Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 4**

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

***з розділу***: «Прості структури даних. Одновимірні масиви. Двовимірні масиви. Алгоритми обробки»

***Виконав:***

студент групи ШІ-13

Сіренко Юрій Сергійович

# **Тема роботи:**

# Розглянути та опрацювати структури даних та їх використання у вигляді одновимірних та двовимірних масивів. Робота з алгоритми обробки, що застосовуються до цих структур.

# **Мета роботи:**

# Детально розглянути та проаналізувати структури даних, зокрема прості структури, одновимірні та двовимірні масиви, а також алгоритми обробки, пов'язані з цими структурами. Вивчення та демонстрація основних понять, властивостей та застосувань цих елементів програмування, їхнє застосування на практиці.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Прості структури даних.
* Тема №2: Одновимірні масиви.
* Тема №3: Двовимірні масиви.
* Тема №4: Алгоритми обробки.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Прості структури даних.
  + Джерела Інформації
    - Відео 1 - [Introduction to Data Structures & Algorithms | Course Details & Prerequisites](https://www.youtube.com/watch?v=XCyuHSJS7XE&ab_channel=SimpleSnippets)
    - Стаття 1 - [C++'s Built-In Data Structures](https://www.codecademy.com/learn/c-plus-plus-for-programmers/modules/cpp-built-in-data-structures/cheatsheet).
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Основні концепцій та призначення простих структур даних, різноманітні типи простих структур, такі як числові, символьні та логічні, а також основні операції, що можна виконувати з цими структурами.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 30.11.2023
* Тема №2: Одновимірні масиви.
  + Джерела Інформації:
    - Відео 1 - [Arrays in C++](https://www.youtube.com/watch?v=ENDaJi08jCU&pp=ygUKYXJyYXlzIGMrKw%3D%3D)
    - Стаття 1 - [C++ Arrays](https://www.w3schools.com/cpp/cpp_arrays.asp).
    - Стаття 2 - [C++ Arrays and Loops](https://www.w3schools.com/cpp/cpp_arrays_loop.asp)
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Розуміння концепції одновимірних масивів, їхніх характеристик та призначення у програмуванні.
    - Індексація та доступ до окремих елементів в одновимірних масивах.
    - Операції та алгоритми, які можна застосовувати до одновимірних масивів, включаючи сортування, пошук елементів, зміну та вилучення даних.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 30.11.2023
* Тема №3: Двовимірні масиви.
  + Джерела Інформації:
    - Відео 1 - [Multidimensional Arrays in C++ (2D arrays)](https://www.youtube.com/watch?v=gNgUMA_Ur0U&pp=ygUYMiBkaW1lbnNpb25hbCBhcnJheXMgYysr).
    - Стаття 1 - [C++ Multi-Dimensional Arrays](https://www.w3schools.com/cpp/cpp_arrays_multi.asp).
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Розуміння концепцій та особливостей двовимірних масивів, їхньої структури та призначення у програмуванні.
    - Здобули навички роботи з двовимірними масивами, включаючи Індексацію та звернення до конкретних елементів.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 30.11.2023
* Тема №4 Алгоритми обробки.
  + Джерела Інформації:
    - Стаття 1 - [C++. STL. Algorithms processing the sequence as a whole](https://www.bestprog.net/en/2023/05/23/c-stl-algorithms-processing-the-sequence-as-a-whole/).
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Різноманітні загальні алгоритми, що застосовуються до обробки даних, такі як сортування та пошук.
    - Алгоритми сортування та їх застосування до масивів.
    - Пошук елементів та алгоритми пошуку.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 30.11.2023

# **Виконання роботи:**

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

Завдання №1 Опрацювання теорії

* Деталі завдання: опрацювання вище перечисленої теорії. Розбір статей, перегляд відео та імплементація теорії на практиці. Вивчення найосновніших тем.

Завдання №2 VNS Lab №4 Task 1

* Варіант завдання – 1
* Деталі завдання

1) Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел.

2) Роздрукувати отриманий масив.

3) Знищити елемент із номером К.

4) Додати після кожного парного елемента масиву елемент зі значенням 0.

5) Роздрукувати отриманий масив.

* Важливі деталі для врахування в імплементації програми
* Використання вектора для отримання бажаного результату

Завдання №3 VNS Lab №5 Task 1

* Варіант завдання – 1
* У двовимірному масиві записані слова, що представляють собою послідовність цифр, що завершуються 0. Необхідно роздрукувати слова через кому, взявши надрукований рядок у дужки. Довжина друкованого рядка 60 символів. Добування слова оформити у вигляді функції. Наприклад: вихідні дані - 123023402303450 234450234567010 234455677670450 результат - (123,234,23,345)(23445,234567,1)(23445567767,45)
* Деталі завдання

