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

***Виконала:***

студент групи ШІ-13

Присліпська Софія Андріївна

# **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:**

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 - Practice# programming: Self Practice Task

Task 10 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

Task 11 - Results Evaluation and Release

# **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>

Стаття.

* + - * <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: Ознайомлена
  + Початок опрацювання теми: 01.12.2023
  + Звершення опрацювання теми: 07.12.2023
* Тема №2: Символи і рядкові змінні
  + Джерела Інформації:
  + Стаття.

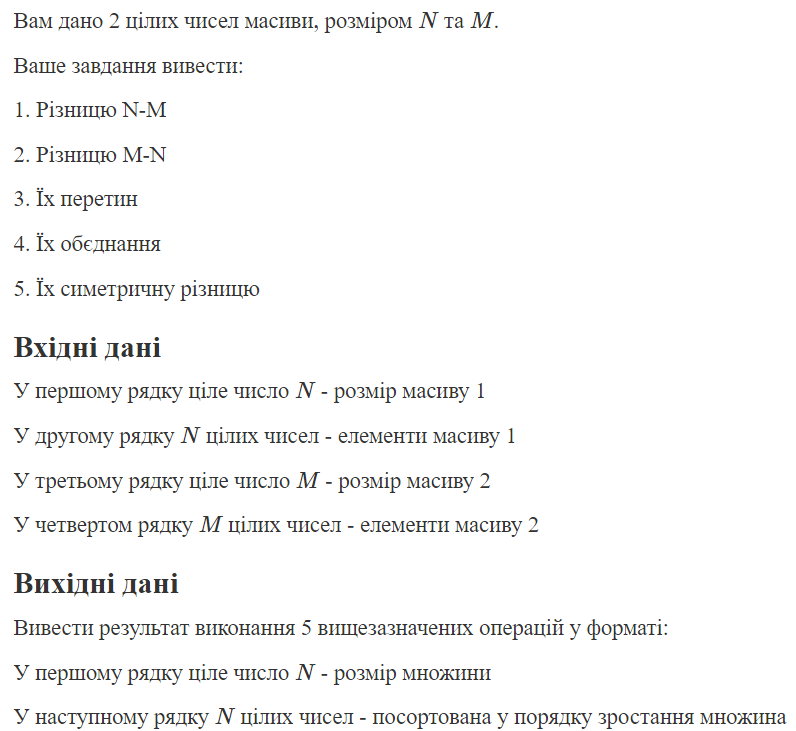
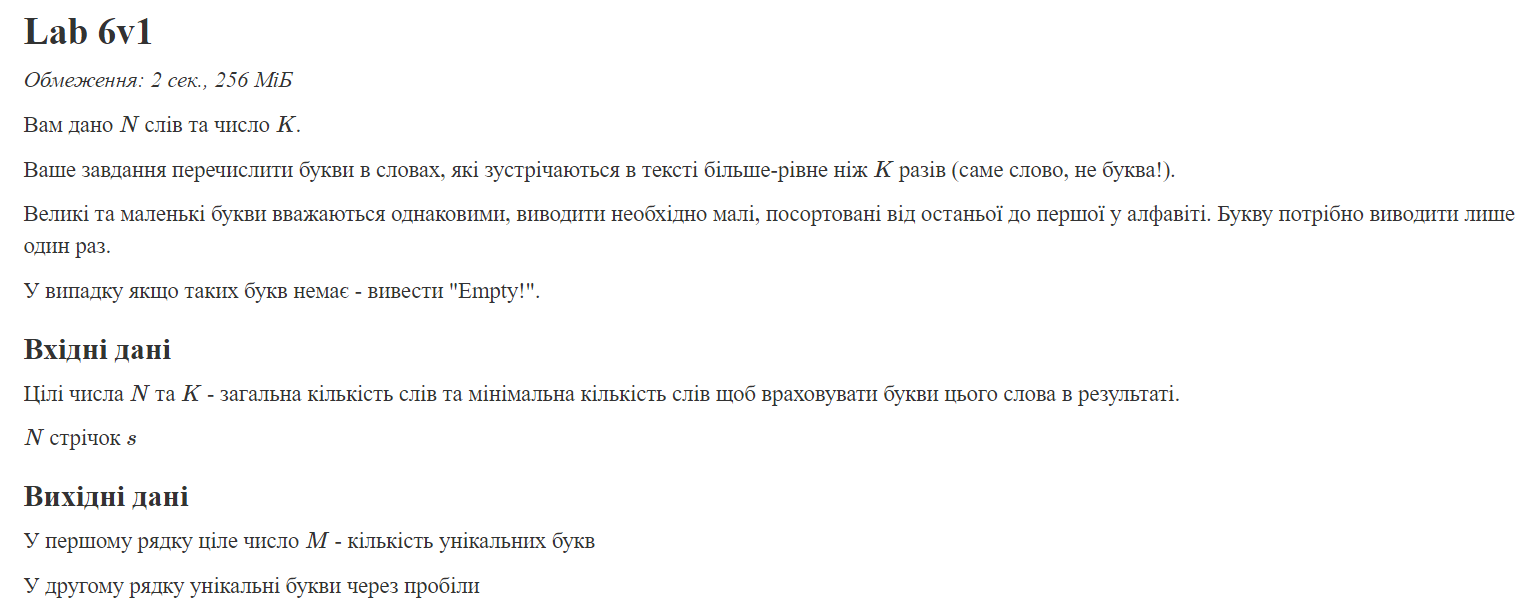
<https://www.geeksforgeeks.org/strings-in-cpp/>

