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-14

Зімнов Олександр Сергійович

# **Тема роботи:**

# Прості структури даних. Одновимірні масиви. Двовимірні Масиви. Алгоритми обробки

# **Мета роботи:**

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

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

1. Індивідуальний план опрацювання теорії:

* Тема №1: Прості структури даних.
  + Джерела Інформації
    - Лекції
    - Лекційний матеріал на ВНС
  + Що опрацьовано:
    - Array.
    - Structure.
    - Vector.
    - List.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.2023
  + Звершення опрацювання теми: 21.11.2023
* Тема №2: Одновимірні масиви. Двовимірні Масиви.
  + Джерела Інформації:
    - Лекції
    - Лекційний матеріал на ВНС
  + Що опрацьовано:
    - Одновимірні масиви.
    - Двовимірні Масиви.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 21.11.2023
  + Звершення опрацювання теми: 24.11.2023
* Тема №3: Алгоритми обробки.
  + Джерела Інформації:
    - Лекції
    - Лекційний матеріал на ВНС
  + Що опрацьовано:
    - [Сортування (бульбашкою, вставкою, злитям)](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210736&displayformat=dictionary).
    - Пошук (лінійний, бінарний).
    - Хешування.
    - Дерева.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 24.11.2023
  + Звершення опрацювання теми: 25.11.2023

# **Виконання роботи:**

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

Завдання №1 - Theory Education Activities

* Вивчення теорії

Завдання №2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)

* Створення діаграм для програм в draw io

Завдання №3 - Lab# programming: VNS Lab 4

* Варіант № 4
* Деталі завдання: Сформувати одновимірний масив цілих чисел, використовуючи генератор випадкових чисел. Роздрукувати отриманий масив. Знищити елементи, індекси яких кратні 3. Додати після кожного від’ємного елемента масиву елемент зі значенням M[I-1 ]+1. Роздрукувати отриманий масив.
* Важливі деталі для врахування в імплементації програми: Для додавання випадкового числа використовується функція rand(). Для створення масиву з заданим користувачем розміром необхідно використати вказівник \*.

Завдання №4 - Lab# programming: VNS Lab 5

* Варіант № 4
* Деталі завдання: Визначити чи є матриця ортонормованою, тобто такою, що скалярний добуток кожної пари різних рядків дорівнює 0, а скалярний добуток рядка самого на себе дорівнює 1.
* Важливі деталі для врахування в імплементації програми: Варто звернути увагу, що для створення двовимірного масив розміром заданим користувачем, потрібно використати два вказівники \*\*, також необхідно в функцію передати не лише матрицю, а й її розміри

Завдання №5 - Lab# programming: Algotester Lab 2

* Варіант № 2
* Деталі завдання: У вас є масив r розміром N. Також вам дано 3 цілих числа. Спочатку ви маєте видалити з масиву ці 3 числа, які вам дані. Після цього перетворити цей масив у масив сум, розміром Nnew−1 (розмір нового масиву після видалення елементів), який буде відображати суми сусідніх елементів нового масиву. Далі необхідно вивести масив сум на екран.
* Важливі деталі для врахування в імплементації програми: 1≤N≤105, 0≤ri≤105, 0≤a,b,c≤105. Для створення масиву розміром заданим користувачем потрібно використати вказівник \*.

Завдання №6 - Practice# programming: Algotester Lab 3

* Варіант № 3
* Деталі завдання: Вам дана стрічка s. Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.
* Важливі деталі для врахування в імплементації програми: 1≤|S|≤105. Для виміру довжини стрічки використати функцію length().

Завдання №7 - Practice# programming: Class Practice Task

* Деталі завдання: Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.
* Важливі деталі для врахування в імплементації програми: Потрібно написати дві функції, в одному випадку функція перевіряє чи значення число і перетворює його в текст, а друга функція рекурсивно робить перевірку на паліндром і повертає true чи false.