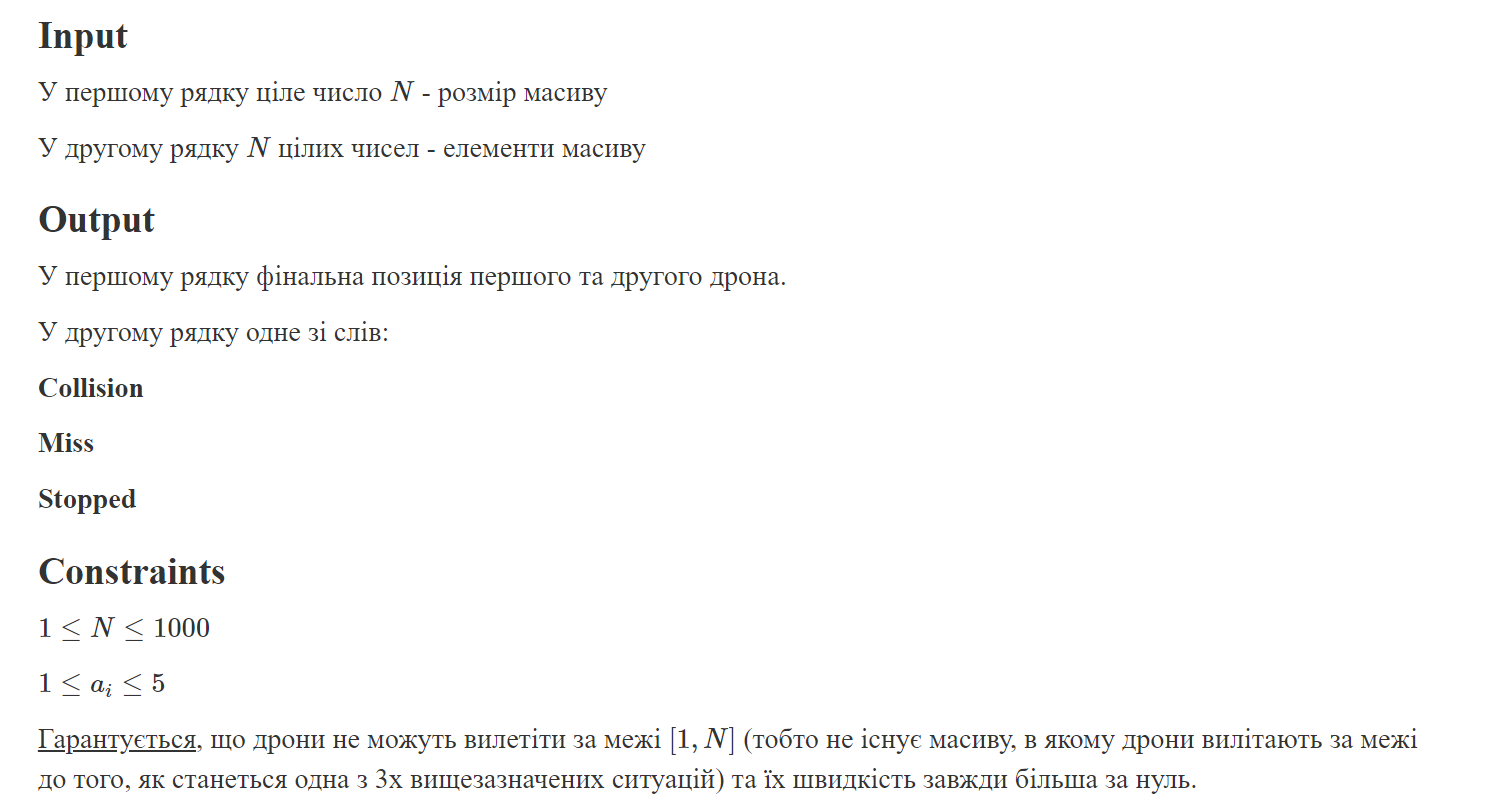
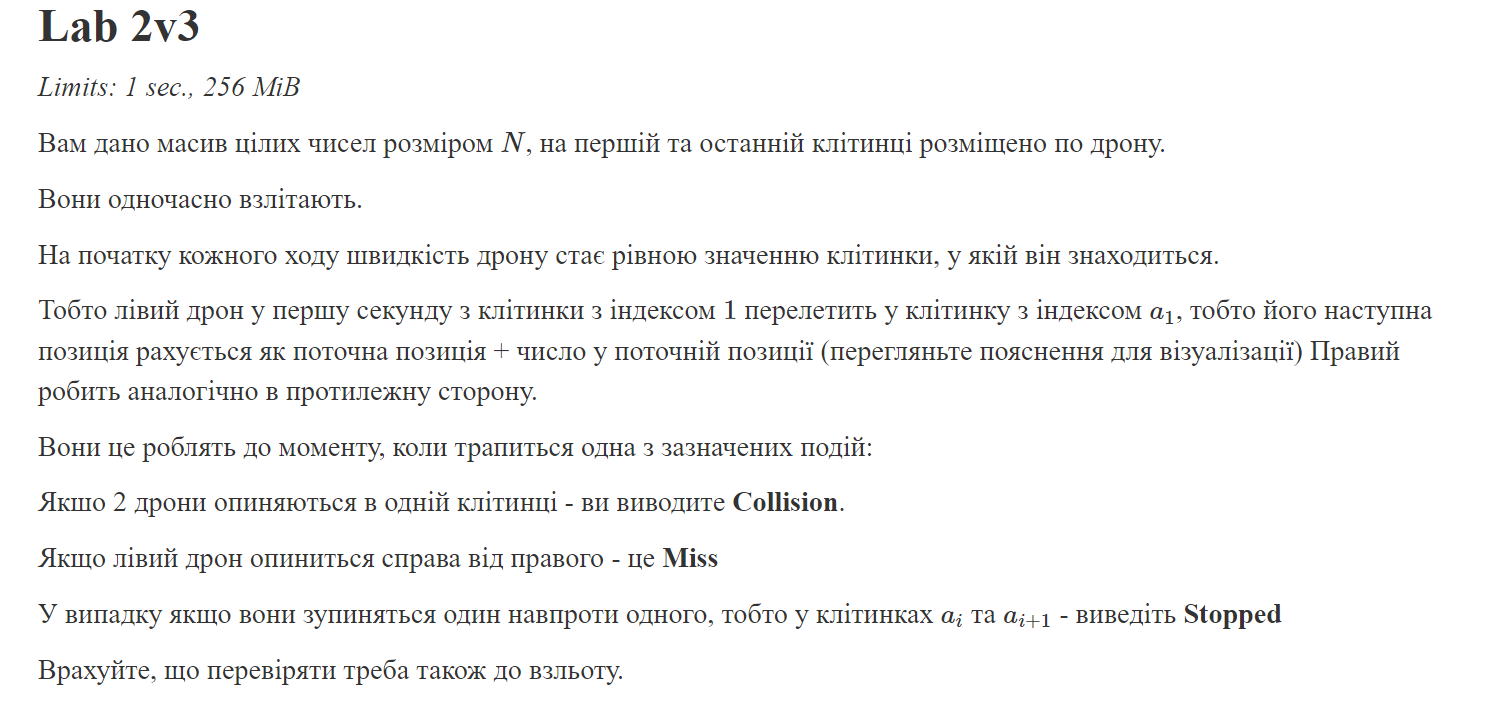
Використовуючи функції, розв’язати зазначене у варіанті завдання. Масив повинен передаватися у функцію як параметр.

