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(ла):**

Студент групи ШІ-11

Попович Олег Васильович

# **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:**

Ознайомитися з роботою з файлами засобами мови C++. Системи числення. Бінарні Файли. Символи і рядкові змінні та текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

Виконати задані лабораторні роботи.

Виконати задане практичне завдання.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Робота з файлами в C++
* Тема №2: Символи і рядкові змінні та текстові файли
* Тема №3: Алгоритми роботи з файлами засобами мови C++

1. Індивідуальний план опрацювання теорії:

* Тема №1: Робота з файлами в C++.
  + Джерела Інформації
* “The C++ Programming Language” by Bjarne Stroustrup
* “C++ For Dummies” by Stephen R. Davis
  + Що опрацьовано:
* Поняття файлу та його типи
* Бібліотека fstream
* Оголошення та ініціалізація об'єктів введення та виведення
* Введення з файлу та запис у файл
* Введення даних до кінця файлу
* Закриття файлів
  + Статус: Ознайомлений
  + Початок опрацювання теми: 16.11.2023
  + Звершення опрацювання теми: 30.11.2023
* Тема №2: Символи і рядкові змінні та текстові файли
  + Джерела Інформації:
* “The C++ Programming Language” by Bjarne Stroustrup
* “C++ For Dummies” by Stephen R. Davis
  + Що опрацьовано:
* Поняття символу
* Типи символів
* Змінні символьного типу
* Поняття рядка
* Рядкові змінні
* Введення даних з клавіатури в символьні змінні
* Виведення даних з символьних змінних на екран
* Введення даних з текстового файлу в рядкові змінні
* Виведення даних з рядкових змінних у текстовий файл
  + Статус: Ознайомлений
  + Початок опрацювання теми: 16.11.2023
  + Звершення опрацювання теми: 30.11.2023
* Тема №3: Алгоритми роботи з файлами засобами мови C++
  + Джерела Інформації:
* “The C++ Programming Language” by Bjarne Stroustrup
* “C++ For Dummies” by Stephen R. Davis
  + Що опрацьовано:
* Алгоритм створення файлу
* Алгоритм запису даних у файл
* Алгоритм читання даних з файлу
* Алгоритм закриття файлу
  + Статус: Ознайомлений
  + Початок опрацювання теми: 16.11.2023
  + Звершення опрацювання теми: 30.11.2023

# **Виконання роботи:**

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

Завдання **№1 VNS Lab 6 Task 1**

* 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. Для рядка знайти кількість слів, що починаються на букву «а».

Завдання **№2  VNS Lab 8 Task 1.**

* 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 "Власник автомобіля":

* прізвище, ім'я, по батькові;
* номер автомобіля;
* номер техпаспорта;
* відділення реєстрації ДАІ.

Знищити елемент із заданим номером, додати 2 елементи перед елементом із

заданим прізвищем.

Завдання **№3  VNS Lab 9 Task 1.**

* Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію Скопіювати з файлу F1 у файл F2 всі рядки, у яких міститься тільки одне слово.

Визначити номер слова, у якому найбільше приголосних букв.

