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. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.»

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

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт до блоку №5

**Виконав:**

Студент групи ШІ-12

Михальчук Антон Євгенійович

# **Тема роботи:**

# Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

# **Мета роботи:**

# Дослідити основи роботи з файлами у мові програмування C++, зокрема розглянути та освоїти принципи роботи з текстовими та бінарними файлами. Опанувати операції введення та виведення символів і рядкових змінних у файл, а також ознайомитися зі стандартною бібліотекою C++ для роботи з файлами. Навчитися створювати власні бібліотеки та використовувати їх у проектах, організовуючи код для повторного використання та покращення його структури.

# **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 стрічки та файли.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.09.2024
  + Звершення опрацювання теми: 12.11.2024

# **Виконання роботи:**

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

Завдання №1 VNS Lab 4 Варіант: 9

* Деталі завдання:

Задано рядок, що складається із символів. Символи поєднуються в слова.

Слова одне від одного відокремлюються одним або декількома пробілами.

Наприкінці тексту ставиться крапка.

Надрукувати всі слова-паліндроми, які є в цьому рядку

Завдання №2 VNS Lab 8 Варіант: 9

* Деталі завдання:

Структура "Пацієнт":

- прізвище, ім'я, по батькові;

- домашня адреса;

- номер медичної карти;

- номер страхового поліса.

Знищити елемент із заданим номером медичної карти, додати 2 елементи в

початок файлу.

Завдання №3 VNS Lab 9 Варіант: 9

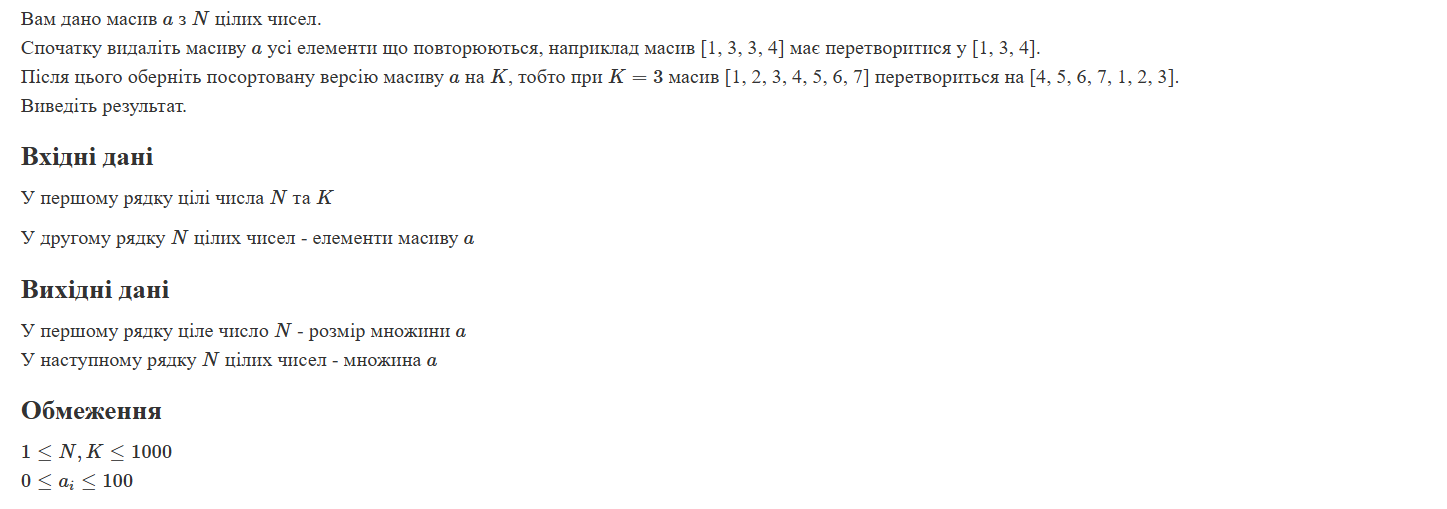
* Деталі завдання:

1) Скопіювати з файлу F1 у файл F2 всі рядки, які містять тільки одне слово.

2) Знайти найдовше слово у файлі F2.

Завдання №4 Algotester Lab 4 Варіант: 2

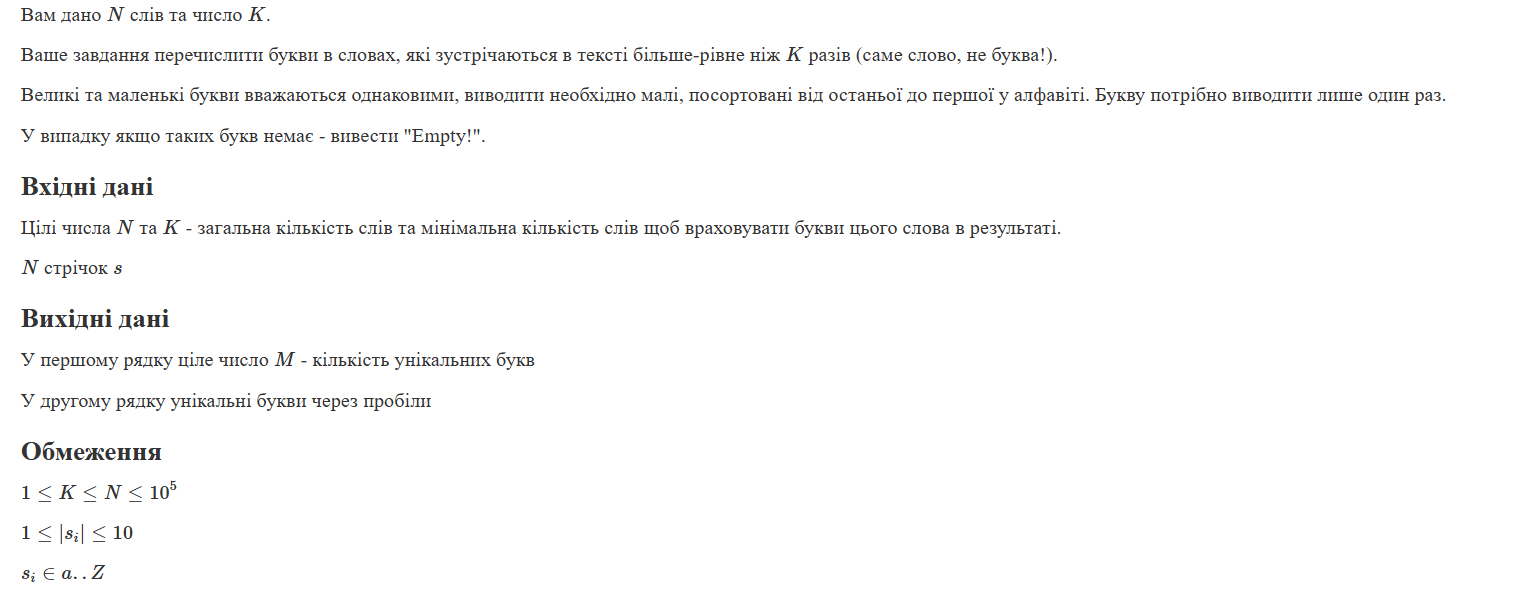
* Деталі завдання:



