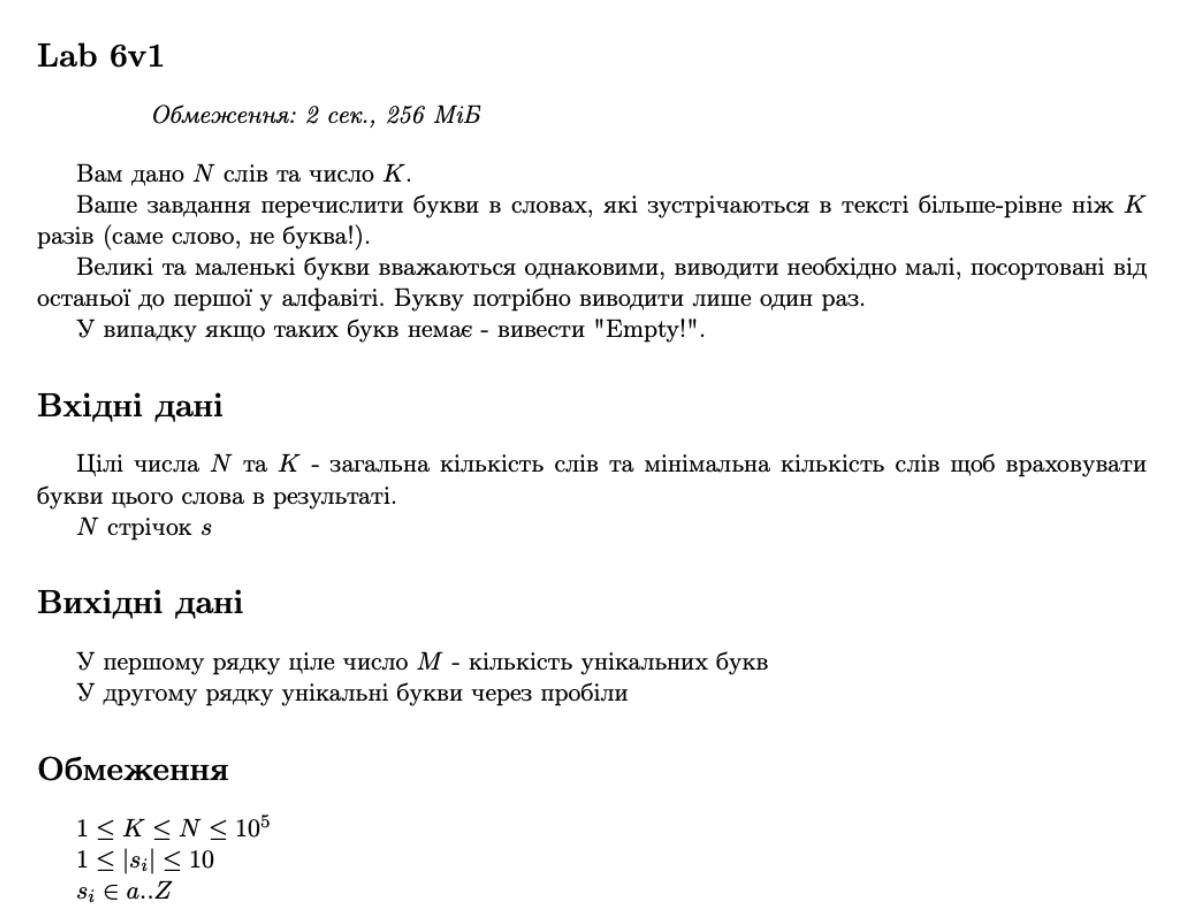
* Деталі завдання
* 

Figure Умова до завдання №5

Завдання №6 **Class Practice Work**

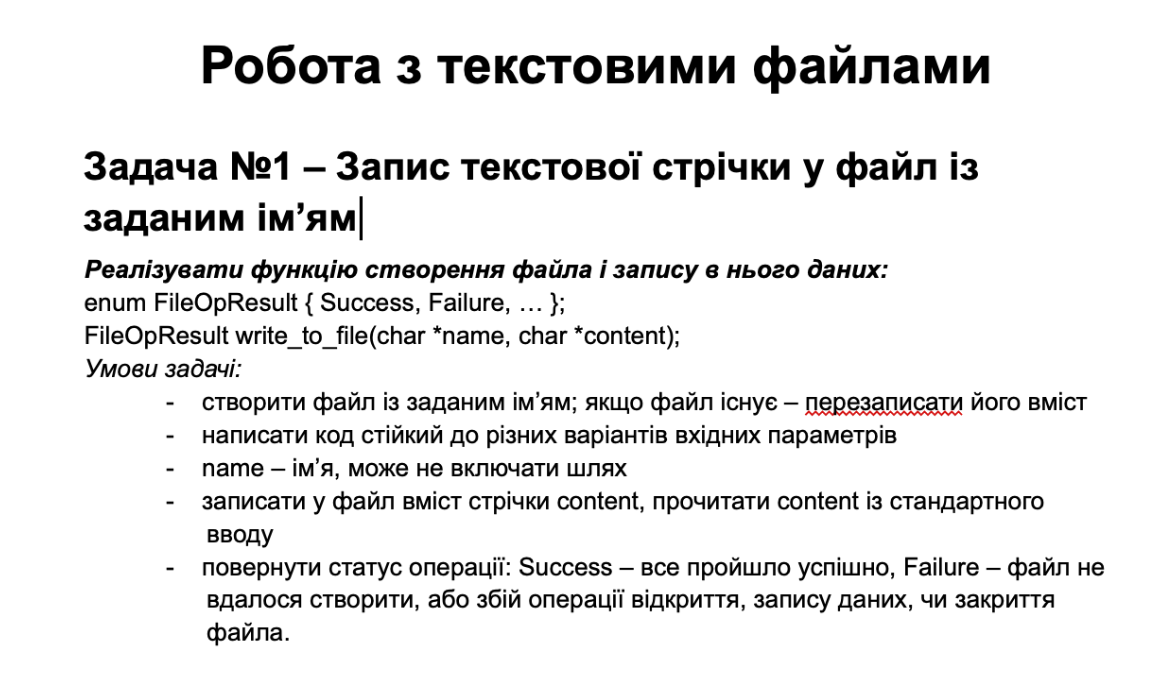
* Деталі завдання
* 

Figure Умова до завдання №6

# **Завдання №7 Self Practice Work – Algotester Lab 4v3 | NULP\_LABS\_MPP\_2023**

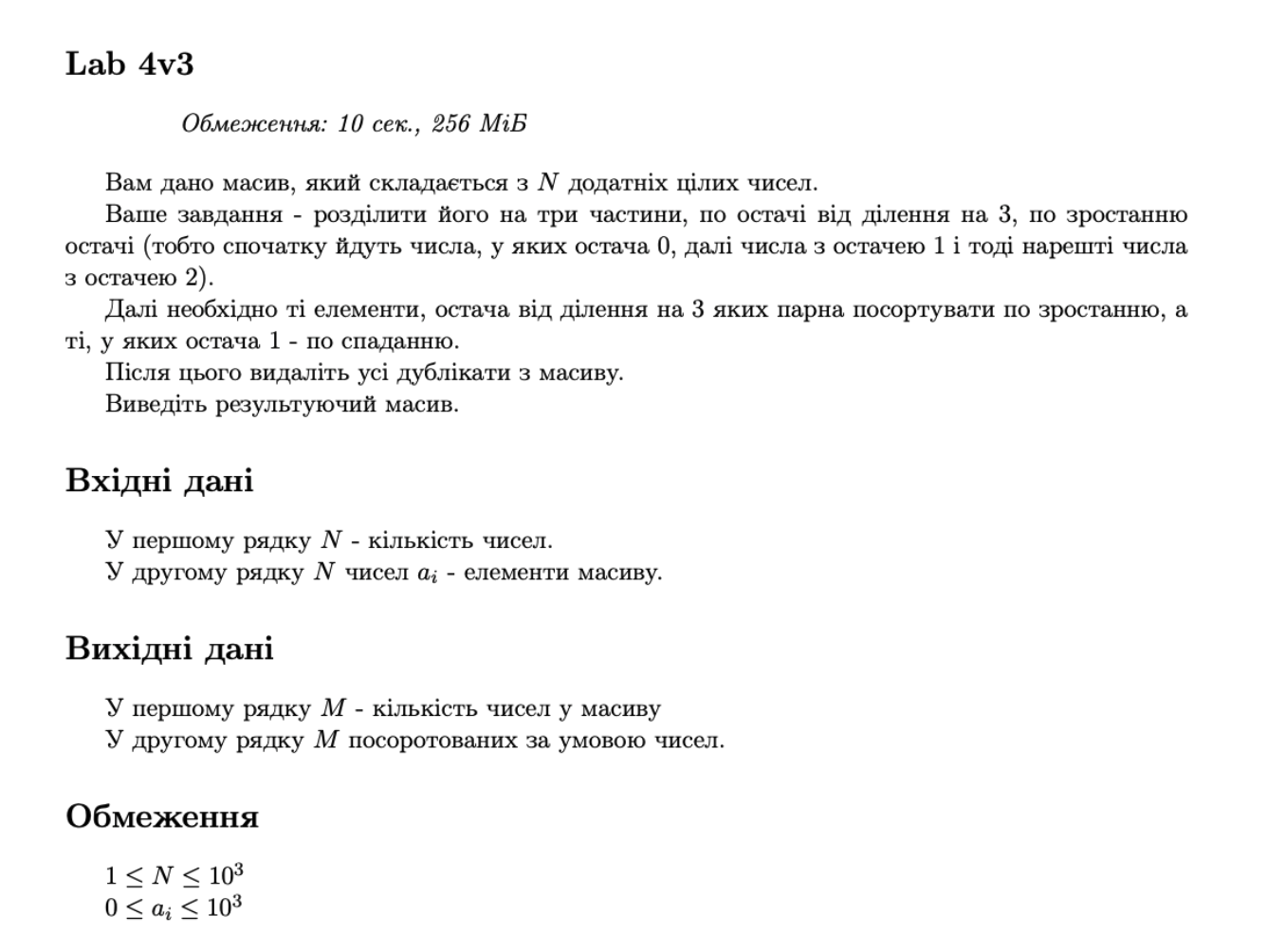
* Деталі завдання
* 

Figure Умова до завдання №7

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

Програма №1 **VNS Lab 6 - Task 1-17**

* Планований час на реалізацію: 30хв
* Важливі деталі для врахування в імплементації: необхідно посортувати незалежно від регістру першої букви, тому пишемо власну функцію сортування.

Програма №2 **VNS VNS Lab 8 - Task 1-17**

* Планований час на реалізацію: 120хв.

Програма №3 **VNS Lab 9 - Task 1-17**

Планований час на реалізацію: 60хв.

