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»

***Виконала:***

студентка групи ШІ-13

Кисців Каріна Володимирівна

# **Тема роботи:**

# Аналіз, реалізація та оптимізація алгоритмів обробки одновимірних та двовимірних масивів в контексті простих структур даних.

# **Мета роботи:**

# Мета даної роботи полягає в опрацюванні таких понять як прості структури даних, різні види масивів(одновимірні та двовимірні) та розглянути використання стандартних алгоритмів обробки.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

- Тема №1: Прості структури даних

- Тема №2: Одновимірні та двовимірні масиви

- Тема №3: Масиви як параметри функцій

- Тема №4: Стандартні операції обробки масивів

- Тема №5: Базові алгоритми сортування масивів

- Тема №6\* Linked List, структури та класи, конструктори та деструктори

1. Індивідуальний план опрацювання теорії:

Тема №1: Прості структури даних

* + Джерела Інформації
    - <https://acode.com.ua/urok-64-struktury/>
    - Конспект лекцій з дисципліни «Алгоритмізація і програмування. Частина 1» Л.І.Мочурад
  + Що опрацьовано:
    - Оголошення і визначення структур, їх ініціалізація
    - Основні операції на структурах даних
    - Різні лінійні та нелінійні структури даних, такі як масив, рядок, зв’язаний список, стек, черга.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 28.11.2023
  + Звершення опрацювання теми: 28.11.2023
* Тема №2: Одновимірні масиви та двовимірні масиви
  + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація і програмування. Частина 1» Л.І.Мочурад
    - <https://acode.com.ua/urok-77-masyvy/>
  + Що опрацьовано:
    - З вищеподаних джерел теоретично опрацьовано що таке одновимірні та двовимірні масиви, як вони зберігаються в пам’яті комп’ютера, та як їх оголосити, вказавши тип даних, назву та розмір масиву.
    - Виконання операцій над одно- та двовимірними масивами.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 28.11.2023
  + Звершення опрацювання теми: 28.11.2023
* Тема №3: Масиви як параметри функцій
  + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація і програмування. Частина 1» Л.І.Мочурад
    - <https://acode.com.ua/urok-78-fiksovani-masyvy/>
  + Що опрацьовано:
    - Передача масивів за значеннм
    - Передача багатовимірних масивів
  + Статус: Ознайомлена
  + Початок опрацювання теми: 29.11.2023
  + Звершення опрацювання теми: 29.11.2023
* Тема №4: Стандартні операції обробки масивів
  + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація і програмування. Частина 1» Л.І.Мочурад

Що опрацьовано:

* + - З поданих вище матеріалів опрацьовано такі операції як: перегляд масиву по одному елементу, формування псевдодинамічного масиву
  + Статус: Ознайомлена
  + Початок опрацювання теми: 29.11.2023
  + Звершення опрацювання теми: 29.11.2023

Тема №5: Базові алгоритми сортування масивів

* + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація і програмування. Частина 1» Л.І.Мочурад
    - <https://acode.com.ua/urok-80-sortuvannya-masyviv-metodom-vyboru/>
  + Що опрацьовано:
    - Теоретично опрацьовано різні типи сортування масивів такі як сортування бульбашкою, сортування вибором, сортування вставками.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 29.11.2023
  + Звершення опрацювання теми: 29.11.2023

Тема №6\* Linked List, структури та класи, конструктори та деструктори

* + Джерела Інформації:
    - [https://acode.com.ua/urok-124-konstruktory/](https://acode.com.ua/urok-124-konstruktory/%20)
    - <https://acode.com.ua/urok-128-destruktory/>
    - <https://www.codementor.io/@codementorteam/a-comprehensive-guide-to-implementation-of-singly-linked-list-using-c_plus_plus-ondlm5azr>
  + Що опрацьовано:
    - Теоретично опрацьовано Linked List, структури та класи, конструктори та деструктори, для чого вони необхідні, як використовувати їх у власних завданнях
  + Статус: Ознайомлена
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 29.11.2023

# **Виконання роботи:**

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

#### Завдання №1 VNS Lab 4 - Task 1

* Варіант завдання: **18**
* Деталі завдання: 1) Реалізувати з використанням масиву однонаправлене кільце (перегляд можливий зліва направо, від останнього елемента можна перейти до першого).

2) Роздрукувати отриманий масив, починаючи з К-ого елемента і до К-1.

3) Додати в кільце перший і останній елементи.

4) Знищити з кільця парні елементи.

5) Роздрукувати отриманий масив, починаючи з К-ого елемента і до К-1.

- Важливі деталі: Для виконання даного завдання, потрібно було додатково опрацювати Linked list та поінтери.

#### Завдання №2 VNS Lab 5 - Task 1

* Варіант завдання: **18**
* Деталі завдання: Задано двовимірний масив N x N. Послідовно розглядаються квадратні підмасиви, правий верхній елемент яких лежить на бічній діагоналі. У кожному такому підмасиві перебуває максимальний елемент. Шляхом перестановок рядків і стовпців (повністю) елемент треба перемістити в правий верхній кут підмасиву. Перевірити чи вийшла на бічній діагоналі спадаюча послідовність елементів.