**Пам’ятайте, ви маєте написати 2 варіанти розвязку, один з використанням засобів STL (std::unique, std::sort, std::rotate), інший зі своєю реалізацією.**

Завдання №5 Algotester Lab 6 Варіант: 1

* Деталі завдання:



Завдання №5 Class Practice Task

* Деталі завдання:

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.

enum FileOpResult { Success, Failure, … };

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

Умови задачі:

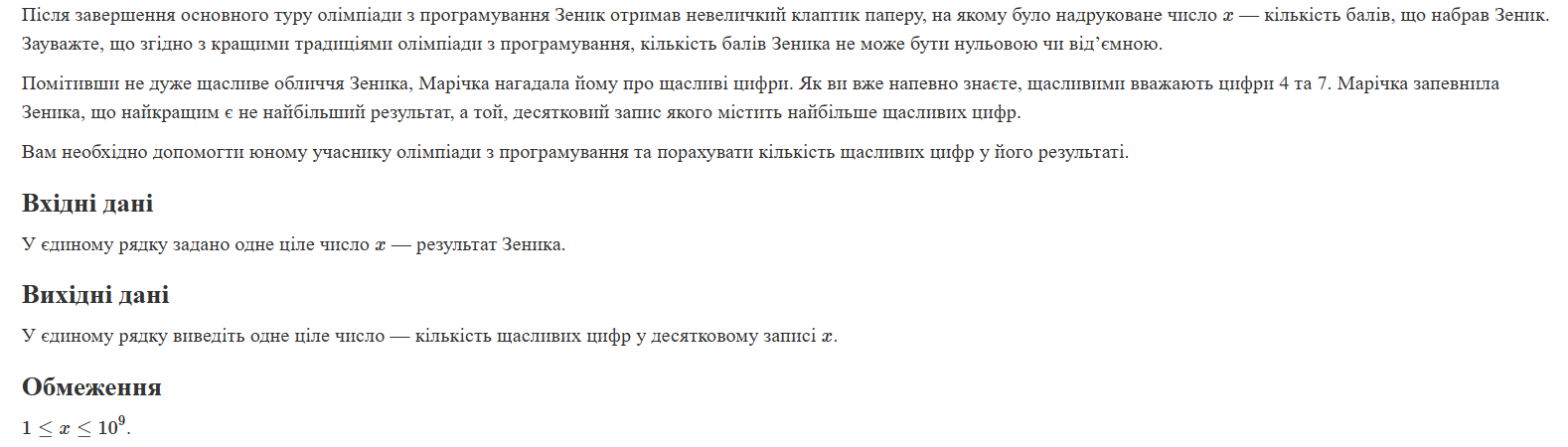
- копіювати вміст файла з ім’ям file\_from у файл з ім’ям file\_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів

- 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 №6 Self Practice Task Щасливий результат

* Деталі завдання:



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

Програма №1 VNS Lab 6 Варіант: 9

* Планований час на реалізацію: 1 год.

Програма №2 VNS Lab 8 Варіант: 9

* Планований час на реалізацію: 1 год.

Програма №3 VNS Lab 8 Варіант: 9

* Планований час на реалізацію: 1 год.