Програма №4 **Algotester Lab 4**

**Блок-схема:**

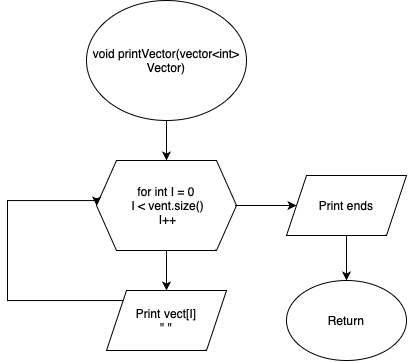
****

Figure Блок-схема до програми №4 - функція printVector

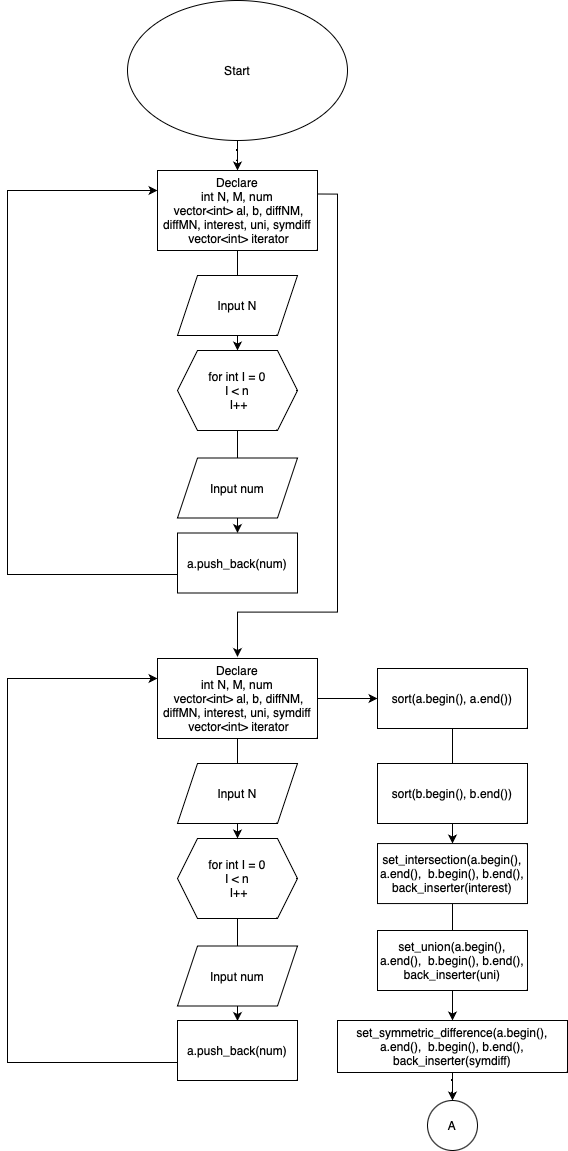


Figure Блок-схема до програми №4 – функція main 1

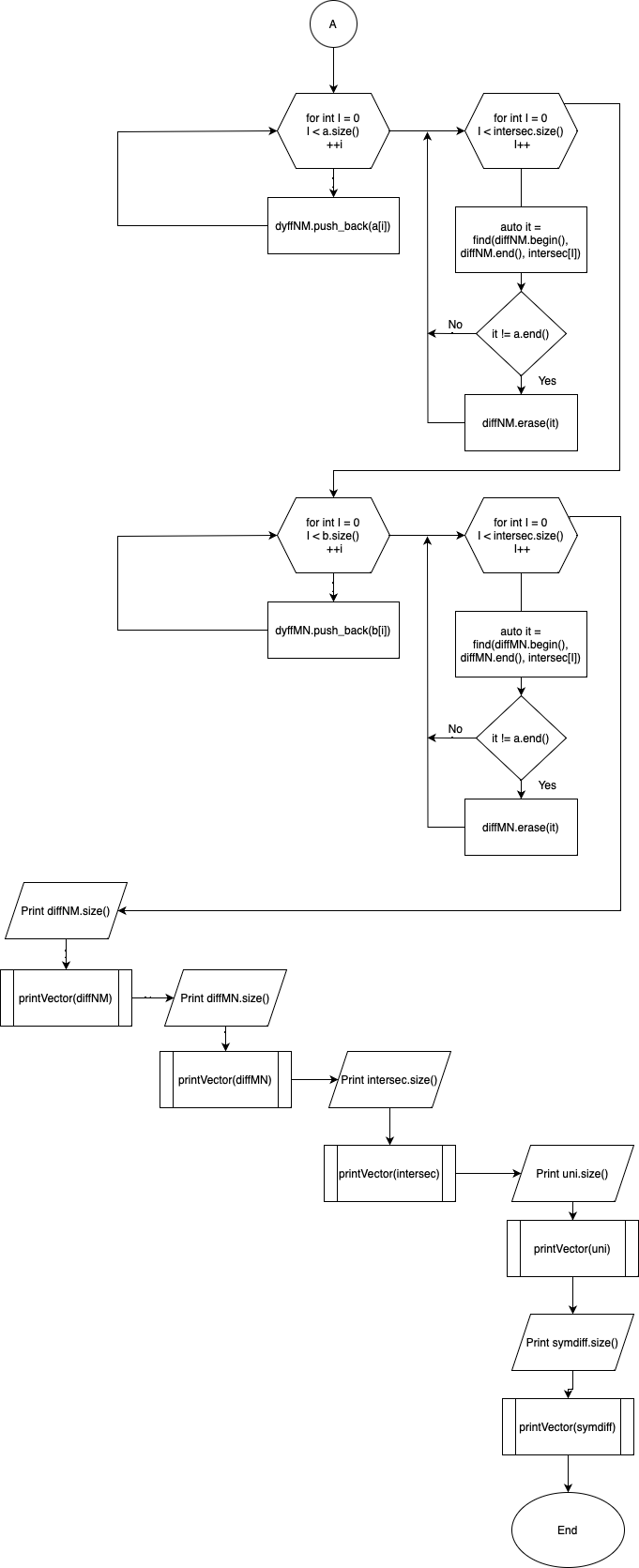


Figure Блок-схема до програми №4 – функція main 2

* Планований час на реалізацію 40 хв.

Програма №5 **Algotester Lab 6**

* Планований час на реалізацію: 60хв.

Програма №6 **Class Practice Work**

* Планований час на реалізацію: 60хв.

Програма №7 **Self Practice Work – Algotester Lab 4v3 | NULP\_LABS\_MPP\_2023**

* Планований час на реалізацію: 40хв.