Завдання **№4  Algotester Lab 4v3.**

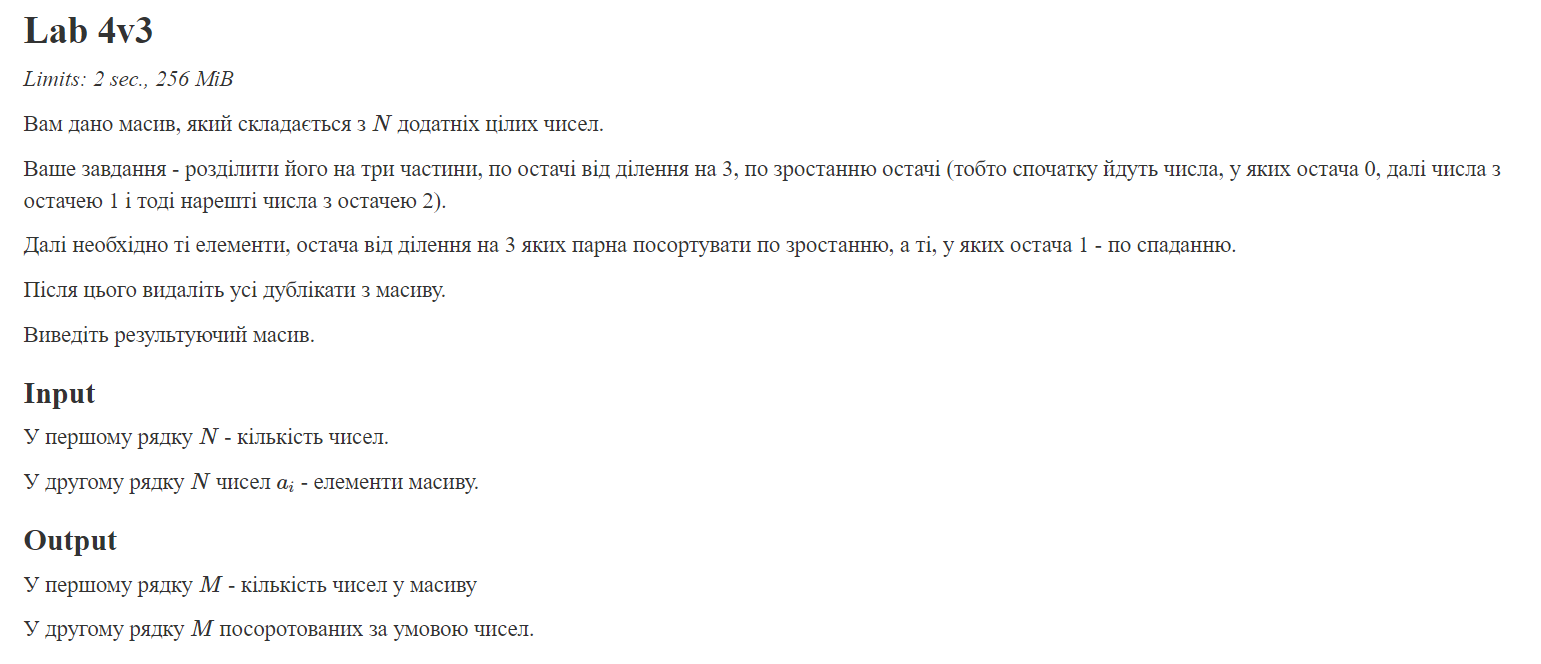


Рисунок 1 Завдання до Algotester Lab 4

Завдання **№5  Algotester Lab 6v3.**

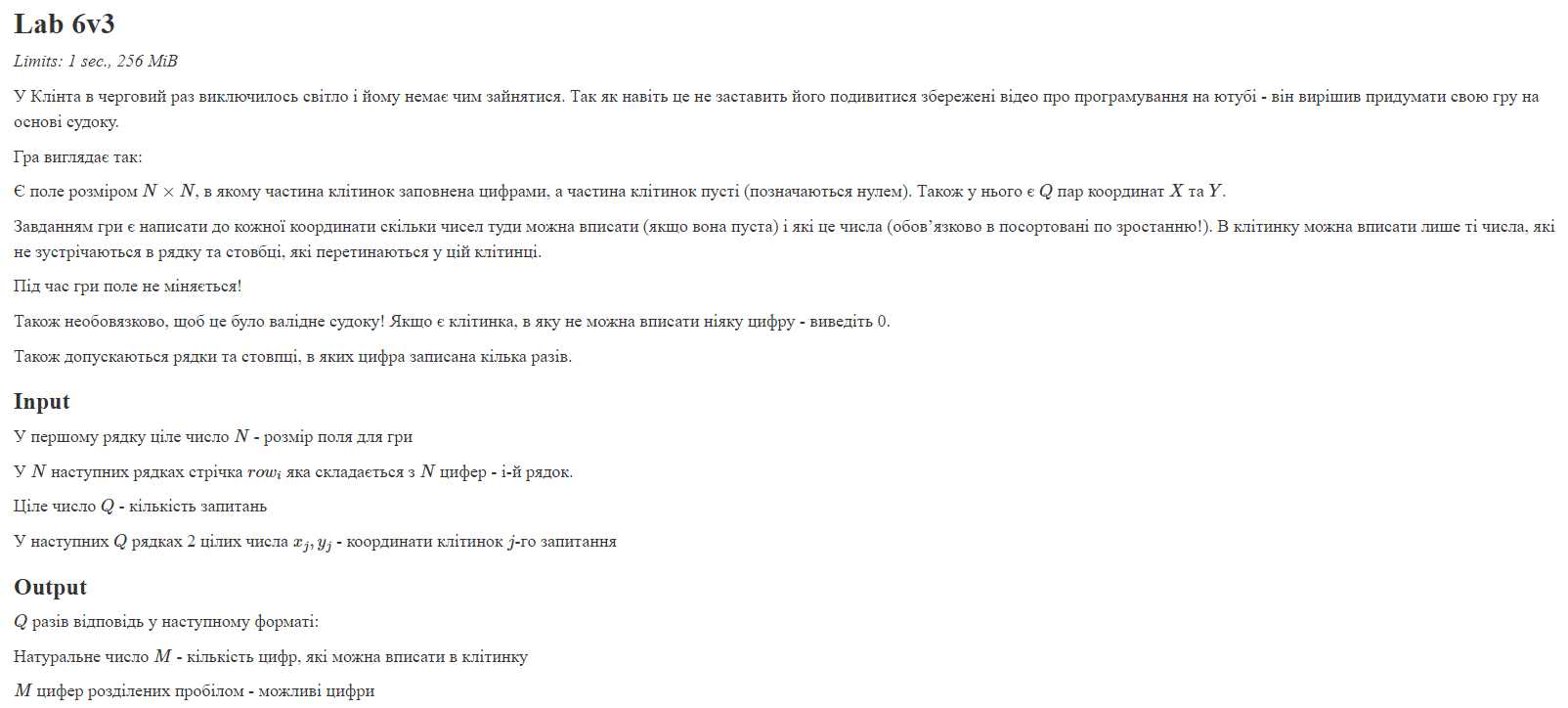


Рисунок 2 Завдання до Algotester Lab 6

Завдання **№6  Practice Work**

* **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.

* **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.