Програма №4 Algotester Lab 3 Варіант: 2

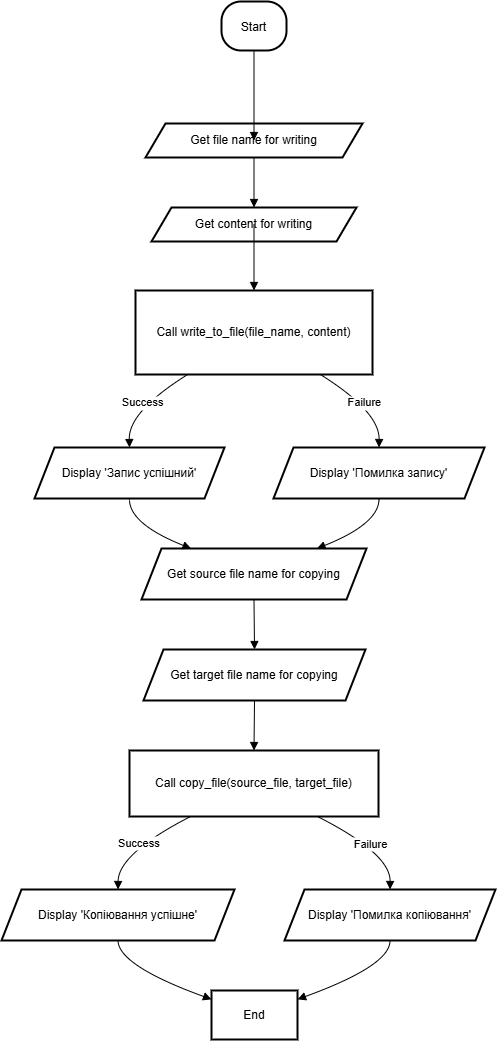
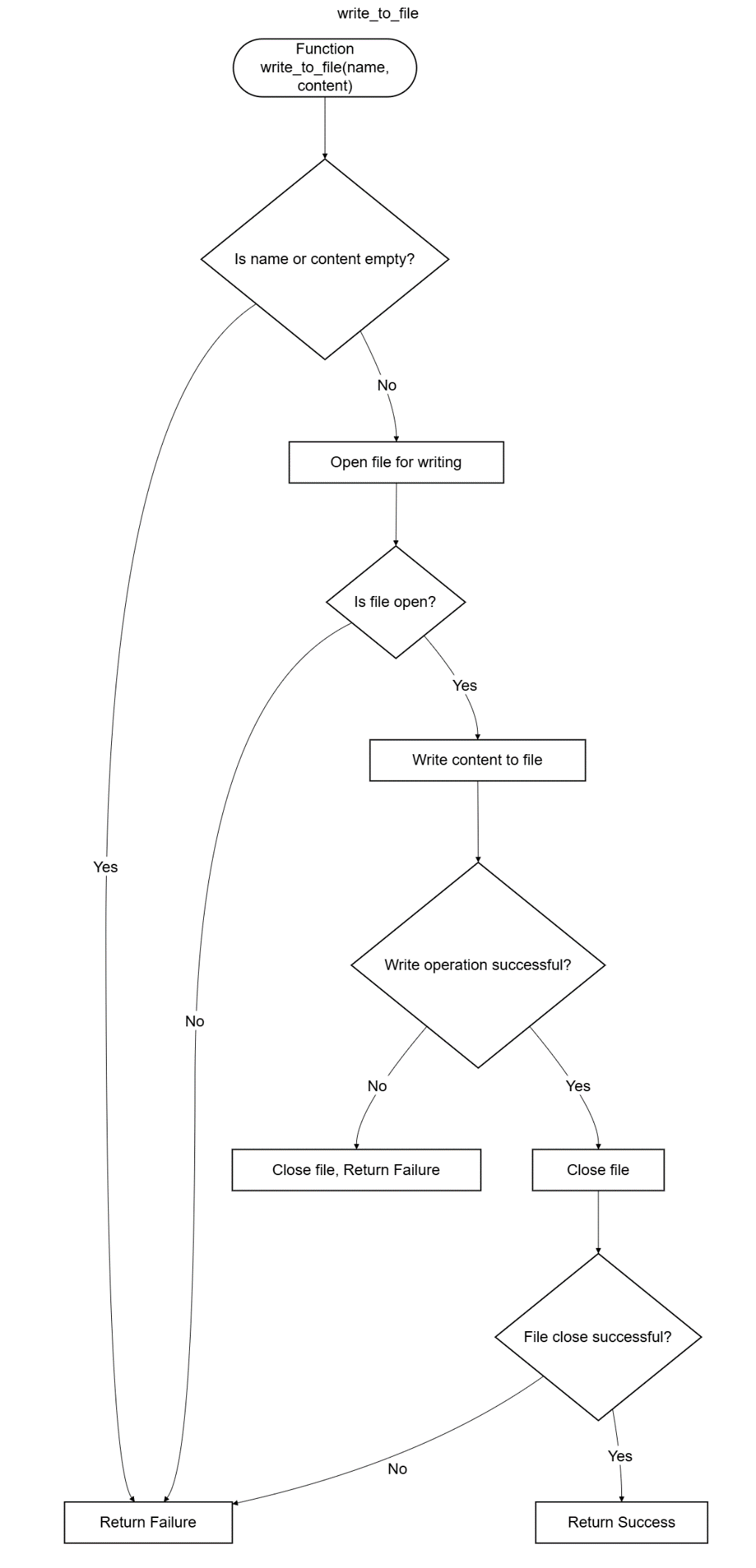
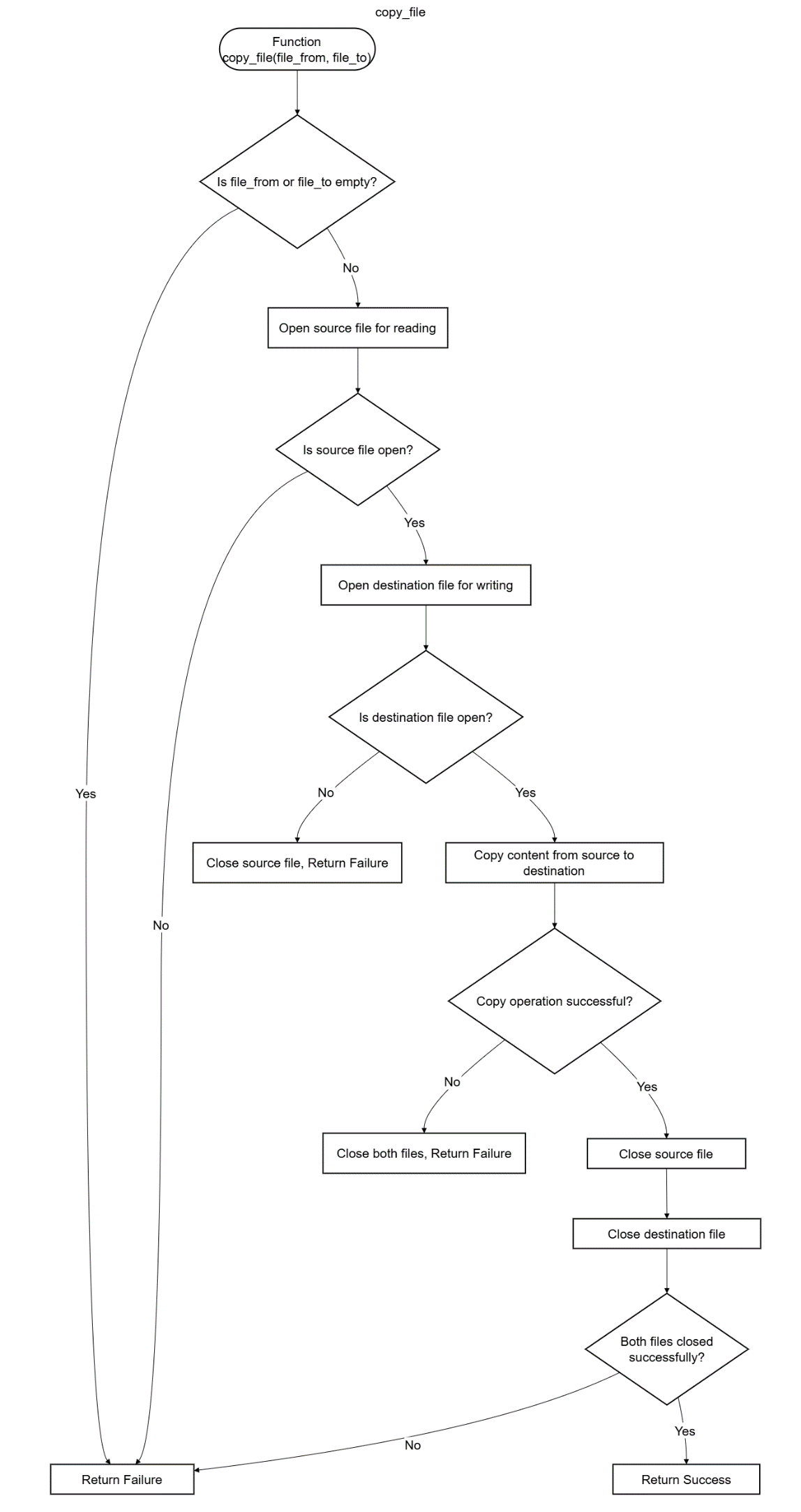
* Планований час на реалізацію: 30 хв.

