Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт (ВНС:лабораторна робота №6, лабораторна робота №8, лабораторна робота №9 Algotester: lab4 v3, lab6 v3)**

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

***з розділу***: «Файли. Системи числення. Бінарні файли. Символи та рядкові змінні. Текстові файли. Стандартна бібліотека. Створення та використання бібліотек»

***Виконала:***

студент групи ШІ-12

Лебединська Яна Олександрівна

# ***Тема роботи:***

Робота з файлами. Створення файлу, внесення в нього інформації, перезапис інформації у файлі. редагування цієї інформації. Створення функцій для полегшення роботи з файлами.

# ***Мета роботи:***

Навчитись вправно користуватись файлами в програмному середовищі. Набути навиків створення файлів, внесення інформації, редагування вмісту файлу, перезапису інформації. Навчитись працювати з файлами через власноруч створені функції.

# ***Теоретичні відомості:***

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Файли
* Тема №2: Текстові файли
* Тема №3: Бінарні файли
* Тема №4: Стек (базове поняття про стек)

1. Індивідуальний план опрацювання теорії:

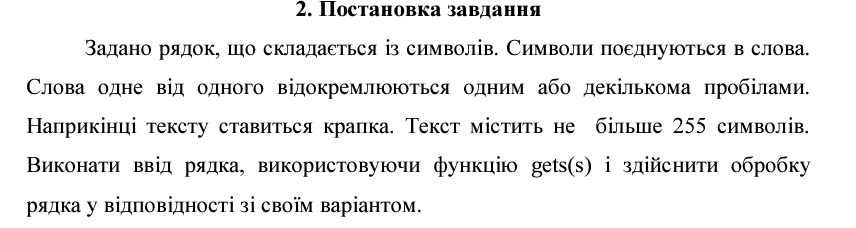
* Тема №1:Заголовкові файли
  + Джерела Інформації
    - <https://acode.com.ua/urok-24-zagolovkovi-fajly/> (**стаття**)
  + Що опрацьовано:
    - Заголовкові файли зі стандартної бібліотеки, що це таке
    - Яка різниця між “<>” і “”””
    - Як написати свій заголовковий файл
  + Статус: Ознайомлена
  + Початок опрацювання теми: 30.11.23
  + Звершення опрацювання теми: 30.11.23
* Тема №2: Текстові файли
  + Джерела Інформації:
    - <https://purecodecpp.com/uk/archives/2751> (**стаття**)
    - <https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/> (**стаття**)
  + Що опрацьовано:
    - методи запису та читання рядків файлу
    - класи fstream, ifstream, ofstream
    - як відкрити і закрити файл
  + Статус: Ознайомлена
  + Початок опрацювання теми: 01.12.23
  + Звершення опрацювання теми: 03.12.23
  + По темі зроблено конспект
* Тема №3: Бінарні файли
  + Джерела Інформації:
    - <https://studfile.net/preview/5994719/page:7/> (**стаття**)
  + Що опрацьовано:
    - що таке бінарний (двійковий) файл
    - особливості роботи з таким типом файлів
  + Статус: Ознайомлена
  + Початок опрацювання теми: 04.12.23
  + Звершення опрацювання теми: 05.12.23
* Тема №4: Стек
  + Джерела Інформації:
    - <https://acode.com.ua/urok-111-stek-i-kupa/> (**стаття**)
  + Що опрацьовано:
    - що таке стек
    - особливості стеку
  + Статус: Ознайомлена
  + Початок опрацювання теми: 29.11.23
  + Звершення опрацювання теми: 30.11.23
  + По темі зроблено конспект.
* Тема №5: Рядкові змінні
  + Джерела Інформації:
    - <https://acode.com.ua/urok-60-vvedennya-v-std-string/> (**стаття**)
    - <https://acode.com.ua/urok-88-symvolni-konstanty-ryadkiv-c-style/> (**стаття**)
  + Що опрацьовано:
    - тип даних string
    - ввід і вивід рядків
    - getline і його використання
    - символьні константи С-типу
  + Статус: Ознайомлена
  + Початок опрацювання теми: 25.11.23
  + Звершення опрацювання теми: 26.11.23

# ***Виконання роботи:***

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

*Завдання №1*: VNS Lab 6

* Варіант №22
* постановка завдання:

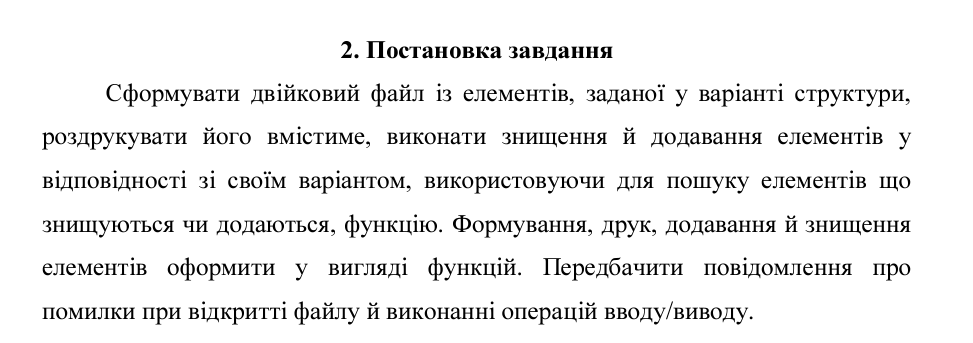


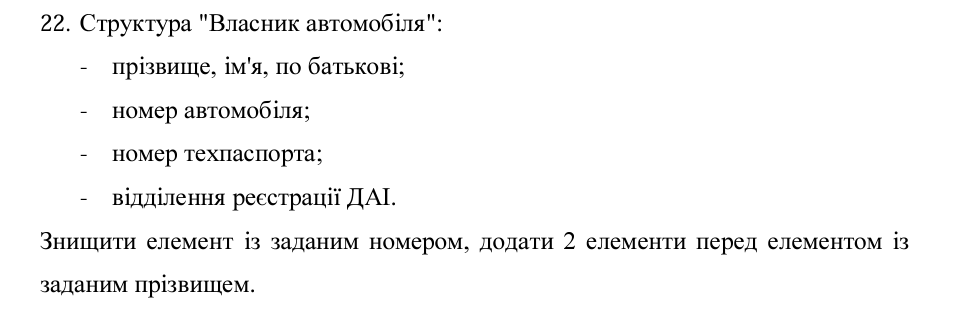


*умова завдання lab 6 v22*

*Завдання №2*: VNS Lab 8

* Варіант №22
* постановка завдання:

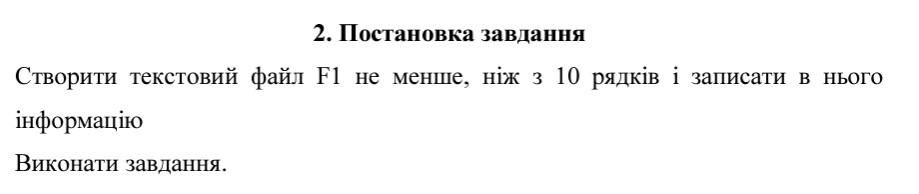


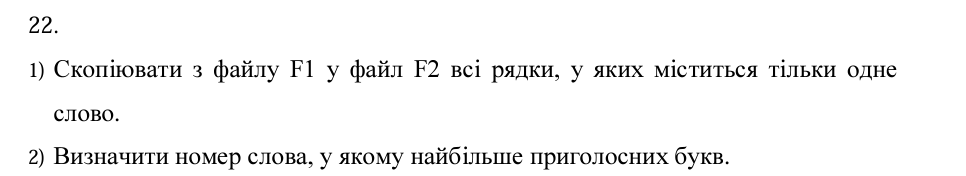


*умова завдання lab 8 v22*

*Завдання №3*: VNS Lab 9

* Варіант №22
* постановка завдання:

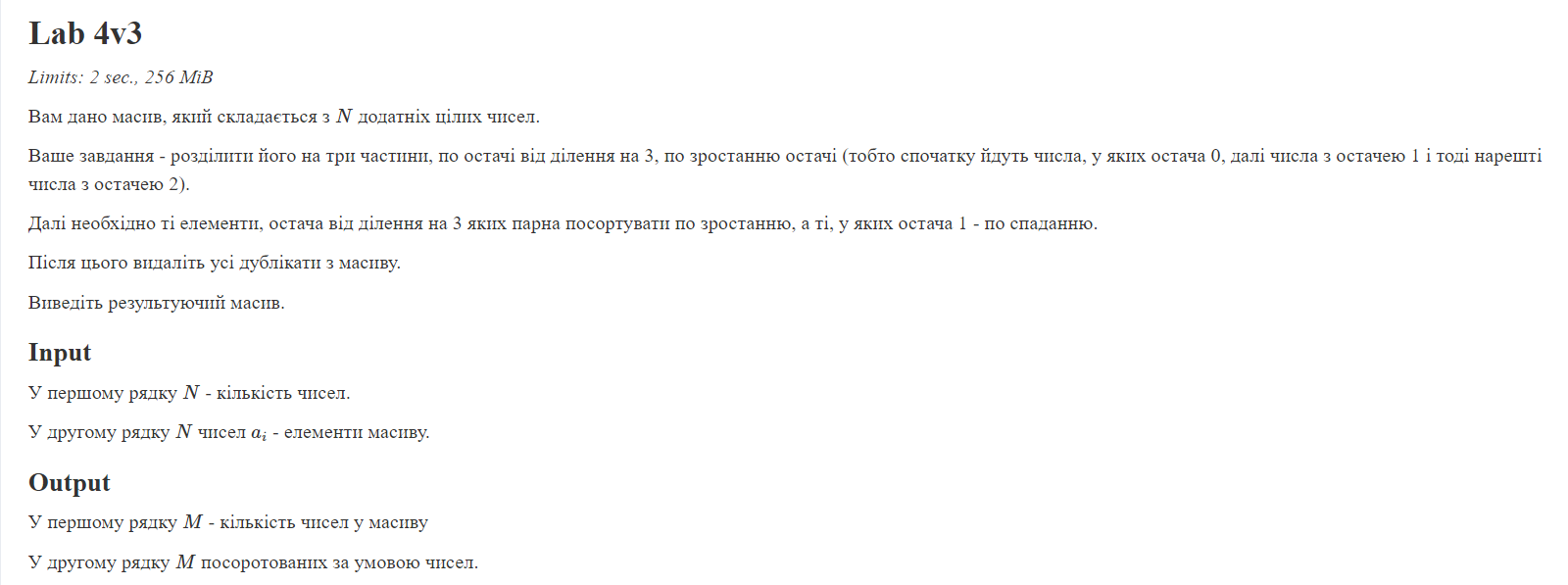




*умова завдання lab 9 v22*

*Завдання №4*: Algotester lab 4

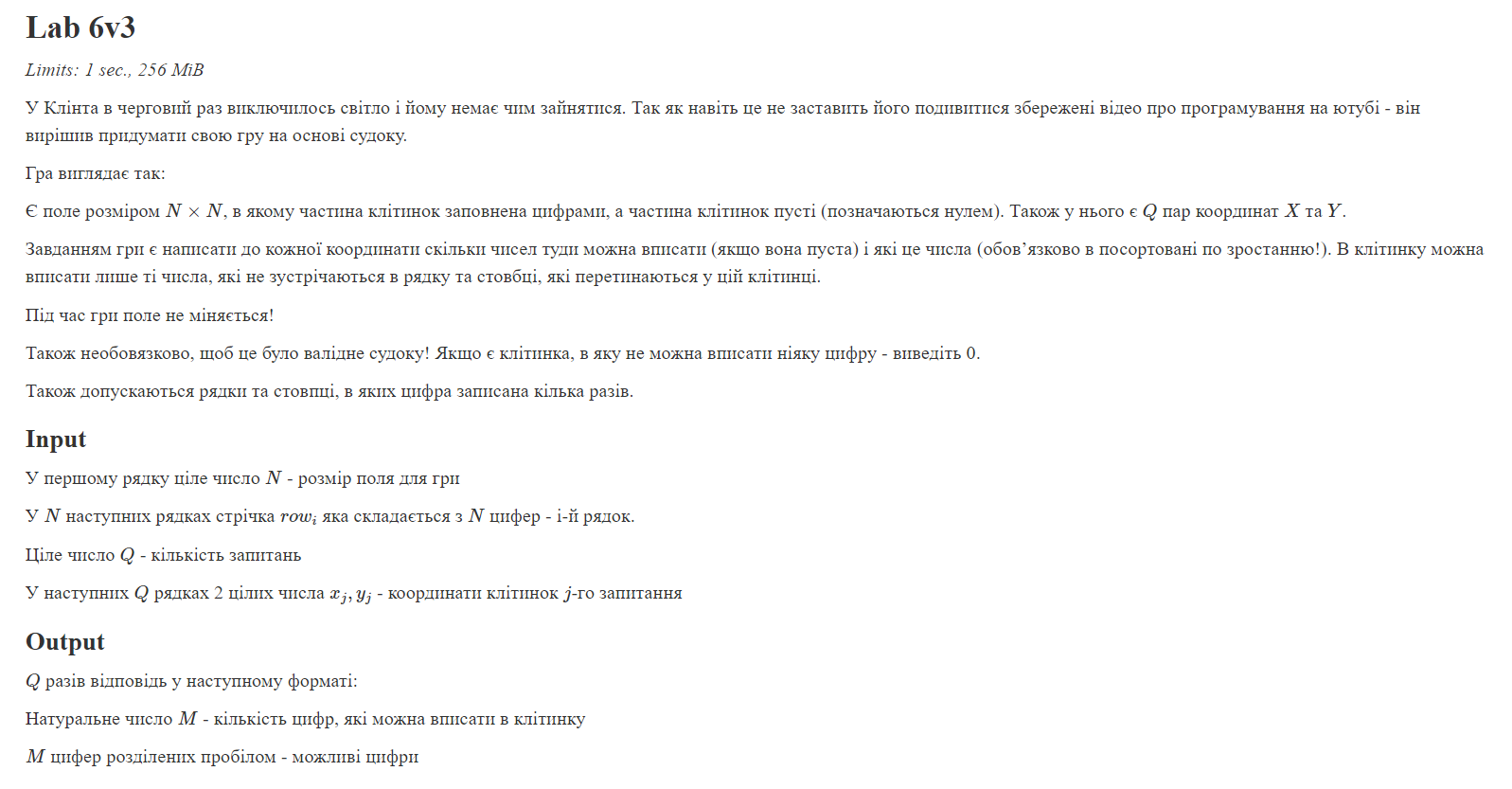
* Варіант №3
* постановка завдання: <https://algotester.com/en/ContestProblem/DisplayWithEditor/134642>



*умова задачі lab 4 v3 Algotester*

*Завдання №5*: Algotester lab 6

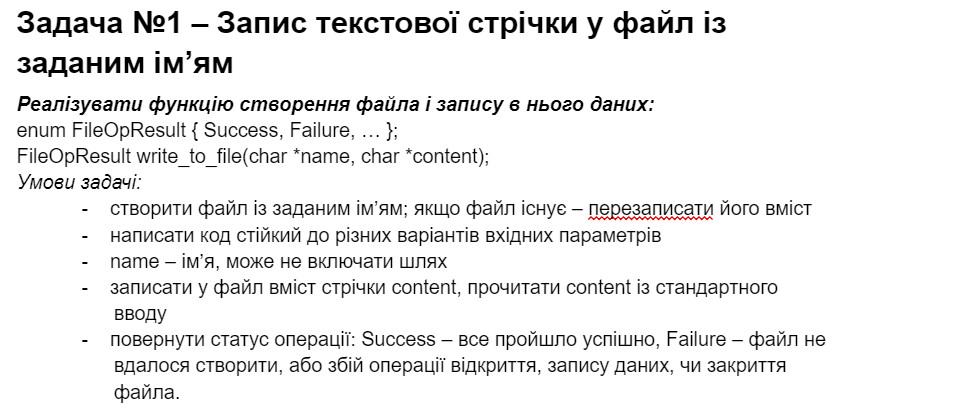
* Варіант №3
* постановка завдання: <https://algotester.com/en/ContestProblem/DisplayWithEditor/134646>