* Важливі деталі для врахування в імплементації програми

Використання string для утримання символів та використання вектора для утримання рядків

Завдання №4 Algotester Lab №2

* Варіант завдання – 3



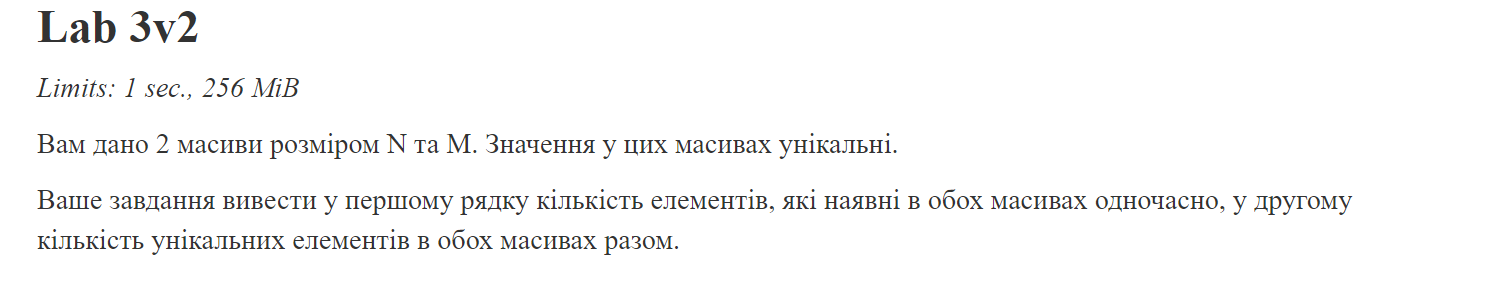
Умова до Алготестера 1

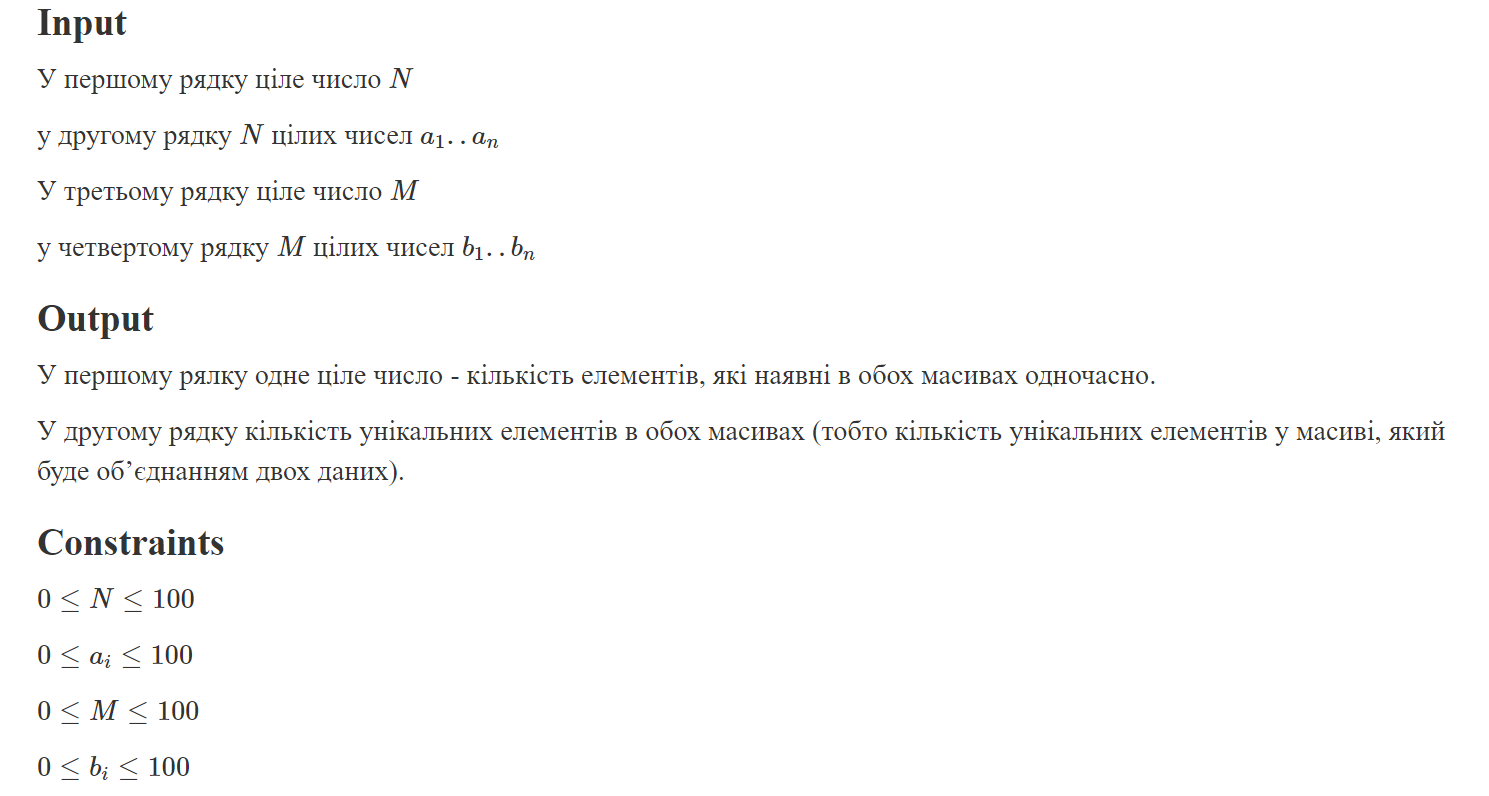
* Важливі деталі для врахування в імплементації програми

Важливість перевірки позицій дронів до взльоту (по інакшому програма не пропустить)

Завдання №5 Algotester Lab №3

* Варіант завдання – 2





Умова до Алготестера 2

* Важливі деталі для врахування в імплементації програми

Немає

Завдання №6 Class Practice Task 4

**Перевірка чи слово або число є паліндромом**

- Деталі завдання

## **Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.**

## **Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).**

## **Мета Задачі**

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

### Вимоги:

1. Визначення функції:

Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.

1. Приклад визначення функції:

*bool isPalindrome(const string& str, int start, int end);*

1. Перевантаження функцій:

Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.