#### Завдання №3 Algotester Lab 2

Варіант завдання: **3**

* Дано масив цілих чисел розміром N, на першій та останній клітинці розміщено по дрону. Вони одночасно взлітають. На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться. Тобто лівий дрон у першу секунду з клітинки з індексом 11 перелетить у клітинку з індексом a1, тобто його наступна позиція рахується як поточна позиція + число у поточній позиції. Правий робить аналогічно в протилежну сторону. Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якшо 2 дрони опиняються в одній клітинці - ви виводите **Collision**.

Якщо лівий дрон опиниться справа від правого - це **Miss**

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках a[i] та a[i+1] - виведіть **Stopped**

- Важливі деталі: Гарантується, що дрони не можуть вилетіти за межі [1,N] (тобто не існує масиву, в якому дрони вилітають за межі до того, як станеться одна з 3х вищезазначених ситуацій) та їх швидкість завжди більша за нуль.

#### Завдання №4 Algotester Lab 3

* Варіант завдання: **3**
* Деталі завдання: зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

#### Завдання №5 Class Practice Palindrome

* Деталі завдання: Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.
* Важливі деталі: Використати рекурсію для перевірки чи слово/набір чисел є паліндромом. Використати перевантаження функції.

#### Завдання №6 Self-Practice (Цікавий масив)

- Деталі завдання: Вам задано масив з n елементів a[i]. Необхідно перевірити, чи існує таке значення k(1≤k<n), що сума перших k елементів масиву рівна сумі останніх n−k елементів.

Іншими словами, необхідно визначити, чи можна розбити масив на дві частини: початок і кінець так, щоб сума чисел в обох частинах співпадала. Вхідні дані необхідно зчитувати з стандартного потоку вводу (з консолі). Вихідні дані: Виведіть TAK, якщо масив можна розбити на дві частини з однаковою сумою, або NI, якщо не можна.

Виводити потрібно у стандартний потік виводу (в консоль). Ніяких додаткових даних виводити не потрібно.

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

#### Програма № 1 VNS Lab 4 - Task 1

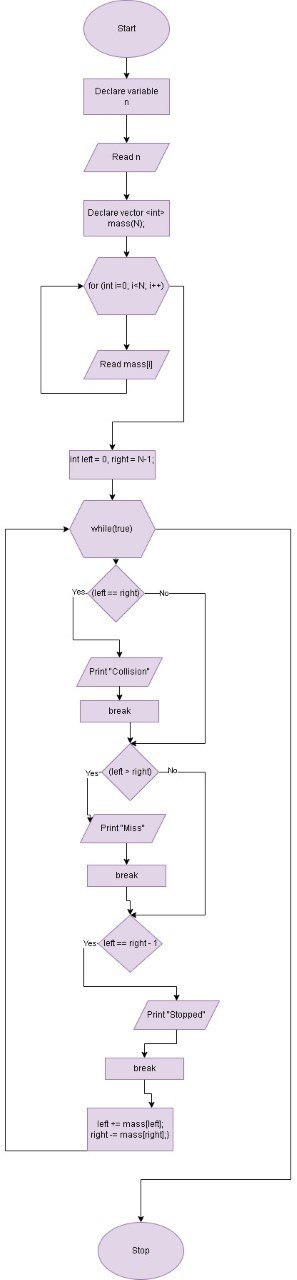
* Планований час на реалізацію: 3 дні

#### Програма №2 VNS Lab 5 - Task 1

* Планований час на реалізацію: 2 дні

**Програма №4 Algotester Lab 3**

* Планований час на реалізацію 1 день



*Рисунок 1 Блок-схема до програми №4 Algotester Lab 3v3*

#### Програма №4 Algotester Lab 3

* Планований час на реалізацію 1 день

#### Програма №5 Class Practice

* Планований час на реалізацію: 1 день

#### Програма №6 Self-Practice

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

Посилання на pull-request: [https://github.com/artificial-intelligence-department/](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/707/files#diff-2b904b457be1d10af6db5e13dd0cb4eeeacf73de9bdb363f09f640bb6e667c9d)

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/707/files#diff-2b904b457be1d10af6db5e13dd0cb4eeeacf73de9bdb363f09f640bb6e667c9d)

#include <iostream>

using namespace std;

struct Node

    {

        int value;

        int index;

        Node\* next;

        Node\* previous;

    };

    class LinkedList

    {

        private:

        int sizeL {0};

        Node\* head;

        Node\* end;

        public:

        void insertLeft(int value) {

            sizeL++;

            Node\* newNode = new Node;

            newNode->value = value;

                 if(head==nullptr) {

                    head = newNode;

                    end = newNode;

                    newNode->next = newNode;

                    newNode->previous = newNode;

                } else {

                    newNode->next = head;

                    head = newNode;

                }

        }

        void insert(int value){

            sizeL++;