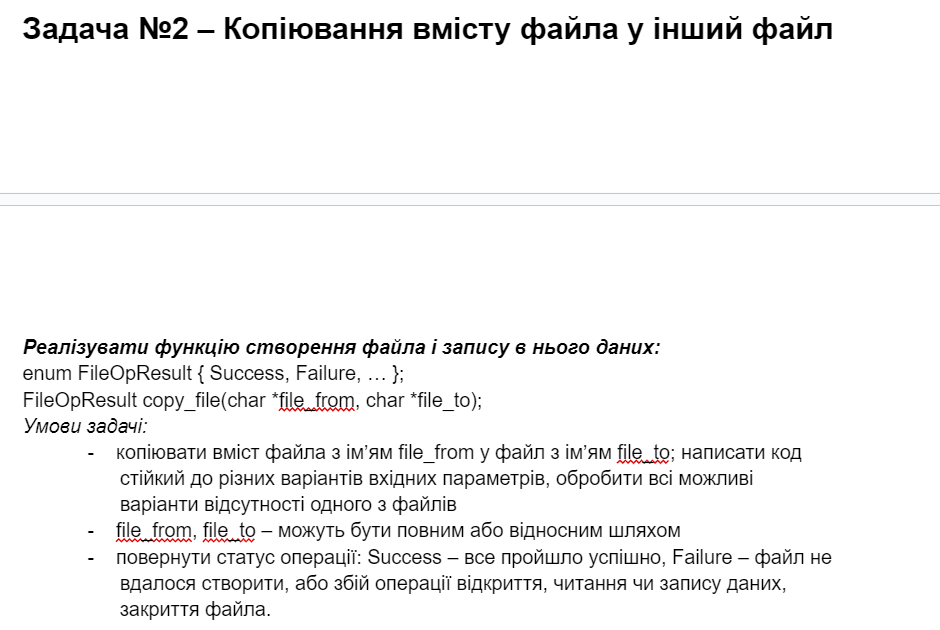


*умова задачі lab 6 v3 Algotester*

*Завдання №6*: Class Practice Task

* Без варіанту
* постановка завдання:

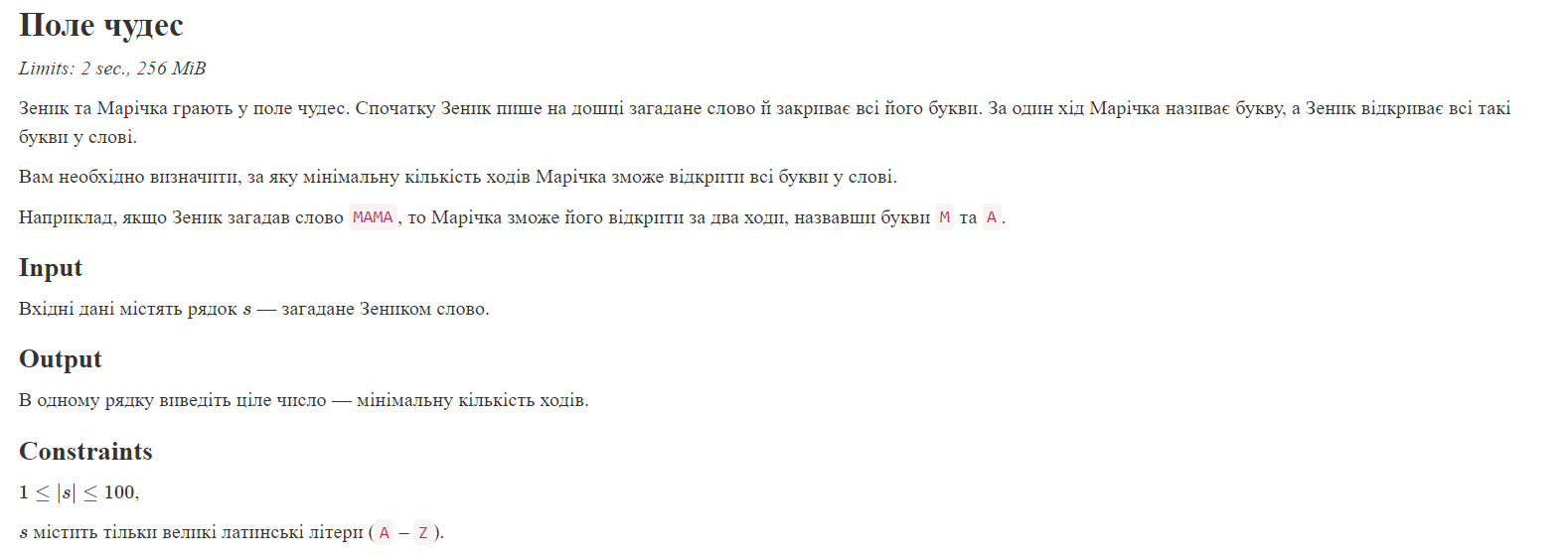




*умова задачі class practice*

*Завдання №*7 Self Practice Task (algotester)

* Без варіанту
* Постановка завдання: <https://algotester.com/en/ArchiveProblem/DisplayWithEditor/40120>



*умова задачі self practice Algotester*

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

***Програма №1*** VNS Lab 6

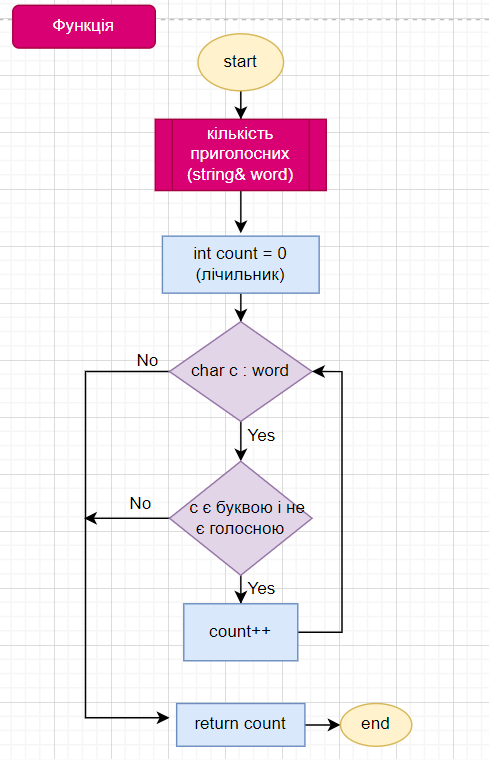
* Планований час на реалізацію: 1 год

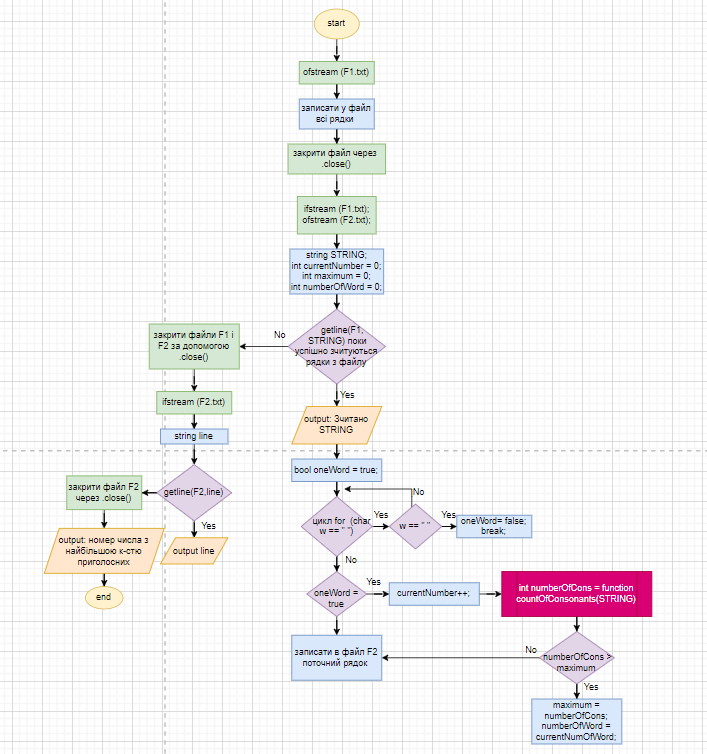
***Програма №2*** VNS Lab 8

* Планований час на реалізацію 1 год 20 хв

***Програма №3*** VNS Lab 9

* Блок-схема





* Планований час на реалізацію 1 год 20 хв

***Програма №4*** Algotester lab 4

* Планований час на реалізацію: 1 год

***Програма №5*** Algotester lab 6

* Планований час на реалізацію: 1 год 30 хв

***Програма №6*** Class Practice Task

* Планований час на реалізацію: 2 год 30 хв

***Програма №7*** Self Practice Task (algotester)

Планований час на реалізацію: 30 хв

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

Додаткової конфігурації не потрібно.