*bool isPalindrome*(ціле число);

1. Рекурсія:

Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

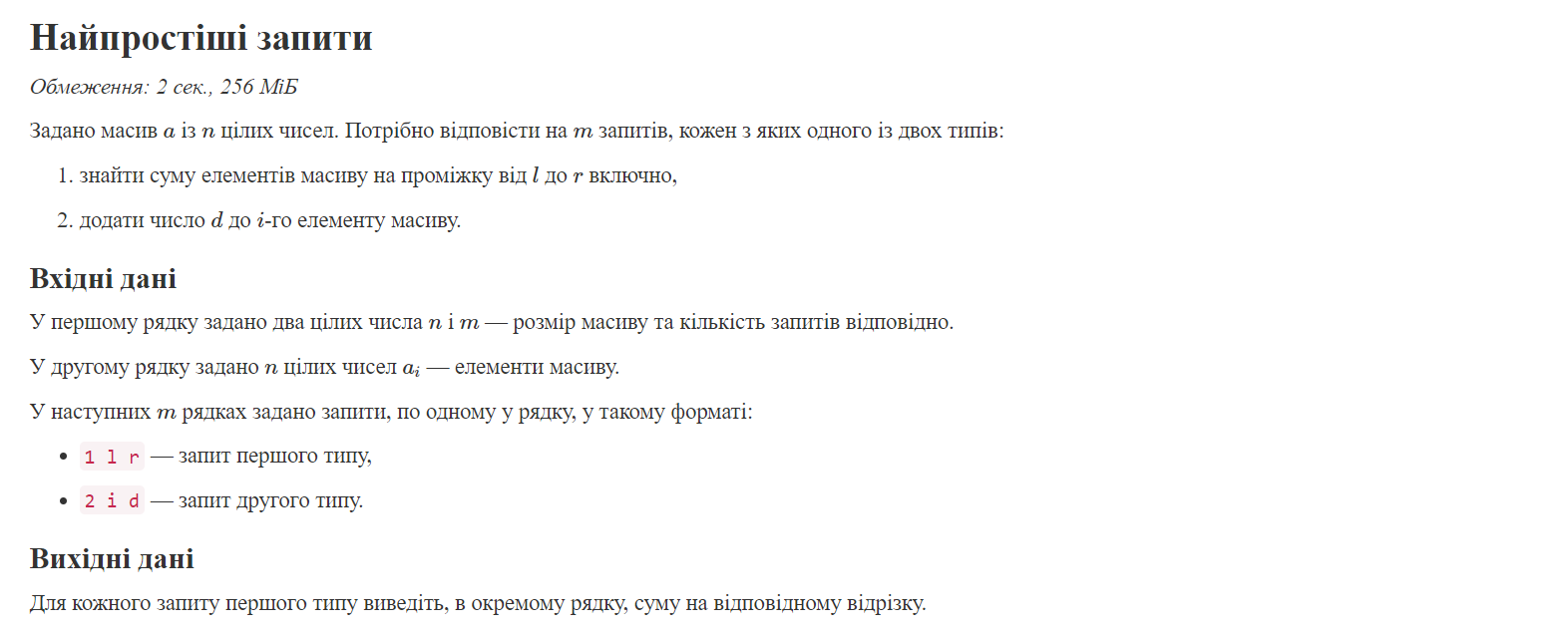
* Важливі деталі для врахування в імплементації програми

Визначте та реалізуйте рекурсивну функцію isPalindrome для рядків.

Визначте та реалізуйте перевантажену функцію isPalindrome для цілих чисел. Перетворити ціле число на рядок і використовуйте рядкову версію функції, щоб перевірити, чи це паліндром.

Завдання №7 Self Practice Algotester Найпростіші запити

* Деталі завдання





Умова до Алготестера 3

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

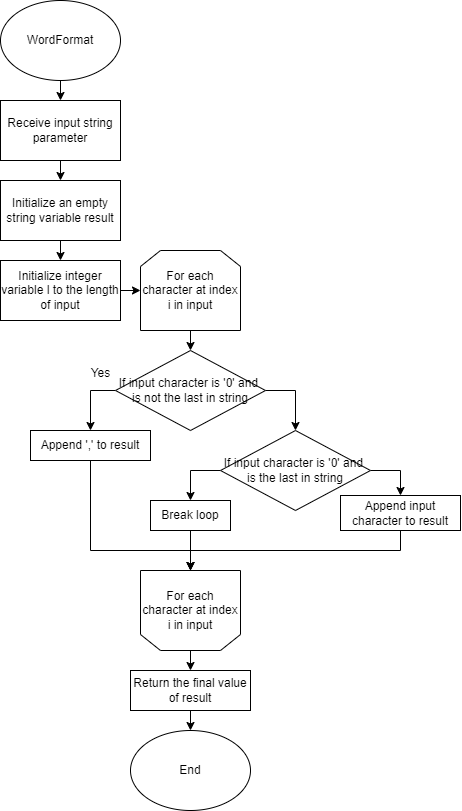
Програма №1 VNS Lab №4

* Блок-схема - відсутня
* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 30 хвилин
* Важливі деталі для врахування в імплементації

Використання вектора для отримання бажаного результату

Програма №2 VNS Lab №5

* Блок-схема



* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 4 години
* Важливі деталі для врахування в імплементації

Використання string для утримання символів та використання вектора для утримання рядків

Програма №3 Algotester Lab №2

* Блок-схема - відсутня
* Планований час на реалізацію – 2 години
* Час затрачений на виконання завдання – 3 години
* Важливі деталі для врахування в імплементації

Врахування всіх обмежень

Програма №4 Algotester Lab №3

* Блок-схема - відсутня
* Планований час на реалізацію – 2 години
* Час затрачений на виконання завдання – 4 години

Важливі деталі для врахування в імплементації

Врахування всіх обмежень

Програма №5 Class Practice Palindrome

* Блок-схема - відсутня
* Планований час на реалізацію – 2 години
* Час затрачений на виконання завдання – 1 година 30 хвилин
* Важливі деталі для врахування в імплементації

Визначте та реалізуйте рекурсивну функцію isPalindrome для рядків.

Визначте та реалізуйте перевантажену функцію isPalindrome для цілих чисел. Перетворити ціле число на рядок і використовуйте рядкову версію функції, щоб перевірити, чи це паліндром.

Програма №5 Self Practice Algotester Найпростіші запити

* Блок-схема - відсутня
* Планований час на реалізацію – 30 хвилин
* Час затрачений на виконання завдання – 3 години
* Важливі деталі для врахування в імплементації

Важливою деталлю є оптимізація програми, адже алготестер перевіряє її з великими значеннями m та n.