* + Що опрацьовано:
    - Опрацьовано поняття символів і рядкових змінних, різницю між рядковими змінними в С і С++, корисні методи рядкових змінних у С++.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 01.11.2023
  + Звершення опрацювання теми: 07.12.2023
* Тема №3: Стандартна бібліотека
  + Джерела Інформації:
  + 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: Ознайомлена
  + Початок опрацювання теми: 08.11.2023
  + Звершення опрацювання теми: 07.12.2023

# **Виконання роботи:**

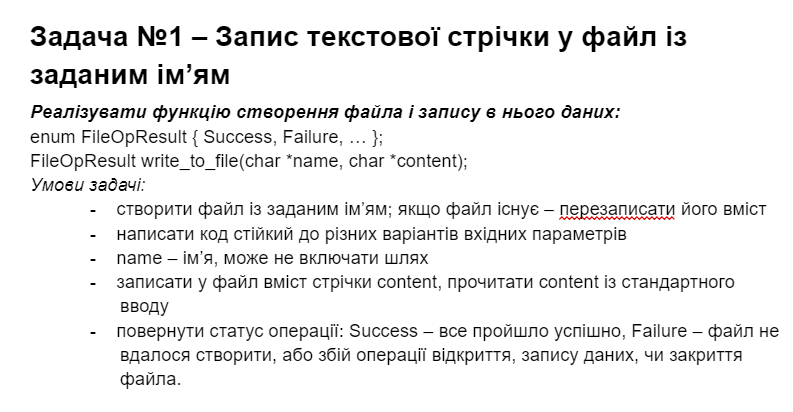
## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 **Algotester Lab 4**

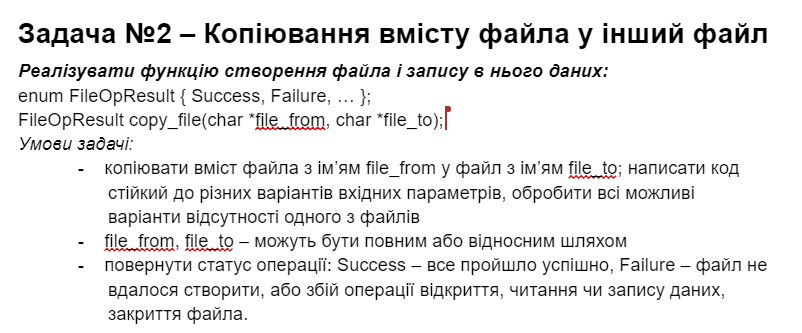
* Варіант завдання:1
* Деталі завдання:
* 
* *Figure 1.*
* Завдання №2 **Algotester Lab 6**
* Варіант завдання:1
* Деталі завдання:
* 

*Figure 1.*

Завдання №3 **Class Practice Work**

****

*Figure 2.*

****

*Figure 3.*

# Завдання №4 **Self Practice Work**

* деталі завдання:
* Створення файлу для запису

Завдання №5 **VNS Lab 6**

* Варіант завдання:11
* 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: Перетворити рядок таким чином, щоб всі слова в ньому були надруковані
* навпаки.