Програма №5 VNS Algotester Lab 6 Варіант: 1

* Планований час на реалізацію: 50 хв.

Програма №5 Class Practice Task

* Блок-схема



* Планований час на реалізацію: 90 хв.

Програма №6 Self Practice Task Щасливий результат

* Планований час на реалізацію: 10 хв

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

Завдання №1

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-92afdec32338c21e40ccb123a692a4d858248aa262b93e76a98f8e4a1ecb7a9c>

#include <iostream>

#include <random>

using namespace std;

int main()

{

    int elementToDelete, indexOfElementToDelete = -1, k, size;

    int arr[100], newArr[100];

    cout << "Enter the size of array: ";

    cin >> size;

    random\_device rd;

    mt19937 gen(rd());

    uniform\_int\_distribution<int> distr(1, 100);

    for (int i = 0; i < size; ++i)

    {

        arr[i] = distr(gen);

    }

    for (int i = 0; i < size; ++i)

    {

        cout << arr[i] << " ";

    }

    cout << endl;

    cout << "Enter the value you want to delete: ";

    cin >> elementToDelete;

    for (int i = 0; i < size; ++i)

    {

        if (arr[i] == elementToDelete)

            indexOfElementToDelete = i;

    }

    if (indexOfElementToDelete != -1)

    {

        for (int i = indexOfElementToDelete; i < size; ++i)

        {

            arr[i] = arr[i + 1];

        }

        --size;

    }

    else

    {

        cout << "No value to delete!" << endl;

    }

    cout << "How many elements shall the array be shifted to the right: ";

    cin >> k;

    int newIndex;

    for (int i = 0; i < size; ++i)

    {

        newIndex = (i + k) % size;

        if (newIndex < 0)

        {

            newIndex = size + newIndex;

        }

        newArr[newIndex] = arr[i];

    }

    for (int i = 0; i < size; ++i)

    {

        cout << newArr[i] << " ";

    }

    cout << endl;

    return 0;

}

Завдання №2

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-d83e20e4ef30c0f1fd7f0c74bbf46dbb56310b423dcddd76f4715906f89f86d1>

#include <iostream>

#include <random>

using namespace std;

int sumOfBelowNumbers(int matrix[5][5])

{

    int sum = 0;

    for (int i = 0; i < 5; i++)

    {

        for (int j = 0; j < i; j++)

        {

            sum += matrix[i][j];

        }

    }

    return sum;

}

int main()

{

    int matrices[10][5][5];

    int sums[10];

    random\_device rd;

    mt19937 gen(rd());

    uniform\_int\_distribution<int> distr(-100, 100);

    for (int i = 0; i < 10; i++)

    {

        for (int j = 0; j < 5; j++)

        {

            for (int k = 0; k < 5; k++)

            {

                matrices[i][j][k] = distr(gen);

            }

        }

        sums[i] = sumOfBelowNumbers(matrices[i]);

    }

    int max = sums[0];

    for (int i = 1; i < 10; i++)

    {

        if (sums[i] > max)

            max = sums[i];

    }

    cout << max;

    return 0;

}

Завдання №3

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-cf6eded9566285684642f3dd4f355a3148e36137ff589f0d872c47ed0f145509>

#include <iostream>

#include <vector>

using namespace std;

int main()

{

    int n;

    cin >> n;

    vector<int> numbers(n);

    for (int i = 0; i < n; i++)

    {

        cin >> numbers[i];

    }

    int indexLeft = 0, indexRight = n - 1;

    while (true)

    {

        if (indexRight - indexLeft == 1)

        {

            cout << indexLeft + 1 << ' ' << indexRight + 1 << endl;

            cout << "Stopped" << endl;

            break;

        }

        else if (indexRight - indexLeft == 0)

        {

            cout << indexLeft + 1 << ' ' << indexRight + 1 << endl;

            cout << "Collision" << endl;

            break;

        }

        else if (indexLeft > indexRight)

        {

            cout << indexLeft + 1 << ' ' << indexRight + 1 << endl;

            cout << "Miss" << endl;

            break;

        }

        indexLeft += numbers[indexLeft];

        indexRight -= numbers[indexRight];

    }

    return 0;

}

Завдання №4

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-555a142b7741ef9ecfef5ddfd85b93585ee9328e7c980c9f60df31447a5c82bb>

#include <iostream>

#include <vector>

using namespace std;

int main()

{

    int n, m;

    cin >> n;

    vector<int> a(n);

    for (int i = 0; i < n; i++)

    {

        cin >> a[i];

    }

    cin >> m;

    vector<int> b(m);

    for (int i = 0; i < m; i++)

    {

        cin >> b[i];

    }

    int intersection = 0, symmetricDifference = 0;

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < m; j++)

        {

            if (a[i] == b[j])

            {

                ++intersection;

            }

        }

    }

    for (int i = 0; i < n; i++)

    {

        bool notFound = true;

        for (int j = 0; j < m; j++)

        {

            if (a[i] == b[j])

            {

                notFound = false;

                b[j] = -1;