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

Завдання №1 VNS Lab №4 Task 1

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/595/files#diff-3ca086f5f34892691b0c0fa1d43bb2c86b0450cac0dbae2985723fab3af965da>

#include <iostream>

#include <vector>

#include <time.h>

using namespace std;

int main()

{

    int n;

    cin>>n;

    //Генеруєм масив з випадкових чисел та друкуєм його

    vector<int> array;

    srand(time(0));

    for(int i = 0; i<n;i++)

    {

        array.push\_back(rand() % 1000);

        cout<<array[i]<<" ";

    }

    cout<<"\n";

    //Обираєм номер числа який стерти

    int k;

    cin>>k;

    k-=1;

    //Стиражмо цей елемент

    auto it = array.begin()+k;

    array.erase(it);

    //Вставляємо нулі після кожного парного числа

    for(int na=2;na<array.size();na+=3)

    {

        it = array.begin()+na;

        array.insert(it,0);

    }

    //Друкуємо результат

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

    {

        cout<<array[i]<<" ";

    }

    cout<<"\n";

}

*Код програми №1 VNS Lab №4*

Завдання №2 VNS Lab №5 Task 1

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/595/files#diff-b1e9b3a89769e2579465ac7c4893bda798f2bd76d93d182c72875e8b12088d4f>

#include <iostream>

#include <vector>

#include <string>

using namespace std;

string wordFormat(string input)

{

    string result;

    int l = input.length();

    for(int i = 0;i<l;i++)

    {

        if(input[i]=='0'&&i!=l-1)

        {

            result+=',';

        }

        else if(input[i]=='0'&&i==l-1)

        {

            break;

        }

        else

        {

            result+=input[i];

        }

    }

    return result;

}

int main()

{

    vector <string> word;

    string input;

    while(true)

    {

        getline(cin,input);

        if(input.empty())

        {

            break;

        }

        word.push\_back(wordFormat(input));

    }

    int lineLength = 0;

    for(int i = 0; i<word.size();i++)

    {

        if (word[i].length() + 1 > 60)

        {

            for(int n = 60,j = 0;n-60<word[i].length();n+=60)

            {

                output+='(';

                for(j = n-60;j<n&&j<word[i].length();j++)

                {

                    if(word[i][j]!=','||j<n-1)

                    {

                        output +=word[i][j];

                    }

                }

                output += ")";

            }

        }

        else

        {

            output+='(';

            output+=word[i];

            output+=')';

        }

    }

    cout<<output<<"\n";

}

*Код програми №2 VNS Lab №5*

Завдання №3 Algotester Lab №2

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/595/files#diff-2e7d3fcdf49b9392613233b18d647a4adc19feb2f2b48916935b001023e6efa1>

#include <iostream>

#include <string>

#include <cmath>

using namespace std;

int main()

{

    int stepright = 0;

    int stepleft = 0;

    int initial = 0;

    int n;

    cin >> n;

    int final = n-1;

    int road[n];

    for(int q = 0; q<n; q++)

    {

        cin >> road[q];

    }

    while(initial+stepright<=final-stepleft)

    {

        if(initial+stepright==final-stepleft)

        {

            cout << initial+stepright+1<<" "<<final-stepleft+1<<"\n";

            cout << "Collision";

            return 0;

        }

        else if(initial+stepright+1==final-stepleft)

        {

            cout << initial+stepright+1<<" "<<final-stepleft+1<<"\n";

            cout << "Stopped";

            return 0;

        }

        initial = initial+stepright;

        stepright = road[initial];

        final = final-stepleft;

        stepleft = road[final];

    }

    cout << initial+stepright+1<<" "<<final-stepleft+1<<"\n";

    cout << "Miss";

    return 0;

}

*Код програми №3 Algotester Lab 2*

Завдання №4 Algotester Lab №3

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/595/files#diff-ca6ffb598abab25a731beaaeb836ac63c60069d51cba9632f910552ee8d20ec5>

#include <iostream>

using namespace std;

int look(int array[], int size, int a)

{

    for(int i = 0;(i<size);i++)

    {

        if(array[i]==a)

        {

            return 1;

        }

    }

    return 0;

}

int main()

{

    int result=0;

    int n,m;

    cin>>n;

    int narray[n];

    for(int i = 0;i<n;i++)

    {

        cin>>narray[i];

    }

    cin>>m;

    int marray[m];

    for(int i = 0;i<m;i++)

    {

        cin>>marray[i];

    }

    for(int i = 0;i<n;i++)

    {

        result += look(marray, m ,narray[i]);

    }

    cout<<result<<"\n";

    cout<<m+n-result;

}

*Код програми №4 Algotester Lab 3*

Завдання №5 Class Practice Palindrome

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/595/commits/b64795ae3e4946ca68322f4b7a08f90c77a33dcb#diff-ca6ffb598abab25a731beaaeb836ac63c60069d51cba9632f910552ee8d20ec5>

#include <iostream>

#include <string>

using namespace std;

bool isPalindrome(const string& str, int start, int end)

{

    if (start >= end)

    {

        return true;

    }

    if (str[start] == str[end])

    {

        return isPalindrome(str, start + 1, end - 1);

    }

    return false;

}

bool isPalindrome(int num)

{

    string numStr = to\_string(num);

    return isPalindrome(numStr, 0, numStr.length() - 1);

}

int main()

{

    string str;

    cout << "Enter a string: ";

    cin >> str;

    if (isPalindrome(str, 0, str.length() - 1))

    {

        cout << str << " is a palindrome.\n";

    } else

    {

        cout << str << " is not a palindrome.\n";

    }

    return 0;

}

*Код програми №5 Class Practice – Palindrome*

Завдання №6 Self Practice Algotester Найпростіші запити

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/595/files#diff-d55bc4be86125707cd02f3f9f8cea10bc08dcef41dcd8a4948e5dd62ceb41ebe>

#include <iostream>

using namespace std;

int main()

{

    int n, m;

    cin >> n >> m;

    int array[n];

    for(int i = 0; i < n; i++)

    {

        cin >> array[i];

    }

    int prefix\_sum[n];

    prefix\_sum[0] = array[0];

    for(int i = 1; i < n; i++)

    {

        prefix\_sum[i] = prefix\_sum[i-1] + array[i];

    }

    int tasks[m][3];

    for(int i = 0; i < m; i++)

    {

        cin >> tasks[i][0] >> tasks[i][1] >> tasks[i][2];

    }

    for(int i = 0; i < m; i++)

    {

        if(tasks[i][0] == 1)

        {

            cout << prefix\_sum[tasks[i][2]-1]-prefix\_sum[tasks[i][1]-2] << "\n";

