Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

**про 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

**Виконала:**

Студентка групи ШІ-12

Лазаревич Юлія Дмитрівна

Львів 2024

**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:**

Ознайомитись з одновимірними, двовимірними та динамічними масивами, вказівниками та посиланнями, структурами даних, вкладеними структурами та алгоритмами обробки.

**Теоретичні відомості:**

Тема №1: Класи пам'яті у C++.

* Джерела Інформації:
* [Класи, Об’єкти і Методи в С++](https://acode.com.ua/urok-121-klasy-ob-yekty-i-metody/)
* Що опрацьовано:
* Статична пам’ять – область пам'яті, де зберігаються глобальні та статичні змінні, які ініціалізуються на етапі компіляції та які існують протягом усього виконання програми. (static int counter = 0;)
* Динамічна пам’ять виділяється під час виконання програми за допомогою new і звільняється з використанням delete, що дозволяє створювати змінні або масиви з довжиною, яка визначається під час виконання програми.( int\* arr = new int[5]; delete[] arr;)
* Стек використовується для зберігання локальних змінних і викликів функцій. Він має обмежену пам’ять і працює за принципом "останнім увійшов — першим вийшов" (LIFO).
* Виділення пам’яті здійснюється через new або malloc, а вивільнення — через delete або free. Неправильне управління може призвести до витоків пам'яті.
* Статус: ознайомлена.
* Початок опрацювання теми: 20.11.24
* Завершення опрацювання теми 25.11.24

Тема №2: Одновимірні та двовимірні масиви.

* Джерела Інформації:
* [Що таке масив у програмуванні: основи та застосування](https://foxminded.ua/masyv-u-prohramuvanni/#:~:text=%D0%9E%D0%B4%D0%BD%D0%BE%D0%B2%D0%B8%D0%BC%D1%96%D1%80%D0%BD%D0%B8%D0%B9%20%E2%80%93%20%D1%82%D0%B0%D0%BA%D0%BE%D0%B6%20%D0%BD%D0%B0%D0%B7%D0%B8%D0%B2%D0%B0%D1%94%D1%82%D1%8C%D1%81%D1%8F%20%D0%BB%D1%96%D0%BD%D1%96%D0%B9%D0%BD%D0%B8%D0%BC%20%D1%96%20%D1%8F%D0%B2%D0%BB%D1%8F%D1%94%20%D1%81%D0%BE%D0%B1%D0%BE%D1%8E,%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D1%87%D0%B0%D1%94%20%D1%80%D1%8F%D0%B4%D0%BE%D0%BA%2C%20%D0%B0%20%D0%B4%D1%80%D1%83%D0%B3%D0%B8%D0%B9%20%E2%80%93%20%D1%81%D1%82%D0%BE%D0%B2%D0%BF%D1%87%D0%B8%D0%BA%20%D0%B0%D0%B1%D0%BE%20%D0%BA%D0%BB%D1%96%D1%82%D0%B8%D0%BD%D0%BA%D1%83.)
* Що опрацьовано:
* Одновимірні:
* Створення та ініціалізація: int arr[5] = {1, 2, 3, 4, 5};
* Основні операції: Індексація (arr[2]), присвоєння (arr[0] = 10), читання.
* Цикли для обходу: for (int i = 0; i < 5; i++) {std::cout << arr[i];}
* Функції: Масиви передаються як вказівники.
* Алгоритми сортування: Наприклад, сортування вставками.
* Двовимірні(представляють таблицю елементів (матрицю)):
* Створення та ініціалізація: int matrix[3][4] = { {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} };
* Вкладені цикли для обходу:

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 4; j++) {

std::cout << matrix[i][j] << " ";

}

std::cout << std::endl;

}

* На практиці двовимірні масиви корисні для задач обробки матриць, наприклад, знаходження суми елементів рядка чи стовпця.
* Масиви передаються через вказівники:

void printMatrix(int matrix[][4], int rows) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < 4; j++) {

std::cout << matrix[i][j] << " ";

}

std::cout << std::endl;

}

}

* Статус: ознайомлена.
* Початок опрацювання теми: 20.11.24
* Завершення опрацювання теми 25.11.24

Тема №3: Вказівники та Посилання.

* Джерела Інформації:
* [Урок №92. Посилання](https://acode.com.ua/urok-92-posylannya/)
* [Урок №84. Вказівники](https://acode.com.ua/urok-84-vkazivnyky/)
* Що опрацьовано:
* Вказівники для доступу до масивів:

int arr[] = {1, 2, 3};

int\* p = arr;

* Арифметика вказівників: Наприклад, p + 1 вказує на наступний елемент.
* Різниця між вказівниками та посиланнями: Посилання є статичними, а вказівники можна змінювати.
* Динамічне виділення пам’яті:

int\* arr = new int[10];

delete[] arr;

* Статус: ознайомлена.
* Початок опрацювання теми: 27.11.24
* Завершення опрацювання теми 28.11.24

Тема №4: Динамічні Масиви.

* Джерела Інформації:
* [Урок №90. Динамічні масиви](https://acode.com.ua/urok-90-dynamichni-masyvy/)
* Що опрацьовано:
* Масив створюється під час виконання за допомогою оператора new:

int\* arr = new int[10];

* Завжди потрібно вивільняти пам’ять:

delete[] arr;

* Для зміни розміру можна створити новий масив, копіювати вміст та видалити старий:

int\* newArr = new int[newSize];

std::copy(oldArr, oldArr + oldSize, newArr);

delete[] oldArr;

* Передаються у функції як вказівники разом із розміром.
* Статус: ознайомлена.
* Початок опрацювання теми: 28.11.24
* Завершення опрацювання теми 29.11.24

Тема №5: Структури Даних

* Джерела Інформації:
* [Урок №64. Структури](https://acode.com.ua/urok-64-struktury/)
* Що опрацьовано:
* Оголошення та використання:

struct Point {

int x, y;

};

Point p = {10, 20};

* Масиви у структурах:

struct Student {

std::string name;

int grades[5];

};

* Структури передаються у функції за значенням або посиланням:

void printPoint(const Point& p) {

std::cout << p.x << ", " << p.y;

}

* Union – тип даних, де всі поля ділять одну область пам'яті:

union Data {

int i;

float f;

};

* Переліки:

enum Color { RED, GREEN, BLUE };

* Статус: ознайомлена.
* Початок опрацювання теми: 28.11.24
* Завершення опрацювання теми 29.11.24

Тема №6: Вкладені структури та їх використання.