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

**Завдання №1** VNS Lab 6

Посилання на файл програми у пул-запиті GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/03d53b1e206a61cb78e147e079cd4de29df8d4f3/ai_12/yana_lebedynska/Epic5/vns_lab_6_v22_yana_lebedynska.cpp>

#include <iostream>

#include <string>

using namespace std;

int numberOfWordsBeginWithA(const string& words) {

int number = 0;

string word;

for (char letter : words)

{

if (isspace(letter))

{

if (tolower(word[0]) == 'a') {

number++;

}

word.clear(); //очищую слово, бо переходжу на наступне, є пробіл

}

else

{

word += letter;

}

}

if (tolower(word[0]) == 'a')

{

number++;

}

return number;

}

int main()

{

string WORD;

cout << "Enter a string: ";

getline(cin, WORD); //щоб також оброблялись пробіли

int NUMBER = numberOfWordsBeginWithA(WORD);

cout << "Number of words that begin with letter 'a' is: " << NUMBER << endl;

return 0;

}

*код задачі lab 6*

**Завдання №2** VNS Lab 8

Посилання на файл програми у пул-запиті GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/03d53b1e206a61cb78e147e079cd4de29df8d4f3/ai_12/yana_lebedynska/Epic5/2vns_lab_8_v22_yana_lebedynska.cpp>

#include <iostream>

#include <vector>

#include <string>

#include <fstream>

using namespace std;

struct OwnerOfCar

{

string lastName;

string firstName;

string middleName;

string numOfCar;

long numOfTechPassport;

int numOfRegistraionOffice;

};

void printInformation(vector<OwnerOfCar>& owner)

{

cout << endl;

for(auto& OWNER : owner)

{

cout << "The last name: " << OWNER.lastName << endl;

cout << "The first name: " << OWNER.firstName << endl;

cout << "The middle name: " << OWNER.middleName << endl;

cout << "The car number: " << OWNER.numOfCar << endl;

cout << "The technical passport number: " << OWNER.numOfTechPassport << endl;

cout << "The num of registration office: " << OWNER.numOfRegistraionOffice << endl;

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

}

}

void deleteFromOwner(vector<OwnerOfCar>& owner, string& numberOfCarToDelete)

{

cout << endl;

for(auto i = owner.begin(); i < owner.end(); i++)

{

if((\*i).numOfCar == numberOfCarToDelete)

{

i = owner.erase(i); //тепер i вказує на наступний елемент після видаленого

break;

}

}

}

void insertOwners(vector<OwnerOfCar>& owner, string& LASTname, OwnerOfCar& owner1, OwnerOfCar& owner2)

{

cout << endl;

for(auto i = owner.begin(); i < owner.end(); i++)

{

if((\*i).lastName == LASTname)

{

i = owner.insert(i, owner1);

i = owner.insert(i, owner2);

break;

}

}

}

int main()

{

vector<OwnerOfCar> carOwners;

ifstream file("owners.dat", ios::binary);

if(file.is\_open())

{

OwnerOfCar tempOwner;

while(file.read(reinterpret\_cast<char\*>(&tempOwner), sizeof(OwnerOfCar)))

{

carOwners.push\_back(tempOwner);

}

file.close();

}

else

{

cout << "Error! File couldn't be opened.";

}

OwnerOfCar owner1 = {"Lebedynska", "Yana", "Olexandrivna", "BC4326", 232535577, 2};

OwnerOfCar owner2 = {"Brychko", "Maria", "Ihorivna", "BC7455", 753855112, 23};

carOwners.push\_back(owner1);

carOwners.push\_back(owner2);

ofstream file\_output("OWNERS.dat", ios::binary);

if(file\_output.is\_open())

{

for(auto& Owners : carOwners)

{

file\_output.write(reinterpret\_cast<const char\*>(&Owners), sizeof(OwnerOfCar));

}

file\_output.close();

}

cout << "The list before editing" << endl;

printInformation(carOwners);

string CarNumToDelete;

cout << "Enter car number to delete: ";

cin >> CarNumToDelete;

deleteFromOwner(carOwners, CarNumToDelete);

cout << "The list after deleting" << endl;

printInformation(carOwners);

OwnerOfCar newOwner1 = {"Borsch", "Anastasia", "Ihorivna", "BC1199", 243647999, 5};

OwnerOfCar newOwner2 = {"Yankiv", "Olesia", "Semenivna", "BC4312", 655578000, 21};

string lastNameToInsertBefore;

cout << "Enter the last name from list to insert before two new owners: ";

cin.ignore();

getline(cin, lastNameToInsertBefore);

insertOwners(carOwners, lastNameToInsertBefore, newOwner1, newOwner2);

cout << "The list after adding new owners: ";

printInformation(carOwners);

return 0;

}

## *код задачі lab 8*

**Завдання №3** VNS Lab 9

Посилання на файл програми у пул-запиті GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/03d53b1e206a61cb78e147e079cd4de29df8d4f3/ai_12/yana_lebedynska/Epic5/vns_lab_9_v22_yana_lebedynska.cpp>

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

int countOfConsonants(const string& word)

{

int count = 0;

for(char c : word)

{

if(isalpha(c) && !strchr("aeiouAEIOU", c))

{

count++;

}

}

return count;

}

int main()

{

ofstream F1\_output("F1.txt");

F1\_output << "My name is Yana" << endl;

F1\_output << "Student" << endl;

F1\_output << "Programming is my passion" << endl;

F1\_output << "Christmas is coming" << endl;

F1\_output << "AI" << endl;

F1\_output << "Let is snow" << endl;

F1\_output << "Some information" << endl;

F1\_output << "The great Gatsby" << endl;

F1\_output << "Winter" << endl;

F1\_output << "Night" << endl;

F1\_output.close();

ifstream F1\_input("F1.txt");

ofstream F2\_output("F2.txt");

string STRING;

int currentNumberOfWord = 0;

int maximum = 0;

int numberOfWord = 0;

while (getline(F1\_input, STRING)) // поки зчитка з файлу успішна

{

cout << "Read: " << STRING << endl;

bool oneWord = true;

for (char w : STRING)

{

if (w == ' ')

{

oneWord = false;

break;

}

}

if(oneWord)

{

currentNumberOfWord++;

int numberOfConsonants = countOfConsonants(STRING);

if(numberOfConsonants > maximum)

{

maximum = numberOfConsonants;

numberOfWord = currentNumberOfWord;

}

F2\_output << STRING << endl;

}

}

F1\_input.close();

F2\_output.close();

cout << endl;

ifstream F2\_input("F2.txt");

string line;

while (getline(F2\_input, line))

{

cout << line << endl;

}

F2\_input.close();

cout << endl;

cout << "The number of word with the biggest count of consonants: " << numberOfWord << endl;

return 0;

}

## *код задачі lab 9*

**Завдання №4** Algotester lab 4 v3

Посилання на файл програми у пул-запиті GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/03d53b1e206a61cb78e147e079cd4de29df8d4f3/ai_12/yana_lebedynska/Epic5/algotester_lab_4_yana_lebedynska.cpp>

#include <iostream>

#include <vector>

#include <algorithm>

#include <unordered\_set>

using namespace std;

int main()

{

unsigned int N;

cin >> N;

vector<unsigned int> array(N);

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

{

cin >> array[i];

}

vector<unsigned int> remainder0, remainder1, remainder2;

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

{

int rem = array[i] % 3;

if(rem == 0)

{

remainder0.push\_back(array[i]);

}

else if(rem == 1)

{

remainder1.push\_back(array[i]);

}

else

{

remainder2.push\_back(array[i]);

}

}

sort(remainder1.begin(), remainder1.end(), greater<int>());

sort(remainder0.begin(), remainder0.end());

sort(remainder2.begin(), remainder2.end());

vector<unsigned int> result;

result.insert(result.end(), remainder0.begin(), remainder0.end());

result.insert(result.end(), remainder1.begin(), remainder1.end());

result.insert(result.end(), remainder2.begin(), remainder2.end());

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

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

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

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

{

cout << result[i] << " ";

}

return 0;

}

## *код задачі algotester lab4 v3*

**Завдання №5** *Algotester lab6 v3*

Посилання на файл програми у пул-запиті GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/03d53b1e206a61cb78e147e079cd4de29df8d4f3/ai_12/yana_lebedynska/Epic5/algotester_lab_6_yana_lebedynska.cpp>

#include <iostream>

#include <algorithm>

#include <vector>

#include <unordered\_set>

using namespace std;

void CellCheck(const vector<string> &NUMS, int X, int Y)

{

X--;

Y--;

if(NUMS[X][Y] == '0')

{

unordered\_set<char> possibleNums;

for(int i = 1; i <= NUMS.size(); i++)

{

possibleNums.insert('0' + i);

}

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

{

possibleNums.erase(NUMS[X][i]);

possibleNums.erase(NUMS[i][Y]);

}

vector<char> totalResult(possibleNums.begin(), possibleNums.end());

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

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

for(char num : totalResult)

{

cout << num << " ";