Завдання **№7 Self Practice Work**

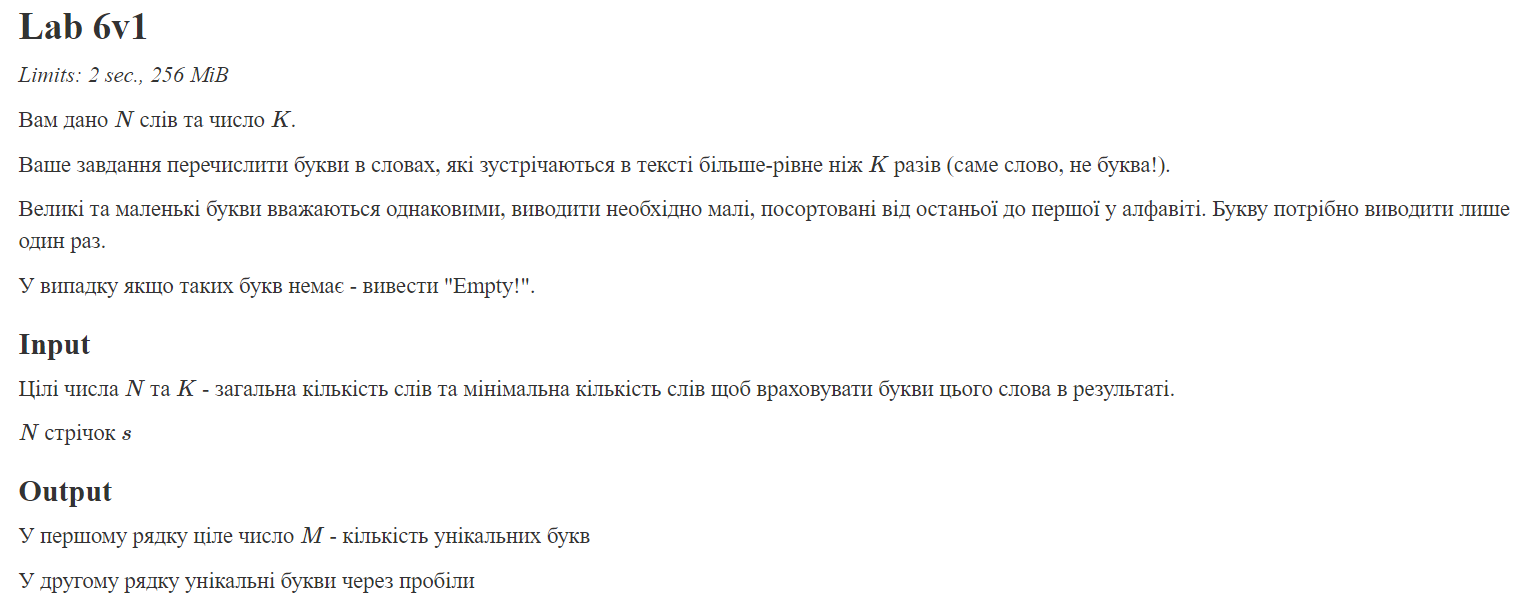


Рисунок 3 Завдання до Self Practice

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

Програма **№1 VNS Lab 6 Task 1**

* Планований час на реалізацію: **10 хвилин**.

Програма **№2 VNS Lab 8 Task 1**

* Планований час на реалізацію: **1 година**.

Програма **№3 VNS Lab 9 Task 1.**

* Блок-схема
* Планований час на реалізацію: **35 хвилин**.

Програма **№4 Algotester Lab 4 Task 1.**

* Блок-схема
* Планований час на реалізацію: **25 хвилин**.

Програма **№5 Algotester Lab 6 Task 1.**

* Блок-схема

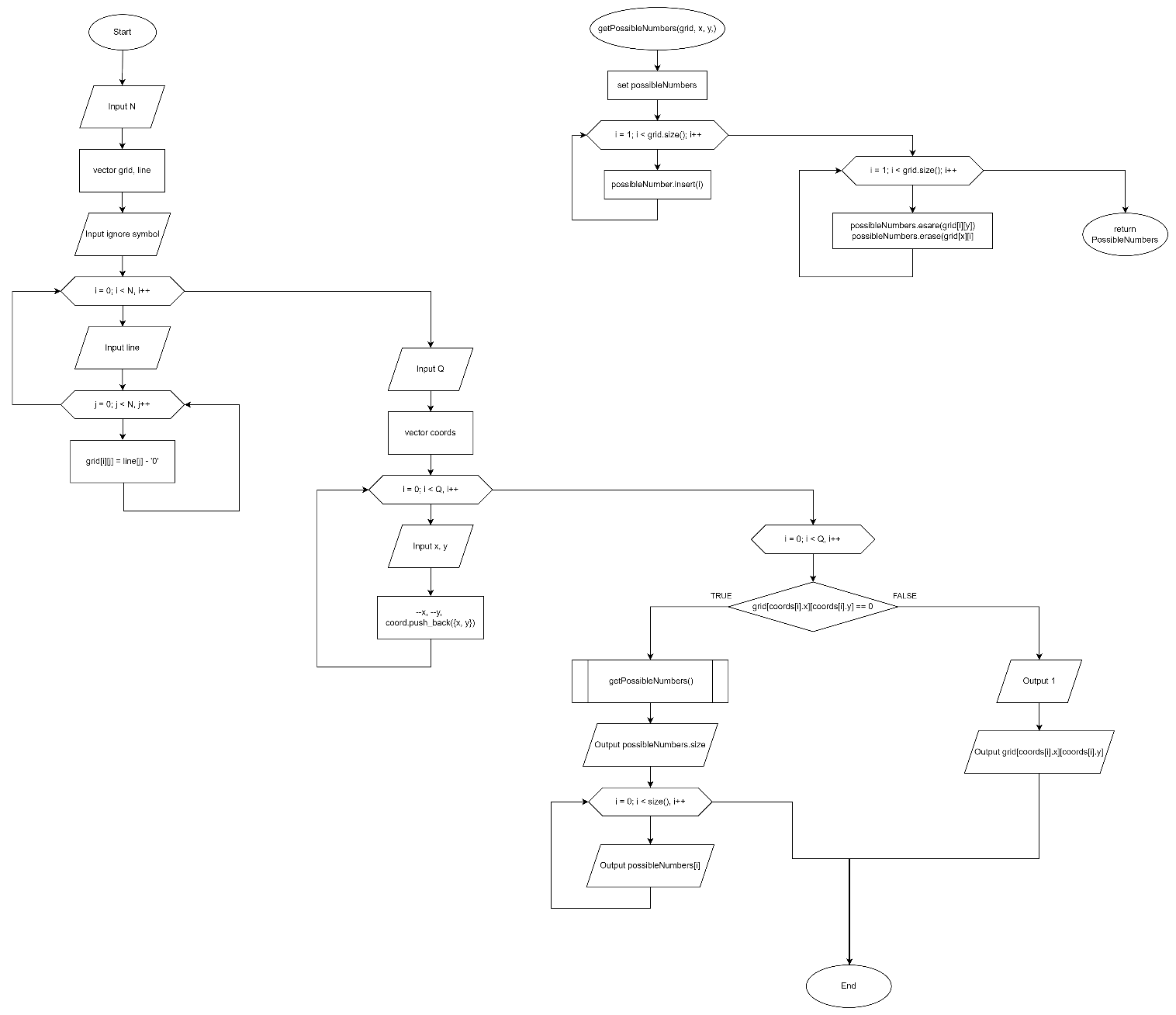


Рисунок 4 Блок схема до програми №3

* Планований час на реалізацію: **35 хвилин**.

