Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

**ро виконання лабораторних та практичних робіт блоку № 4**

На тему:  «Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.»

***з дисципліни:*** «Основи програмування»

до:

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт до блоку №4

**Виконав:**

Студент групи ШІ-12

Михальчук Антон Євгенійович

# **Тема роботи:**

# Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.

# **Мета роботи:**

# Метою даної роботи є вивчення та закріплення теоретичних і практичних знань щодо використання та маніпулювання одновимірними та двовимірними масивами, а також розуміння принципів роботи з вказівниками, посиланнями та динамічними масивами. Окрім того, передбачається розгляд та застосування основних структур даних, включаючи вкладені структури, а також реалізація алгоритмів обробки масивів та структур для ефективного вирішення практичних задач програмування.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №\*.1: С++ Arrays, Data structures

1. Індивідуальний план опрацювання теорії:

* Тема №\*.1: С++ Basics
  + Джерела Інформації
    - Відео. <https://www.youtube.com/watch?v=2UDMGCcRCjo>
    - Стаття. <https://www.w3schools.com/cpp/>
  + Що опрацьовано:
    - Вивчив базовий синтаксис та семантику мови C++.
    - Особливу увагу приділяв таким темам, як масиви
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.09.2024
  + Звершення опрацювання теми: 11.11.2024

# **Виконання роботи:**

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

Завдання №1 VNS Lab 4 Варіант: 9

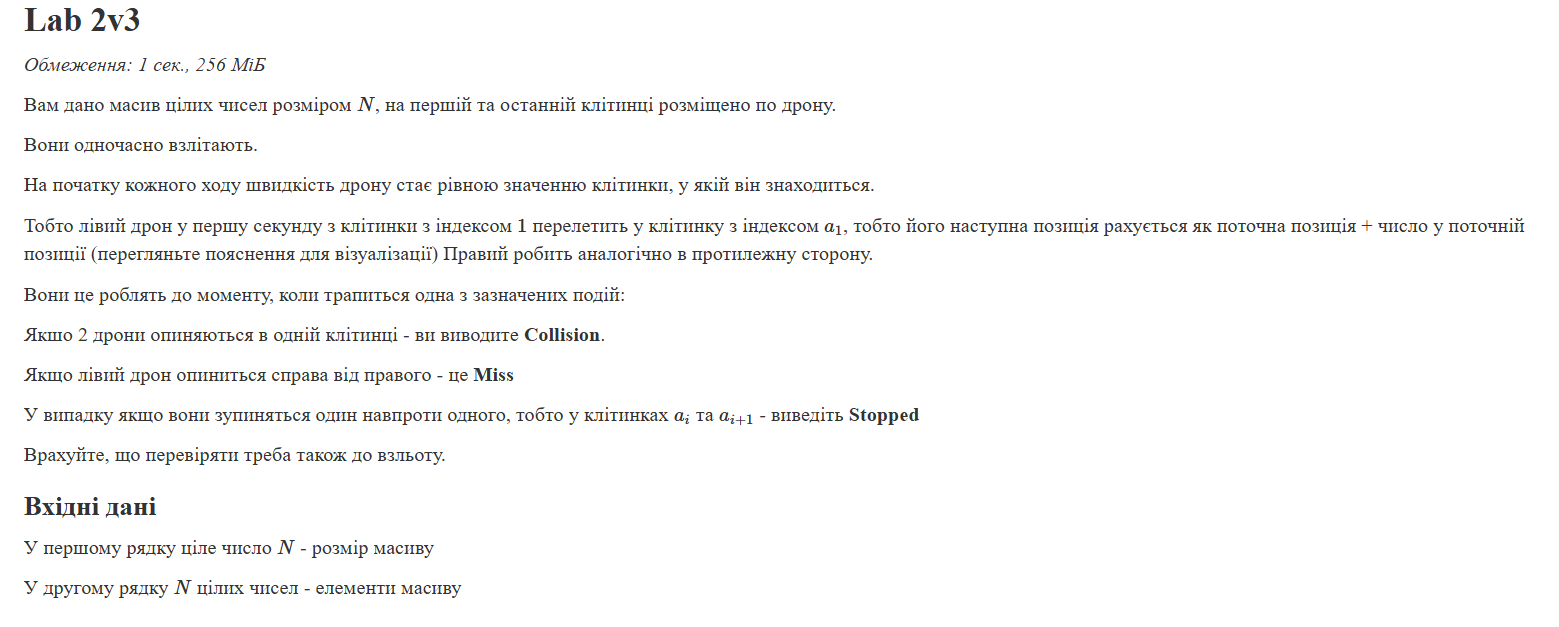
* Деталі завдання:
  + Сформувати одновимірний масив цілих чисел, використовуючи генератор
  + випадкових чисел.
  + Роздрукувати отриманий масив.
  + Знищити перший елемент із заданим значенням.
  + Зсунути масив циклічно на К елементів вправо.
  + Роздрукувати отриманий масив.
* Важливі деталі для врахування в імплементації програми:
  + При виконанні роботи використовуються статичні масиви. Для організації статичних масивів із псевдозмінними межами необхідно оголосити масив досить великої довжини, наприклад, 100 елементів.
  + При зменшенні або збільшенні довжини масиву необхідно змінювати його реальну довжину.