            Node\* newNode = new Node;

            newNode->value = value;

            newNode->next = nullptr;

                if(head==nullptr){

                    head = newNode;

                    end = newNode;

                    newNode->next = newNode;

                    newNode->previous = newNode;

                } else {

                    newNode->next = end;

                    newNode->previous = head;

                    end->next = newNode;

                    end = newNode;

                    end->next = head;

                }

            }

            LinkedList(int size) {

                head = nullptr;

                end = nullptr;

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

            {

                insert(i+1);

            }

            }

            ~LinkedList() {

                Node\* temp;

                while(head != nullptr) {

                    temp= head;

                    head = head->next;

                    delete temp;

                }

            }

            void displayFromKToKMinusOne(int K) {

    Node\* temp = head;

    for (int i = 0; i < K; i++) {

        temp = temp->next;

    }

    for (int i = 0; i < sizeL; i++) {

        cout << temp->value << " ";

        temp = temp->next;

    }

    cout << endl;

}

    void delEven() {

    Node\* tr = new Node;

    tr->value = -1;

    tr->next = head;

    tr->previous = end;

    head->previous = tr;

    end->next = tr;

    Node\* temp = tr;

    do {

        if (temp->next->value % 2 == 0) {

            sizeL--;

            Node\* toDelete = temp->next;

            temp->next = toDelete->next;

            toDelete = temp;

            if (toDelete == end) {

            end = temp;

            }

            delete toDelete;

        } else {

            temp = temp->next;

        }

    } while (temp != tr);

    head = tr->next;

     delete tr;

}

         void display() {

            Node\* temp = head;

            while (temp != end) {

                cout << temp->value << " ";

                temp = temp->next;

            }

            cout << endl;

        }

    };

int main(int argc, char const \*argv[]) {

    LinkedList List{10};

    List.displayFromKToKMinusOne(5);

    List.insertLeft(5);

    List.insert(7);

    List.displayFromKToKMinusOne(0);

    List.delEven();

    List.displayFromKToKMinusOne(0);

    return 0;

}

*Рисунок 2 Програмний код №1 VNS Lab 4 - Task 1*

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/707/files#diff-2b904b457be1d10af6db5e13dd0cb4eeeacf73de9bdb363f09f640bb6e667c9d)

#include <iostream>

using namespace std;

void move\_max(int\*\* arr, int n, int i) {

    // Перевіряємо, чи не досягли ми кінця бічної діагоналі

    if (i >= n) return;

    // Знаходимо максимальний елемент в підмасиві, який починається з (i, i)

    int max = arr[i][i];

    int max\_row = i;

    int max\_col = i;

    for (int r = i; r < n; r++) {

        for (int c = i; c < n; c++) {

            if (arr[r][c] > max) {

                max = arr[r][c];

                max\_row = r;

                max\_col = c;

            }

        }

    }

    if (max\_row != i || max\_col != n-1-i) {

        for (int c = i; c < n; c++) {

            int temp = arr[i][c];

            arr[i][c] = arr[max\_row][c];

            arr[max\_row][c] = temp;

        }

        for (int r = i; r < n; r++) {

            int temp = arr[r][n-1-i];

            arr[r][n-1-i] = arr[r][max\_col];

            arr[r][max\_col] = temp;

        }

    }

    move\_max(arr, n, i+1);

}

// Функція, яка перевіряє, чи вийшла на бічній діагоналі спадаюча послідовність елементів

bool is\_decreasing(int\*\* arr, int n) {

    // Перебираємо всі елементи бічної діагоналі, починаючи з другого

    for (int i = 1; i < n; i++) {

        // Якщо поточний елемент більший або рівний попередньому, то повертаємо false

        if (arr[i][n-1-i] >= arr[i-1][n-i]) {

            return false;

        }

    }

    return true;

}

void print\_array(int\*\* arr, int n) {

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            cout << arr[i][j] << " ";

        }

        cout << "\n";

    }

}

int main() {

    int n;

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

    cin >> n;

    int\*\* arr = new int\*[n];

    for (int i = 0; i < n; i++) {

        arr[i] = new int[n];

    }

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

           cin >> arr[i][j] ;

        }

    }

    cout << "start"<<endl;

    print\_array(arr, n);

    move\_max(arr, n, 0);

    cout << "end"<<endl;

    print\_array(arr, n);

    if (is\_decreasing(arr, n)) {

        cout << "decrease sequance";

    } else {

        cout << "no decrece sequance";

    }

    for (int i = 0; i < n; i++) {

        delete[] arr[i];

    }

    delete[] arr;

    return 0;

}

*Рисунок 3 Програмний код №2 VNS Lab 5 - Task 1*

Завдання №3 Algotester Lab 2

[Посилання на файл програми у пул-запиті GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/707/files#diff-e4339edac44f532227bc5cd7a8f4760a332b431c58c0b2751f256c169456816e)

#include <iostream>

#include <vector>

using namespace std;

int main() {

    int N;

    cin >> N;

    vector <int> mass(N);

    for(int i=0; i<N; i++){

        cin >> mass[i];