}

}

else

{

cout << "1" << endl << NUMS[X][Y];

}

cout << endl;

cout << endl;

}

int main()

{

int N, M;

cin >> N;

vector<string> numbers(N);

for(string &ROW : numbers)

{

cin >> ROW;

}

cin >> M;

int arr[M][2];

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

{

cin >> arr[i][0] >> arr[i][1];

}

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

{

CellCheck(numbers, arr[i][0], arr[i][1]);

}

return 0;

}

## *код задачі algotester lab 6 v3*

**Завдання №6** Class Practice Task

Посилання на файл програми у пул-запиті GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/03d53b1e206a61cb78e147e079cd4de29df8d4f3/ai_12/yana_lebedynska/Epic5/practice_class_task_yana_lebedynska.cpp>

#include <iostream>

#include <fstream>

using namespace std;

enum FileOpResult {Success, Failure};

FileOpResult write\_to\_file(const char \*fileName, const char \*fileContent)

{

cout << endl;

ofstream file\_output(fileName);

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

{

cout << "Error! (File couldn't be open)";

return Failure;

}

file\_output << fileContent;

if(file\_output.fail())

{

cout << "Error! (Content couldn't be written to file)";

file\_output.close();

return Failure;

}

file\_output.close();

return Success;

}

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

{

cout << endl;

ifstream fileToCopy(file\_from);

if(!fileToCopy.is\_open())

{

cout << "Error! (File couldn't be open)";

return Failure;

}

ofstream fileInWhichCopy(file\_to);

if(!fileInWhichCopy.is\_open())

{

cout << "Error! (File couldn't be opened)";

fileInWhichCopy.close();

return Failure;

}

fileInWhichCopy << fileToCopy.rdbuf();

if(fileToCopy.fail() || fileInWhichCopy.fail())

{

cout << "Error! (Coping isn't done)";

fileToCopy.close();

fileInWhichCopy.close();

return Failure;

}

fileToCopy.close();

fileInWhichCopy.close();

return Success;

}

int main()

{

string fileNameToCopy, fileNameInWhichCopy, FILEcontent;

cout << "Enter the name of file you want to copy here -> ";

cin >> fileNameToCopy;

cout << "Enter information you want to write to the file here -> ";

cin.ignore();

getline(cin, FILEcontent);

FileOpResult resultOf1Operation = write\_to\_file(fileNameToCopy.c\_str(), FILEcontent.c\_str());

if(resultOf1Operation == Success)

{

cout << "Success!";

}

else

{

cout << endl;

cout << "Failure!";

return 1;

}

cout << endl;

cout << "Enter the name of file you want to copy information in here -> ";

cin >> fileNameInWhichCopy;

FileOpResult resultOf2Operation = copy\_file(fileNameToCopy.c\_str(), fileNameInWhichCopy.c\_str());

if(resultOf2Operation == Success)

{

cout << "Success!";

}

else

{

cout << endl;

cout << "Failure!";

return 1;

}

return 0;

}

## *код задачі class practice*

**Завдання №7** Self Practice Task (Algotester)

Посилання на файл програми у пул-запиті GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/blob/03d53b1e206a61cb78e147e079cd4de29df8d4f3/ai_12/yana_lebedynska/Epic5/practice_self_task_yana_lebedynska.cpp>

#include <iostream>

#include <set>

#include <string>

using namespace std;

int main() {

string word;

cin >> word;

set<char> LETTERS;

for (char letter : word)

{

LETTERS.insert(letter);

}

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

return 0;