        }

        else if(tasks[i][0] == 2)

        {

            for(int j = tasks[i][1]-1; j < n; j++)

            {

                prefix\_sum[j] += tasks[i][2];

            }

        }

    }

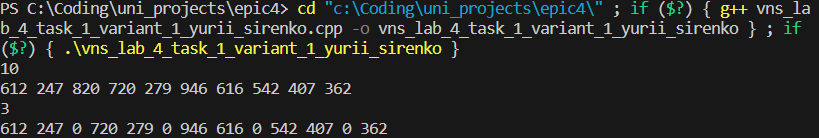
}

*Код програми №6 Self Practice Algotester Найпростіші запити*

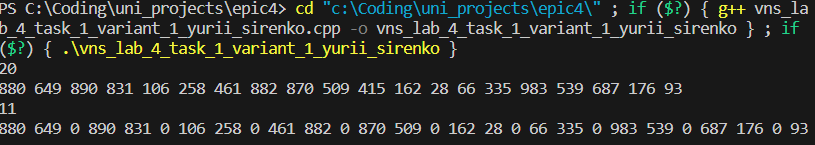
## **4. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 VNS Lab №4

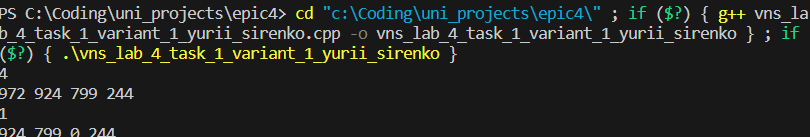
Деталі по виконанню і тестуванню програми



VNS Lab №4 Test 1



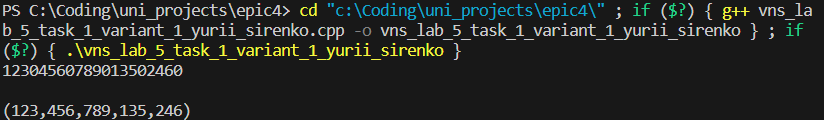
VNS Lab №4 Test 2



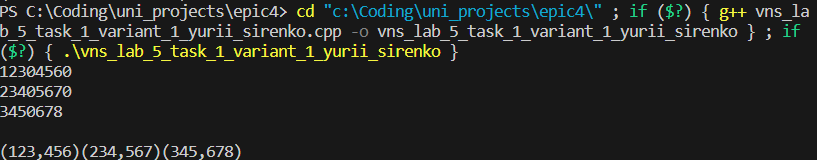
VNS Lab №4 Test 3

Завдання №2 VNS Lab №5

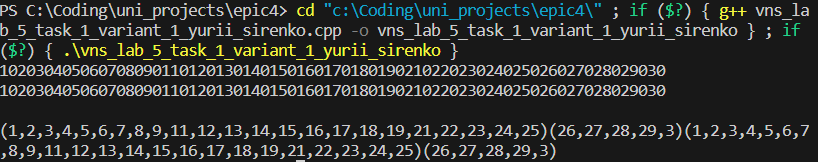
Деталі по виконанню і тестуванню програми



VNS Lab №5 Test 1



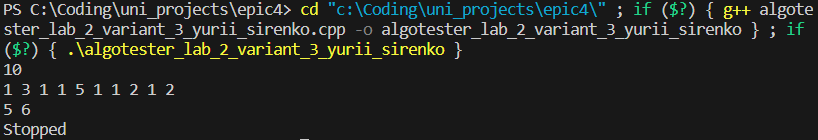
VNS Lab №5 Test 2



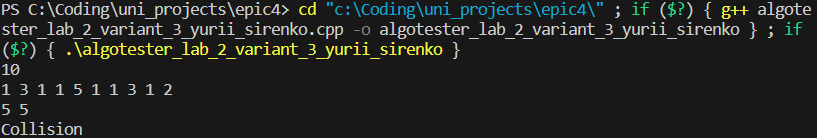
VNS Lab №5 Test 3

Завдання №3 Algotester Lab №2

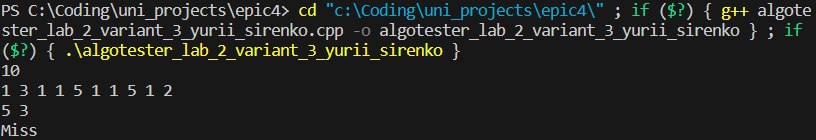
Деталі по виконанню і тестуванню програми



Algotester Lab №2 Test 1



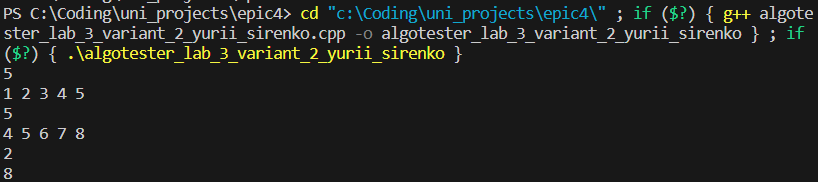
Algotester Lab №2 Test 2



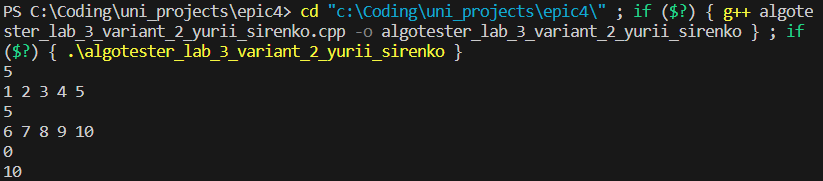
Algotester Lab №2 Test 3

Завдання №4 Algotester Lab №3

Деталі по виконанню і тестуванню програми



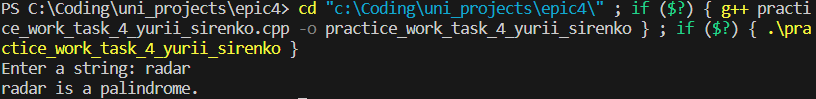
Algotester Lab №3 Test 1



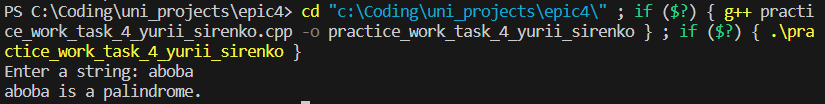
Algotester Lab №3 Test 2

Завдання №5 Class Practice Palindrome

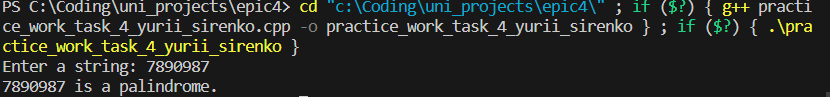
Деталі по виконанню і тестуванню програми