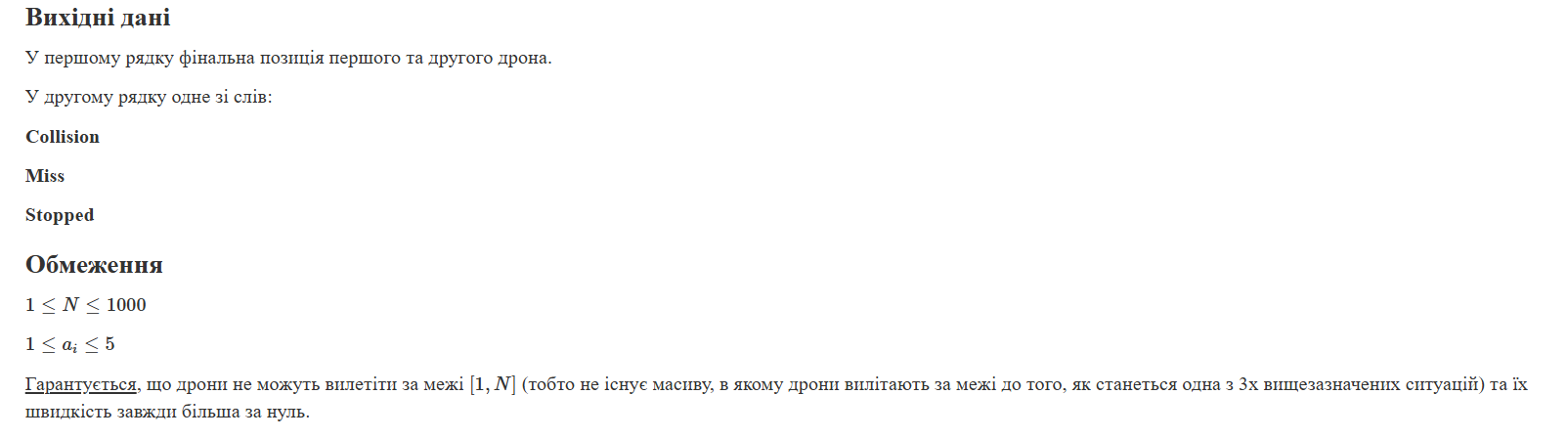
Завдання №2 VNS Lab 5 Варіант: 9

* Деталі завдання:
  + Написати функцію для обчислення суми елементів квадратної матриці, які розташовані нижче головної діагоналі. З її допомогою знайти максимальне значення такої суми в n матрицях.

Завдання №3 Algotester Lab 2 Варіант: 3

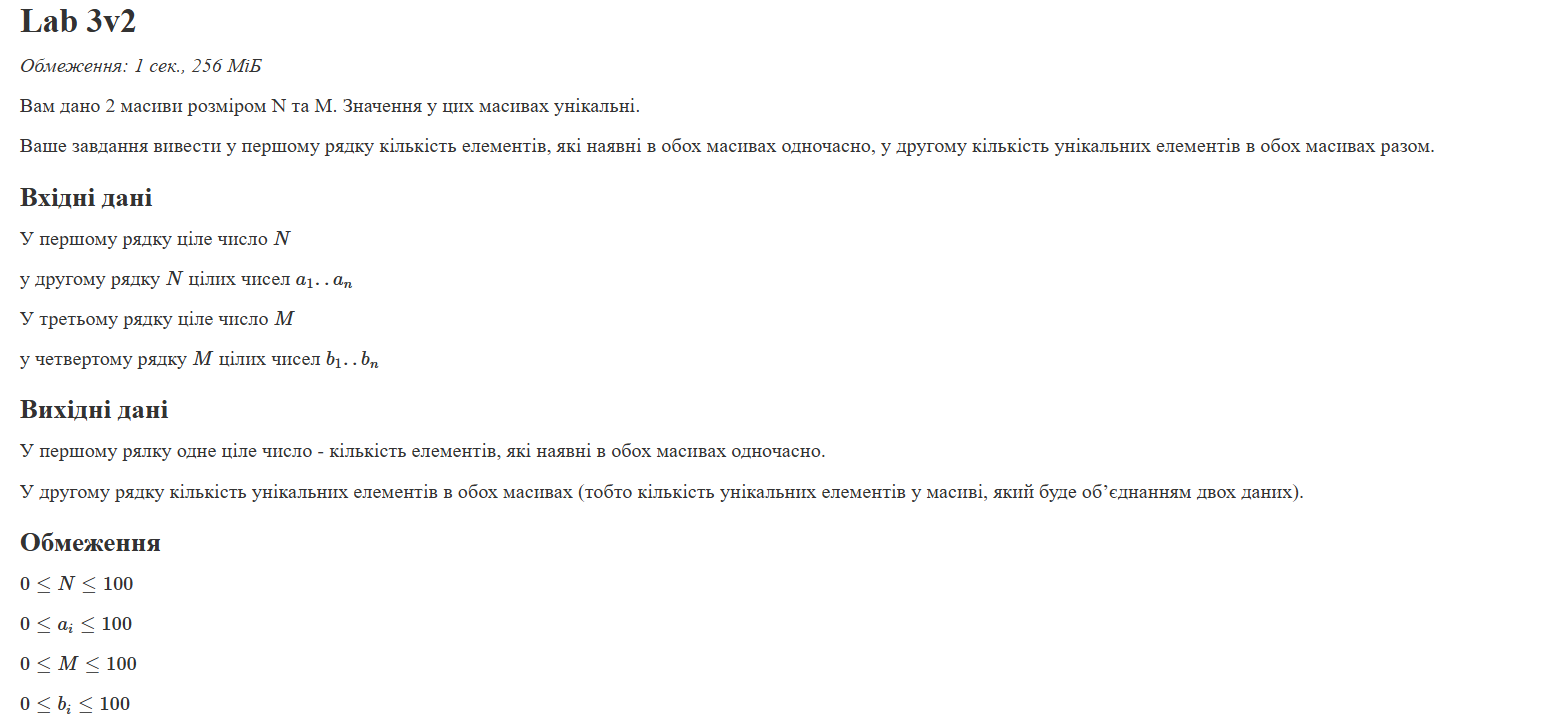
* Деталі завдання:





Завдання №4 Algotester Lab 3 Варіант: 2

* Деталі завдання:



Завдання №5 Class Practice Task

* 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»).

Вимоги:

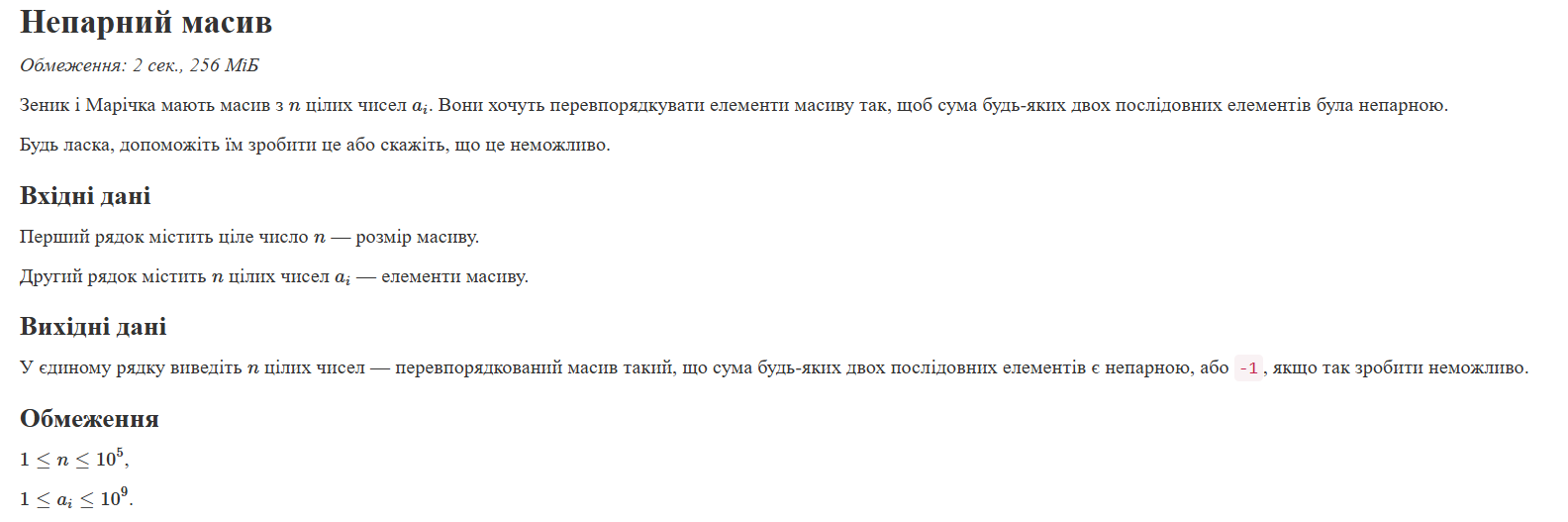
* Визначення функції:
  + Реалізуйте рекурсивну функцію isPalindrome, яка перевіряє, чи заданий рядок є паліндромом.
* Приклад визначення функції:
  + bool isPalindrome(const string& str, int start, int end);
* Перевантаження функцій:
  + Перевантажте функцію isPalindrome для роботи з цілими значеннями.
* bool isPalindrome(ціле число);
* 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.

Кроки реалізації:

* Визначте та реалізуйте рекурсивну функцію isPalindrome для рядків.
* Визначте та реалізуйте перевантажену функцію isPalindrome для цілих чисел.
* Використати математичний підхід щоб перевірити чи число є паліндромом.

Завдання №6 Self Practice Task Непарний масив

* Деталі завдання:



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

Програма №1 VNS Lab 4 Варіант: 9

* Планований час на реалізацію: 20 хв

Програма №2 VNS Lab 5 Варіант: 9

* Планований час на реалізацію: 30 хв.

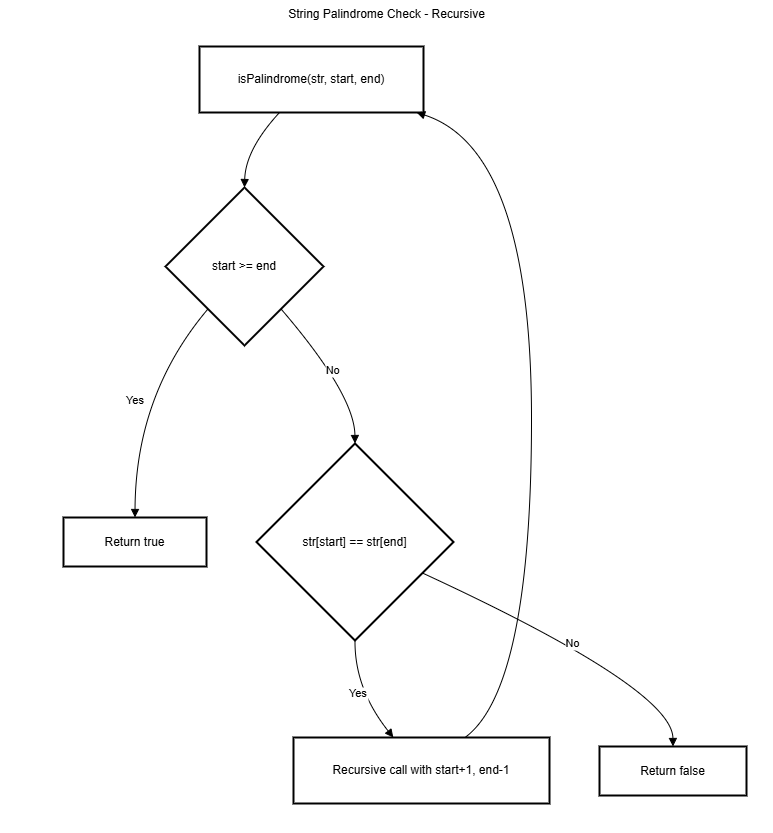
Програма №3 Algotester Lab 3 Варіант: 2

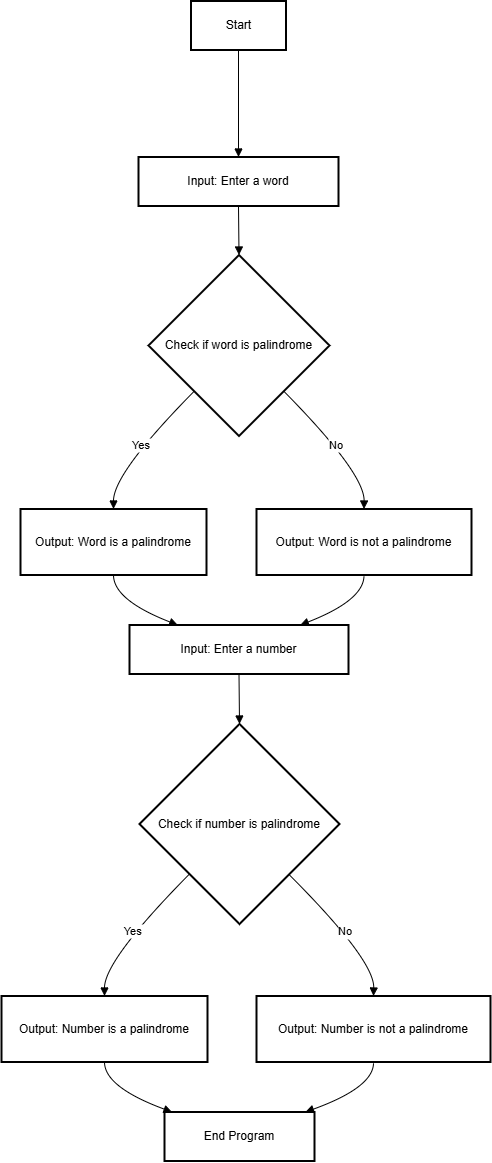
* Планований час на реалізацію: 20 хв.