            }

        }

        if (notFound)

        {

            ++symmetricDifference;

        }

    }

    for (int i = 0; i < m; i++)

    {

        bool notFound = true;

        for (int j = 0; j < n; j++)

        {

            if (b[i] == a[j])

            {

                notFound = false;

            }

        }

        if (notFound)

        {

            ++symmetricDifference;

        }

    }

    cout << intersection << endl

         << symmetricDifference;

    return 0;

}

Завдання №5

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-84167e79a2609ef992bc476033248f7366f932435119a75b3ac550405ef0164a>

#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 false;

    }

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

}

bool isPalindrome(int num)

{

    int original = num, reversedNumber = 0, digit;

    while (num > 0)

    {

        digit = num % 10;

        reversedNumber = reversedNumber \* 10 + digit;

        num = (num - digit) / 10;

    }

    return original == reversedNumber;

}

int main()

{

    string word;

    cout << "Enter a word: ";

    cin >> word;

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

    {

        cout << word << " is a palindrome." << endl;

    }

    else

    {

        cout << word << " is not a palindrome." << endl;

    }

    int number;

    cout << "Enter a number: ";

    cin >> number;

    if (isPalindrome(number))

    {

        cout << number << " is a palindrome." << endl;

    }

    else

    {

        cout << number << " is not a palindrome." << endl;

    }

    return 0;

}

Завдання №6

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/266/files#diff-89b746c0b22a94d4a6cec53f873c9c8cd4e7c0da325cb2ac7af4fd906a84ce52>

#include <iostream>

#include <vector>

#include <cmath>

using namespace std;

int main()

{

    int n;

    cin >> n;

    vector<int> oddNumbers, evenNumbers;

    int t;

    for (int i = 0; i < n; i++)

    {

        cin >> t;

        if (t % 2 == 0)

        {

            evenNumbers.push\_back(t);

        }

        else

        {

            oddNumbers.push\_back(t);

        }

    }

    int sizeOdd = oddNumbers.size(), sizeEven = evenNumbers.size();

    if (sizeOdd == 0 || sizeEven == 0)

    {

        cout << -1;

    }

    else if (abs(sizeOdd - sizeEven) <= 1)

    {

        if (sizeOdd >= sizeEven)

        {

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

            {

                cout << oddNumbers[i] << ' ' << evenNumbers[i] << ' ';

            }

            if (sizeOdd > sizeEven)

                cout << oddNumbers[sizeOdd - 1];

        }

        else

        {

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

            {

                cout << evenNumbers[i] << ' ' << oddNumbers[i] << ' ';

            }

            if (sizeEven > sizeOdd)

                cout << evenNumbers[sizeEven - 1];

        }

    }

    else

    {

        cout << -1;

    }

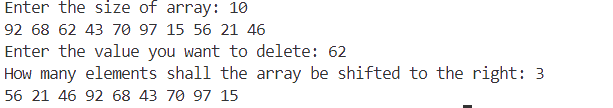
    return 0;

}

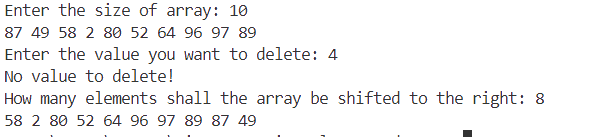
## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1

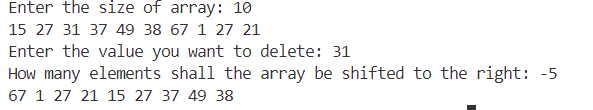
Створив псевдодинамічний масив за допомогою оголошення його розмірності, як 100. Та утворив псевдозмінні межі.



Також додав вивід, коли елемент для видалення не виявлено:



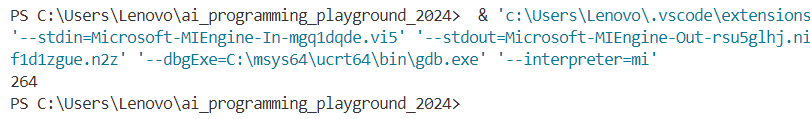
Додав можливість зсунення вліво за допомогою від’ємних значень



Час затрачений на виконання завдання: 35 хв.

Завдання №2

Помістив 10 матриць 5\*5 в один масив.  
Заповнив їх випадковими значеннями за допомогою псевдовипадкових чисел.



Час затрачений на виконання завдання: 40 хв.

Завдання №3



Завдання №4

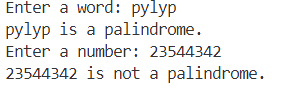


Завдання №5

Ввід спершу слова, а потім числа.

Функція для рядків є рекурсивною, що постійно звужує діапазон рядка та перевіряє, чи є символи на кінцях однаковими.

Перевантажена функція для цілочисельних значень використовує спосіб ділення на остачу та цілочисельне ділення для пошуку протилежного числа. Після чого звіряємо, чи є число паліндромом.



Завдання №6

Розбиваємо масив на два підмасиви: один для парних чисел, інший для непарних.