    }

    int left = 0, right = N-1;

    while(true) {

        if(left == right) {

            cout << left+1 << " " << right+1 << endl;

            cout << "Collision" << endl;

            break;

        }

        else if(left > right) {

            cout << left+1 << " " << right+1 << endl;

            cout << "Miss" << endl;

            break;

        }

        else if( left == right - 1) {

            cout << left+1 << " " << right+1 << endl;

            cout << "Stopped" << endl;

            break;

        }

            left += mass[left];

        right -= mass[right];}

    return 0;

}

*Рисунок 4 Програмний код №3 Algotester Lab 2v3*

Завдання №4 Algotester Lab 3v3

[Посилання на файл програми у пул-запиті GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/707/files#diff-8be82bac076fae8541201fa570980bbf8e21bf5be6e10617b7dad567c7ff3c68)

#include <string>

#include <iostream>

using namespace std;

int main(){

    string s;

    cin >> s;

    if (s.empty()) {

        cout << "String is empty." << endl;

        return 1;

    }

    string compressedS = "";

    char currentChar = s[0];

    int repNum = 1;

    for (size\_t i = 1; i < s.length(); ++i){

        if (s[i] == currentChar){

            repNum++;

        } else {

            compressedS += currentChar;

            if (repNum > 1){

                compressedS += to\_string(repNum);

            }

            currentChar = s[i];

            repNum = 1;

        }

    }

    compressedS += currentChar;

    if (repNum > 1){

        compressedS += to\_string(repNum);

    }

    cout << compressedS << endl;

    return 0;

}

*Рисунок 5 Програмний код №4 Algotester Lab 3v3*

Завдання №5 Class Practice Palindrome

[Посилання на файл програми у пул-запиті GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/707/files#diff-48a49e828548bcb2a5e37c83a4e80ce6f018f1965f8bd4bb6d6d6e31653ca3b5)

#include <iostream>

#include <vector>

#include <sstream>

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(const vector<int> &vec, int start, int end) {

    if (start >= end) {

        return true;

    }

    if (vec[start] != vec[end]) {

        return false;

    }

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

}

int main() {

    string inputStr;

    cout << "Enter a word: ";

    getline(cin >> ws, inputStr);

    if (isPalindrome(inputStr, 0, inputStr.length() - 1)) {

        cout << "The string is a palindrome.\n";

    } else {

        cout << "The string is not a palindrome.\n";

    }

    string inputNumbers;

    cout << "Enter numbers: ";

    getline(cin >> ws, inputNumbers);

    vector<int> numbers;

    istringstream iss(inputNumbers);

    int num;

    while (iss >> num) {

        numbers.push\_back(num);

    }

    if (isPalindrome(numbers, 0, numbers.size() - 1)) {

        cout << "The vector of numbers is a palindrome.\n";

    } else {

        cout << "The vector of numbers is not a palindrome.\n";

    }

    return 0;

}

*Рисунок 6 Програмний код №5 Class Practice Palindrome*

Завдання №6 Self practice (Цікавий масив)

[Посилання на файл програми у пул-запиті GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/707/files#diff-f8c84e57dba75bb2d9ea42b85de1192888862fe36831ba6059f00d177fb71fc4)

#include <iostream>

#include <vector>

int main() {

    int n;

    std::cin >> n;

    std::vector<int> arr(n);

    for (int i = 0; i < n; ++i) {

        std::cin >> arr[i];

    }

    int totalSum = 0;

    for (int i : arr) {

        totalSum += i;

    }

    int currentSum = 0;

    for (int k = 1; k < n; ++k) {

        currentSum += arr[k - 1];

        if (currentSum == totalSum - currentSum) {

            std::cout << "TAK\n";

            return 0;

        }

    }

    std::cout << "NI\n";

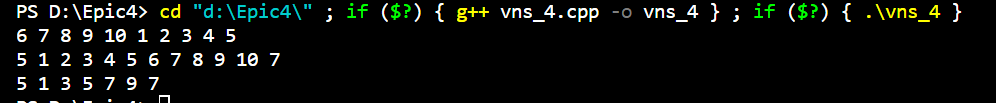
    return 0;