* Джерела Інформації:

* [Урок №64. Структури](https://acode.com.ua/urok-64-struktury/)
* Що опрацьовано:
* Оголошення:

struct Address {

std::string city;

std::string street;

};

struct Person {

std::string name;

Address address;

};

* Передача у функції:

void printPerson(const Person& p) {

std::cout << p.name << ", " << p.address.city;

}

* Застосування: Використовуються для моделювання складних об’єктів, таких як бази даних.
* Статус: ознайомлена.
* Початок опрацювання теми: 12.10.24
* Завершення опрацювання теми 22.10.24

Тема №7: Алгоритми обробки та робота з масивами та структурами.

* Джерела Інформації:
* [Урок №77. Масиви](https://acode.com.ua/urok-77-masyvy/)
* Що опрацьовано:
* Сортування: Наприклад, сортування бульбашкою:

void bubbleSort(int arr[], int n) {

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

std::swap(arr[j], arr[j + 1]);

}

}

}

}

* Пошук: Лінійний або бінарний пошук у масиві.
* Маніпуляції зі структурами: Наприклад, обчислення середнього балу студента з використанням масиву оцінок
* Статус: ознайомлена.
* Початок опрацювання теми: 12.10.24
* Завершення опрацювання теми 22.10.24

**Виконання роботи:**

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

Завдання №1 – VNS Lab 4 – Task 1 – Variant 10.

1) Сформувати одновимірний масив цілих чисел, використовуючи генератор

випадкових чисел.

2) Роздрукувати отриманий масив.

3) Знищити 5 перших елементів масиву.

4) Додати в кінець масиву 3 нових елементи.

5) Роздрукувати отриманий масив.

Методичні вказівки:

1) При виконанні роботи використовуються статичні масиви. Для організації статичних масивів із псевдозмінними межами необхідно оголосити масив досить великої довжини, наприклад, 100 елементів: int N=100; int a[N];

Потім користувач вводить реальну довжину масиву (не більше N) і працює з масивом тієї довжини, що він сам вказав. Інші елементи (хоча пам'ять під них і буде виділена) не розглядаються.

2) При зменшенні або збільшенні довжини масиву необхідно змінювати його реальну довжину.

Завдання №2 – VNS Lab 5 – Task 1 – Variant 10.

Використовуючи функції, розв’язати зазначене у варіанті завдання. Масив

повинен передаватися у функцію як параметр.

Написати функцію, що перевіряє чи є від’ємні елементи в зазначеному рядку

двовимірного масиву. Знищити з масиву всі рядки з від’ємними елементами,

знищений рядок заповнюється 0 і переноситься в кінець масиву.

Завдання №3 – Algotester Lab 2 – Variant 3.

Вам дано масив цiлих чисел розмiром N, на першiй та останнiй клiтинцi розмiщено по дрону. Вони одночасно взлiтають.

На початку кожного ходу швидкiсть дрону стає рiвною значенню клiтинки, у якiй вiн знаходиться. Тобто лiвий дрон у першу секунду з клiтинки з iндексом 1 перелетить у клiтинку з iндексом a1, тобто його наступна позицiя рахується як поточна позицiя + число у поточнiй позицiї (перегляньте пояснення для вiзуалiзацiї) Правий робить аналогiчно в протилежну сторону.

Вони це роблять до моменту, коли трапиться одна з зазначених подiй:

Якшо 2 дрони опиняються в однiй клiтинцi - ви виводите Collision.

Якщо лiвий дрон опиниться справа вiд правого - це Miss

У випадку якщо вони зупиняться один навпроти одного, тобто у клiтинках ai та ai+1 – виведiть «Stopped» Врахуйте, що перевiряти треба також до взльоту.

Завдання №4 – Algotester Lab 3 – Variant 3.

Вам дана стрiчка s.

Ваше завдання зробити компресiю стрiчки, тобто якщо якась буква йде бiльше одного разу пiдряд у стрiчцi замiнити її на букву + кiлькiсть входжень пiдряд.

Вхiднi данi

У першому рядку стрiчка S

Вихiднi данi

Стрiчка Scompressed

Обмеження

1 ≤ |S| ≤ 105

Завдання №5**–** Class Practice Work – Перевірка чи слово або число є паліндромом.

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:
   1. Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.
2. Приклад визначення функції:
   1. *bool isPalindrome(const string& str, int start, int end);*
3. Перевантаження функцій:
   1. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.
   2. *bool isPalindrome*(ціле число);
4. Рекурсія:
   1. 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 Work – Algotester Lab 3 – Variant 2.

Вам дано 2 масиви розміром N та M. Значення у цих масивах унікальні.

Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

**Вхідні дані**

У першому рядку ціле число NN

у другому рядку NN цілих чисел a1..ana1..an

У третьому рядку ціле число MM

у четвертому рядку MM цілих чисел b1..bnb1..bn

**Вихідні дані**

У першому рялку одне ціле число - кількість елементів, які наявні в обох масивах одночасно.

У другому рядку кількість унікальних елементів в обох масивах (тобто кількість унікальних елементів у масиві, який буде об’єднанням двох даних).

Додаткове завдання №7– Algotester Lab 3 – Variant 1.