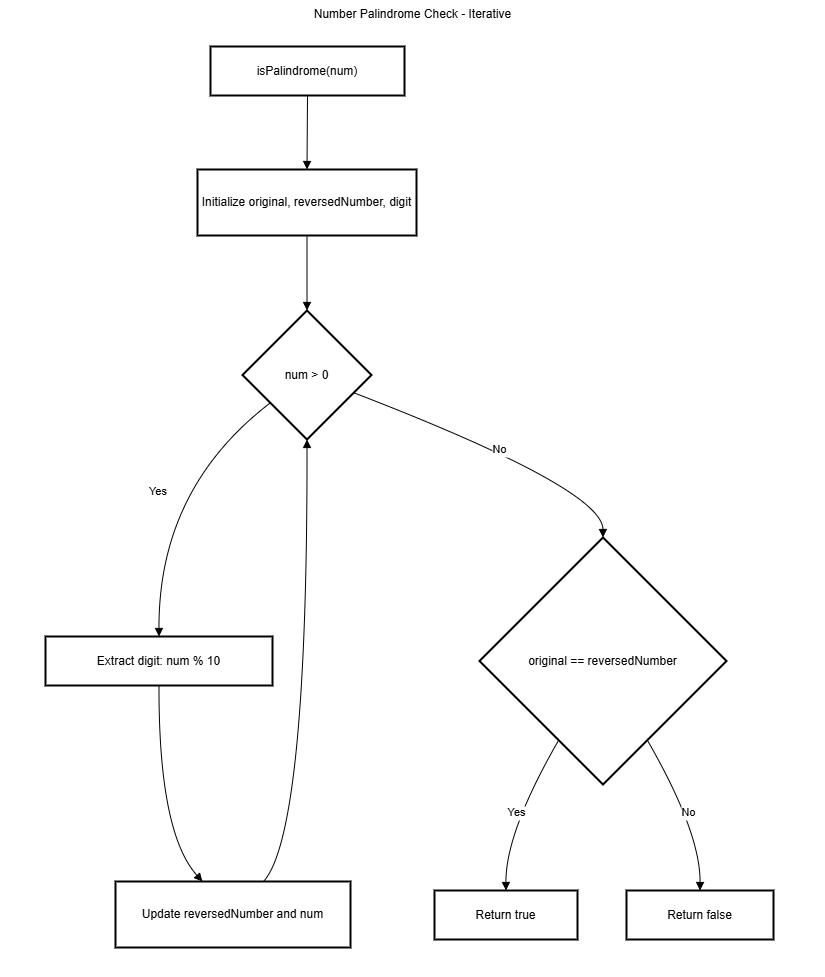
Програма №4 VNS Algotester Lab 2 Варіант: 3

* Планований час на реалізацію: 30 хв.

Програма №5 Class Practice Task

* Блок-схема





* Планований час на реалізацію: 50 хв.

Програма №6 Self Practice Task Непарний масив

* Планований час на реалізацію: 50 хв

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

Завдання №1

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-6c05650ce74c77e06758635acd9b094333fca212b26733039bb5a3e757b76535>

#include <iostream>

using namespace std;

int main()

{

    const double eps = 0.0001;

    double an = 1, s = 0, d, n = -1;

    while (an >= eps)

    {

        n++;

        d = 10 / (n + 1);

        an \*= d;

        s += an;

    }

    cout << "The sum of the series with an accuracy of epsilon = " << eps << " is " << s << " The value is reached at an = " << an << " when n = " << n;

    return 0;

}

Завдання №2

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-415abdda3dfca6073b5be835e7de758b17dd9d99c8d2d62ef64948fc8359bfcf>

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

    const double eps = 0.0001;

    double exactValueY, xn, sn, xe, se;

    int i;

    for (double k = 0.1; k <= 0.8 + eps; k += 0.07)

    {

        exactValueY = 0.25 \* log(((1 + k) / (1 - k))) + 0.5 \* atan(1 / k);

        sn = 0;

        for (int n = 0; n <= 3; n++)

        {

            xn = pow(k, (4 \* n + 1)) / (4 \* n + 1);

            sn += xn;

        }

        xe = k;

        se = 0;

        i = -1;

        while (xe >= eps)

        {

            i++;

            xe = pow(k, (4 \* i + 1)) / (4 \* i + 1);

            se += xe;

        }

        cout << "X = " << k << " SN = " << sn << " SE = " << se << " Y = " << exactValueY << endl;

    }

    return 0;

}

Завдання №3

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-fd57a95df204ab21a978c82bcea0552fcac383c5f6758a63ac5801542b6ba880>

#include <iostream>

#include <cstdarg>

using namespace std;

template<typename T>

T min(T argType, int count, ...)

{

    va\_list args;

    va\_start(args, count);

    T minValue = va\_arg(args, T);

    for (int i = 1; i < count; i++)

    {

        T current = va\_arg(args, T);

        if (current < minValue)

        {

            minValue = current;

        }

    }

    va\_end(args);

    return minValue;

}

int main()

{

    cout << min(1, 5, 5, 3, -4, 6, 7) << endl;

    cout << min(1.1, 10, 2.1234, 5.23, 3, -4.3, -69.23, -4.4, 2, 9, -14.3, 7.56) << endl;

    cout << min(1, 12, 0, 1, -12, 5, 5, 3, -4, 6, 7, 99, -14, -45) << endl;

    return 0;

}

Завдання №4

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-bf12ec01580da6ffba449f66411411c1665a5ce581376ca108c9177703a70581>

#include <iostream>

#include <utility>

using namespace std;

double DecimalAddition(double a, double b)

{

    return a + b;

}

float DecimalAddition(float a, float b)

{

    return a + b;

}

int gcd(int a, int b)

{

    while (b != 0)

    {

        int temp = b;

        b = a % b;

        a = temp;

    }

    return a;

}

pair<int, int> fractionAddition(pair<int, int> f, pair<int, int> s)

{

    int numerator = f.first \* s.second + s.first \* f.second;

    int denominator = f.second \* s.second;

    int d = gcd(numerator, denominator);

    numerator /= d;

    denominator /= d;

    return make\_pair(numerator, denominator);

}

pair<long long, long long> fractionAddition(pair<long long, long long> f, pair<long long, long long> s)

{

    long long numerator = f.first \* s.second + s.first \* f.second;

    long long denominator = f.second \* s.second;

    long long d = gcd(numerator, denominator);

    numerator /= d;

    denominator /= d;

    return make\_pair(numerator, denominator);

}

int main()