}

*Рисунок 7 Програмний код №6 Class Practice Palindrome*

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

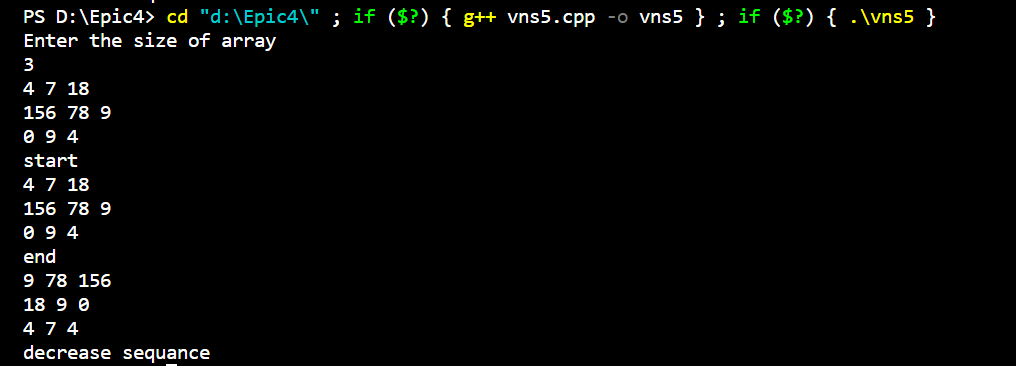
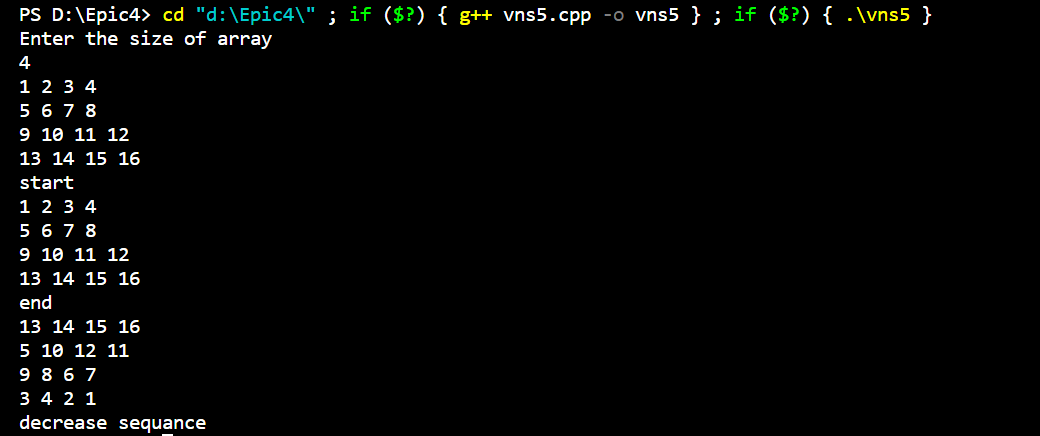
#### Завдання №1 VNS Lab 4 - Task 1. Реалізовано однонаправлене кільце за допомогою структур, класів, поняття Linked list та поінтерів.



*Рисунок 8 Результат виконання VNS Lab 4 - Task 1*

Час затрачений на виконання завдання – 3 дні

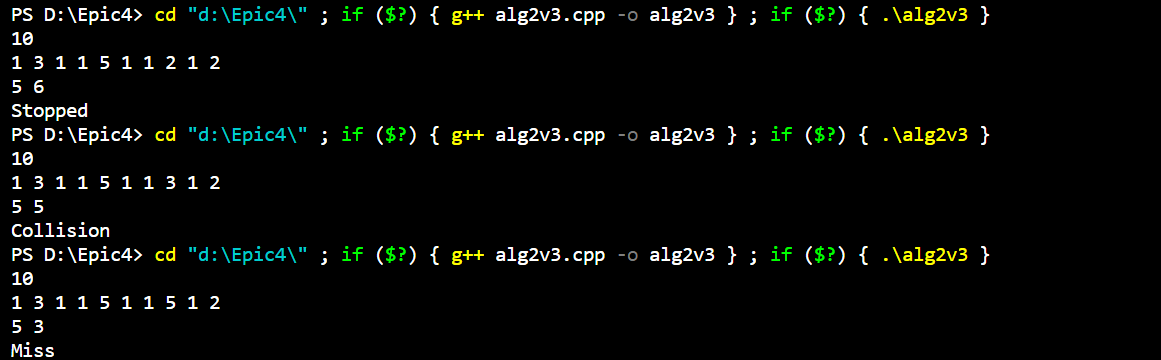
Завдання №2 VNS Lab 5 - Task 1. Для реалізації завдання спочатку створено функцію, яка знаходить максимальний елемент в квадратному підмасиві і переміщає його в правий верхній кут підмасиву за допомогою перестановок рядків і стовпців, з допомогою циклів та рекурсії. А також функцію, яка перевіряє, чи вийшла на бічній діагоналі спадаюча послідовність елементів



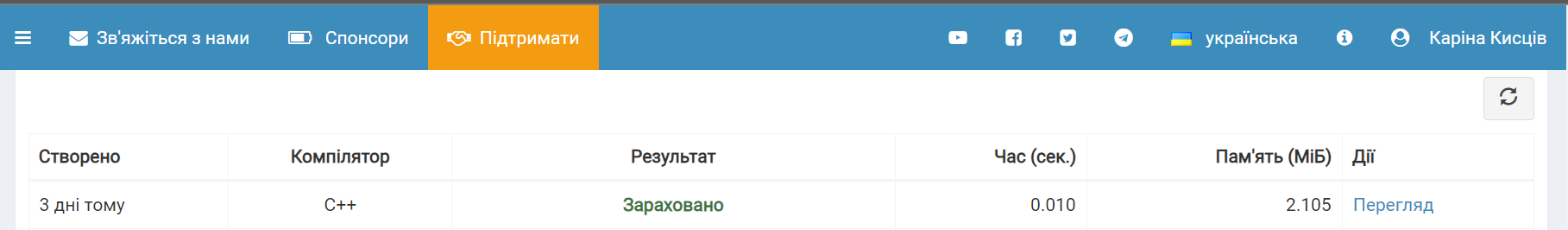
*Рисунок 9 Результат виконання VNS Lab 5 - Task 1*