Завдання №\_6 **VNS Lab 8**

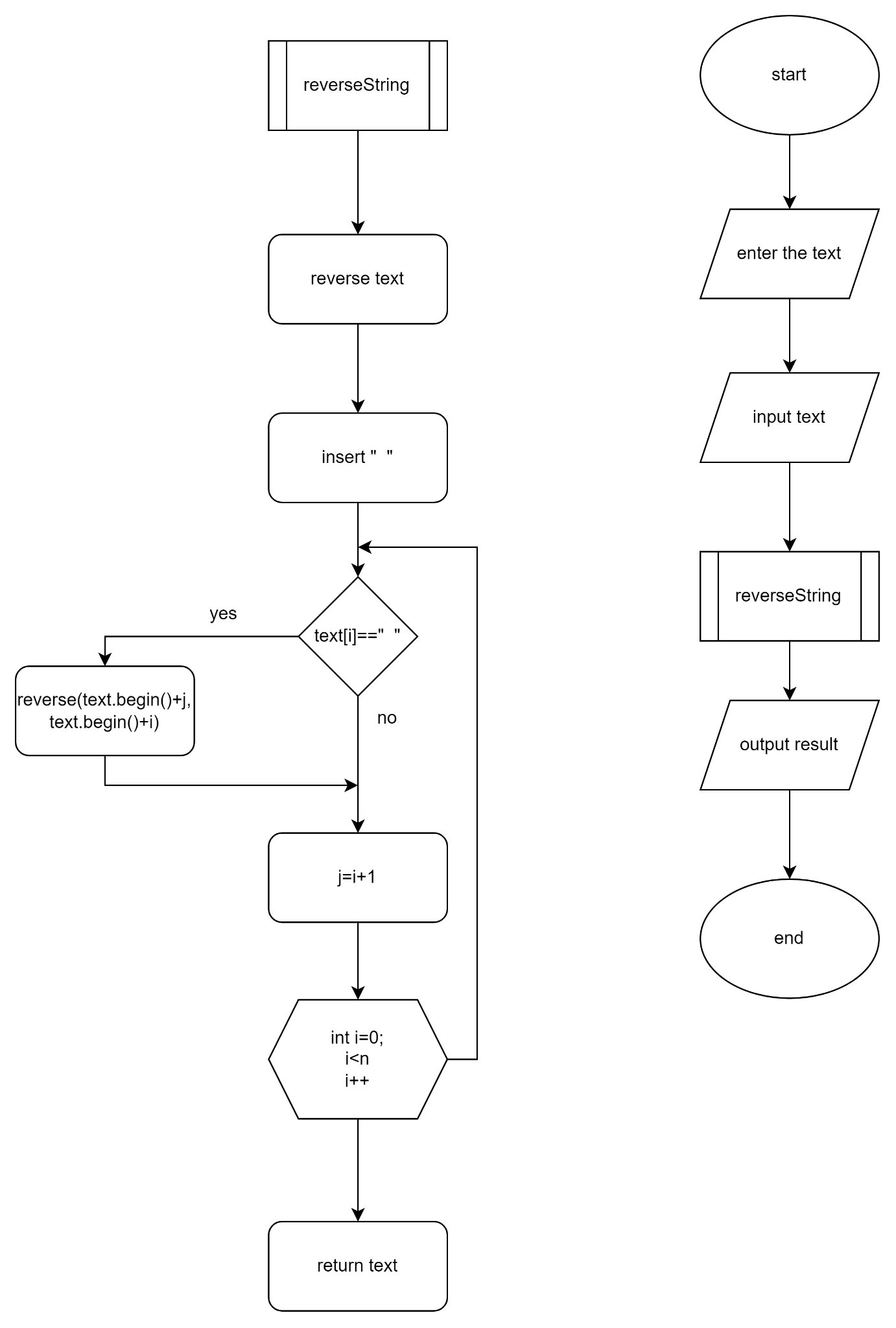
* Варіант завдання:11
* Деталі завдання:
* Сформувати двійковий файл із елементів, заданої у варіанті структури,
* роздрукувати його вмістиме, виконати знищення й додавання елементів у
* відповідності зі своїм варіантом, використовуючи для пошуку елементів що
* знищуються чи додаються, функцію. Формування, друк, додавання й знищення
* елементів оформити у вигляді функцій. Передбачити повідомлення про
* помилки при відкритті файлу й виконанні операцій вводу/виводу.
* Структура "Відеокасета":
* - назва фільму;
* - режисер;
* - тривалість;
* - ціна.
* Знищити всі елементи із ціною вищою заданої, додати 3 елементи в кінець
* файлу.

Завдання №\_7 **VNS Lab 9**

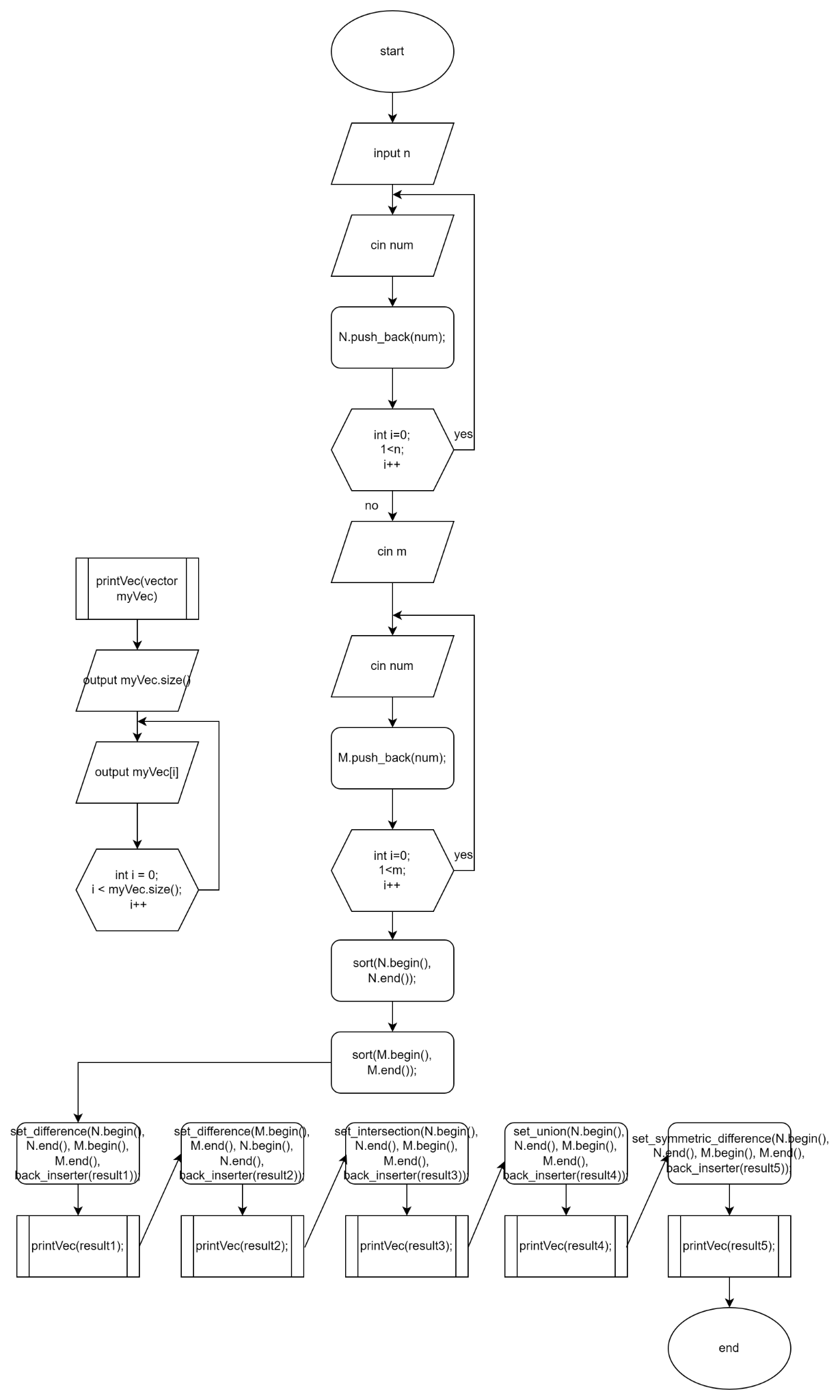
* Варіант завдання:11
* Деталі завдання:
* Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього
* інформацію.
* 1) Скопіювати з файлу F1 у файл F2 всі рядки, крім того рядка, що містить
* найкоротше слово.
* 2) Надрукувати номер цього рядка.