Програма **№6 Practice Work.**

* Планований час на реалізацію: **15 хвилин**.

Програма **№7 Self Practice Work.**

* Планований час на реалізацію: **25 хвилин**.

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

Завдання **№1 VNS Lab 6 Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 12/6/2023.  
//  
#include <iostream>  
#include <cstring>  
  
int startsWith(char\*, char);  
  
int main(int argc, char const \*argv[])  
{  
 char src[255];  
 std::cout << "Please enter the line: ";  
 std::cin.getline(src, 256);  
 printf("There are %d words that starts with %c\n", startsWith(src, 'a'), 'a');  
 return 0;  
}  
  
int startsWith(char\* line, char letter) {  
 int counter = 0;  
 char\* token;  
 token = strtok\_s(line, " ");  
 while(token) {  
 if(\*token == 'a') {  
 counter++;  
 }  
 token = strtok\_s(nullptr, " ");  
 }  
 return counter;  
}

Рисунок 5 Код до програми №1

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/vns_lab_4_task_1_variant_22_oleh_popovych.cpp>

Завдання **№2 VNS Lab 8 Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 12/6/2023.  
//  
  
#include <iostream>  
#include <fstream>  
#include <vector>  
#include <string>  
#include <algorithm>  
  
  
struct CarOwner{  
 std::string firstName;  
 std::string lastName;  
 std::string carNumber;  
 std::string techPassportNumber;  
 std::string registrationNumber;  
};  
  
void deleteByCarNumber(std::vector<CarOwner>&, const std::string&);  
void print(const std::vector<CarOwner>&);  
void addOwner(std::vector<CarOwner>&, const std::string&, const CarOwner&, const CarOwner&);  
  
int main() {  
 std::vector<CarOwner> carOwners = **{** {"Ivanov", "Ivan", "ABC123", "TP123", "Division1"},  
 {"Petrov", "Petr", "XYZ789", "TP456", "Division2"},  
 {"Oleh", "Oleh", "ABD252", "TA225", "Division3"}  
 **}**;  
  
 std::ofstream outFile("file1.bin", std::ios::binary);  
 if (!outFile.is\_open()) {  
 std::cerr << "Unable to open this file" << std::endl;  
 return -1;  
 }  
  
 outFile.write(reinterpret\_cast<const char\*>(carOwners.data()), carOwners.size() \* sizeof(CarOwner));  
 outFile.close();  
  
  
 std::ifstream inFile("file1.bin", std::ios::binary);  
 if (!inFile.is\_open()) {  
 std::cerr << "Unable to open this file" << std::endl;  
 return -1;  
 }  
  
 std::vector<CarOwner> readOwners;  
 CarOwner tempOwner;  
  
 while (inFile.read(reinterpret\_cast<char\*>(&tempOwner), sizeof(CarOwner))) {  
 readOwners.push\_back(tempOwner);  
 }  
 inFile.close();  
  
 std::cout << "Content of file: " << std::endl;  
 print(readOwners);  
  
 deleteByCarNumber(readOwners, "XYZ789");  
  
 CarOwner newOwner1 = {"New1", "Ivanovich", "123ABC", "TP456", "Division3"};  
 CarOwner newOwner2 = {"New2", "Ivanovich", "456DEF", "TP789", "Division4"};  
 addOwner(readOwners, "Ivanov", newOwner1, newOwner2);  
  
 std::ofstream outResFile("file1.bin", std::ios::binary);  
 if (!outResFile.is\_open()) {  
 std::cerr << "Unable to open this file" << std::endl;  
 return -1;  
 }  
  
 outResFile.write(reinterpret\_cast<const char\*>(readOwners.data()), readOwners.size() \* sizeof(CarOwner));  
 outResFile.close();  
  
 std::cout << "Updated content of file:" << std::endl;  
 std::ifstream inResFile("file1.bin", std::ios::binary);  
 if (!inResFile.is\_open()) {  
 std::cerr << "Unable to open this file" << std::endl;  
 return -1;  
 }  
  
 std::vector<CarOwner> newOwners;  
  
 while (inResFile.read(reinterpret\_cast<char\*>(&tempOwner), sizeof(CarOwner))) {  
 newOwners.push\_back(tempOwner);  
 }  
 inResFile.close();  
 print(newOwners);  
  
 return 0;  
}  
  
  
void print(const std::vector<CarOwner>& owners) {  
 for (const auto& owner : owners) {  
 std::cout << "First name: " << owner.firstName << std::endl;  
 std::cout << "Second name: " << owner.lastName << std::endl;  
 std::cout << "Car number: " << owner.carNumber << std::endl;  
 std::cout << "Tech passport number: " << owner.techPassportNumber << std::endl;  
 std::cout << "Registration number: " << owner.registrationNumber << std::endl;  
 std::cout << "--------------------------------------" << std::endl;  
 }  
}  
  
void addOwner(std::vector<CarOwner>& owners,  
 const std::string& targetLastName,  
 const CarOwner& newOwner1,  
 const CarOwner& newOwner2) {  
  
 const std::vector<CarOwner>::iterator &it = std::find\_if(owners.begin(), owners.end(),  
 [&targetLastName](const CarOwner &owner) {  
 return owner.lastName == targetLastName;  
 });  
  
 owners.insert(it, newOwner1);  
 owners.insert(it, newOwner2);  
}  
  
  
void deleteByCarNumber(std::vector<CarOwner>& owners, const std::string& targetNumber) {  
  
 const std::vector<CarOwner>::iterator &it = remove\_if(owners.begin(), owners.end(),  
 [&targetNumber](const CarOwner& owner) {  
 return owner.carNumber == targetNumber;  
 });  
 owners.erase(it, owners.end());  
}