Завдання №8 - Practice# programming: Self Practice Task

* Деталі завдання: Вам дано масив a з N цілих чисел. Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4]. Після цього оберніть посортовану версію масиву a на K, тобто при K=3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3]. Виведіть результат.
* Важливі деталі для врахування в імплементації програми: 1≤N,K≤1000, 0≤ai≤100. Так як ми не можемо одразу видалити всі однакові букви, бо нам потрібно порахувати їх кількість, тому варто використати додаткове значення для провірки повторюваності

Завдання №9 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

* Створення звіту і завантажити його з програмами на GitHub

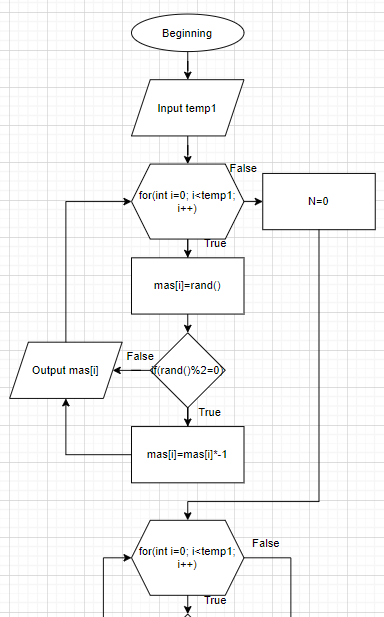
Завдання №10 - Results Evaluation and Release

* Захист звіту.

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

Програма №1 VNS Lab4

* Блок-схема



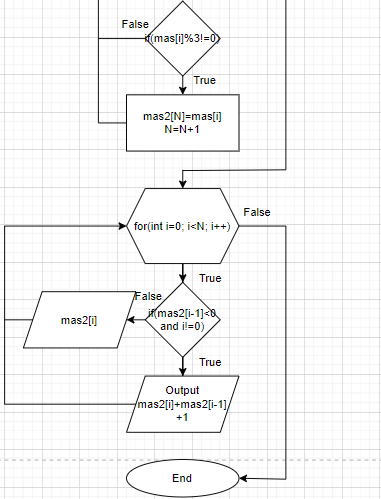
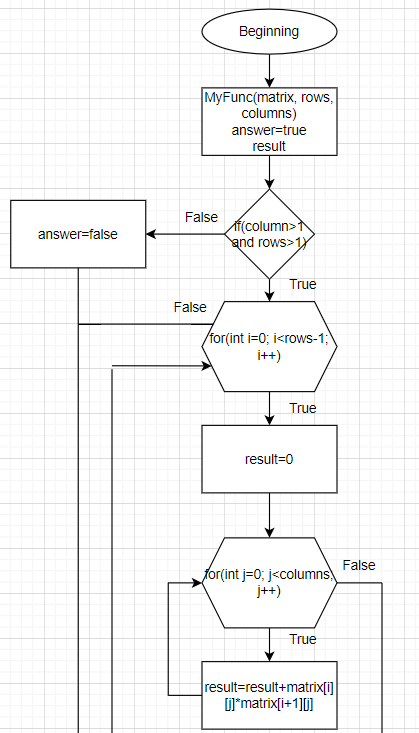


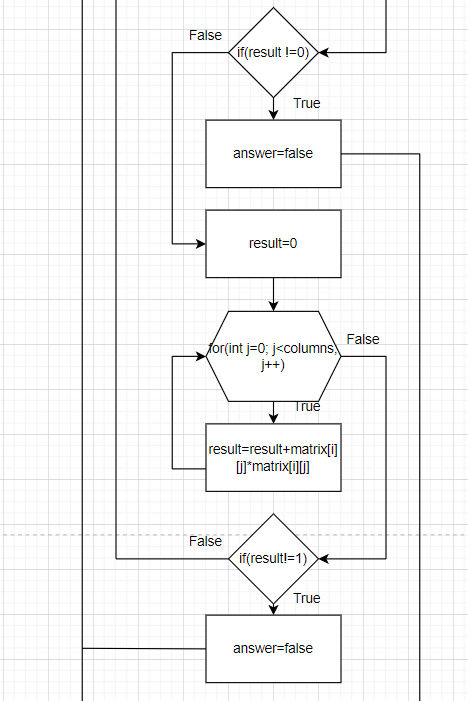
Рисунок 1: Блок схема до програми №1

* Час на виконання: 1 година
* Важливі деталі для врахування в імплементації програми: Функція rand() не генерує від`ємних чисел, тому я зроблю деякі числа від`ємними, завдяки rand()%2 це перевірить чи функція є парна чи не парна, що дорівнює 50-ти % шансу.