{

    double ad = 45.434, bd = 2234.345;

    float af = 45.434, bf = 2234.345;

    cout << DecimalAddition(ad, bd) << endl;

    cout << DecimalAddition(af, bf) << endl;

    pair<int, int> firstFraction(15, 4), secondFraction(23, 6);

    pair<int, int> result = fractionAddition(firstFraction, secondFraction);

    cout << result.first << "/" << result.second << endl;

    pair<long long, long long> firstFractionl(4036854775807, 46476546475), secondFractionl(2342342352453, 345675364446);

    pair<long long, long long> resultl = fractionAddition(firstFractionl, secondFractionl);

    cout << resultl.first << "/" << resultl.second << endl;

    return 0;

}

Завдання №5

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-85cddba83a0396e0ccc52b78ea051c8e0d61827275a4573407dcf3b40ac618a7>

#include <iostream>

#include <utility>

#include <string>

using namespace std;

int main()

{

    char choice;

    string books[10] = {"1984", "To Kill a Mockingbird", "The Great Gatsby", "Moby-Dick", "Pride and Prejudice",

                        "The Catcher in the Rye", "Brave New World", "Kobzar", "Taras Bulba", "The Hobbit"};

    bool bookAvailability[10] = {true, true, true, true, true, true, true, true, true, true};

    int borr, ret;

    bool continueOperation;

    mainMenu:

    while (choice != 'q')

    {

        continueOperation = true;

        cout << "Choose what you want to do in library ('b' for borrowing, 'r' for returning, 'e' for enumerating, 'q' for quitting): ";

        cin >> choice;

        switch (choice)

        {

        case 'b':

        {

            do

            {

                cout << "Write index number of book that you want to borrow, or '100' for returning to menu: ";

                cin >> borr;

                if (borr >= 0 && borr < 10)

                {

                    if (bookAvailability[borr])

                    {

                        bookAvailability[borr] = false;

                        cout << "You have borrowed \"" << books[borr] << "\"." << endl;

                    }

                    else

                    {

                        cout << "The book \"" << books[borr] << "\" is already borrowed." << endl;

                    }

                }

                else if (borr == 100)

                {

                    continueOperation = false;

                }

                else

                {

                    cout << "Invalid book index!" << endl;

                }

            } while (continueOperation);

            break;

        }

        case 'r':

        {

            do

            {

                cout << "Write index number of book that you want to return or '100' for returning to menu: ";

                cin >> ret;

                if (ret >= 0 && ret < 10)

                {

                    if (!bookAvailability[ret])

                    {

                        bookAvailability[ret] = true;

                        cout << "You have returned \"" << books[ret] << "\"." << endl;

                    }

                    else

                    {

                        cout << "The book \"" << books[ret] << "\" was not borrowed." << endl;

                    }

                }

                else if (ret == 100)

                {

                    continueOperation = false;

                }

                else

                {

                    cout << "Invalid book index!" << endl;

                }

                break;

            } while (continueOperation);

            break;

        }

        case 'e':

            for (int i = 0; i < 10; i++)

            {

                cout << "Book \"" << books[i] << "\" with index " << i << " is ";

                if (bookAvailability[i])

                    cout << "available" << endl;

                else

                    cout << "not available!" << endl;

            }

        case 'q':

            cout << "See you later!" << endl;

            break;

        default:

            cout << "Invalid choice! Please try again." << endl;

            goto mainMenu;

            break;

        }

    }

}

Завдання №6

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/185/files#diff-f01da8425fbc0753b1ea2edb71602a9eddad0ad4fc972f56034bfb3df4a5ce56>

#include <iostream>

#include <string>

using namespace std;

int main() {

    int n, winVasyl = 0, winKolia = 0, fullWinVasyl = 0, fullWinKolia = 0;

    string wins;

    cin >> n;

    cin >> wins;

    for (char i : wins) {

        (i == 'V') ? winVasyl += 1 : winKolia += 1;

        if ((winVasyl >= 11) && (winVasyl - winKolia >= 2)) {

            winVasyl = 0;

            winKolia = 0;

            fullWinVasyl += 1;

        }

        else if ((winKolia >= 11) && (winKolia - winVasyl >= 2)) {

            winVasyl = 0;

            winKolia = 0;

            fullWinKolia += 1;

        }

    }

    cout << fullWinKolia<< ':' << fullWinVasyl << endl;

    if ((winVasyl > 0) || (winKolia > 0))

        cout << winKolia << ':' << winVasyl;

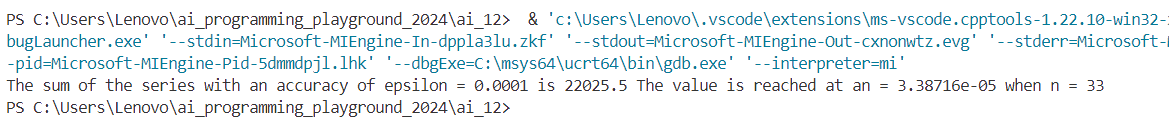
}

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

Завдання №1

Виведемо рекурентну формулу

Тепер щоб отримати наступний елемент потрібно помножити попередній елемент на d.

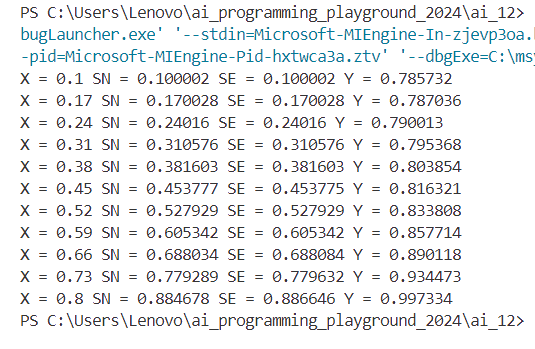


Час затрачений на виконання завдання: 35 хв.

Завдання №2

Отримані результати можуть навести на висновок, що задане n, чи розмір епсілона не дуже змінюють результат. Проте самі степеневі ряди дуже сильно відрізняються від значення функції.

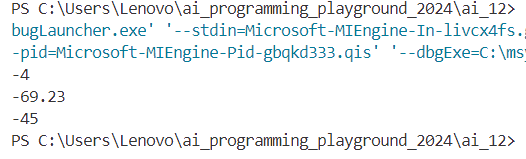
Особливості виконання роботи було те що, бібліотека cmath не надає функції arccot(x), тому я її замінив рівною функцією arctan(1/x).



Час затрачений на виконання завдання: 40 хв.