Ви з’явились у світі під назвою Атод посеред Пустелі Безправ’я. Так сталось, що Ви попали саме в той час і місце, де ведеться битва між чаклункою Ліною і темними силами, які хочуть знищити цей світ. На жаль, трапилась халепа, бо деякі слова із книги чар були пошкоджені під час битви. Одне таке слово можна відновити виконавши ритуал зцілення над пошкодженими буквами. Ритуал зцілення можна виконати на всіх **підряд** розташованих **пошкоджених** буквах. Вам не залишається нічого іншого як допомогти Ліні відновити ці слова і сказати скільки мінімально треба провести таких ритуалів, щоб прочитати одне з наймогутніших у цьому світі заклять - Поневолення Дракона!

**Вхідні дані**

У першому рядку NN - кількість рядків у заклятті.

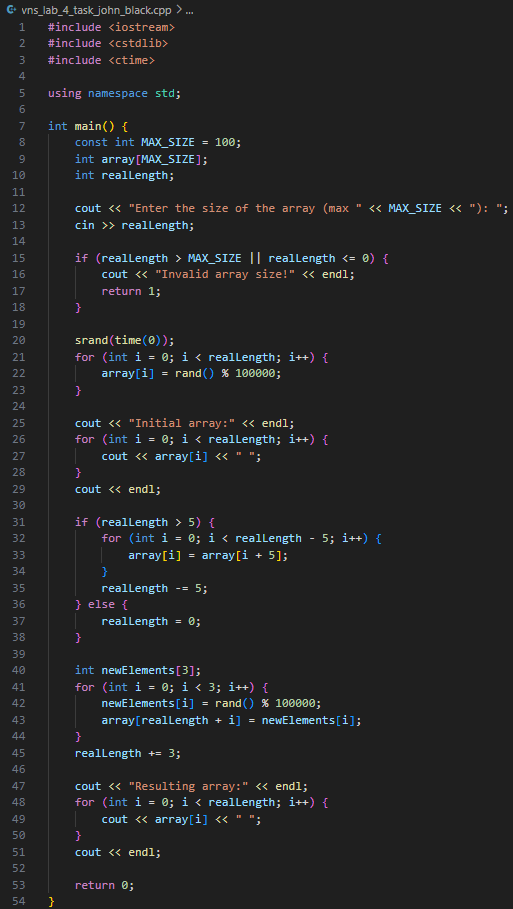
В наступних NN рядкях - набір слів w1,…,wMw1,…,wM, розділених пробілами, де кожне слово може містити малі латинські літери та символ ##, який позначає пошкоджену букву.

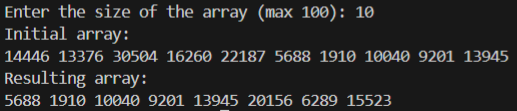
**Вихідні дані**

Єдине ціле число - мінімальна кількість ритуалів, які потрібно провести, щоб відновити закляття.

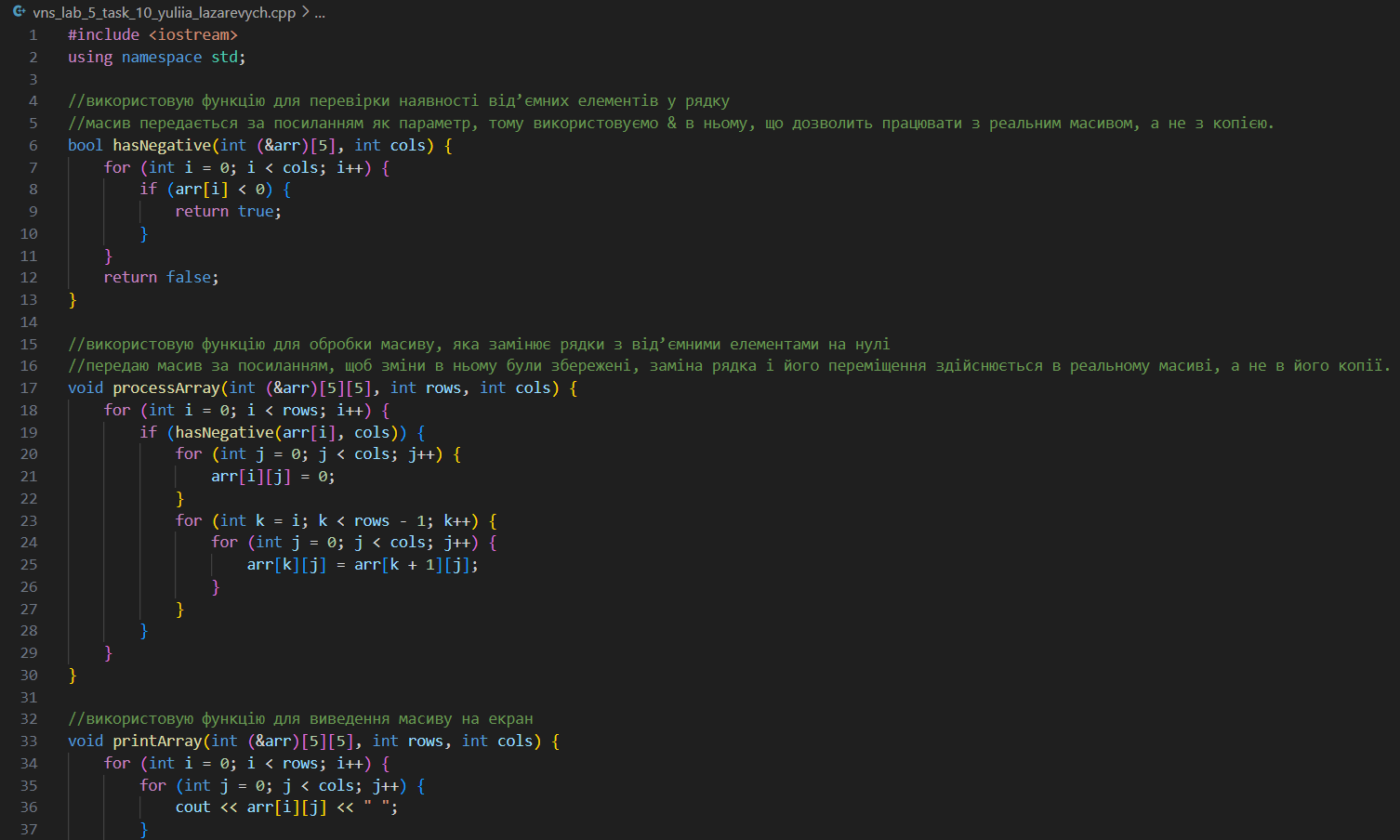
1. **Код програм з посиланням на зовнішні ресурси:**