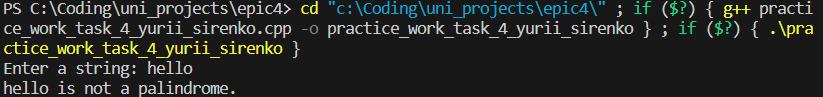
Class Practice Palindrome Test 1



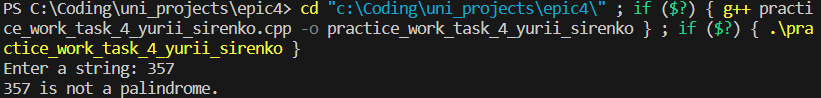
Class Practice Palindrome Test 2



Class Practice Palindrome Test 3



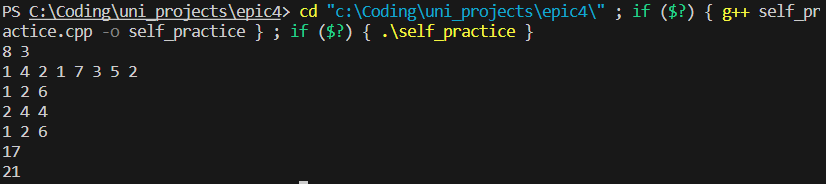
Class Practice Palindrome Test 4



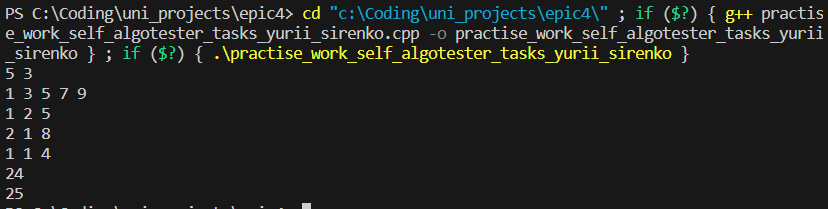
Class Practice Palindrome Test 5

Завдання №6 Self Practice Algotester Найпростіші запити

Деталі по виконанню і тестуванню програми



Self Practice Algotester Найпрос... Test 1

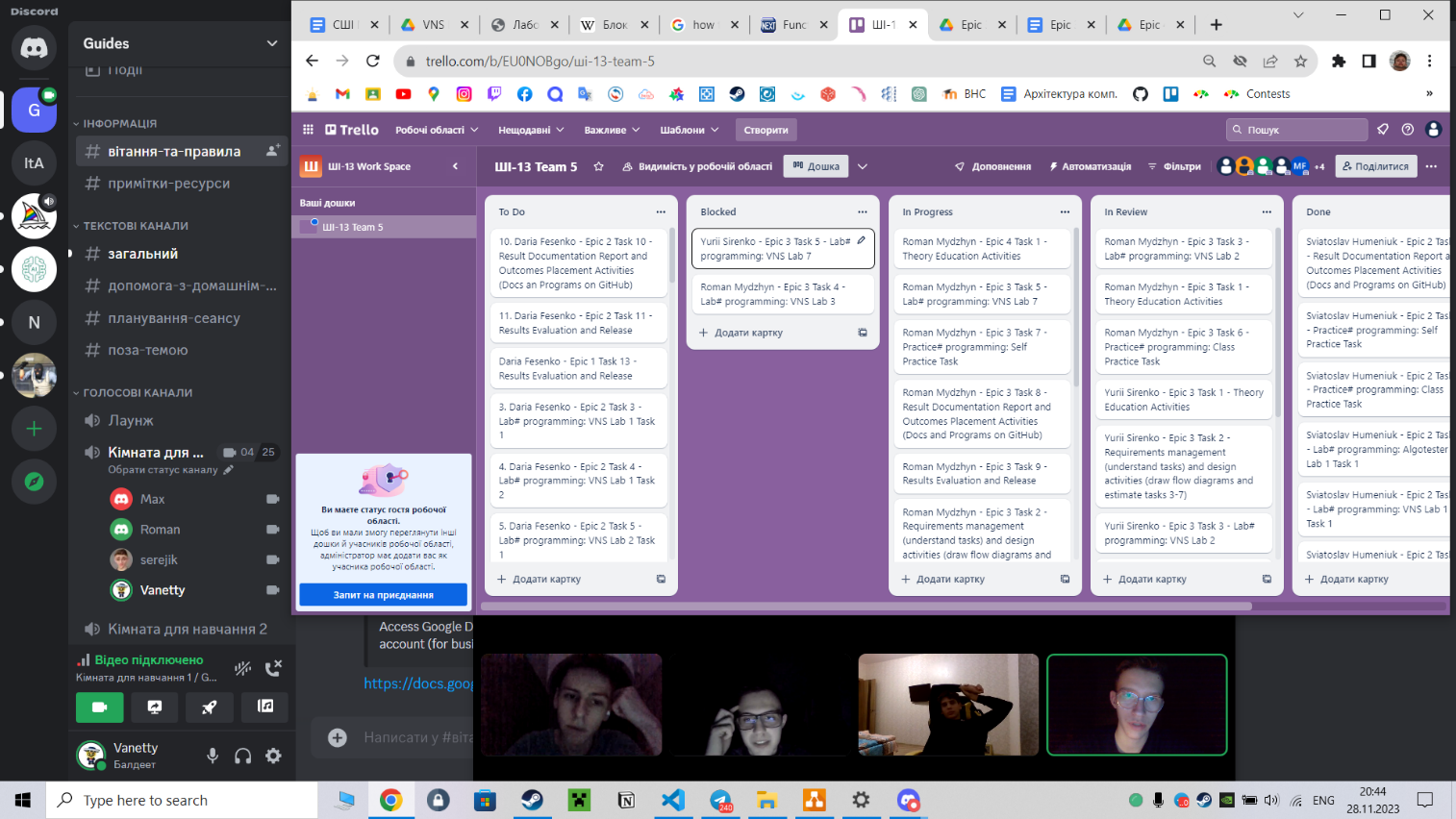


Self Practice Algotester Найпрос... Test 2

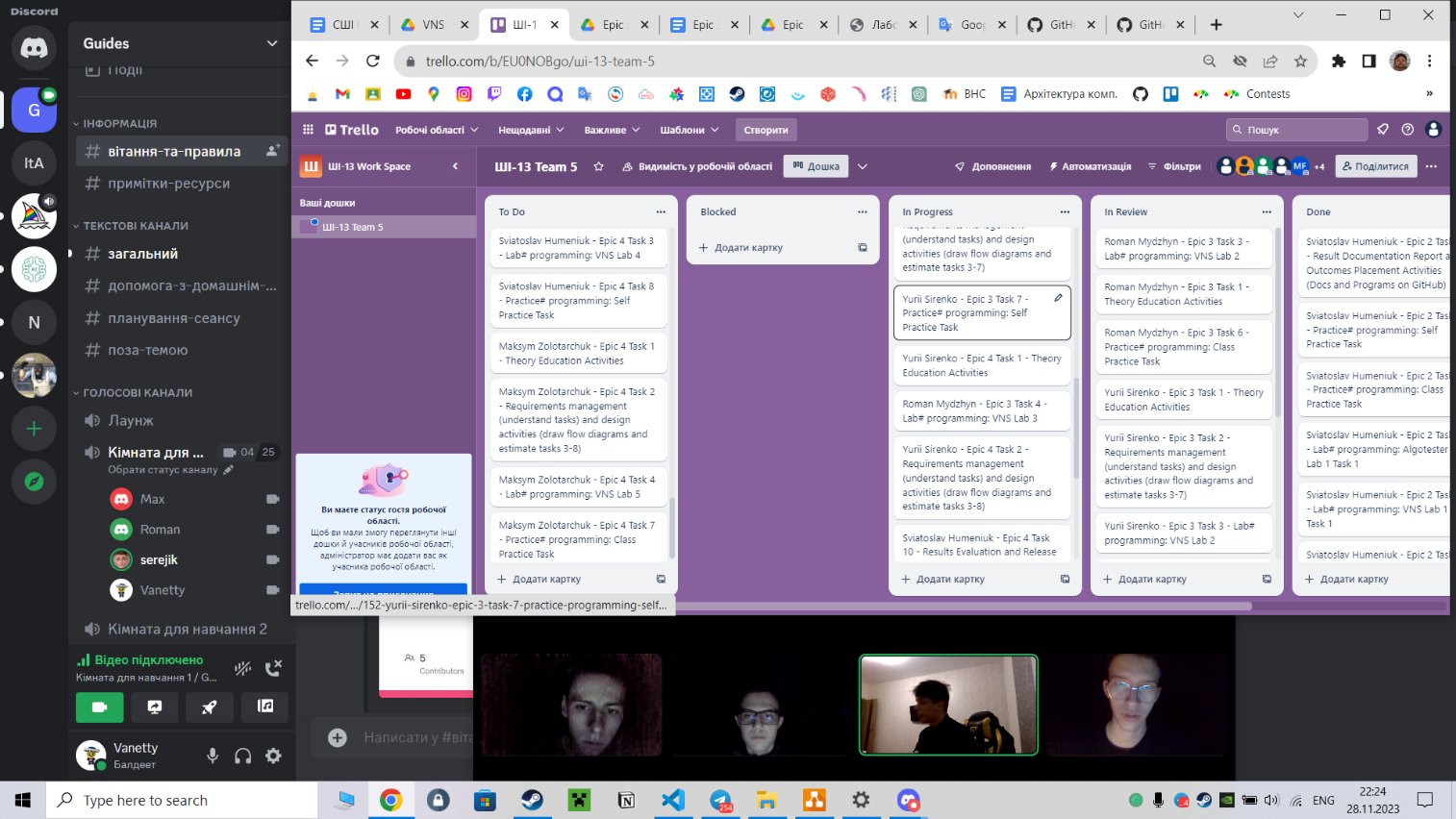
## 

## **5. Кооперація з командою:**

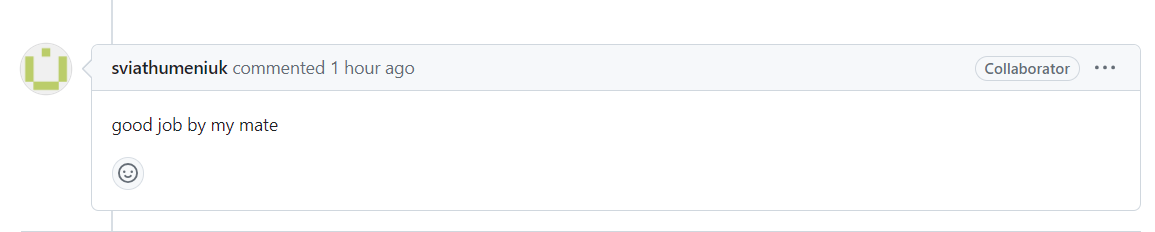
* Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло



* Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло



* Скрін з 2-му коментарями від учасників команди на пул реквесті з Ревю Роботи



# **Висновки:**

У ході цієї лабораторної роботи були ретельно розглянуті та проаналізовані основні структури даних, такі як прості структури, одновимірні та двовимірні масиви, а також алгоритми обробки, пов'язані з цими структурами. Наша мета полягала у вивченні та демонстрації ключових концепцій, властивостей та практичних застосувань цих елементів програмування.

Починаючи з простих структур даних, ми зрозуміли різні типи даних та операції, які можна виконувати з ними. Ознайомившись із світом одновимірних масивів, ми набули навичок роботи з індексами, доступу до елементів та застосування різноманітних алгоритмів обробки даних до цих структур.

Далі, при вивченні двовимірних масивів, ми освоїли складніші концепції індексації та доступу до елементів, а також розглянули реальні застосування, такі як обробка зображень чи математичні операції над матрицями.

Останній розділ лабораторної роботи присвячено алгоритмам обробки даних. Ми вивчили загальні алгоритми сортування, пошуку та їх використання для ефективної роботи з масивами.

Загалом, ця лабораторна робота дозволила не лише засвоїти теоретичні аспекти структур даних та алгоритмів, але й успішно застосувати ці знання на практиці. Отримані навички стануть важливим інструментом у подальших програмних розробках, де оптимальне використання та обробка даних є ключовими завданнями.