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

Програма №1 **VNS Lab 6 - Task 1**

* Блок-схема
* 
* Планований час на реалізацію:30 хв

Програма №1 **Algotester Lab 4**

* Блок-схема
* 
* Планований час на реалізацію:30 хв

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

Завдання №1 **Algotester Lab 4**

#include <bits/stdc++.h>

using namespace std;

int n, m;

vector<int> N, M;

void printVec(vector<int> myVec)

{

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

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

{

cout << myVec[i] << " ";

}

cout << endl

<< endl;

}

int main()

{

cin >> n;

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

{

int num;

cin >> num;

N.push\_back(num);

}

cin >> m;

for (int i = 0; i < m; i++)

{

int num;

cin >> num;

M.push\_back(num);

}

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

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

vector<int> result1;

set\_difference(N.begin(), N.end(), M.begin(), M.end(), back\_inserter(result1));

printVec(result1);

vector<int> result2;

set\_difference(M.begin(), M.end(), N.begin(), N.end(), back\_inserter(result2));

printVec(result2);

vector<int> result3;

set\_intersection(N.begin(), N.end(), M.begin(), M.end(), back\_inserter(result3));

printVec(result3);

vector<int> result4;

set\_union(N.begin(), N.end(), M.begin(), M.end(), back\_inserter(result4));

printVec(result4);

vector<int> result5;

set\_symmetric\_difference(N.begin(), N.end(), M.begin(), M.end(), back\_inserter(result5));

printVec(result5);

return 0;

}

Завдання №2 **Algotester Lab 6**

#include <iostream>

#include <string>

#include <vector>

#include <cctype>

#include <algorithm>

#include <unordered\_map>

#include <set>

using namespace std;

int main()

{

int k, n;

cin >> k >> n;

vector<string> words;

unordered\_map<string, int> count\_words;

// Введення слів та підрахунок їх кількості

for (int i = 0; i < k; ++i)

{

string word;

cin >> word;

transform(word.begin(), word.end(), word.begin(), ::tolower); // Перетворення слова в нижній регістр

words.push\_back(word);

count\_words[word]++;

}

vector<string> word\_result;

// Визначення слів, що зустрічаються принаймні n разів

for (const pair<string, int> &entry : count\_words)

{

if (entry.second >= n)

{

word\_result.push\_back(entry.first);

}

}

if (word\_result.empty())

{

cout << "Empty!" << endl;

return 0;

}

set<char> letters;

// Збір унікальних букв зі слів, що задовольняють умові

for (const string &word : word\_result)

{

letters.insert(word.begin(), word.end());

}

// Сортування букв у зворотньому алфавітному порядку

vector<char> sorted\_letters(letters.begin(), letters.end());

sort(sorted\_letters.begin(), sorted\_letters.end(), greater<char>());

cout << sorted\_letters.size() << endl;

for (const char &letter : sorted\_letters)

{

cout << letter << " ";

}

cout << endl;

return 0;

}

Завдання №3 **Class Practice Work**

**#include <iostream>**

**#include <fstream>**

**#include <string>**

**using namespace std;**

**enum FileOpResult**

**{**

**Success,**

**Failure**

**};**

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

**{**

**ofstream file(name);**

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

**{**

**return Failure;**

**}**

**file << content;**

**file.close();**

**return Success;**

**}**

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

**{**

**ifstream source(file\_from, ios::binary);**

**if (!source.is\_open())**

**{**

**return Failure;**

**}**

**ofstream destination(file\_to, ios::binary);**

**if (!destination.is\_open())**

**{**

**source.close();**

**return Failure;**

**}**

**destination << source.rdbuf();**

**source.close();**

**destination.close();**

**return Success;**

**}**

**int main()**

**{**

**// Записати стрічку у файл**

**string name = "test.txt";**

**string content;**

**cout << "Enter the text to write to the file: ";**

**getline(cin, content);**

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

**if (result == Success)**

**{**

**cout << "File created and saved successfully." << endl;**

**}**

**else**

**{**

**cout << "An error occurred while creating or writing to the file." << endl;**

**}**

**// Скопіювати вміст файлу**

**string file\_from, file\_to;**

**cout << "Enter the path to the file you want to copy: ";**

**getline(cin, file\_from);**

**cout << "Enter the path to the new file to copy to: ";**

**getline(cin, file\_to);**

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

**if (result == Success)**

**{**

**cout << "File copied successfully." << endl;**

**}**

**else**

**{**

**cout << "An error occurred while copying the file." << endl;**

**}**

**return 0;**

# Завдання №4 **Self Practice Work**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main()

{

// Відкриття файлу для запису

ofstream file("text.txt");