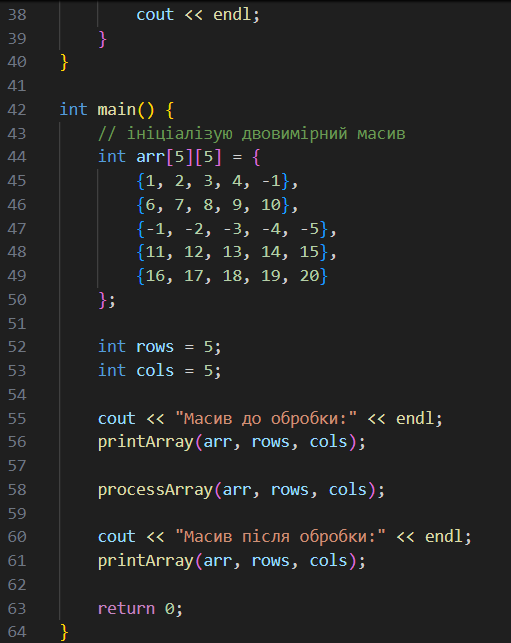
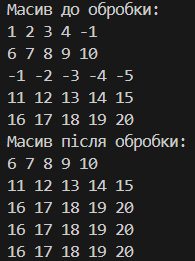
Завдання №1 – VNS Lab 4 – Task 1 – Variant 10.



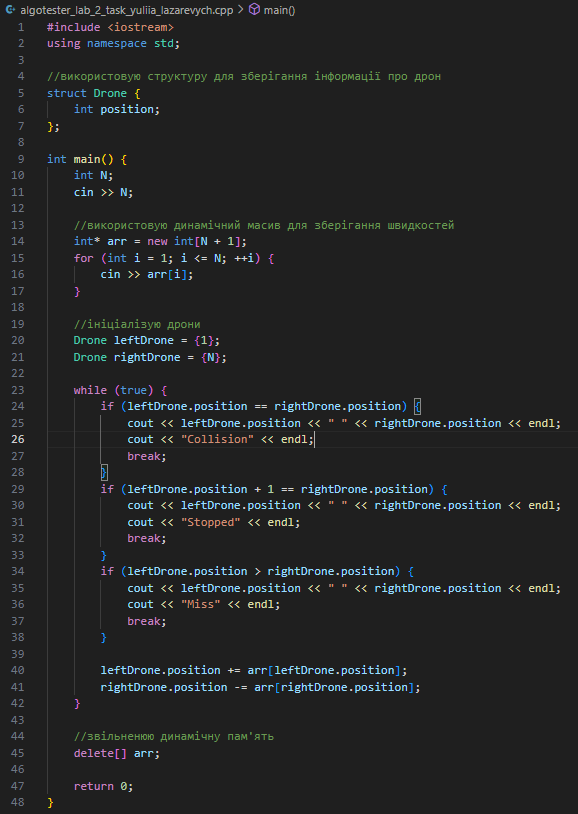


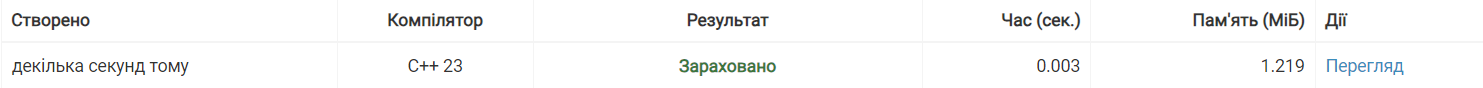
Завдання №2 – VNS Lab 5 – Task 1 – Variant 10.