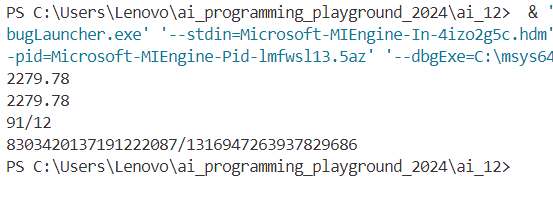
Завдання №3

Наведені приклади з методички використовуть не правильні методи для с++, тому довелося використати va\_list для задання змінної кількості аргументів в функції та шаблони для використання різних типів змінних в функції.



Завдання №4

Для додавання десяткових чисел створив перевантажені функцію, що сприймає double і float типи змінних. Звичайні дроби вирішив задати, як пари int або long long, та створив відповідну перевантажену функцію.



Завдання №5

 **Ініціалізація**:

* Програма починається з ініціалізації масивів books та bookAvailability, де зберігаються назви книг та їх доступність відповідно. Масив books містить назви 10 книг, а масив bookAvailability має булеві значення, які вказують, чи доступна відповідна книга (значення true означає, що книга доступна).

 **Меню**:

* Користувачу пропонується вибрати одну з опцій (створено за допомогою while та switch case):
  + 'b' - взяти книгу.
  + 'r' - повернути книгу.
  + 'e' - переглянути доступність книг.
  + 'q' - завершити роботу програми.
* Після вибору користувачем опції, програма виконує відповідну дію.

 **Позичання книги (створено за допомогою do while):**

* Якщо обрано 'b', користувачу пропонується ввести індекс книги, яку він хоче позичити.
* Якщо книга доступна, то її статус змінюється на false, і користувач отримує повідомлення про успішне позичання.
* Якщо книга вже позичена, програма повідомляє про це.
* Для повернення до меню користувач може ввести 100.

 **Повернення книги (створено за допомогою do while):**

* Якщо обрано 'r', користувачу пропонується ввести індекс книги, яку він хоче повернути.
* Якщо книга позичена (статус false), вона змінюється на true, і програма виводить повідомлення про успішне повернення.
* Якщо книга не була позичена, програма повідомляє про це.
* Для повернення до меню користувач може ввести 100.

 **Перегляд наявності книг (створено за допомогою for)**

* Опція 'e' дозволяє користувачу переглянути поточний статус кожної книги, вказуючи, чи доступна книга.

 **Вихід із програми (створено за допомогою goto)**:

* Вибір 'q' завершує роботу програми та виводить повідомлення "See you later!".

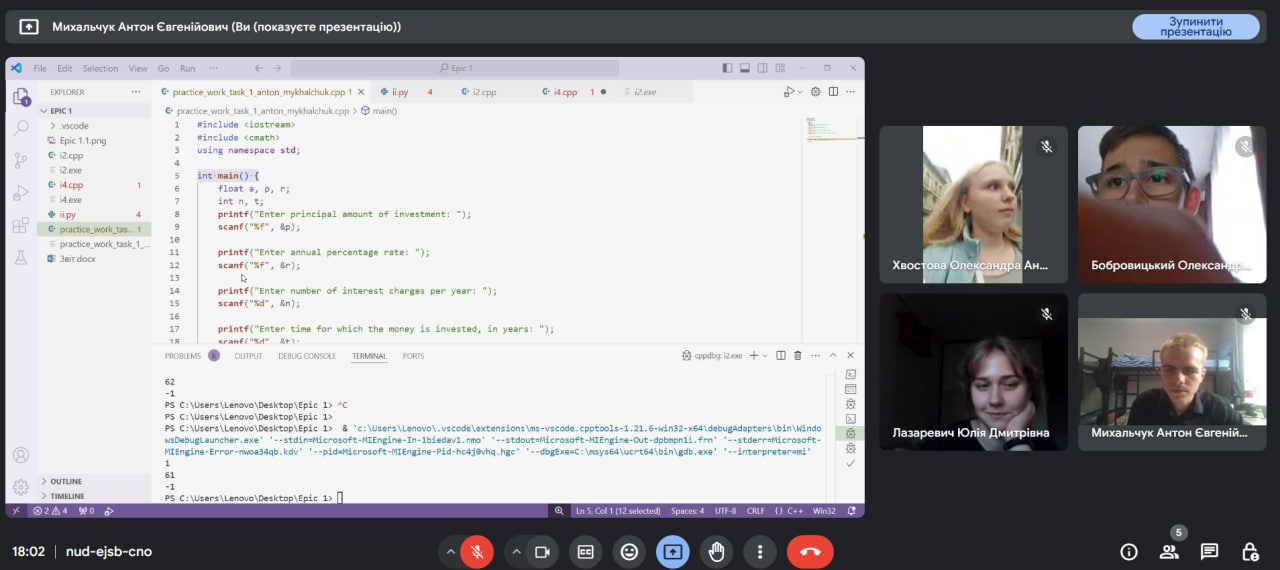
Завдання №6

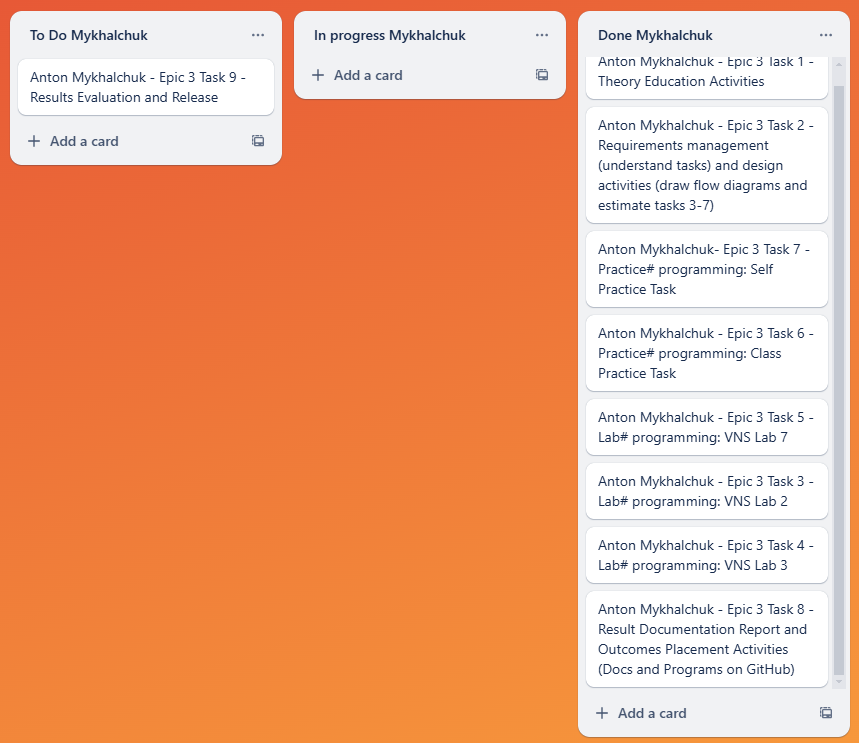
При кожному виграші очка ('V' для Василя і 'K' для Колі) програма перевіряє, чи є переможець партії. Якщо так, очки обнуляються, і збільшується лічильник виграних партій. Виводяться результати виграних партій і, якщо є незакінчена партія, поточні очки.