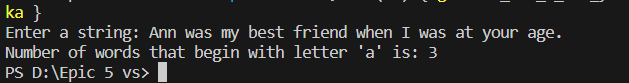
}

## *код задачі self practice (algotester)*

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

***Завдання №1*** VNS Lab 6

**Результат програми:**

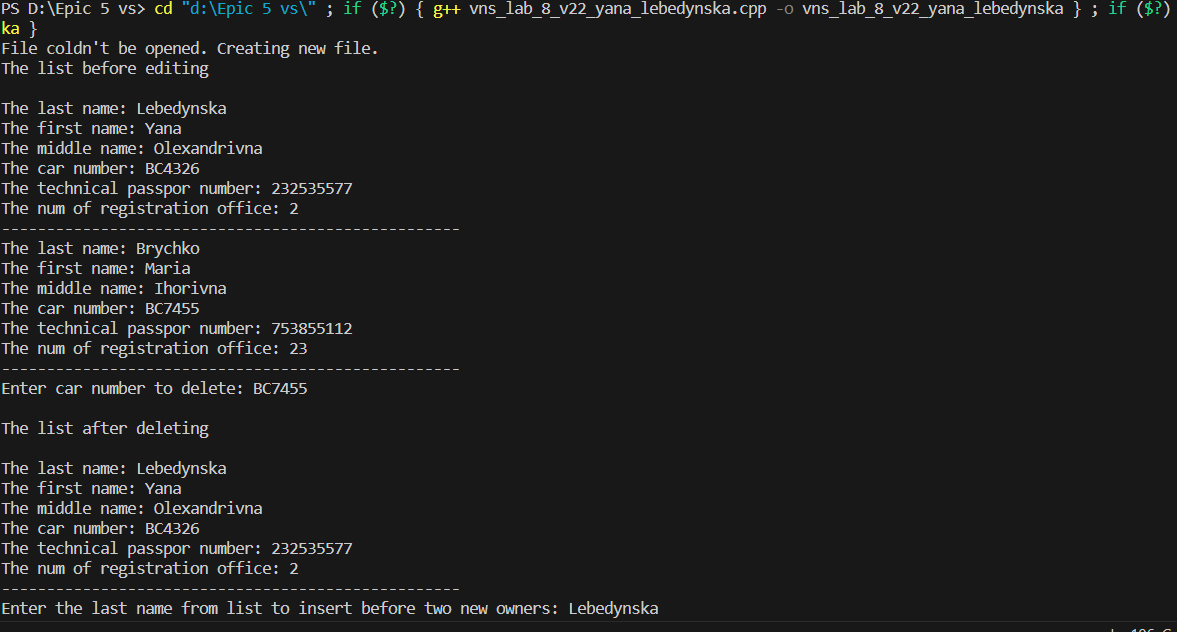
****

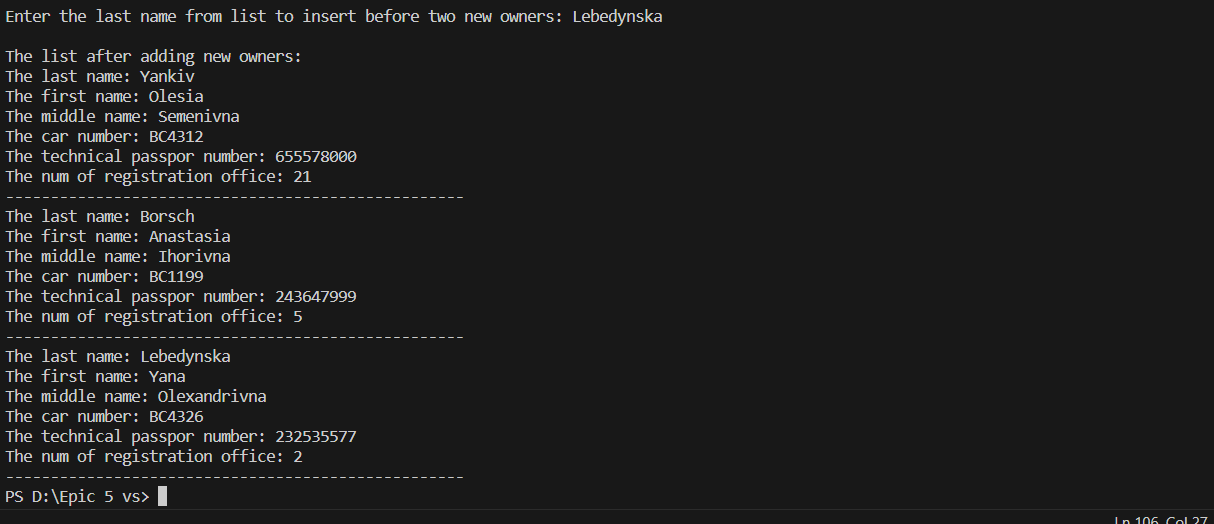
*робота програми*

Час затрачений на виконання: 1 год 20 хв

***Завдання №2*** VNS Lab 8

**Робота програми:**

****

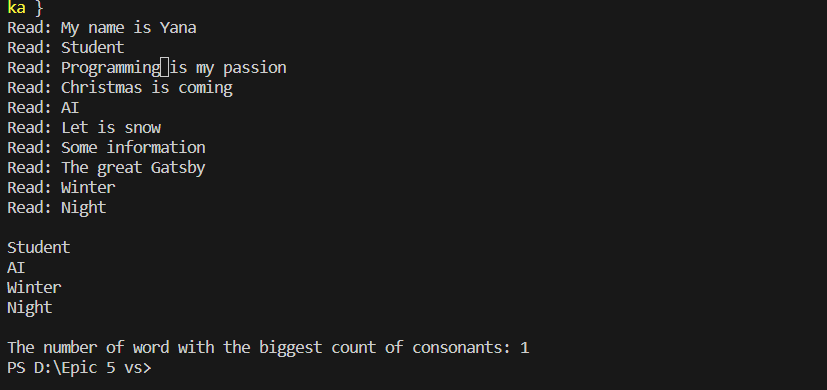
****

*результат програми*

Час затрачений на виконання завдання: 1 год 30 хв

***Завдання №3***VNS Lab 9

**Результат програма:**

****

Час затрачений на виконання завдання: 1 год 45 хв

***Завдання №4*** Algotester lab4 v3

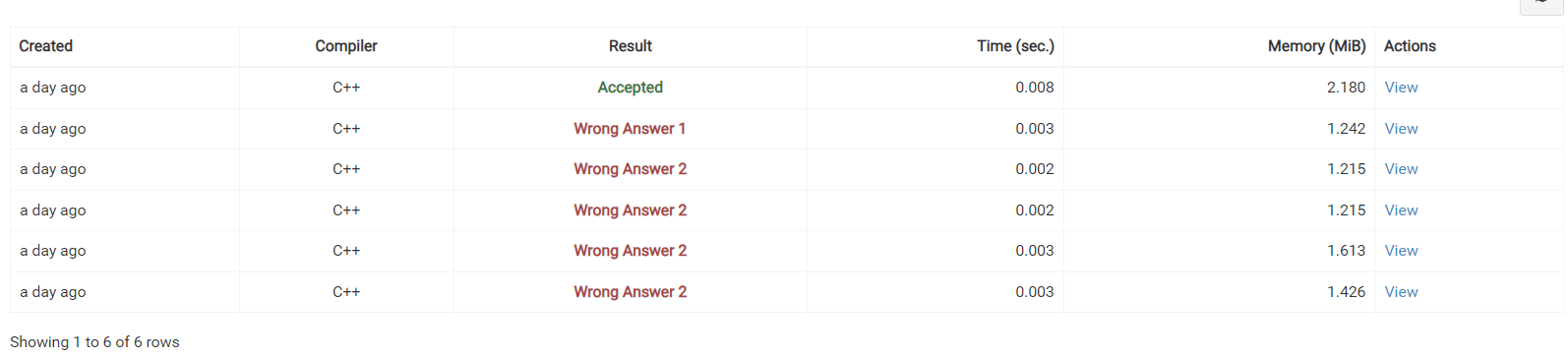
**Input:**

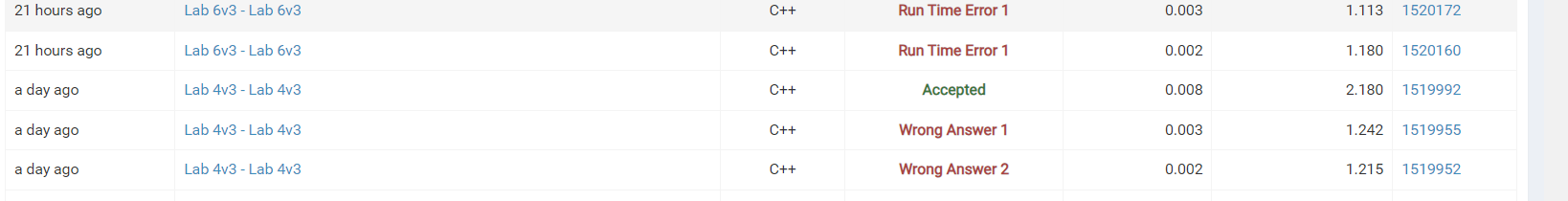


**Output:**



**Evidences:**

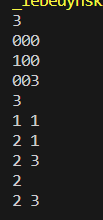
****

****

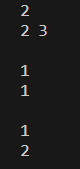
Час затрачений на виконання: 1 год 30 хв

***Завдання №4*** Algotester lab 6 v3

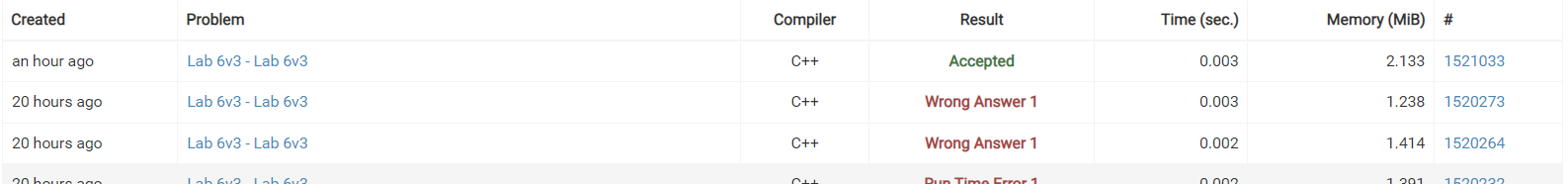
**Input:**

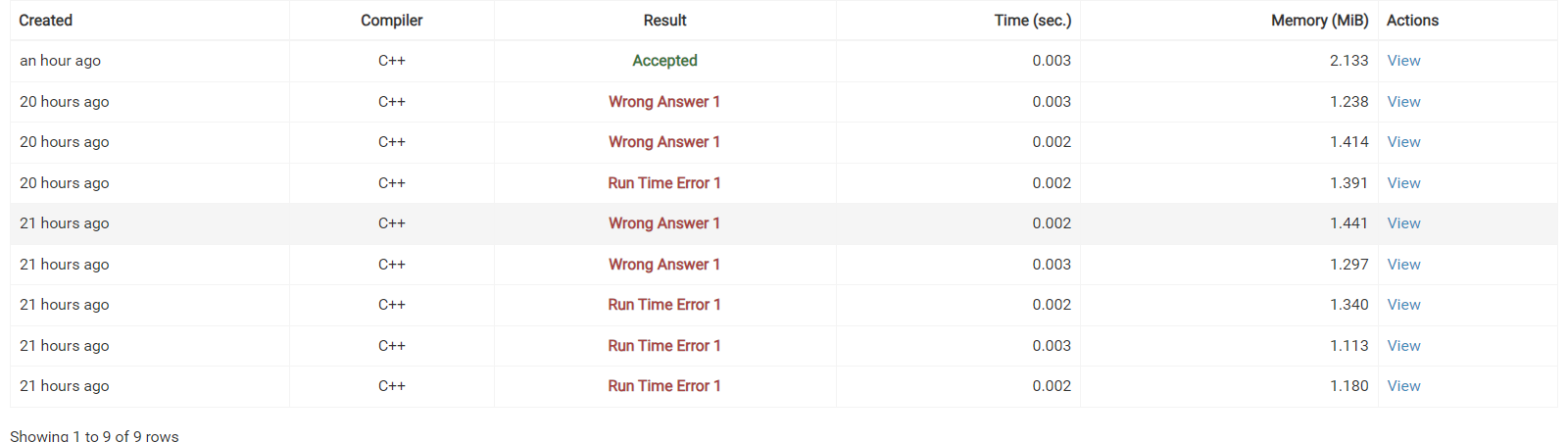
****

**Output:**

****

**Evidences:**

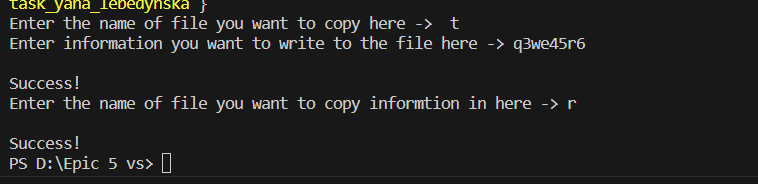
****

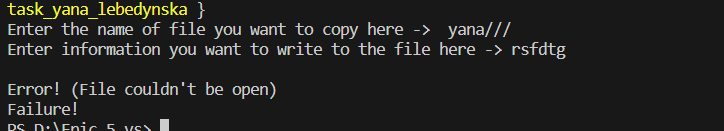
****

Час затрачений на виконання: 2 год

***Завдання №6*** Class Practice Task

**Результат програми:**

****

****

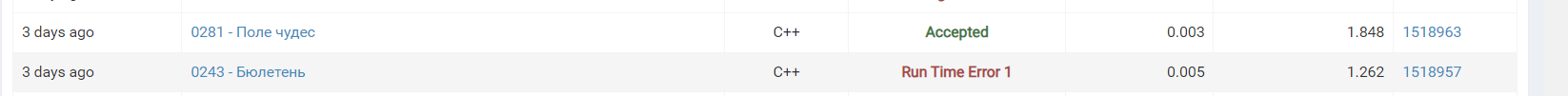
Час затрачений на виконання: 1 год 40 хв