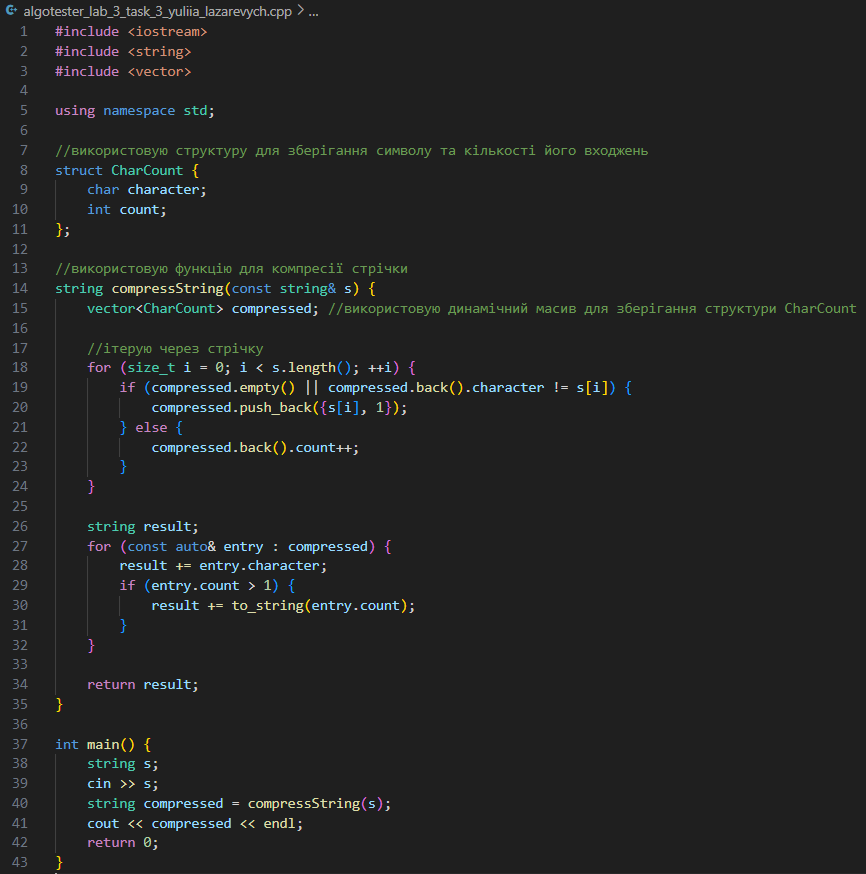
 

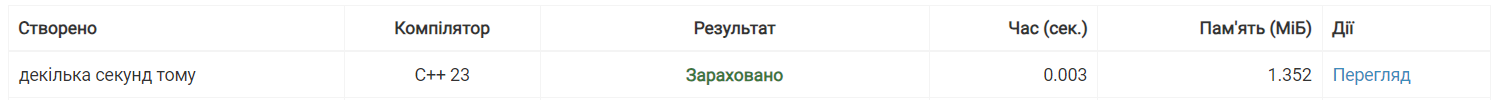
Завдання №3 – Algotester Lab 2 – Variant 3.



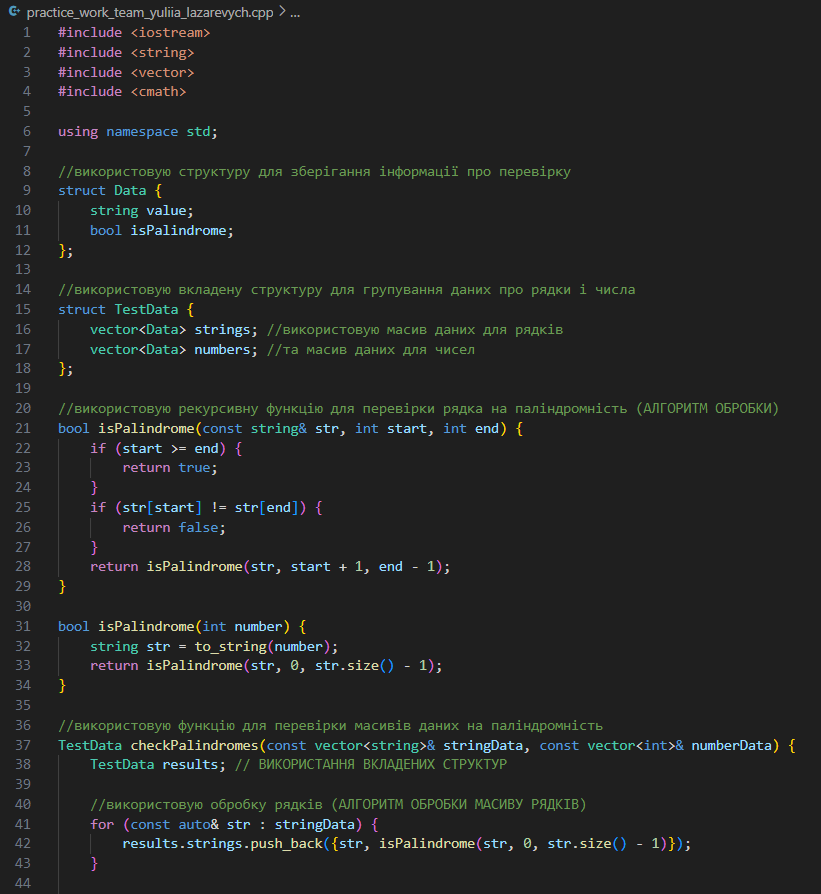


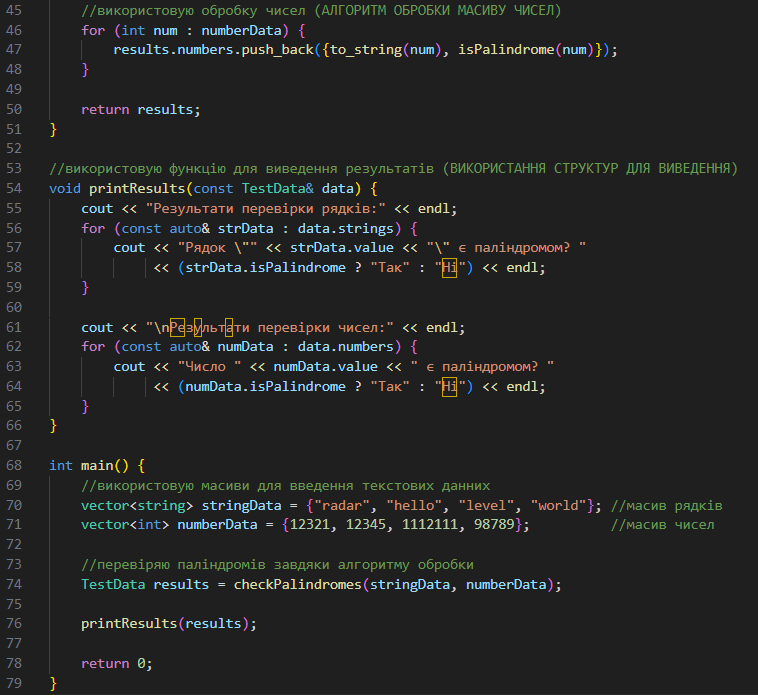
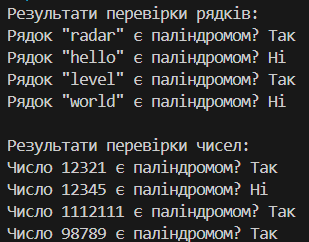
Завдання №4 – Algotester Lab 3 – Variant 3.