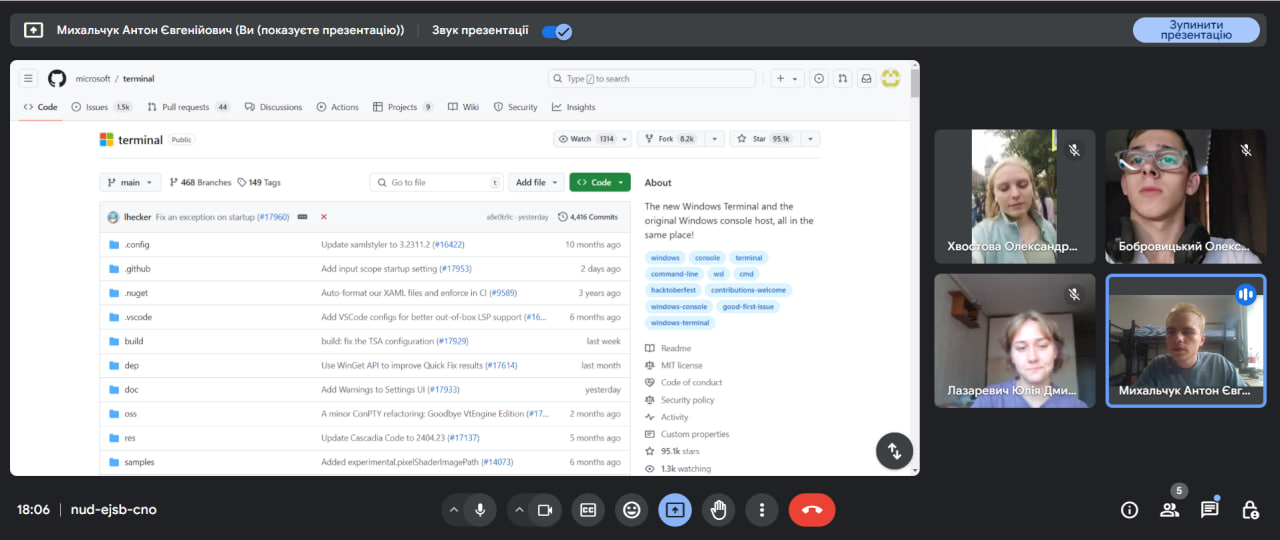
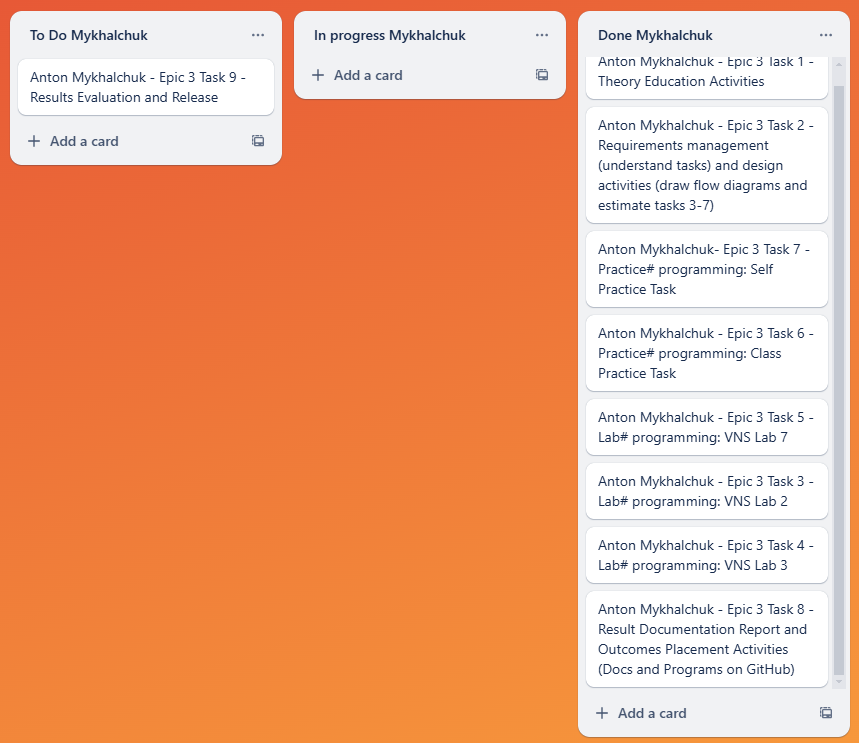
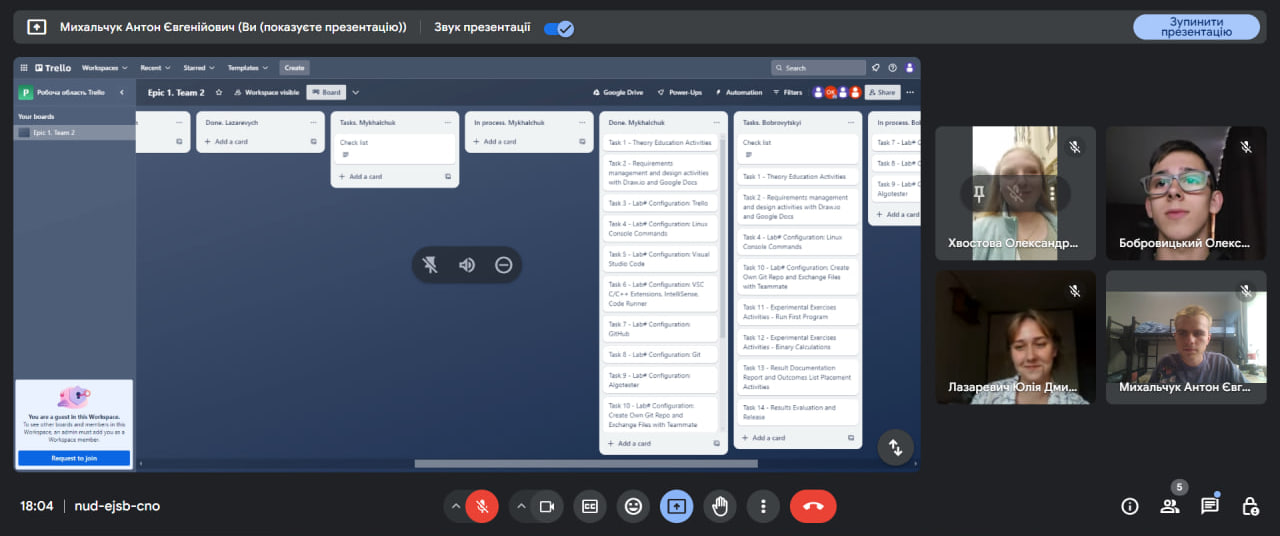