***Завдання №7*** Self Practice Task (Algotester)

**Результат програми:**

****

**Evidences:**

****

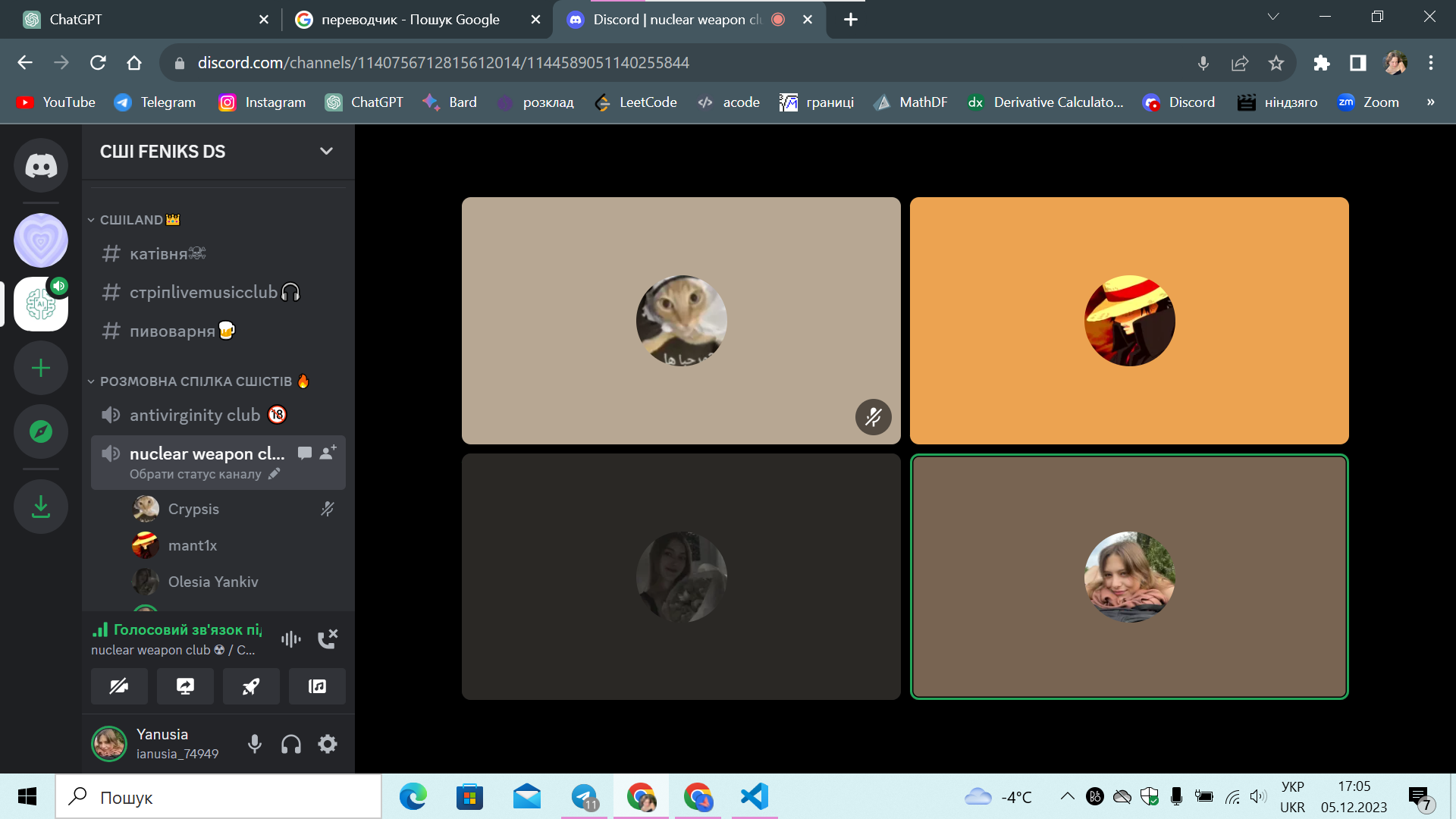
****

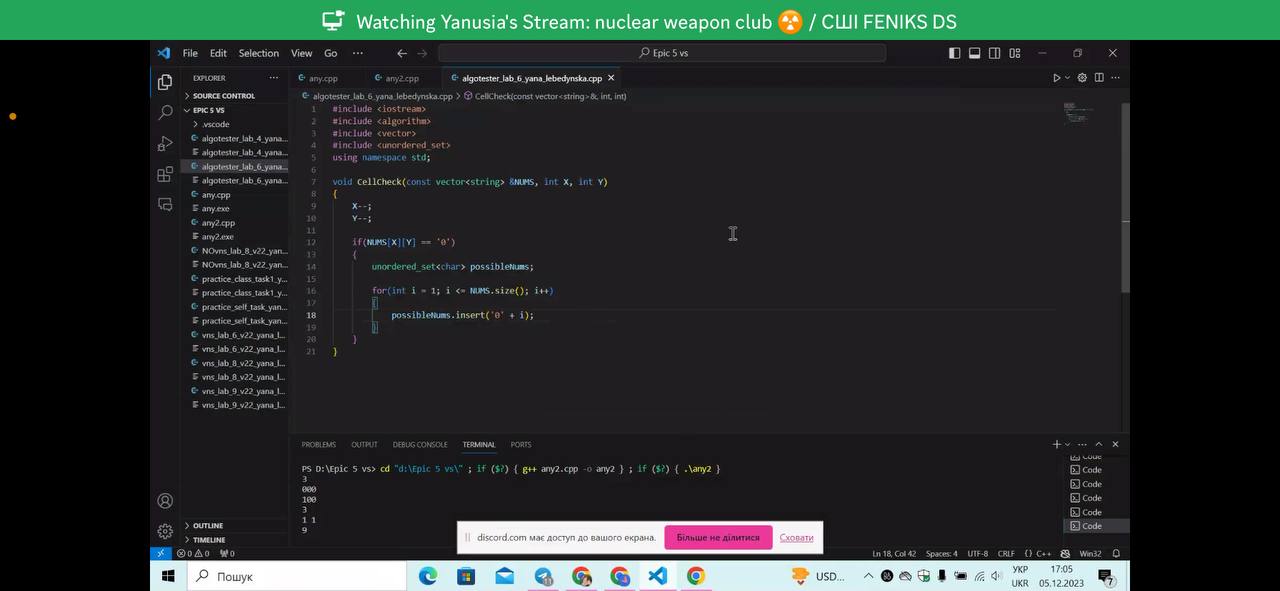
Час затрачений на виконання: 30 хв

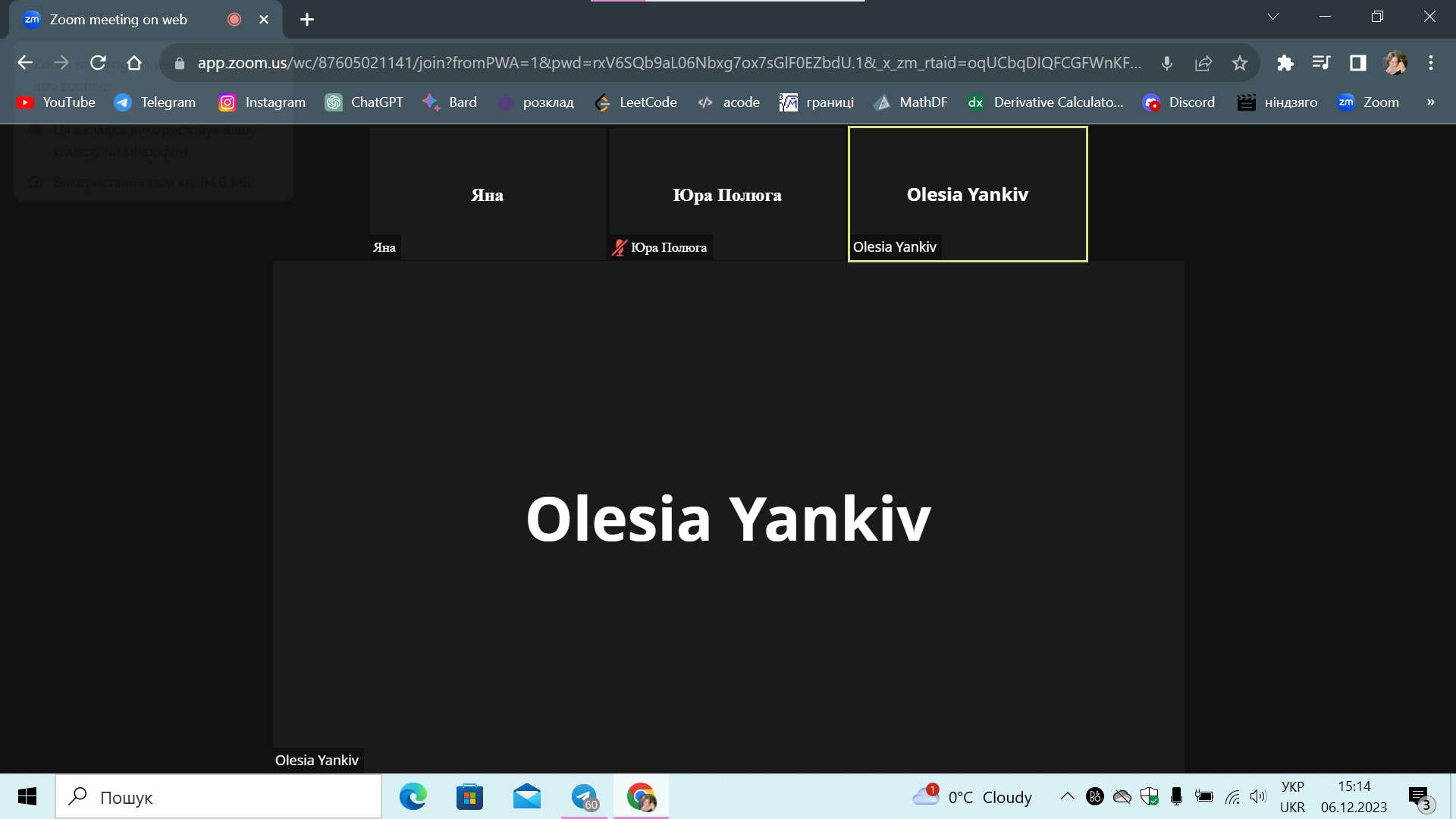
# 

# **Робота команди:**

**Міти**

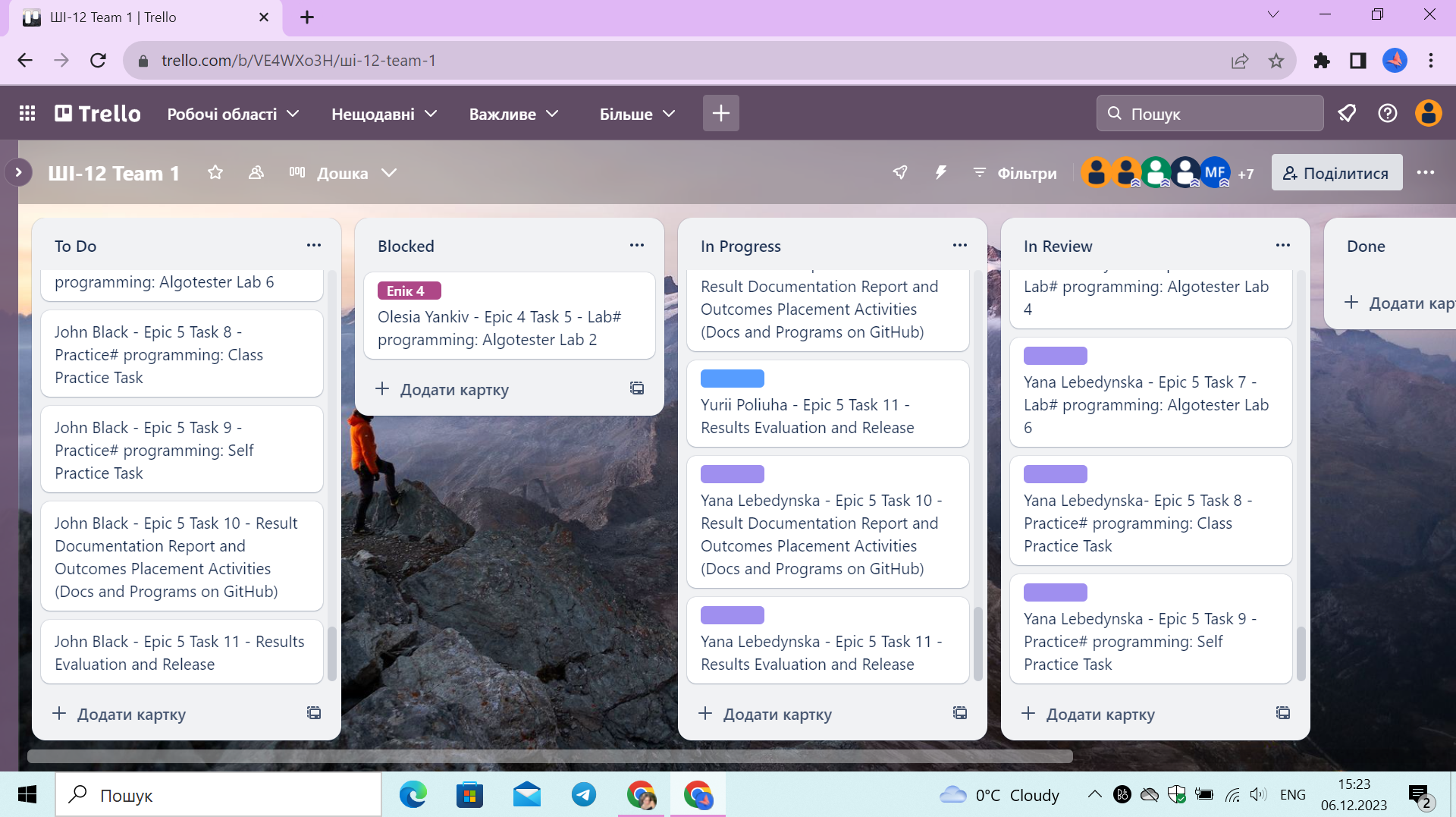
****

****

****

*скріни мітів у зумі*

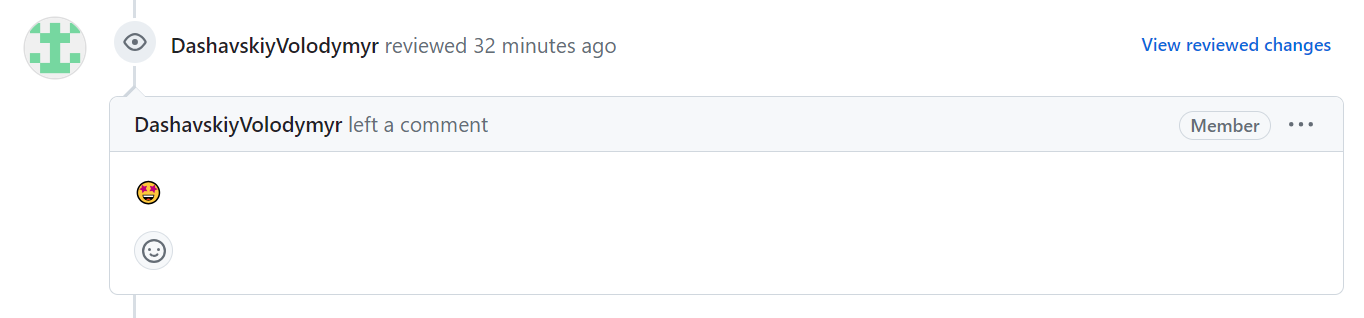
**Трелло**

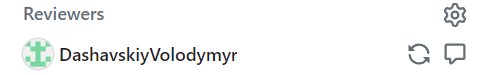
****

*скрін робочої області в трелло*

**Пул-реквест:**

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/704/commits**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/704/commits)

****

****

*коментар до пул-реквесту в гітХабі*

**Висновки:**

Протягом епіка 5 я змогла набути навиків роботи з файлами в різних аспектах. Також навчилась працювати з символьними і рядковими змінними. Поверхнево ознайомилась зі стеком і методом його роботи. Практично закріпила всі набуті нові знання.