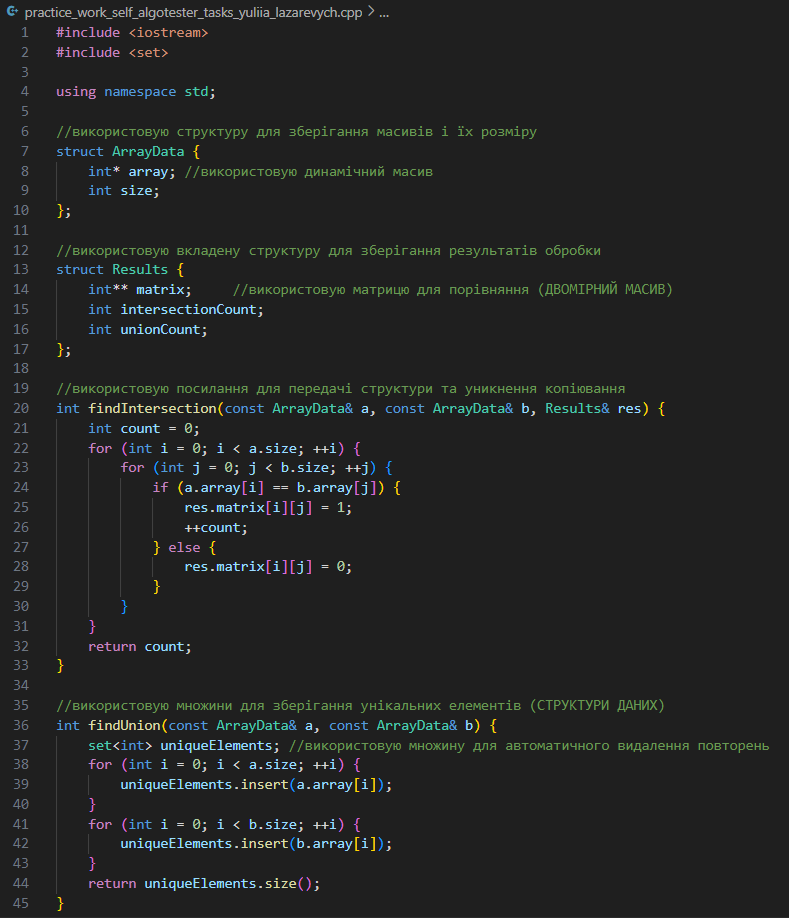


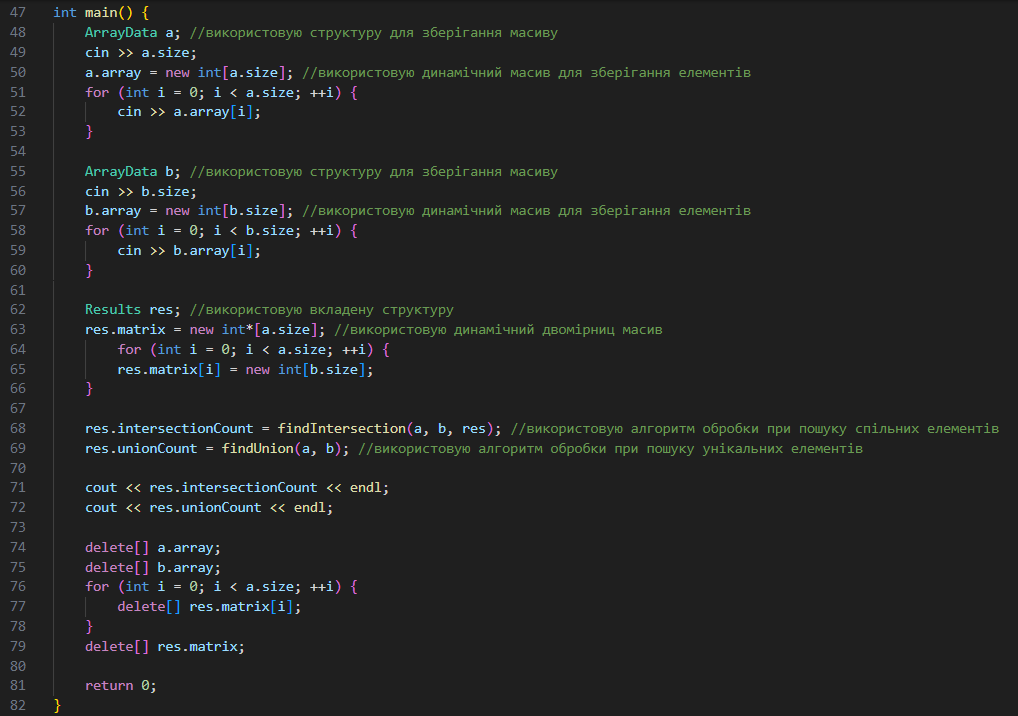
Завдання №5 – Class Practice Work – Перевірка чи слово або число є паліндромом.