Рисунок 6 Код до програми №2

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/vns_lab_4_task_1_variant_22_oleh_popovych.cpp>

Завдання **№3 VNS Lab 9 Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 12/7/2023.  
//  
#include <fstream>  
#include <iostream>  
#include <sstream>  
  
int createFile(const std::string&, const std::string&);  
void copyLines(const std::string&, const std::string&);  
void printFile(const std::string&);  
  
int main(int argc, char const \*argv[])  
{  
 std::string input = "Hello, world\nHeaven\nBrotherhood is everything\nHome";  
 std::string file1 = "f1.txt";  
 std::string file2 = "f2.txt";  
 createFile(file1, input);  
 copyLines(file1, file2);  
  
 std::cout << "The content of file 1:" << std::endl;  
 printFile(file1);  
 std::cout << "----------------------------------------------------" << std::endl;  
 std::cout << "The content of file 2:" << std::endl;  
 printFile(file2);  
 return 0;  
}  
  
int createFile(const std::string& filename, const std::string& content) {  
 std::ofstream outFile(filename);  
 if (!outFile.is\_open()) {  
 std::cerr << "Unable to open this file" << std::endl;  
 return -1;  
 }  
 outFile.write(content.c\_str(), content.length());  
 return 0;  
}  
  
void copyLines(const std::string& inputFile, const std::string& outputFile) {  
 std::ifstream inFile(inputFile);  
 std::ofstream outFile(outputFile);  
  
 if (!inFile.is\_open() || !outFile.is\_open()) {  
 std::cerr << "Unable to open the files." << std::endl;  
 return;  
 }  
  
 std::string line;  
 std::string word;  
 int counter;  
 while (std::getline(inFile, line)) {  
 counter = 0;  
 std::stringstream ss(line);  
 while(std::getline(ss, word, ' ')) {  
 counter++;  
 }  
 if(counter == 1) {  
 line += "\n";  
 outFile.write(line.c\_str(), line.length());  
 }  
 }  
  
 inFile.close();  
 outFile.close();  
}  
  
void printFile(const std::string& filename) {  
 std::ifstream inFile(filename);  
 if (!inFile.is\_open()) {  
 std::cerr << "Unable to open this file" << std::endl;  
 return;  
 }  
  
 std::string line;  
 while (std::getline(inFile, line)) {  
 std::cout << line << std::endl;  
 }  
 inFile.close();  
}

Рисунок 7 Код до програми №3

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/vns_lab_4_task_1_variant_22_oleh_popovych.cpp>

Завдання **№4 Algotester Lab 4 Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 12/7/2023.  
//  
#include <iostream>  
#include <vector>  
#include <algorithm>  
  
bool comparator(int, int);  
  
int main(int argc, char const \*argv[])  
{  
 int N;  
 std::cin >> N;  
  
 std::vector<int> numbers(N);  
 for (int i = 0; i < N; ++i) {  
 std::cin >> numbers[i];  
 }  
  
 const std::vector<int>::iterator &partitionIt =  
 std::partition(numbers.begin(), numbers.end(),  
 [](int num) {  
 return num % 3 == 0;  
 });  
  
  
 std::sort(numbers.begin(), partitionIt, comparator);  
 std::sort(partitionIt, numbers.end(), comparator);  
  
  
 const std::vector<int>::iterator &uniqueIt =  
 std::unique(numbers.begin(), numbers.end());  
 numbers.erase(uniqueIt, numbers.end());  
  
 std::cout << numbers.size() << std::endl;  
 for (int num : numbers) {  
 std::cout << num << " ";  
 }  
  
 return 0;  
}  
  
bool comparator(int a, int b) {  
 if (a % 3 != b % 3) {  
 return a % 3 < b % 3;  
 }  
  
 if (a % 3 == 0) {  
 return a < b;  
 } else if (a % 3 == 1) {  
 return a > b;  
 }  
  
 return a < b;  
}

Рисунок 8 Код до програми №4

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/vns_lab_4_task_1_variant_22_oleh_popovych.cpp>

Завдання **№5 Algotester Lab 6 Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 12/7/2023.  
//  
#include <iostream>  
#include <vector>  
#include <set>  
#include <algorithm>  
  
struct Pair{  
 int x;  
 int y;  
};  
  
std::set<int> getPossibleNumbers(const std::vector<std::vector<int>>&, int, int, int);  
  
int main(int argc, char const \*argv[])  
{  
 int N;  
 std::cin >> N;  
  
 std::vector<std::vector<int>> grid(N, std::vector<int>(N));  
 std::string line;  
 std::cin.ignore();  
 for (int i = 0; i < N; i++) {  
 std::getline(std::cin, line);  
 for(int j = 0; j < N; j++) {  
 grid[i][j] = line[j] - '0';  
 }  
 }  
  
 int Q;  
 std::cin >> Q;  
 std::vector<Pair> coords;  
 for(int i = 0; i < Q; i++) {  
 int x, y;  
 std::cin >> x >> y;  
 --x;  
 --y;  
 coords.push\_back({x, y});  
 }  
  
 for (int i = 0; i < Q; i++) {  
 if (grid[coords[i].x][coords[i].y] == 0) {  
 std::set<int> possibleNumbers = getPossibleNumbers(grid, coords[i].x, coords[i].y, N);  
  
 std::cout << possibleNumbers.size() << std::endl;  
 for (int number : possibleNumbers) {  
 std::cout << number << " ";  
 }  
 std::cout << std::endl;  
 } else {  
 std::cout << 1 << std::endl;  
 std::cout << grid[coords[i].x][coords[i].y] << std::endl;  
 }  
 std::cout << std::endl;  
 }  
  
 return 0;  
}  
  
std::set<int> getPossibleNumbers(const std::vector<std::vector<int>>& grid, int x, int y, int N) {  
 std::set<int> possibleNumbers;  
  
  
 for (int i = 1; i <= grid.size(); i++) {  
 possibleNumbers.insert(i);  
 }  
  
 for (int i = 0; i < grid.size(); i++) {  
 possibleNumbers.erase(grid[i][y]);  
 possibleNumbers.erase(grid[x][i]);  
 }  
  
 return possibleNumbers;  
}

Рисунок 9 Код до програми №5

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/vns_lab_4_task_1_variant_22_oleh_popovych.cpp>