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

Без змін

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

Завдання №1 **VNS Lab 6 - Task 1-17**

Посилання на Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/epic\_5\_practice\_and\_labs\_artem\_laskavyi/ai\_13/artem\_laskavyi/Epic%205/vns\_lab\_6\_task\_artem\_laskavyi.cpp

#include <algorithm>

#include <string>

#include <vector>

#include <iostream>

#include <sstream>

using namespace std;

bool sortAlpha(const string& a, const string& b) {

string aUpper = a;

string bUpper = b;

transform(a.begin(), a.end(), aUpper.begin(), ::toupper);

transform(b.begin(), b.end(), bUpper.begin(), ::toupper);

return aUpper < bUpper;

}

int main() {

string input;

vector<string> v;

getline(cin, input);

stringstream ss(input);

while (getline(ss, input, ' ')){

if(isalpha(input[0])){

v.push\_back(input);

}

}

sort(v.begin(), v.end(), sortAlpha);

cout << "-----------------\n";

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

cout << v[i] << endl;

}

return 0;

Figure Код програми №1

Завдання №2 **VNS Lab 8 - Task 1-17**

Посилання на Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/epic\_5\_practice\_and\_labs\_artem\_laskavyi/ai\_13/artem\_laskavyi/Epic%205/vns\_lab\_8\_task\_artem\_laskavyi.cpp

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

struct Movie{

string name;

string director;

int releaseYear;

int price;

};

void printMovie(const Movie& movie) {

cout << "Name of the movie: " << movie.name << endl;

cout << "Director of the movie: " << movie.director << endl;

cout << "Year of release: " << movie.releaseYear << endl;

cout << "Price: " << movie.price << endl;

}

void structToBinary(const vector<Movie>& movies){

ofstream outf("movies.bin", ofstream::binary | ofstream::trunc);

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

cout << "Error while opening output file" << endl;

return;

}

for (const Movie& movie : movies) {

outf.write(reinterpret\_cast<const char\*>(&movie), sizeof(Movie));

}

outf.close();

}

void delByPrice(int setPrice) {

ifstream inFile("movies.bin", fstream::binary);

if (!inFile) {

cerr << "Error opening input file." << endl;

return;

}

vector<Movie> movies;

Movie movie;

while (inFile.read(reinterpret\_cast<char\*>(&movie), sizeof(Movie))) {

if (movie.price <= setPrice) {

movies.push\_back(movie);

}

}

inFile.close();

structToBinary(movies);

}

void printBinary(int size) {

ifstream inf("movies.bin", ifstream::binary);

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

cout << "Error while opening file for printing" << endl;

return;

}

Movie movie;

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

inf.read(reinterpret\_cast<char\*>(&movie), sizeof(Movie));

cout << "Movie " << i + 1 << ":\n";

printMovie(movie);

cout << endl;

}

inf.close();

}

void addToFront(const Movie& newMovie) {

vector<Movie> movies;

ifstream inf("movies.bin", ifstream::binary);

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

cout << "Error while opening binary file for reading." << endl;

return;

}

Movie movie;

while (inf.read(reinterpret\_cast<char\*>(&movie), sizeof(Movie))) {

movies.push\_back(movie);

}

inf.close();

ofstream outf("movies.bin", ofstream::binary | ofstream::trunc);

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

cout << "Error while opening binary file for writing." << endl;

return;

}

movies.insert(movies.begin(), newMovie);

for (const Movie& m : movies) {

outf.write(reinterpret\_cast<const char\*>(&m), sizeof(Movie));

}

outf.close();

}

int main() {

vector<Movie> movies = {

{"The Godfather", "Francis Coppola", 1972, 170},

{"Pulp Fiction", "Quentin Tarantino", 1994, 150},

{"Forrest Gump", "Robert Zemeckis", 1994, 200},

{"Fight Club", "David Fincher", 1999, 150},

};

structToBinary(movies);

printBinary(movies.size());

cout << "------------------\n";

cout << "Enter highest price that is acceptable in the movie collection\n";

int nprice;

cin >> nprice;

cout << "------------------\n";

delByPrice(nprice);

Movie newMovie = {"Interstellar", "Christopher Nolan", 2014, 165};

addToFront(newMovie);

cout << "Modified binary file:\n";

printBinary(movies.size());

return 0;

}

Figure Код програми №2

Завдання №3 **VNS Lab 9 - Task 1-17**

Посилання на Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/epic\_5\_practice\_and\_labs\_artem\_laskavyi/ai\_13/artem\_laskavyi/Epic%205/vns\_lab\_9\_task\_artem\_laskavyi.cpp

#include <string>

#include <vector>

#include <iostream>

#include <fstream>

#include <sstream>

using namespace std;

int main() {

string line;

vector<string> newlines, v;

int N1, N2;

int line\_pos = 0;

cout << "Enter starting line:\n";

cin >> N1;

cout << "Enter ending line:\n";

cin >> N2;

ifstream inf("file1.txt", ifstream::in);

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

cout << "Error while opening input file" << endl;

return 1;

}

while (getline(inf, line)){

line\_pos++;

if (line\_pos >= N1 && line\_pos <= N2){

if (!line.empty() && line[0] == 'A'){

newlines.push\_back(line);

}

} else if (line\_pos > N2 + 3){

newlines.push\_back(line);

}

}

inf.close();

ofstream outf("file2.txt", ofstream::out | ofstream::trunc);

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

cout << "Error while opening output file" << endl;

return 1;

}

for (const string& newline : newlines) {

outf << newline << endl;

}

outf.close();

stringstream ss(newlines.back());

string word;

while (getline(ss, word, ' ')) {

v.push\_back(word);

}

cout << "Number of words in the last line of file 2: " << v.size() << endl;

return 0;

}

Figure Код програми №3

Завдання №4 **Algotester Lab 4**

Посилання на Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/epic\_5\_practice\_and\_labs\_artem\_laskavyi/ai\_13/artem\_laskavyi/Epic%205/algotester\_lab\_4\_task\_artem\_laskavyi.cpp