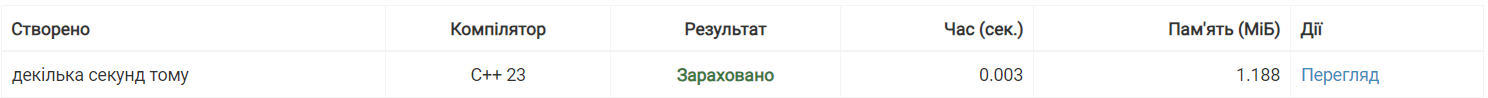
****

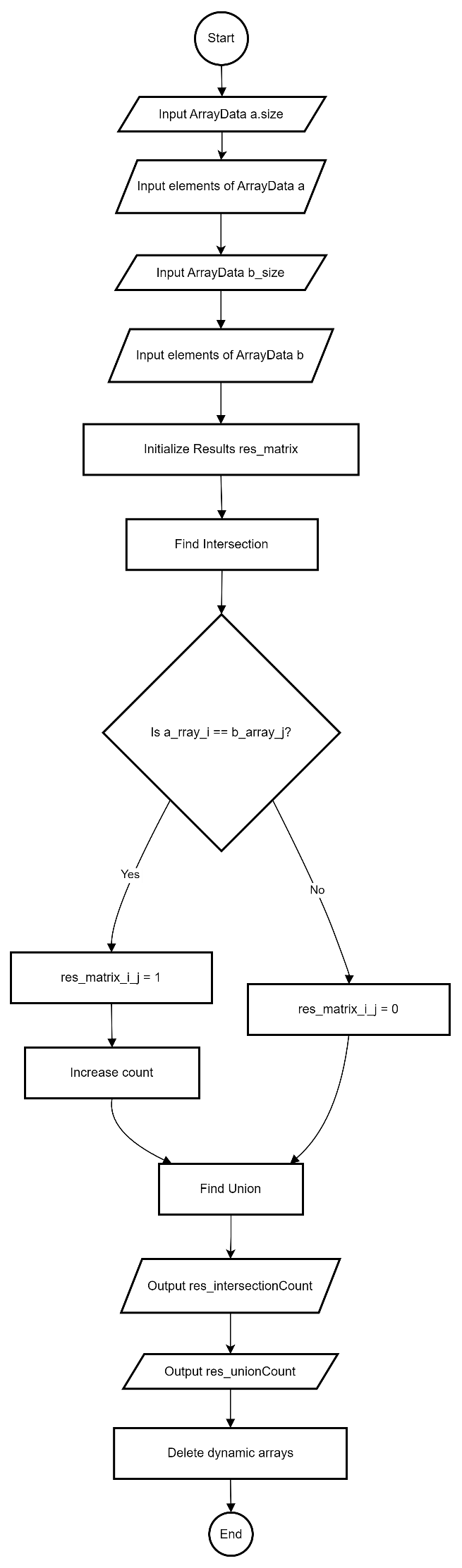
** **

Завдання №6- Self Practice Work – Algotester Lab 3 – Variant 2.