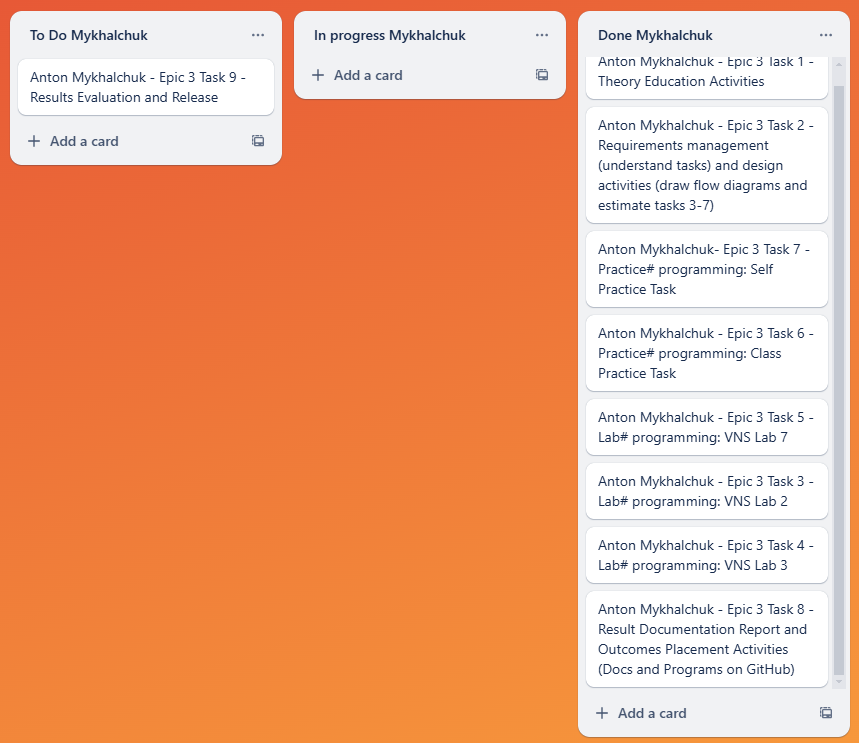
## **6. Кооперація з командою:**

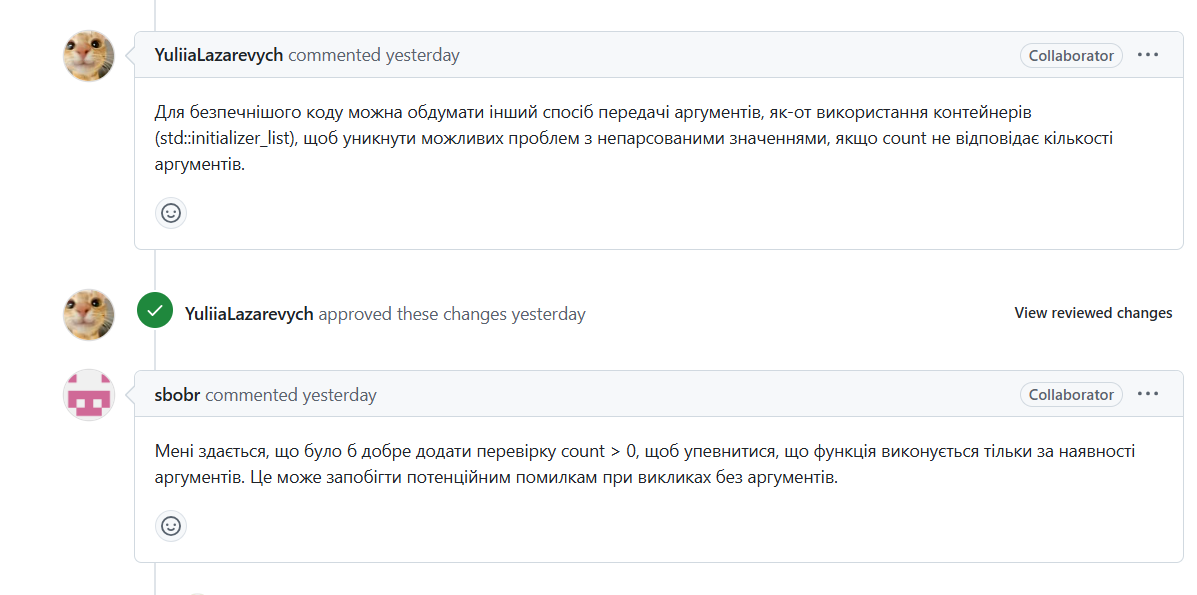
* Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло





* Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло
* 
* 
* Скрін з 3-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло (опційно)
* 



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

# **Висновки:**

У ході роботи було розглянуто та опрацьовано основні концепції програмування в мові C++, пов'язані з циклами та функціями. Особливу увагу приділено вкладеним циклам, умовам завершення їх виконання, а також реалізації різноманітних завдань за допомогою циклічних структур.

Засвоєно принципи роботи з функціями, включаючи базові виклики та перевантаження для підтримки аргументів різного типу. Досліджено використання функцій зі змінною кількістю параметрів, рекурсивні виклики та вбудовані функції для оптимізації коду. Виконані завдання дозволили поглибити розуміння застосування теоретичних знань на практиці, включаючи реалізацію математичних розрахунків через ряди, обчислення з точністю до ε, а також створення програм із використанням структурованих та перевантажених функцій для специфічних сценаріїв.

Загалом, робота дала змогу оволодіти важливими аспектами циклів і функцій, що є основою для створення ефективних програм на C++.