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

void printVector( vector<int> vect ){

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

cout << vect[i] << " ";

}

cout << endl;

}

int main(){

int N, M, num;

vector<int> a, b, diffNM, diffMN, intersec, uni, symdiff;

vector<int>::iterator it;

cin >> N;

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

cin >> num;

a.push\_back(num);

}

cin >> M;

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

cin >> num;

b.push\_back(num);

}

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

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

set\_intersection(a.begin(), a.end(), b.begin(), b.end(), back\_inserter(intersec));

set\_union(a.begin(), a.end(), b.begin(), b.end(), back\_inserter(uni));

set\_symmetric\_difference(a.begin(), a.end(), b.begin(), b.end(), back\_inserter(symdiff));

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

diffNM.push\_back(a[i]);

}

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

auto it = find(diffNM.begin(), diffNM.end(), intersec[i]);

if (it != a.end()){

diffNM.erase(it);

}

}

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

diffMN.push\_back(b[i]);

}

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

auto it = find(diffMN.begin(), diffMN.end(), intersec[i]);

if (it != b.end()){

diffMN.erase(it);

}

}

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

printVector(diffNM);

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

printVector(diffMN);

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

printVector(intersec);

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

printVector(uni);

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

printVector(symdiff);

return 0;

}

Figure Код програми №4

Програма №5 **Algotester Lab 6**

Посилання на Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/epic\_5\_practice\_and\_labs\_artem\_laskavyi/ai\_13/artem\_laskavyi/Epic%205/algotester\_lab\_6\_task\_artem\_laskavyi.cpp

#include <algorithm>

#include <string>

#include <vector>

#include <iostream>

#include <unordered\_map>

using namespace std;

bool sortZA(char a, char b){

return a > b;

}

int main(){

int n, k, m = 0;

cin >> n >> k;

string t, res;

vector<string> s;

unordered\_map<string, int> map;

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

cin >> t;

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

s.push\_back(t);

map[t]++;

}

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

int count = map[s[i]];

if (count >= k) {

m += count;

res += s[i];

}

}

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

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

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

sort(res.begin(), res.end(), sortZA);

if(m >= k){

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

for (auto ch : res){

cout << ch << " ";

}

}else{

cout << "Empty!";

}

return 0;

}

Figure Код програми №5

Програма №6 **Class Practice Work**

Посилання на Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/epic\_5\_practice\_and\_labs\_artem\_laskavyi/ai\_13/artem\_laskavyi/Epic%205/practice\_work\_team\_tasks\_artem\_laskavyi.cpp

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

enum FileOpResult {Success, Failure};

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

ofstream file(name);

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

return Failure;

}

file << content;

file.close();

if (file.fail()) {

return Failure;

}

return Success;

}

int main(){

string filename, content;

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

cin >> filename;

cin.ignore();

cout << "Enter the content to write to the file: ";

getline(cin, content);

if (write\_to\_file(filename, content) == Success) {

cout << "Success\n";

} else {

cout << "Failure\n";

}

return 0;

}

Figure Код програми №6

Програма №7 **Self Practice Work – Algotester Lab 4v3 | NULP\_LABS\_MPP\_2023**

Посилання на Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/epic\_5\_practice\_and\_labs\_artem\_laskavyi/ai\_13/artem\_laskavyi/Epic%205/practice\_work\_self\_algotester\_tasks\_artem\_laskavyi.cpp

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

bool sortDescending(const int &a, const int &b){

return (a > b);

}

int main(){

int N, num, remainder;

vector<int> general, rem0, rem1, rem2, result;

cin >> N;

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

cin >> num;

general.push\_back(num);

}

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

auto last = std::unique(general.begin(), general.end());

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

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

remainder = general[i] % 3;

if(remainder == 0){

rem0.push\_back(general[i]);

}else if(remainder == 1){

rem1.push\_back(general[i]);

}else{

rem2.push\_back(general[i]);

}

}

sort(rem0.begin(), rem0.end());

sort(rem1.begin(), rem1.end(), sortDescending);

sort(rem2.begin(), rem2.end());

result.reserve( rem0.size() + rem1.size() + rem2.size()); // preallocate memory

result.insert( result.end(), rem0.begin(), rem0.end() );

result.insert( result.end(), rem1.begin(), rem1.end() );

result.insert( result.end(), rem2.begin(), rem2.end() );

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

for (auto i : result){

cout << i << ' ';

}

cout << endl;

return 0;

}

Figure Код програми №7

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

Завдання №1 **VNS Lab 6 - Task 1-17**

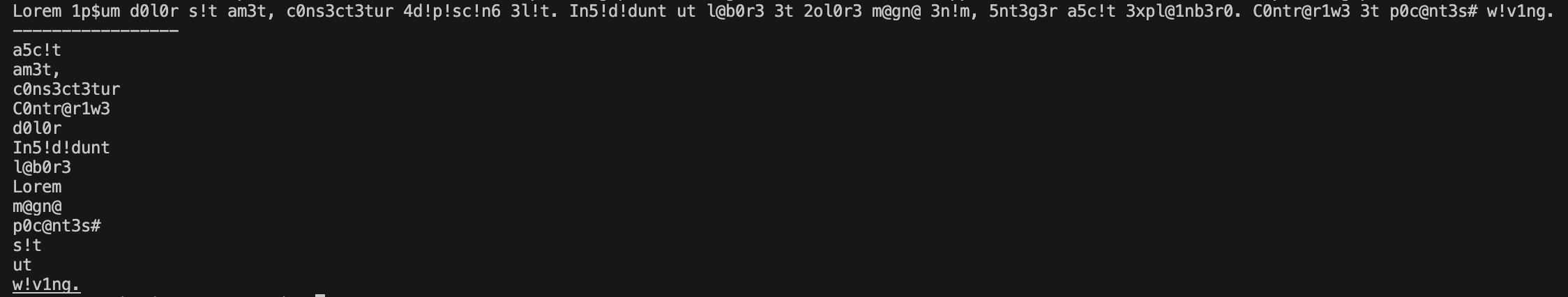


Figure Тестування програми 1

Час затрачений на виконання завдання: 30хв.