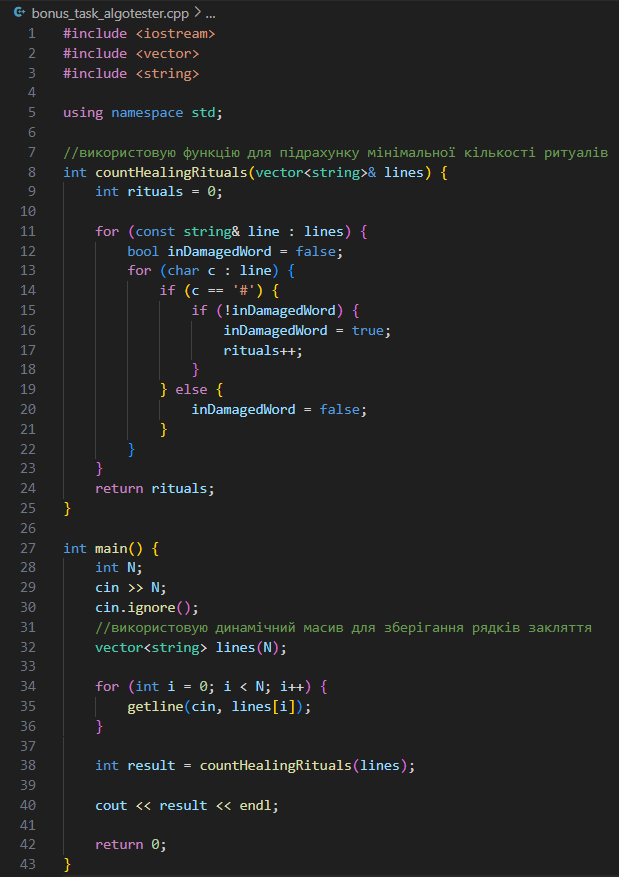
****

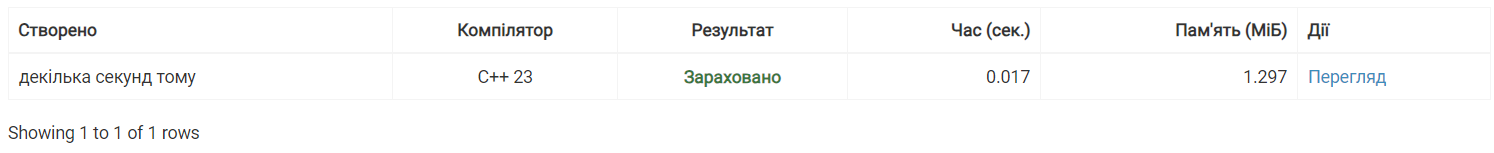






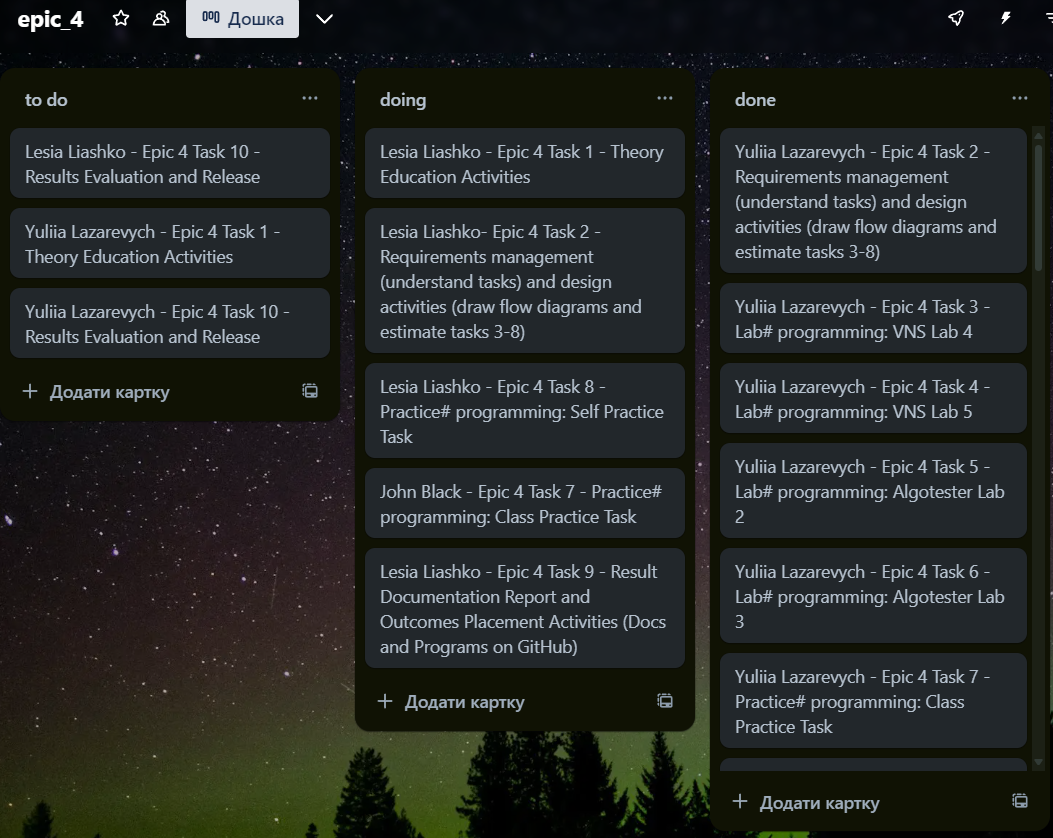
Додаткове завдання №7– Algotester Lab 3 – Variant 1.





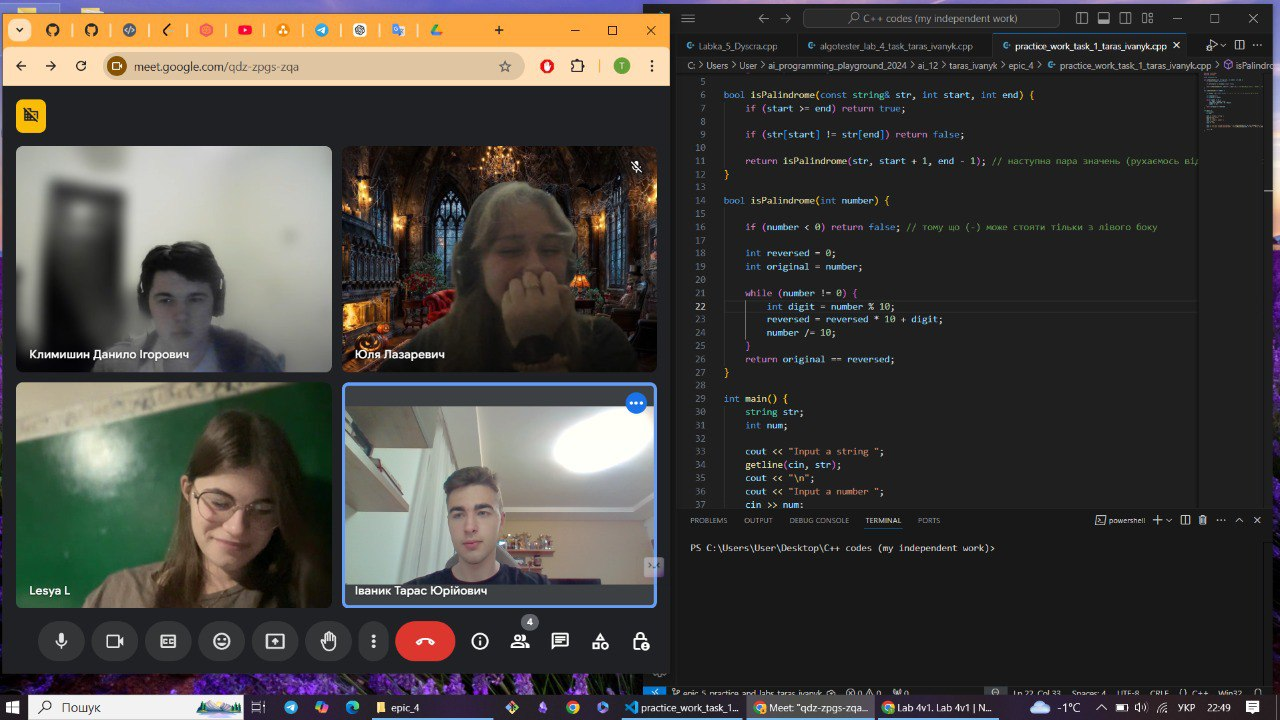
1. **Кооперація з командою:**

* Скрін прогресу по Трелло



Відстежували прогрес всієї команди завдяки дошці Trello

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



Зустрічалися багато разів для обговорення та спільного виконання епіків.

**Висновки:** Виконуючи цей епік я ознайомилась з одновимірними, двовимірними та динамічними масивами, вказівниками та посиланнями, структурами даних, вкладеними структурами та алгоритмами обробки. Також я написала шість кодів де застосувала нові знання на практиці.