// Перевірка, чи вдалося відкрити файл

if (!file.is\_open())

{

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

return 1;

}

int n;

string wish;

cin >> n;

cin.ignore();

cout << "enter your wishes:" << endl;

file << " MY WISHLIST" << endl;

for (size\_t i = 1; i <= n; i++)

{

getline(cin, wish);

file << "wish № " << i << "\t" << wish << endl;

}

// Закриваємо файл

file.close();

cout << "The file has been successfully created and data has been written to it." << endl;

return 0;

}

Завдання №5 **VNS Lab 6 - Task 1**

#include <bits/stdc++.h>

#include <string>

using namespace std;

string reverseString(string text)

{

reverse(text.begin(), text.end());

// Add space at the end so that the last word is also reversed

text.insert(text.end(), ' ');

int n = text.length();

int j = 0;

// Find spaces and reverse all words before that

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

{

if (text[i] == ' ')

{

reverse(text.begin() + j, text.begin() + i);

j = i + 1;

}

}

// Remove spaces from the end of the word that we appended

text.pop\_back();

return text;

}

int main()

{

string text;

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

getline(cin, text);

cout << "You entered: " << text << endl;

string reverse = reverseString(text);

cout << "reversed text: " << reverse;

return 0;

}

Завдання №\_6 **VNS Lab 8 - Task 1**

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

struct Videotape

{

string title;

string director;

int duration;

int price;

};

void printVideotape(const Videotape &tape)

{

cout << "Title: " << tape.title << endl;

cout << "Director: " << tape.director << endl;

cout << "Duration: " << tape.duration << endl;

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

}

void writeToFile(const vector<Videotape> &tapes)

{

ofstream outFile("videotapes.bin", ofstream::binary | ofstream::trunc);

if (!outFile.is\_open())

{

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

return;

}

for (const Videotape &tape : tapes)

{

outFile.write(reinterpret\_cast<const char \*>(&tape), sizeof(Videotape));

}

outFile.close();

}

void deleteByPrice(int setPrice)

{

ifstream inFile("videotapes.bin", ifstream::binary);

if (!inFile.is\_open())

{

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

return;

}

vector<Videotape> tapes;

Videotape tape;

while (inFile.read(reinterpret\_cast<char \*>(&tape), sizeof(Videotape)))

{

if (tape.price <= setPrice)

{

tapes.push\_back(tape);

}

}

inFile.close();

ofstream outFile("videotapes.bin", ofstream::binary | ofstream::trunc);

if (!outFile.is\_open())

{

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

return;

}

for (const Videotape &t : tapes)

{

outFile.write(reinterpret\_cast<const char \*>(&t), sizeof(Videotape));

}

outFile.close();

}

void printBinary()

{

ifstream inFile("videotapes.bin", ifstream::binary);

if (!inFile.is\_open())

{

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

return;

}

Videotape tape;

int i = 0;

while (inFile.read(reinterpret\_cast<char \*>(&tape), sizeof(Videotape)))

{

cout << "Videotape " << ++i << ":\n";

printVideotape(tape);

cout << endl;

}

inFile.close();

}

void addToEnd(const Videotape &newTape)

{

ofstream outFile("videotapes.bin", ofstream::binary | ofstream::app);

if (!outFile.is\_open())

{

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

return;

}

outFile.write(reinterpret\_cast<const char \*>(&newTape), sizeof(Videotape));

outFile.close();

}

int main()

{

vector<Videotape> tapes = {

{"Home Alone", "John Hughes", 100, 170},

{"The Devil Wears Prada", "David Frankel", 90, 150},

{"Jumanji", "Joe Johnston", 110, 200},

{"Spiderman", "Sam Raimi", 80, 180},

};

writeToFile(tapes);

cout << "Initial binary file content:\n";

printBinary();

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

cout << "Enter the highest acceptable price for videotapes in the collection: ";

int nPrice;

cin >> nPrice;

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

deleteByPrice(nPrice);

cout << "Videotapes after deleting by price:\n";

printBinary();

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

cout << "Adding 3 new videotapes to the end of the file...\n";

for (int i = 0; i < 3; ++i)

{

Videotape newTape;

cout << "Enter details for new videotape " << i + 1 << ":\n";

cout << "Title: ";

cin.ignore();

getline(cin, newTape.title);

cout << "Director: ";

getline(cin, newTape.director);

cout << "Duration: ";

cin >> newTape.duration;

cout << "Price: ";

cin >> newTape.price;

addToEnd(newTape);

}

cout << "Updated binary file content:\n";

printBinary();

return 0;