Завдання **№6 Practice Work Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 12/6/2023.  
//  
#include <fstream>  
#include <cstring>  
#include <iostream>  
  
enum FileOpResult {  
 Success, Failure  
};  
  
FileOpResult writeToFile(const char\*, const char\*);  
FileOpResult copyFile(const char\*, const char\*);  
  
int main(int argc, char const \*argv[])  
{  
 std::string name;  
 std::string line;  
 std::string content;  
  
 std::cout << "Please enter the file name: ";  
 std::cin >> name;  
 std::cout << "Please enter the content to write into the file: ";  
 std::cin.ignore();  
 while(std::getline(std::cin, line) && line != "q") {  
 content += line + "\n";  
 }  
 FileOpResult result = writeToFile(name.c\_str(), content.c\_str());  
 std::cout << result << std::endl;  
  
  
 std::string src;  
 std::string dest;  
  
 std::cout << "Please enter the name of file which to copy: ";  
 std::cin >> src;  
 std::cout << "Please enter the name of file where to copy: ";  
 std::cin >> dest;  
 result = copyFile(src.c\_str(), dest.c\_str());  
 std::cout << result << std::endl;  
 return 0;  
}  
  
FileOpResult writeToFile(const char\* name, const char\* content) {  
 std::ofstream file(name);  
 if(file.is\_open()) {  
 file.write(content, strlen(content));  
 } else {  
 return Failure;  
 }  
 file.close();  
 return Success;  
}  
  
FileOpResult copyFile(const char\* fileFrom, const char\* fileTo) {  
 std::ifstream src(fileFrom);  
 std::ofstream dest(fileTo);  
 std::string line;  
  
 if(src.is\_open() && dest.is\_open()) {  
 while(std::getline(src, line)) {  
 line += '\n';  
 dest.write(line.c\_str(), line.length());  
 }  
 } else {  
 return Failure;  
 }  
  
 src.close();  
 dest.close();  
 return Success;  
}

Рисунок 10 Код до програми №6

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/vns_lab_4_task_1_variant_22_oleh_popovych.cpp>

Завдання **№7 Self Practice Task 1**

Деталі по програмі + Вставка з кодом з підписами до вставки

//  
// Created by olehio-p on 12/7/2023.  
//  
#include <iostream>  
#include <unordered\_map>  
#include <set>  
  
  
int main() {  
 int N, K;  
 std::cin >> N >> K;  
  
 std::unordered\_map<std::string, int> wordCount;  
  
 std::string word;  
 for (int i = 0; i < N; i++) {  
 std::cin >> word;  
 for(char& j : word) {  
 j = tolower(j);  
 }  
  
 wordCount[word]++;  
 }  
  
 auto cmp = [](char a, char b) { return a > b; };  
 std::set<char, decltype(cmp)> resultLetters(cmp);  
  
 for (const auto& pair : wordCount) {  
 if (pair.second >= K) {  
 std::set<char> uniqueLetters(pair.first.begin(), pair.first.end());  
  
 for (char letter : uniqueLetters) {  
 resultLetters.insert(letter);  
 }  
 }  
 }  
  
 if (resultLetters.empty()) {  
 std::cout << "Empty!" << std::endl;  
 } else {  
 std::cout << resultLetters.size() << std::endl;  
 for (char letter : resultLetters) {  
 std::cout << letter << " ";  
 }  
 std::cout << std::endl;  
 }  
 return 0;  
}

Рисунок 11 Код до програми №7

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/epic_4_practice_and_labs_oleh_popovych/ai_11/oleh_popovych/epic_4/vns_lab_4_task_1_variant_22_oleh_popovych.cpp>

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

Завдання **№1 VNS Lab 6 Task 1**

Деталі виконання і тестуванню програми

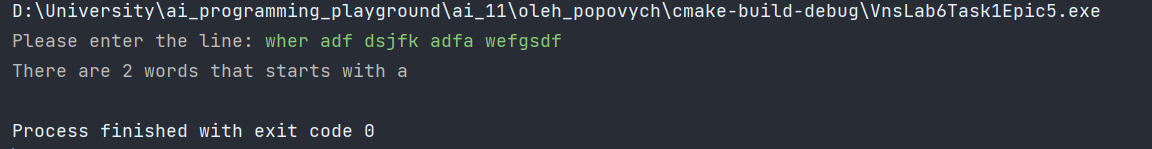


Рисунок 12 Результат виконання програми №1

Час затрачений на виконання завдання: **10 хвилин**.