Завдання №2 **VNS Lab 8 - Task 1-17**

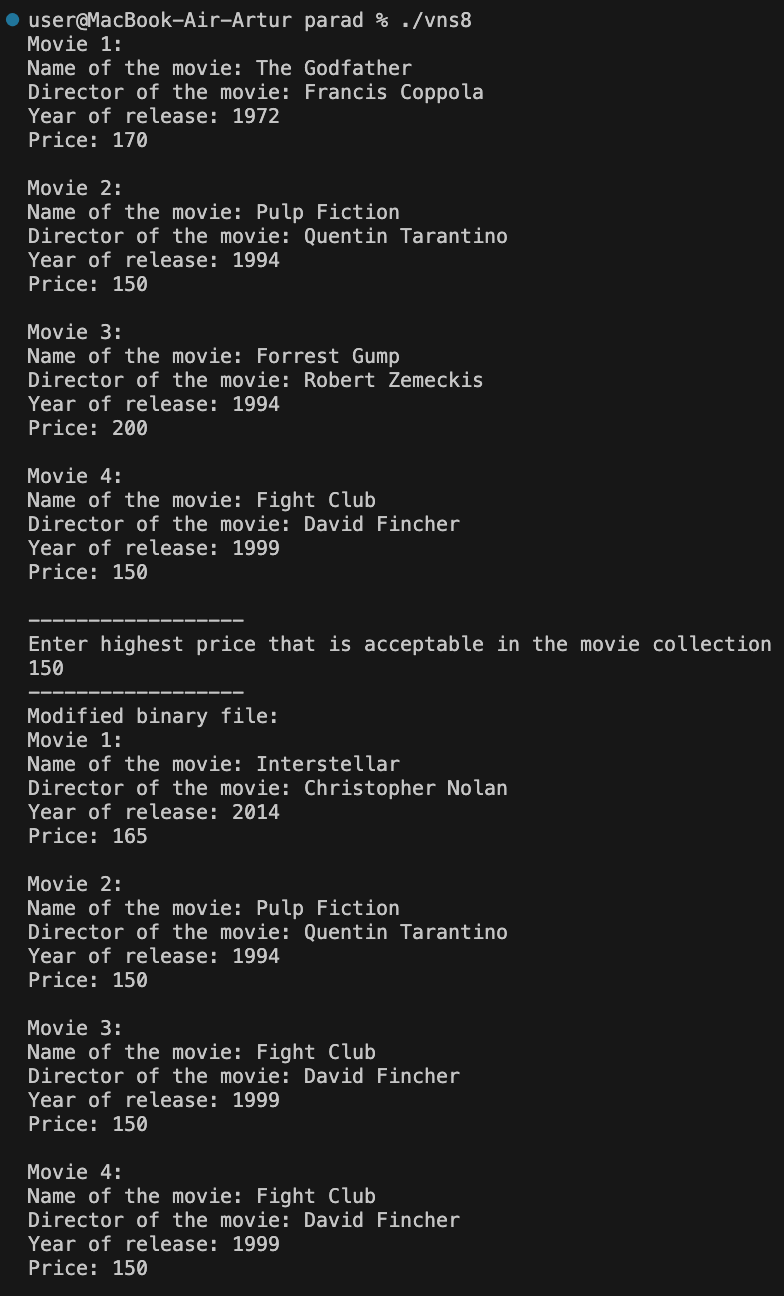


Figure Тестування програми 2

Час затрачений на виконання завдання: 80хв.