}

Завдання №\_7 **VNS Lab 9 - Task 1**

#include <iostream>

#include <fstream>

#include <sstream>

#include <vector>

#include <algorithm>

#include <limits>

using namespace std;

// Функція для знаходження довжини найкоротшого слова у рядку

int shortestWordLength(const string &line)

{

istringstream iss(line);

string word;

int shortest = numeric\_limits<int>::max();

while (iss >> word)

{

shortest = min(shortest, static\_cast<int>(word.length()));

}

return shortest;

}

int main()

{

string filename1 = "F1.txt";

string filename2 = "F2.txt";

// Створення файлу F1 з рядками

ofstream file1(filename1);

if (!file1.is\_open())

{

cout << "Error creating file F1.txt." << endl;

return 1;

}

file1 << "Методичні вказівки\n";

file1 << "Методичні вказівки\n";

file1 << "Постановка завдання.\n";

file1 << "Опис використовуваних типів даних.\n";

file1 << "Текст функцій для:\n";

file1 << "формування файлу,\n";

file1 << "друку файлу,\n";

file1 << "копіювання файлів,\n";

file1 << "виконання завдання.\n";

file1 << "Результат розв’язання конкретного варіанту.\n";

file1.close();

// Відкриття файлу F1 для читання та F2 для запису

ifstream fileIn(filename1);

ofstream fileOut(filename2);

if (!fileIn.is\_open() || !fileOut.is\_open())

{

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

return 1;

}

vector<string> lines;

string line;

// Читання рядків з файлу F1 та зберігання їх у векторі

while (getline(fileIn, line))

{

lines.push\_back(line);

}

fileIn.close();

// Знаходження довжини найкоротшого слова у кожному рядку

vector<int> shortestLengths;

for (const auto &str : lines)

{

shortestLengths.push\_back(shortestWordLength(str));

}

// Знаходження індексу рядка, у якому слово має мінімальну довжину

auto minPos = min\_element(shortestLengths.begin(), shortestLengths.end());

int minIndex = distance(shortestLengths.begin(), minPos);

// Копіювання рядків (крім рядка з найкоротшим словом) до файлу F2

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

{

if (i != minIndex)

{

fileOut << lines[i] << endl;

}

}

fileOut.close();

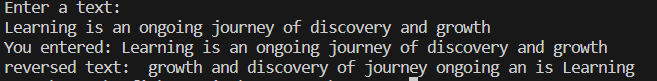
cout << "Successfully copied data from file F1.txt to file F2.txt, except for the line with the shortest word." << endl;

return 0;

}

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

Завдання №1 **VNS Lab 6 - Task 1**

****

*Figure 4.*

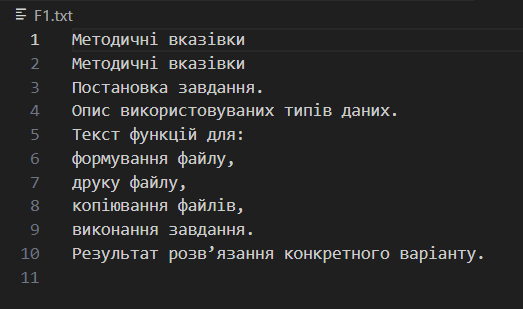
Завдання №\_2 **VNS Lab 8 - Task 1**

Завдання №\_3 **VNS Lab 9 - Task 1**

**output:**

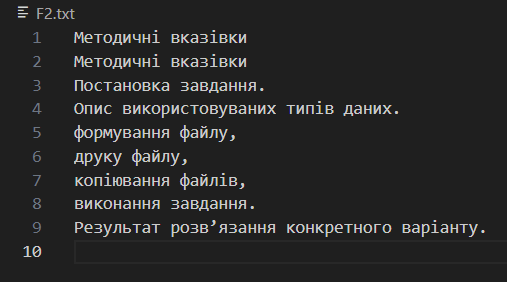
Successfully copied data from file F1.txt to file F2.txt, except for the line with the shortest word.

**file F1:**



*Figure 5.*

**file F2:**



*Figure 6.*

Завдання №4 **Algotester Lab 4**

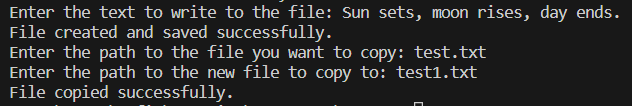
<https://algotester.com/uk/ProblemSolution/Display/1523970>

Завдання №5 **Algotester Lab 6**

<https://algotester.com/uk/ProblemSolution/Display/1523892>

Завдання №6 **Class Practice Work**

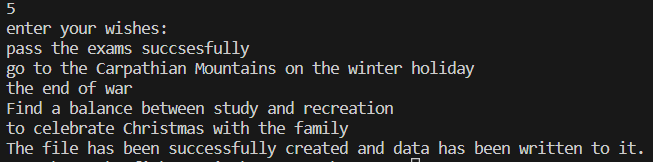
**output:**

****

*Figure 7.*

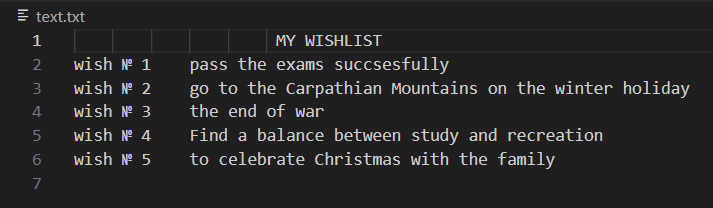
# Завдання №7 **Self Practice Work**

output:



*Figure 8.*

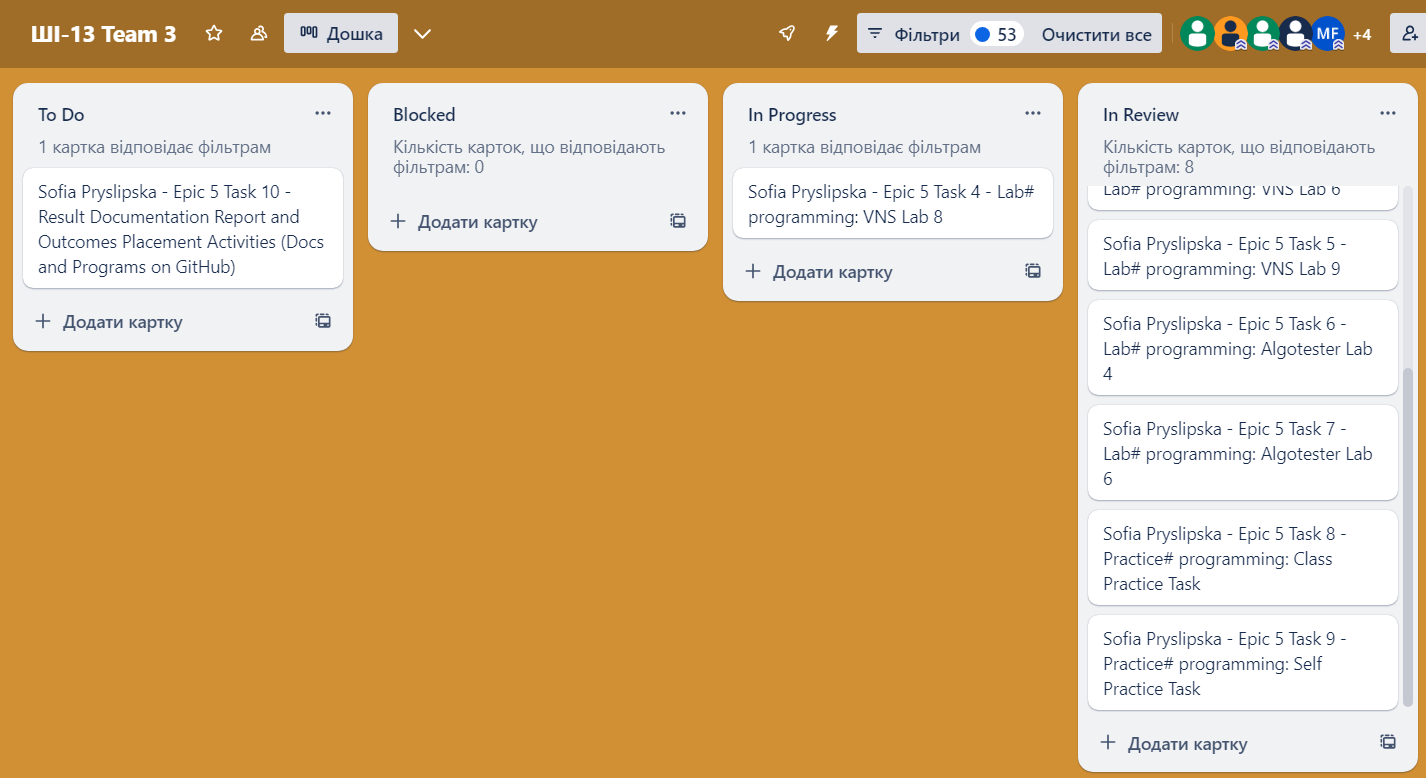
file:



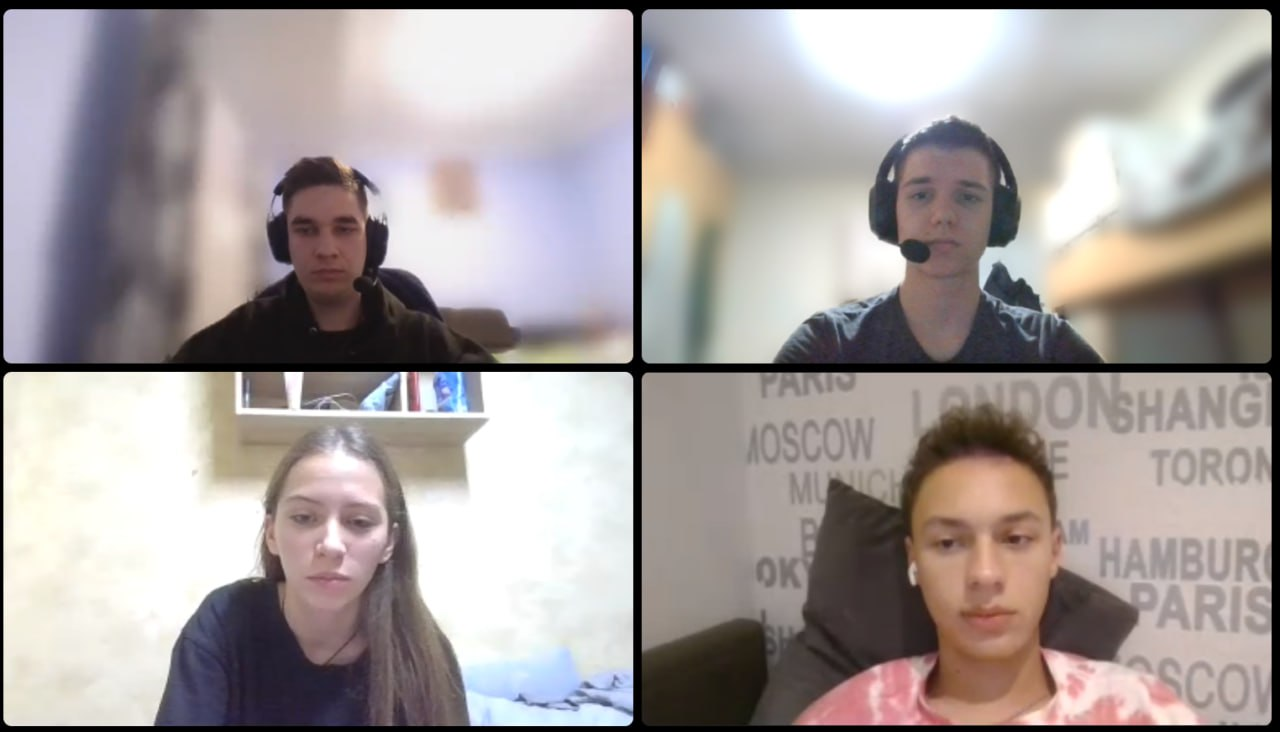
*Figure 9.*

Час затрачений на виконання завдання: 1 година

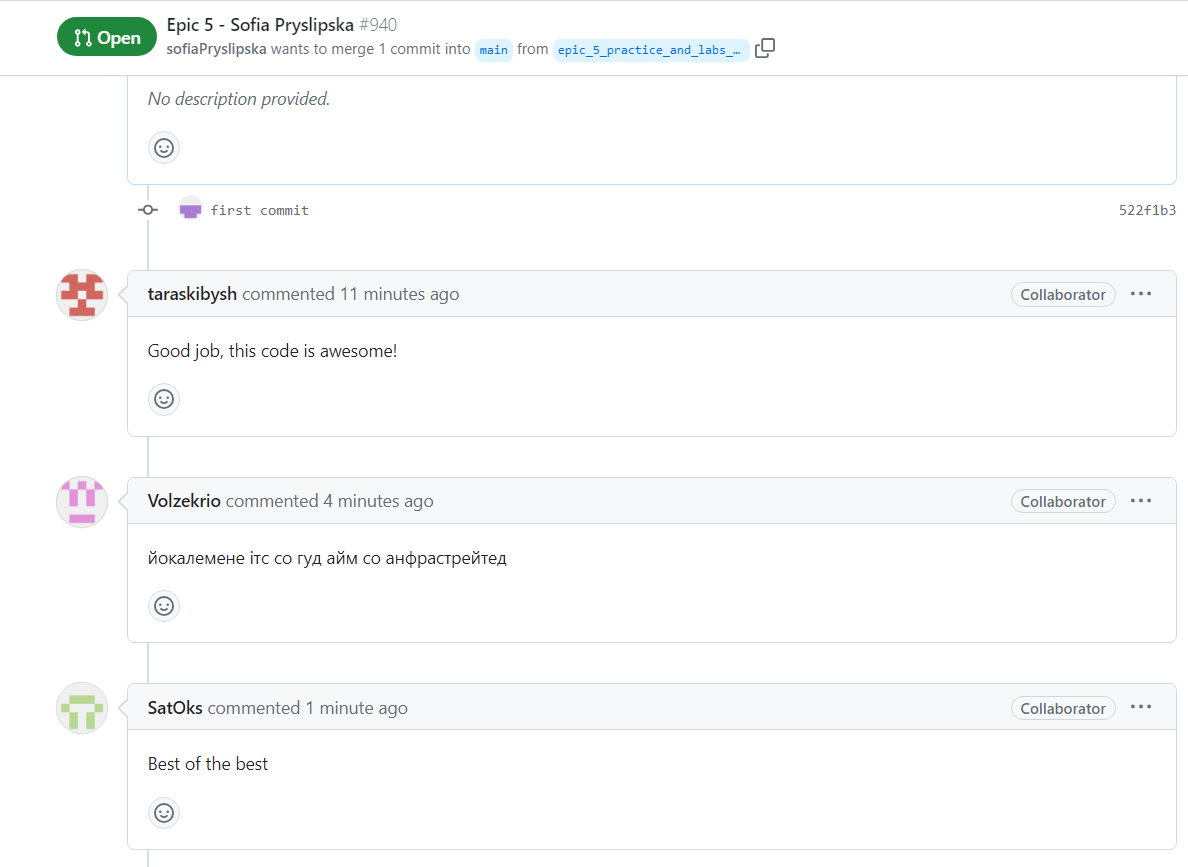
## **5. Кооперація з командою:**



*Figure 10.*



*Figure 11.*

**

*Figure 12.*

# **Висновки:**

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