Програма №2 VNS Lab5

* Блок-схема





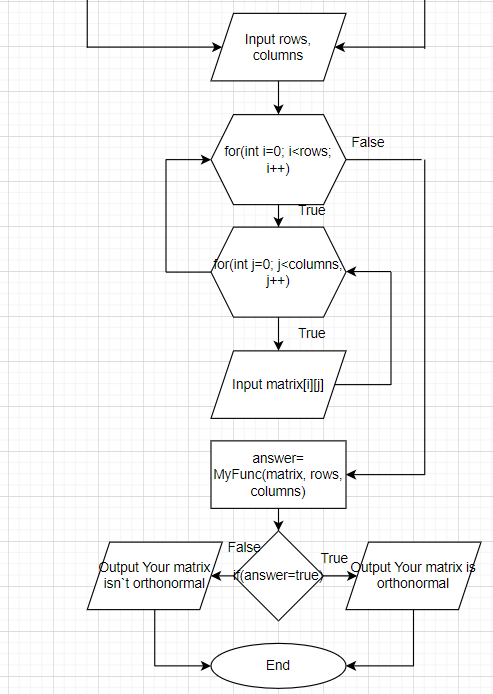


Рисунок 2: Блок схема до програми №2

* Час на виконання: 3 годин
* Важливі деталі для врахування в імплементації програми: Необхідно запам`ятовувати скалярні рядки, достатньо провірити чи вони рівні 1, або 0 відповідно, якщо вони рівні 1, або відповідно 0, то перевірки можна завершити, бо матриця не ортогональна.

Програма №3 Algotester Lab 2

* Блок-схема

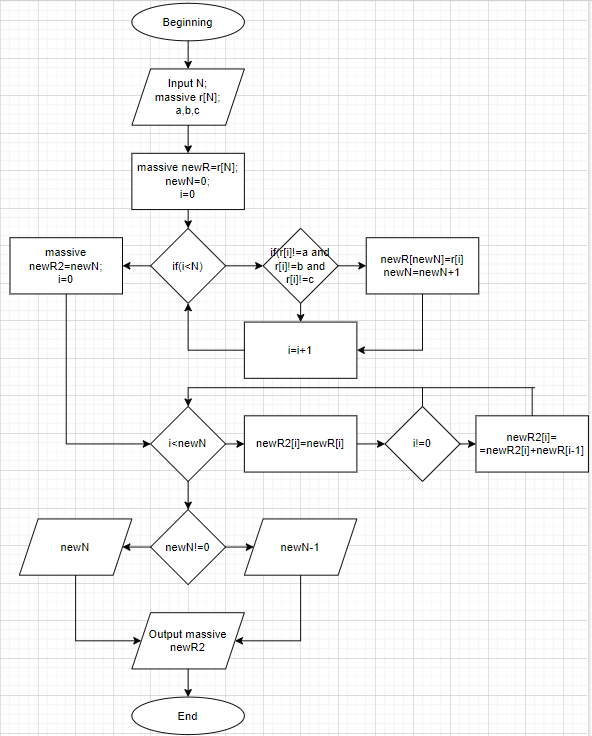


Рисунок 3: Блок схема до програми №3

* Час на виконання: 1 година
* Важливі деталі для врахування в імплементації програми: Для того, щоб видалити елементи масиву можна зробити новий масив в який вносити елементи старого масиву, тільки при умові, що даний елемент не рівний a, b і c.