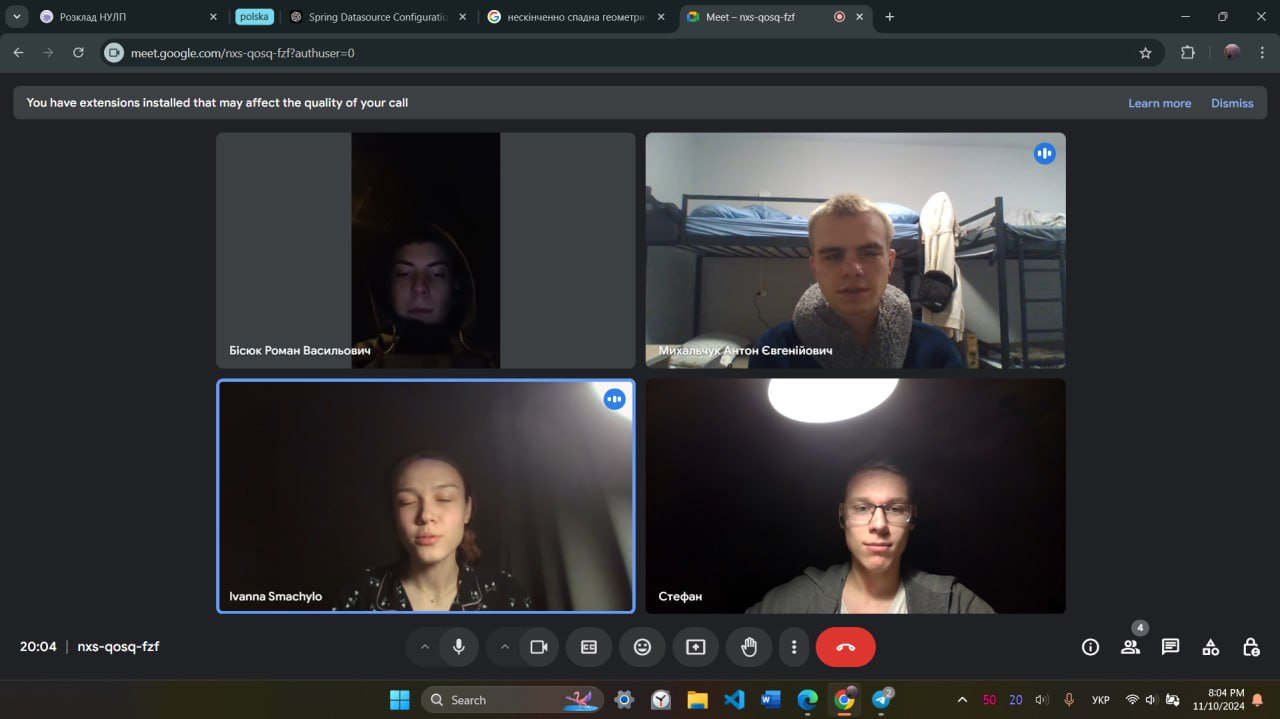
Якщо кількість парних або непарних чисел дорівнює нулю, виводимо -1.

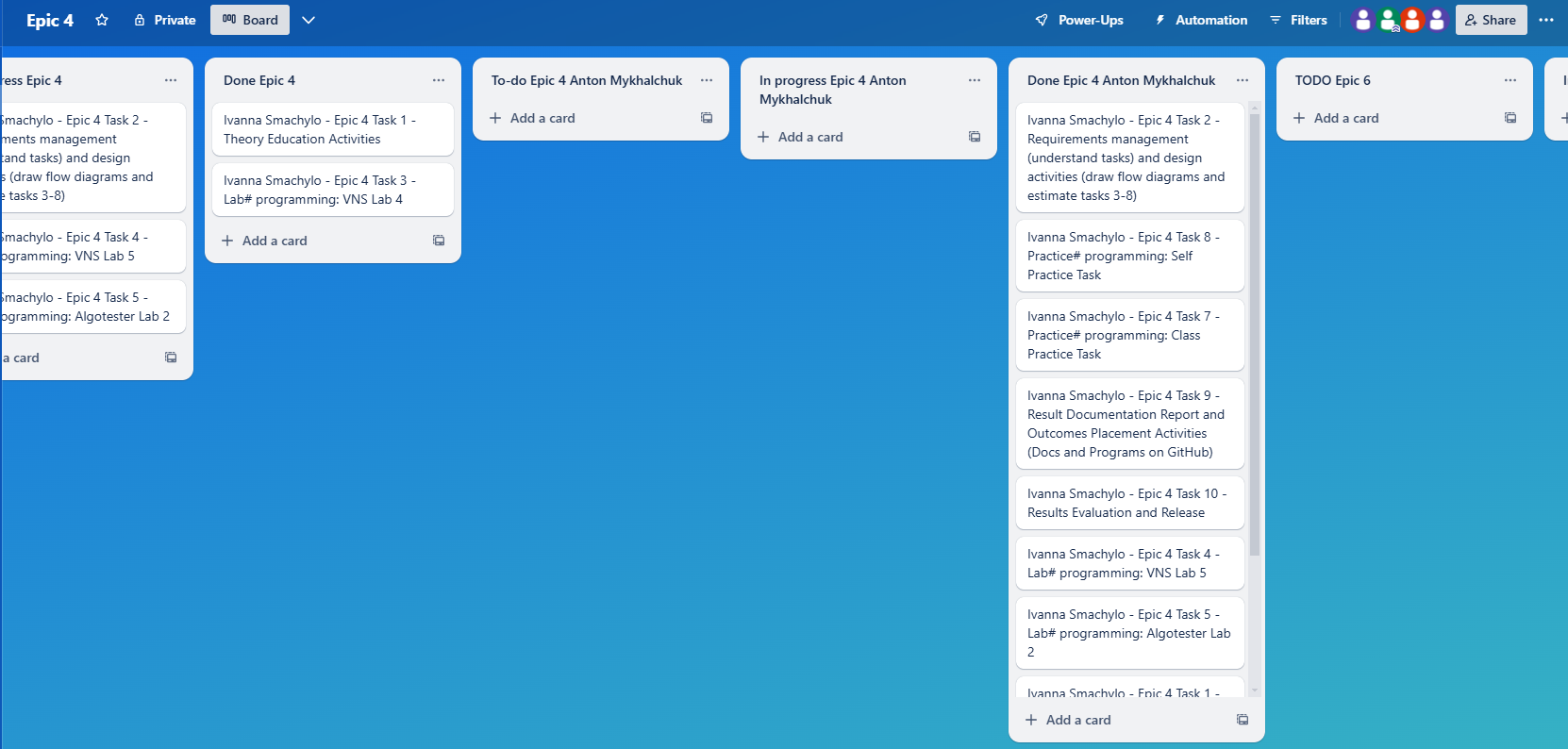
Якщо кількість елементів кожного типу відрізняється не більш аніж на 1, то можемо чергувати їх для досягнення бажаного результату.