Час затрачений на виконання завдання – 2 дні

**Завдання №3 Algotester Lab 2v3**



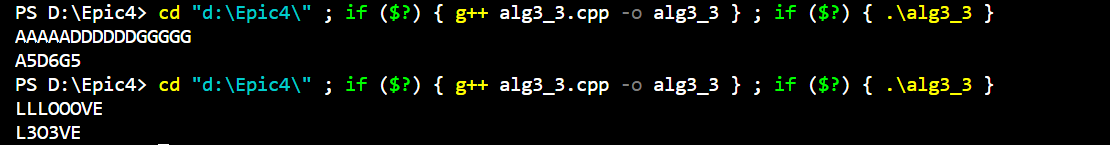
*Рисунок 10 Результат виконання Algotester Lab 2v3*



*Рисунок 11 Скріншот з Algotester Lab 2v3*

Час затрачений на виконання завдання – 1 день

**Завдання №4 Algotester Lab 3v3**. Реалізовано компресію стрічки за допомогою вбудованих функцій бібліотеки “string”, циклів та одновимірних масивів.



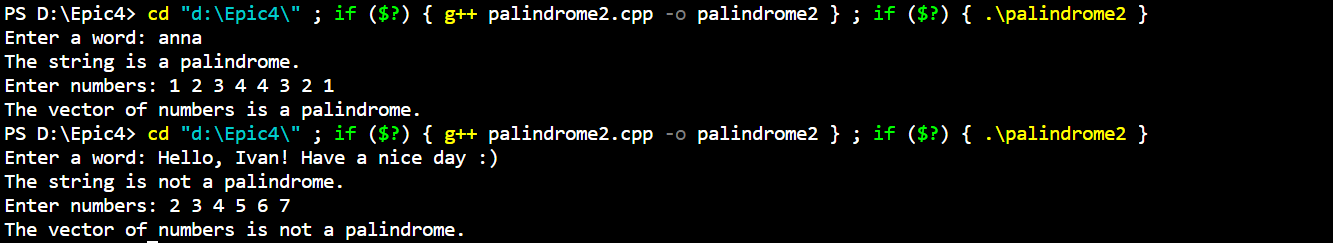
*Рисунок 12 Результат виконання Algotester Lab 3v3*

#### 

*Рисунок 13 Скріншот з Algotester Lab 3v3*

Час затрачений на виконання завдання – 1 день

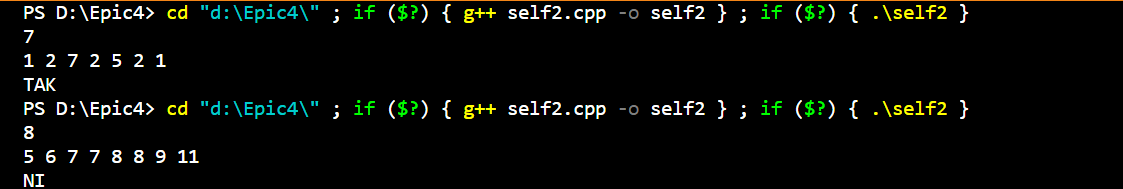
**Завдання №5 Class Practice Palindrome**. Реалізовано програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою використання рекурсії та перевантаження функції.



*Рисунок 14 Результат виконання Class Practice Palindrome*

Час затрачений на виконання завдання – 1 день

#### Завдання №6 Self-Practice (Цікавий масив) Реалізовано програму, яка перевіряє, чи можна розбити масив на дві частини: початок і кінець так, щоб сума чисел в обох частинах співпадала.



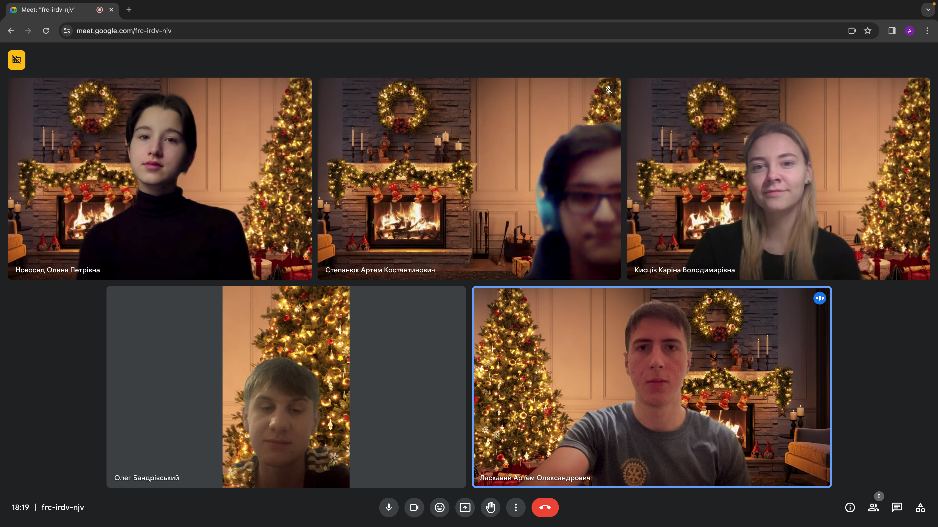
*Рисунок 15 Результат виконання Self-Practice (Цікавий масив)*