Завдання №3 **VNS Lab 9 - Task 1-17**

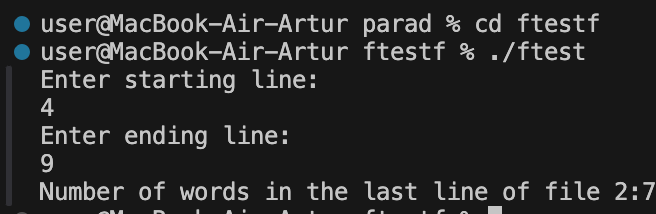


Figure Тестування програми 1 - рис.1

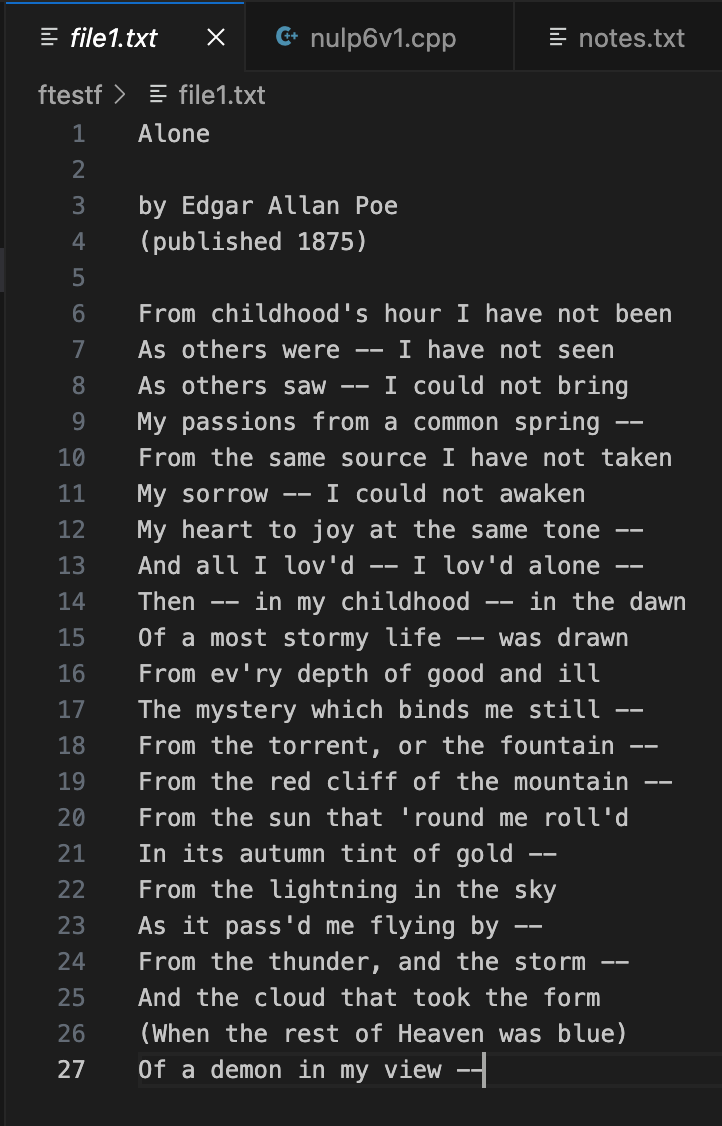


Figure Тестування програми 1 - рис.2

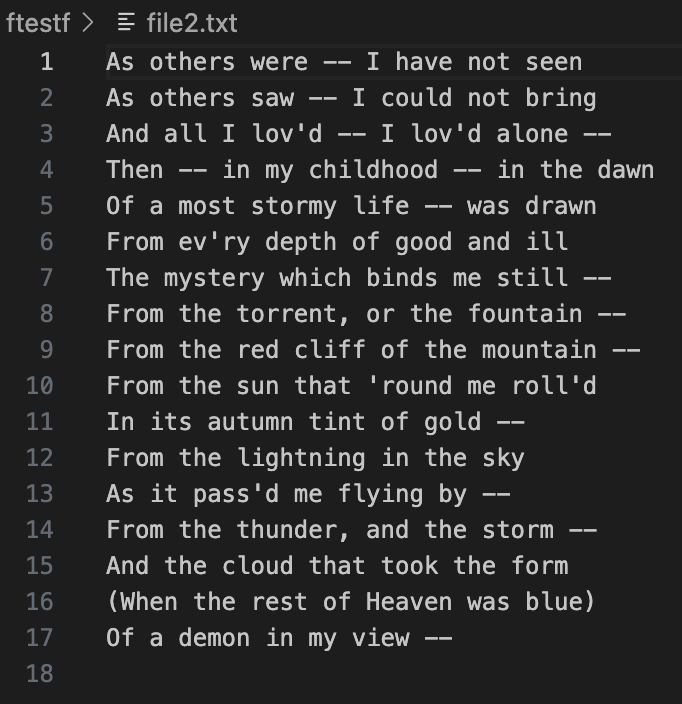


Figure Тестування програми 1 - рис.3

Час затрачений на виконання завдання: 150хв.

Завдання №4 **Algotester Lab 4**

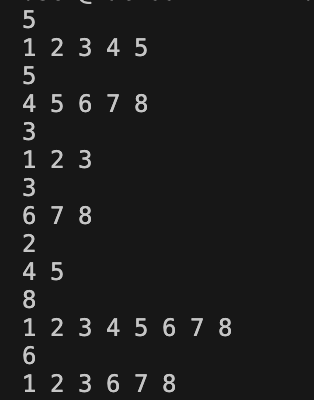


Figure Тестування програми 4

Час затрачений на виконання завдання: 20хв.

Програма №5 **Algotester Lab 6**

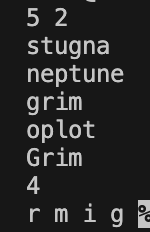


Figure Тестування програми 5

Час затрачений на виконання завдання: 60хв.

Програма №6 **Class Practice Work**

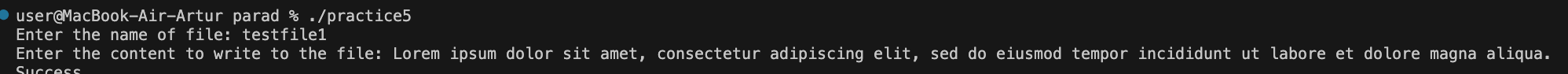


Figure Тестування програми 6 - рис.1

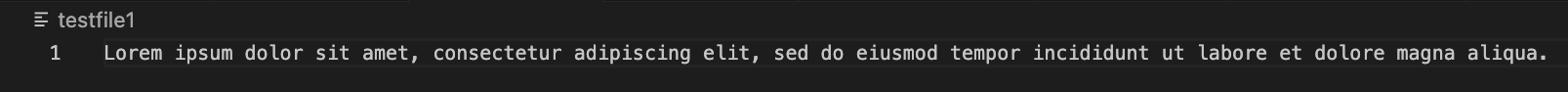


Figure Тестування програми 6 - рис.2

Час затрачений на виконання завдання: 45хв.

Програма №7 **Self Practice Work – Lab 4v3 | NULP\_LABS\_MPP\_2023**

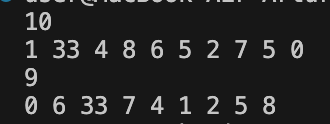


Figure Тестування програми 7

Час затрачений на виконання завдання: 40хв.