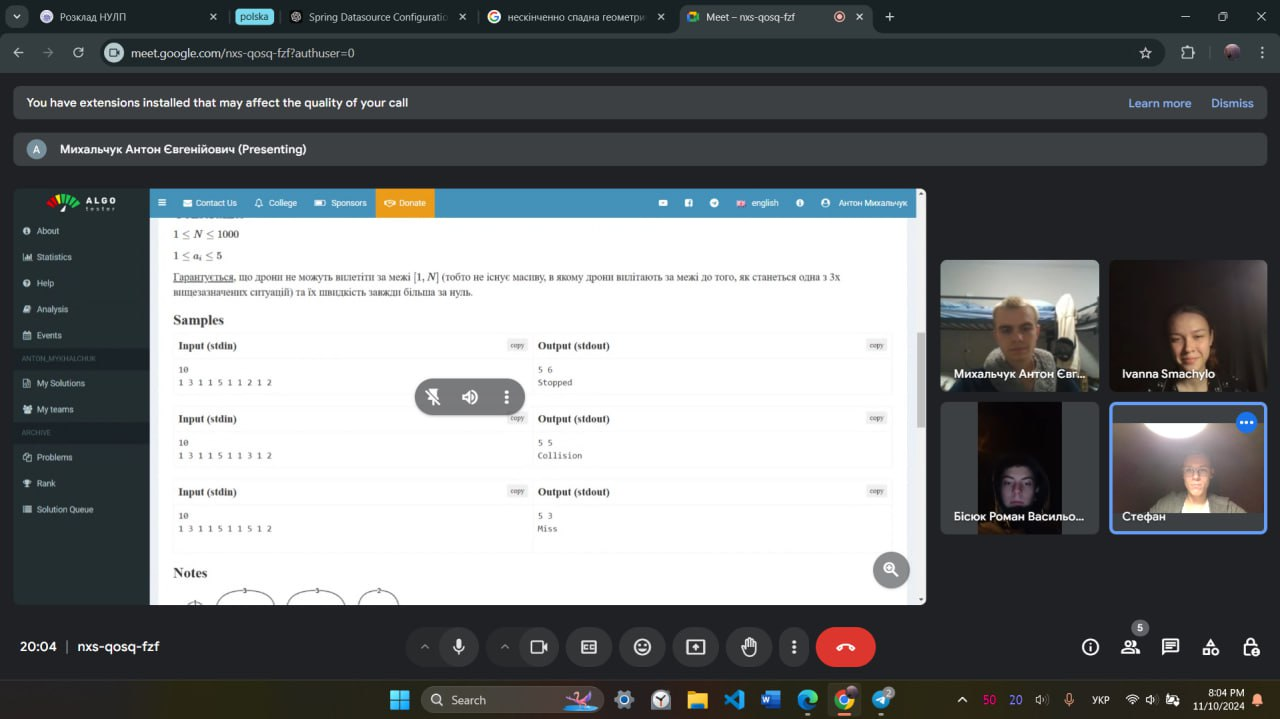
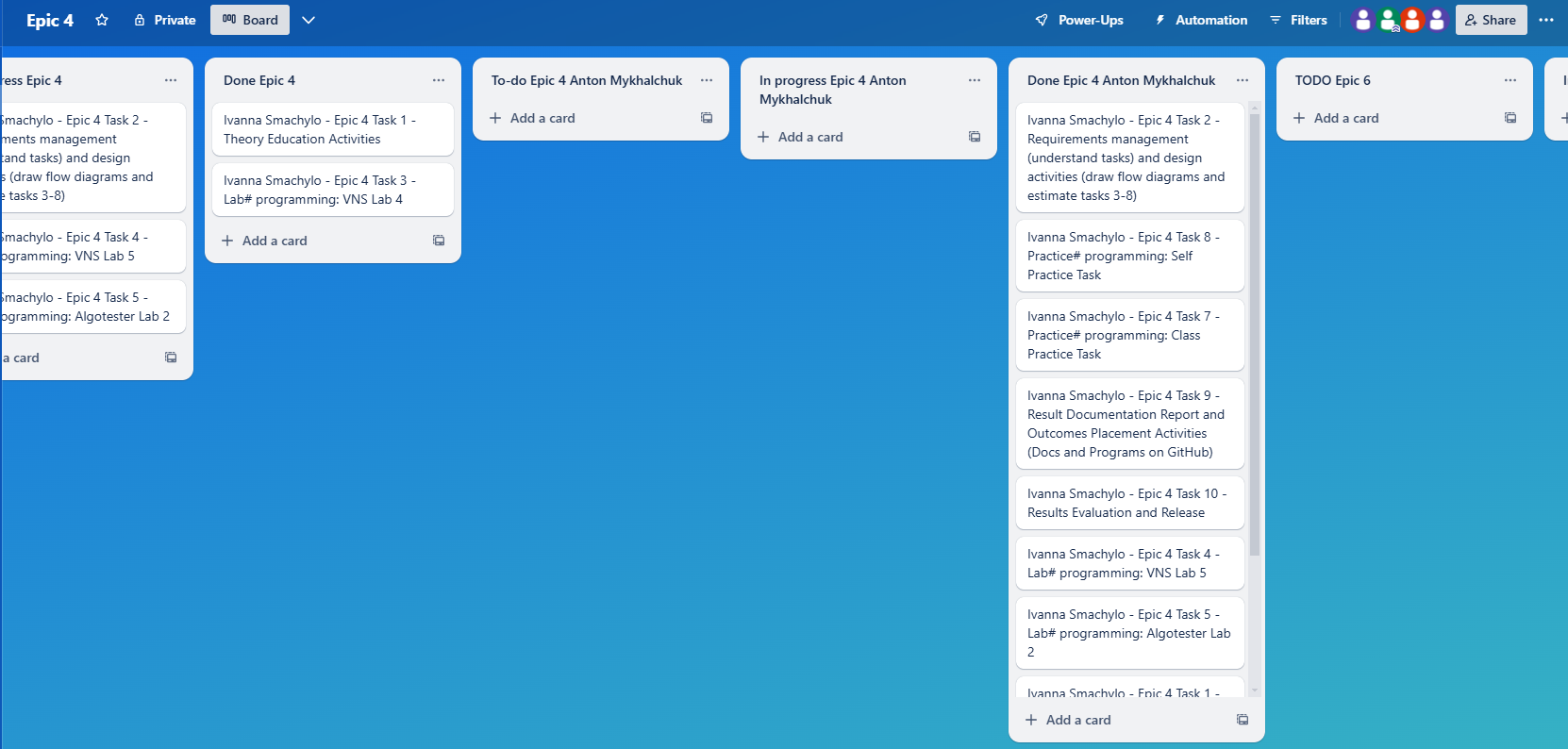
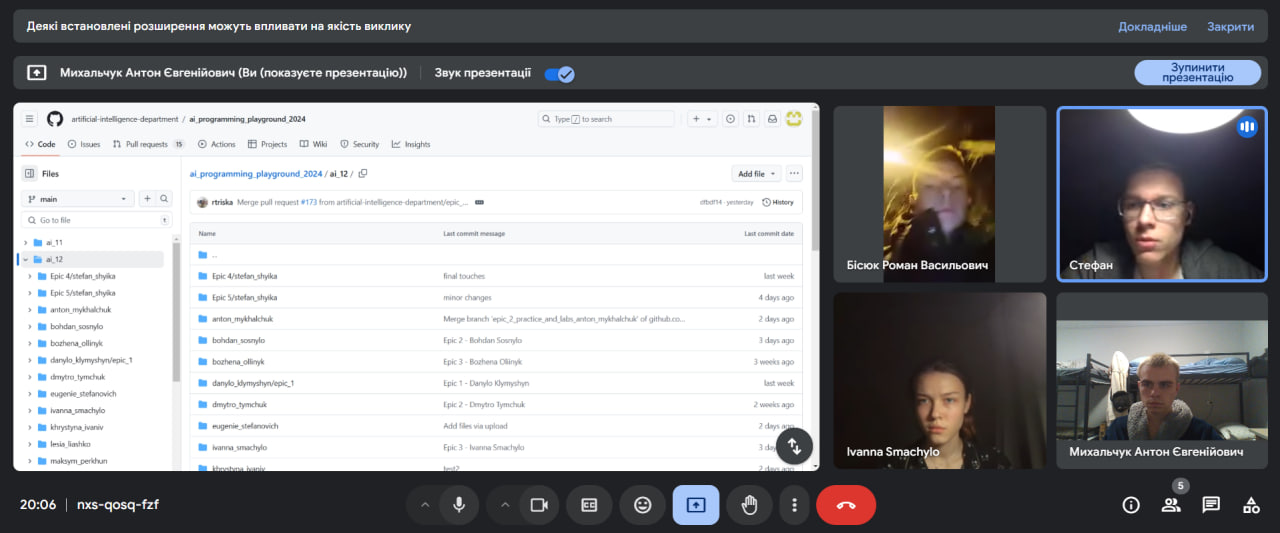