*Рисунок 16 Скріншот з Algotester Self-Practice (Цікавий масив)*

Час затрачений на виконання завдання – 1 день

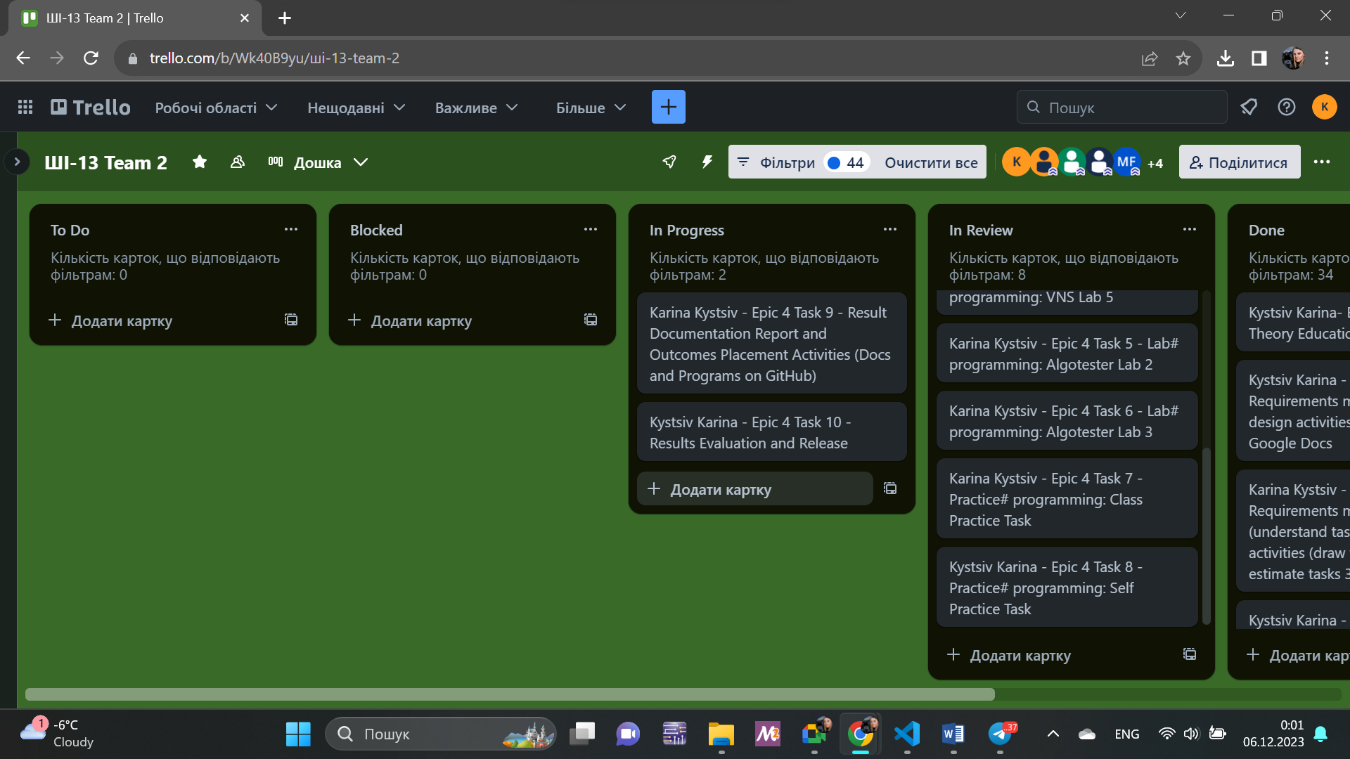
**5. Кооперація з командою:**

* 

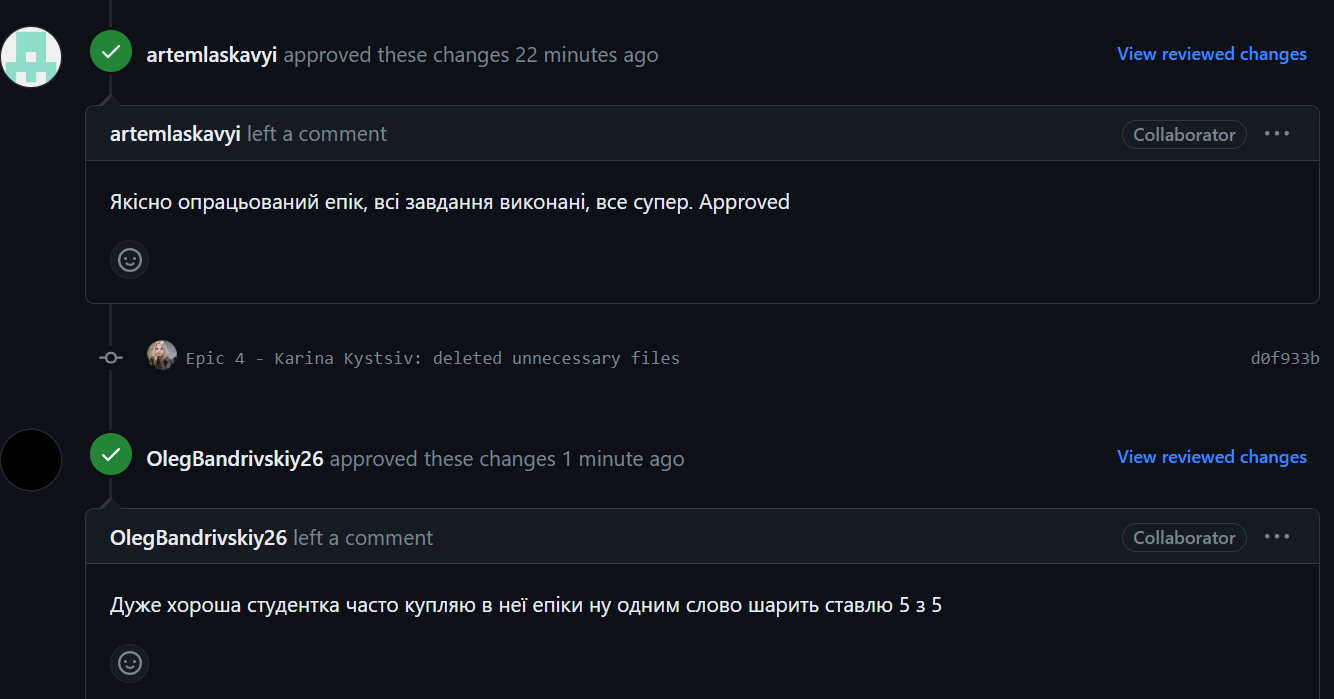
*Рисунок 17 Team 2 Meeting #1*

* 

*Рисунок 18 Team 2 Meeting #2*

* 

*Рисунок 19 Скріншот прогресу в Trello*



# **Висновки:**

Виконавши дану лабораторну роботу я поглибила свої знання про основні концепції роботи з масивами та алгоритмами обробки даних, отримала нові знання про структури даних, та те, що одним із основних завдань структур даних є забезпечення оптимального використання пам'яті та швидкого доступу до них.

Тема №1 охопила основні аспекти роботи з простими структурами даних, включаючи їх оголошення, ініціалізацію та основні операції. Ознайомлення з лінійними та нелінійними структурами, такими як масив, рядок, зв'язаний список, дозволило отримати розуміння їх призначення та використання.

Тема №2 вивчила використання одновимірних та двовимірних масивів, включаючи їхнє оголошення та виконання операцій над ними. Опрацювання цієї теми дозволило здобути навички роботи з масивами в мові програмування.

Тема №3 розглянула передачу масивів як параметрів функцій, включаючи передачу передачу багатовимірних масивів. Це розширило розуміння роботи з функціями та їх взаємодії з масивами в програмах.

Тема №4 зосередилася на стандартних операціях обробки масивів, таких як перегляд по одному елементу та формування псевдодинамічних масивів. Це надало практичні навички в роботі з масивами та обробці їх елементів.