Завдання **№2  VNS Lab 8 Task 1.**

Деталі виконання і тестуванню програми

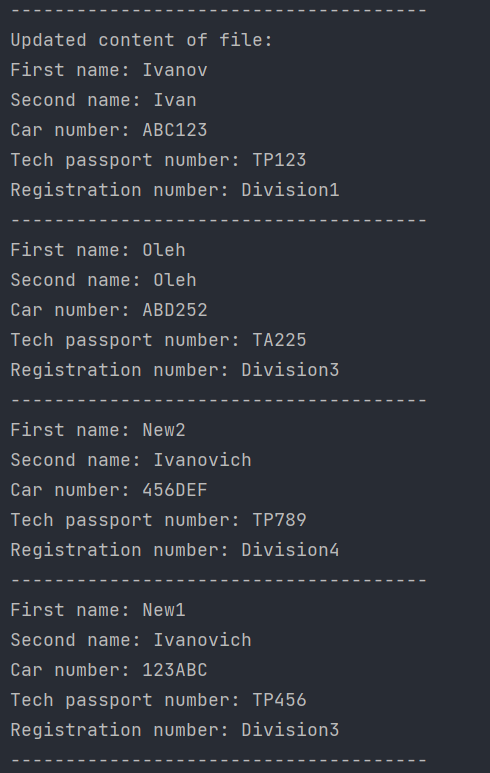


Рисунок 13 Результат виконання програми №2

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

Завдання **№3 VNS Lab 9 Task 1**

Деталі виконання і тестуванню програми

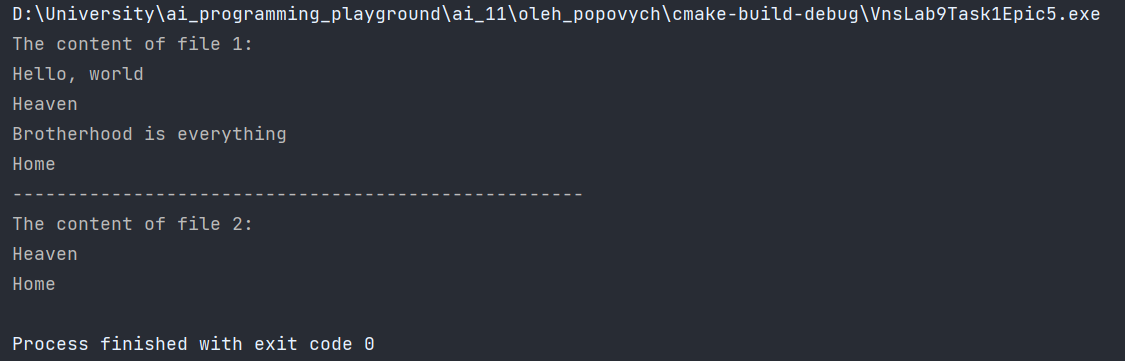


Рисунок 14 Результат виконання програми №3

Час затрачений на виконання завдання: **35 хвилин**.