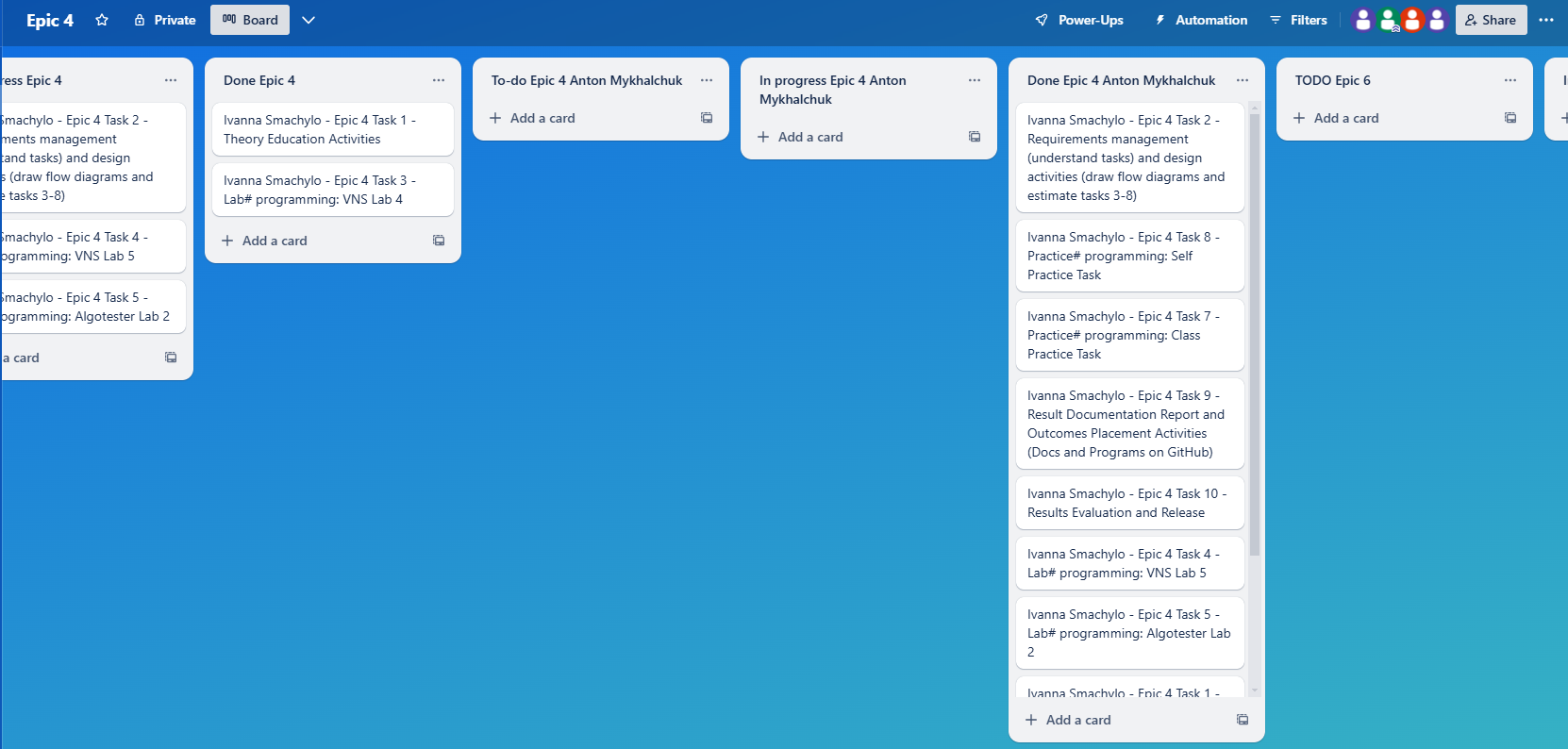
## **6. Кооперація з командою:**

* 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)
* 



# **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++.