Тема №5 розглянула базові алгоритми сортування масивів, представивши теоретичний огляд сортування бульбашкою, сортування вибором та сортування вставками.

Додатково, була ознайомлена з темою №6, яка включала вивчення Linked List, структур та класів, конструкторів та деструкторів.

У першій частині лабораторної роботи ми зосередили увагу на роботі з одновимірними масивами. Освоїли техніку їх ініціалізації, заповнення та здійснення різноманітних операцій, таких як сортування чи пошук елементів. Це надає можливість здійснювати ефективну обробку великої кількості даних.

У другій частині - перейшли до двовимірних масивів, де практично вивчили їхню структуру та основні операції. Робота з такими масивами важлива для моделювання багатовимірних структур даних, таких як матриці.

В ході роботи виникли неабиякі труднощі, зокрема, перше завдання у цій роботі, а саме VNS Lab 4 - Task 1 було дуже складно виконати з використанням масивів, адже завдання полягало у тому, щоб реалізувати з використанням масиву однонаправлене кільце, додати в кільце перший і останній елементи, а потім знищити з кільця парні елементи. Команда порадила використовувати Linked List як альтернативний алгоритм вирішення цього завдання.

Також доволі складною була VNS Lab 5 - Task 1, адже потрібно було працювати із багатовимірними масивами та створенням функцій для них.

Найлегшою для мене, була задача про паліндроми, адже програмний код вийшов доволі структуризованим та чітким, в даному завданні використовувались логічні змінні, рекурсія, вектори, перевантаження функції та одновимірні масиви, а також вбудовані функції бібліотеки «sstream».