Завдання **№4  Algotester Lab 4 Task 1**

Деталі виконання і тестуванню програми



Рисунок 15 Результат виконання програми №4

Час затрачений на виконання завдання: **25 хвилин.**

Завдання **№5  Algotester Lab 6 Task 1**

Деталі виконання і тестуванню програми

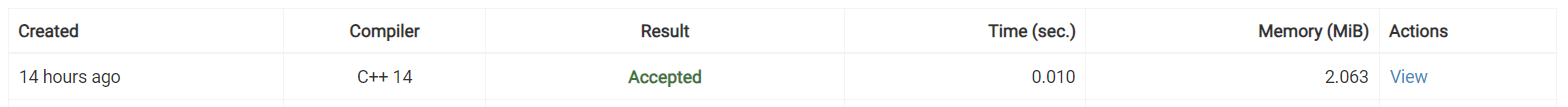


Рисунок 16 Результат виконання програми №5

Час затрачений на виконання завдання: **45 хвилин.**

Завдання **№6  Practice Work.**

Деталі виконання і тестуванню програми

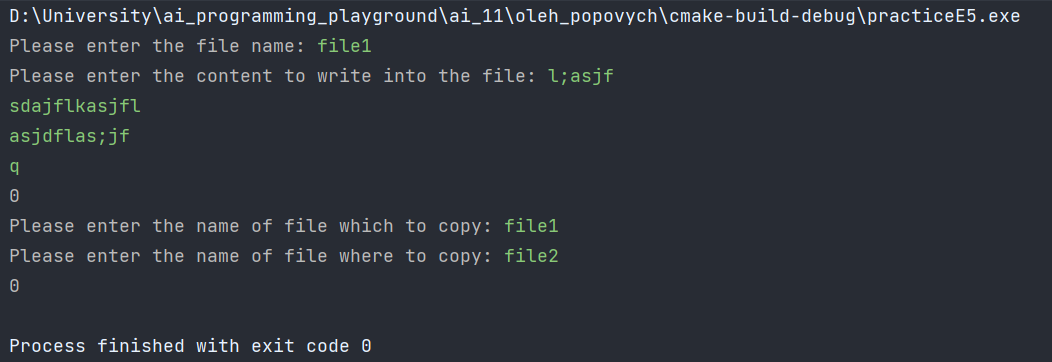


Рисунок 17 Результат виконання програми №6

Час затрачений на виконання завдання: **15 хвилин.**

Завдання **№7  Self Practice Work.**

Деталі виконання і тестуванню програми

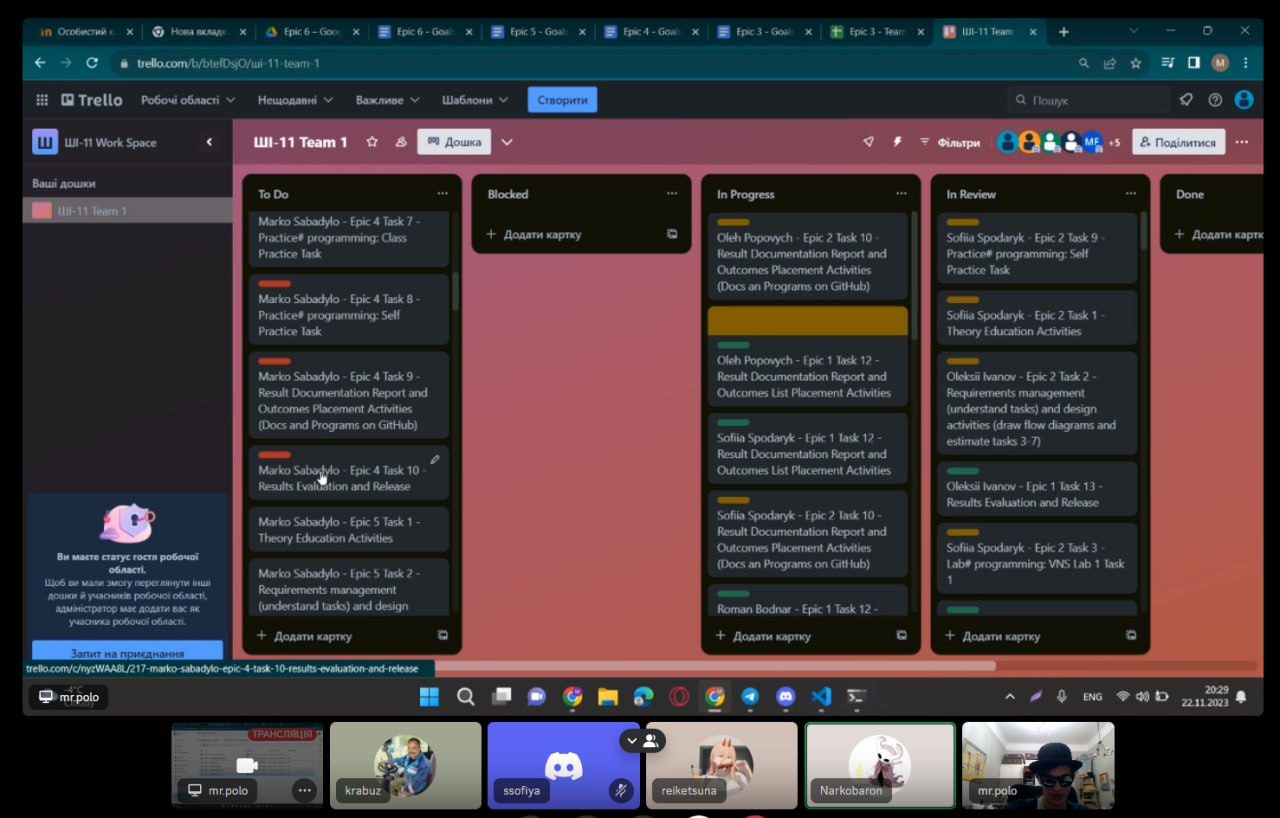


Рисунок 18 Результат виконання програми №7

Час затрачений на виконання завдання: **25 хвилин.**

## **6. Кооперація з командою:**

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



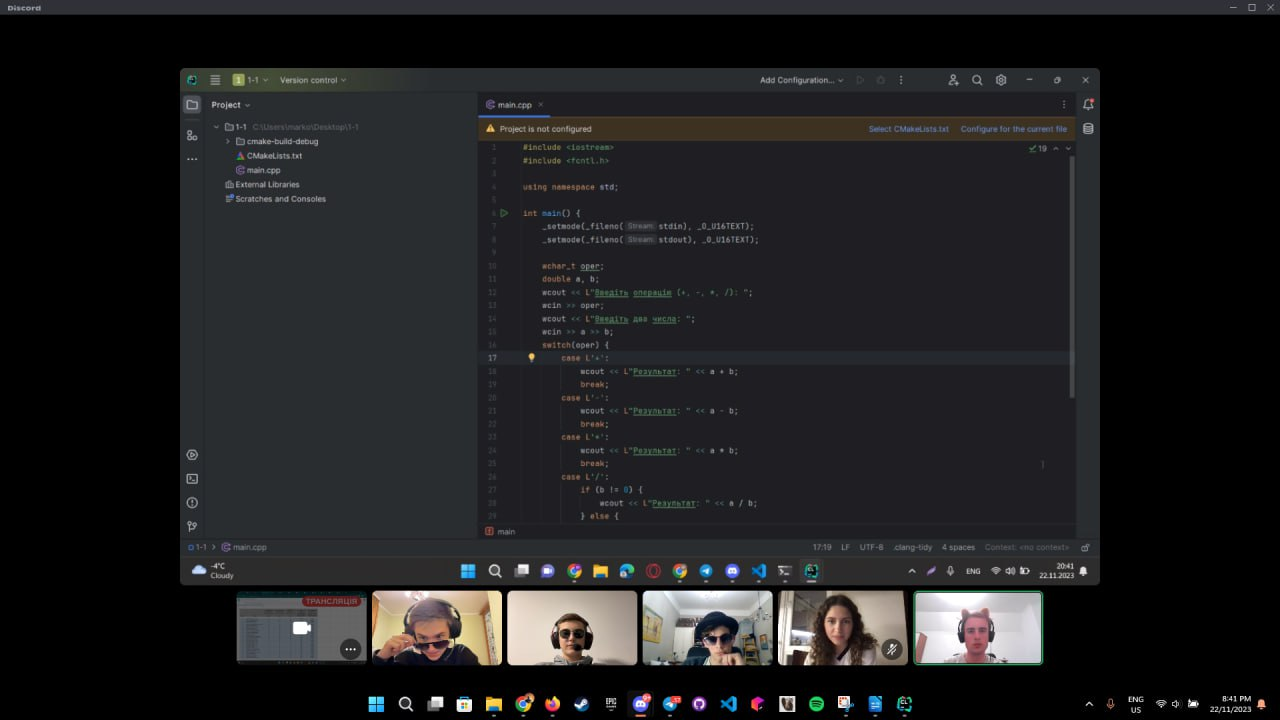


Рисунок 14 Cкріни зустрічі по обговоренню задач Епіку 4

# **Висновки:**

Ознайомився з роботою з файлами засобами мови C++. Системи числення. Бінарні Файли. Символи і рядкові змінні та текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

Виконав задані лабораторні роботи.

Виконав задане практичне завдання.