## **6. Кооперація з командою:**

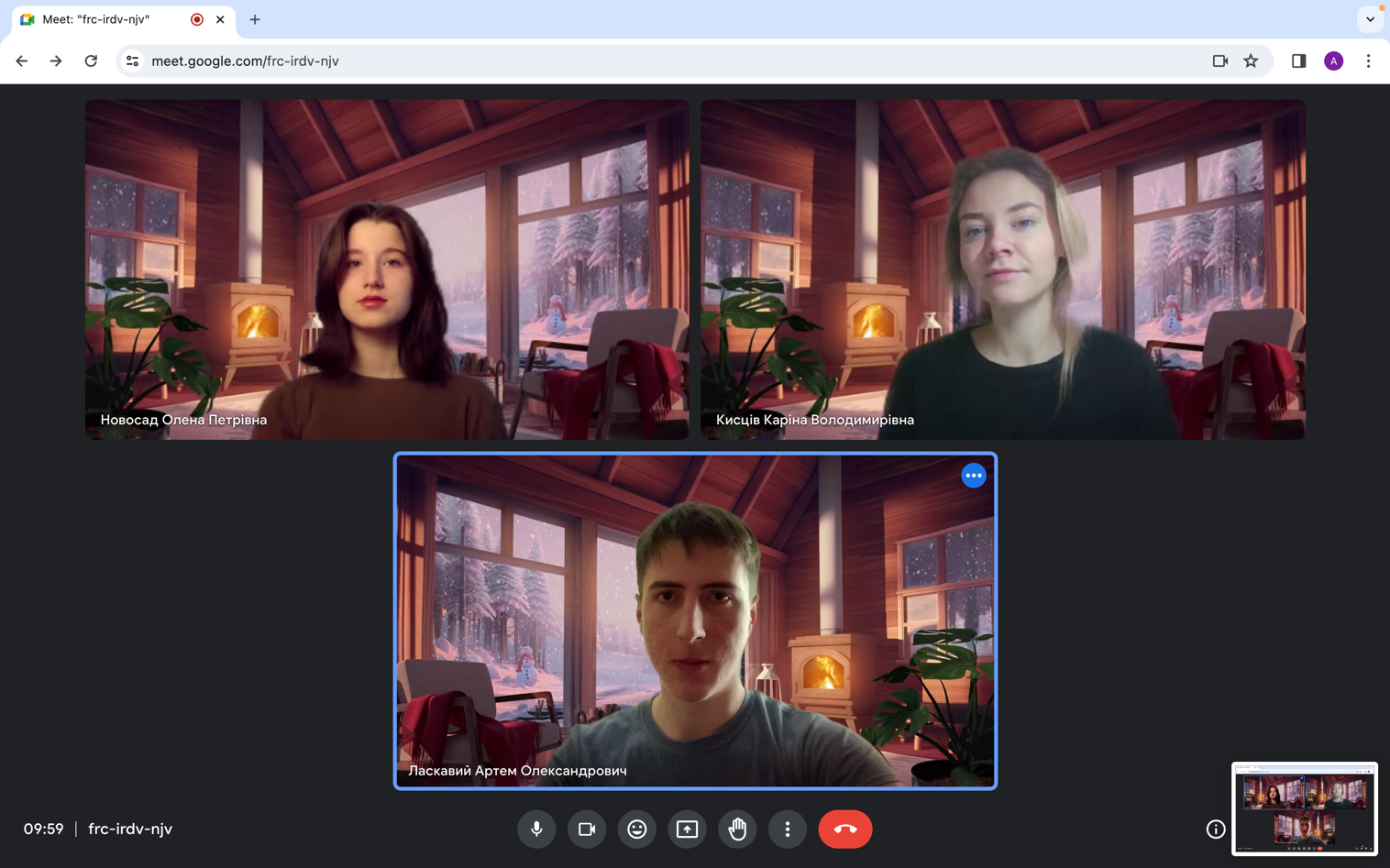
* Скрін з 1-ї зустрічі по обговоренню задач Епіку:
* 

Figure Скріншот зустрічі команди

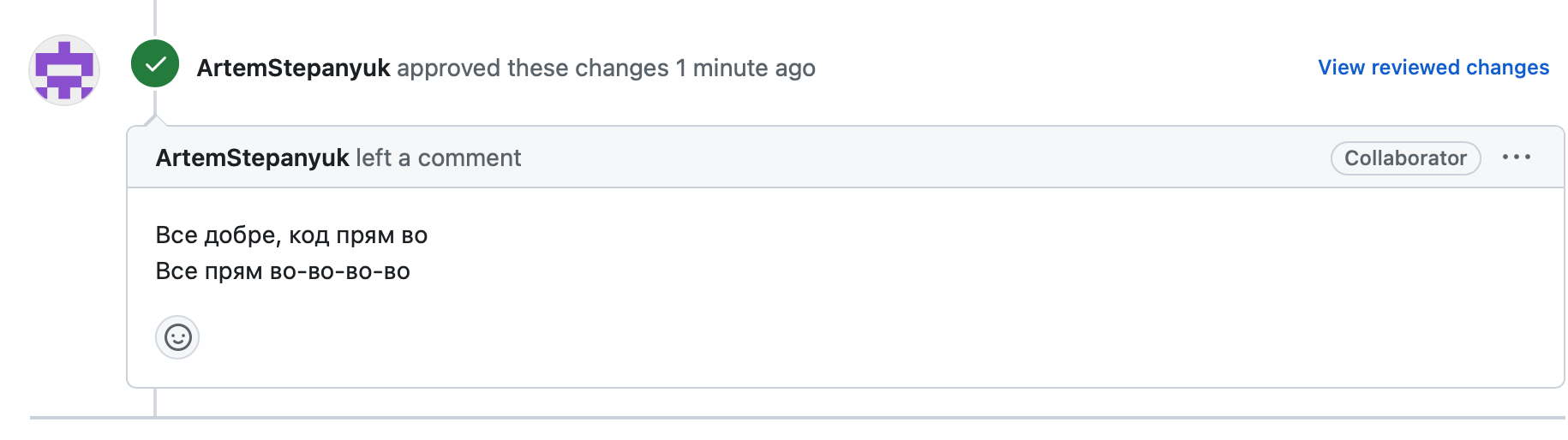
* Скрін з коментарем від учасника команди на пул реквесті з Ревю Роботи
* 

Figure скріншот з коментарями від учасників команди

# **Висновки:**

Під час виконання епіку 5 я ознайомився з бінарними та текстовими файлами, символьними та рядковими змінними, стандартною бібліотекою та системами числення.