Програма №4 Algotester Lab 3

* Блок-схема

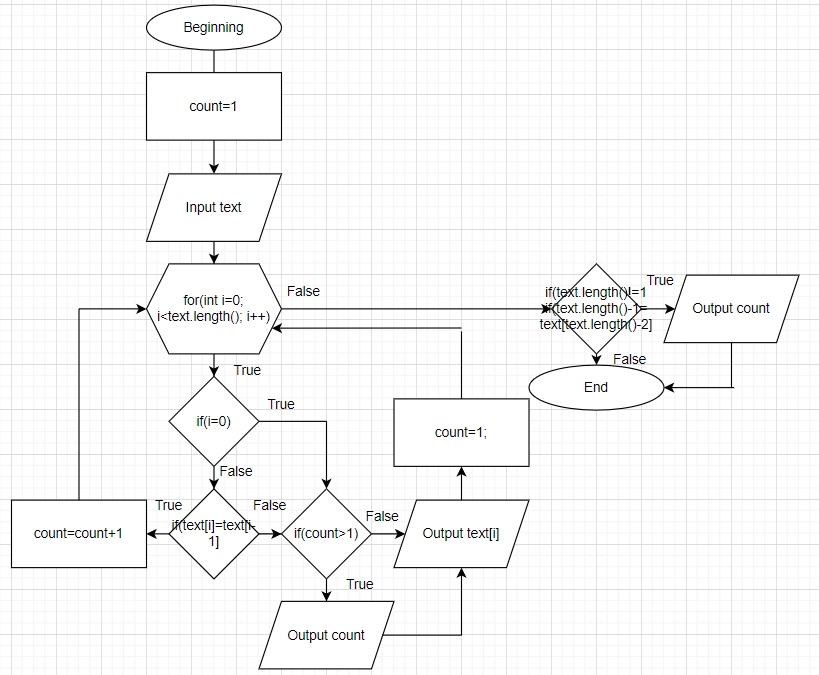


Рисунок 4: Блок схема до програми №4

* Час на виконання: 1 година
* Важливі деталі для врахування в імплементації програми: При провірці чи рівні даний елемент і попередній важливо завчасно перевірити чи є попередній елемент, якщо елемент перший, то потрібно пропустити момент з перевіркою. Аналогічна ситуація з кінцевою перевіркою, якщо довжина тексту буде всього буква, то потрібно пропустити останню перевірку.

Програма №5 Class Practice Task

* Блок-схема

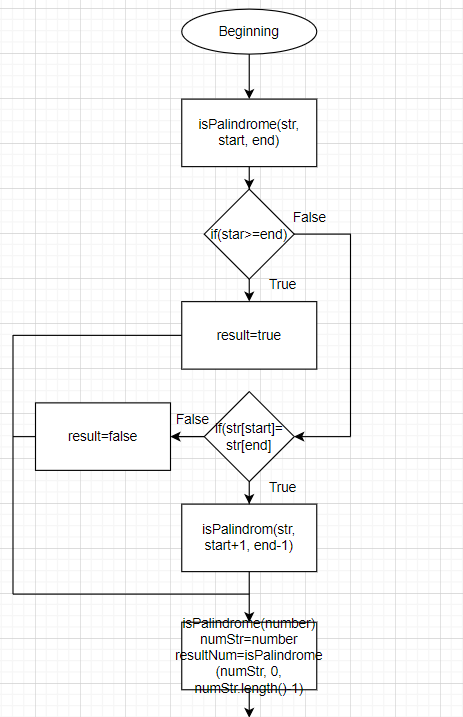
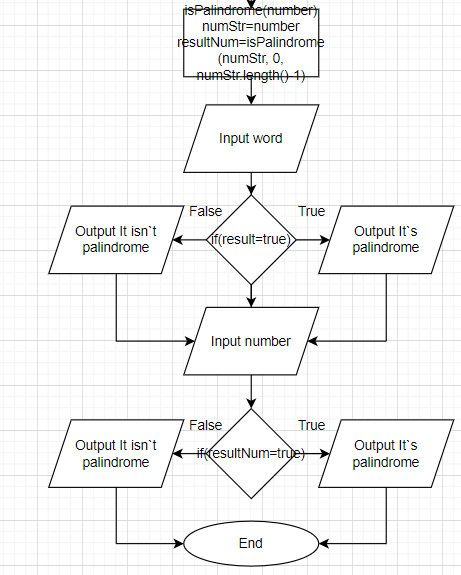
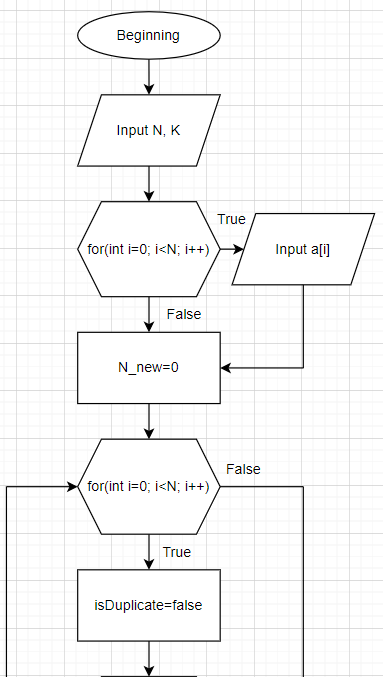


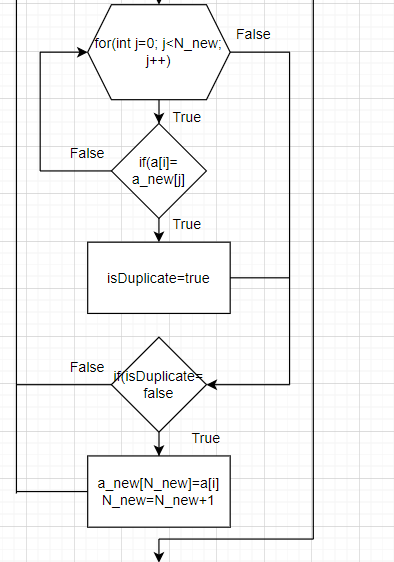
Рисунок 5: Блок схема до програми №5

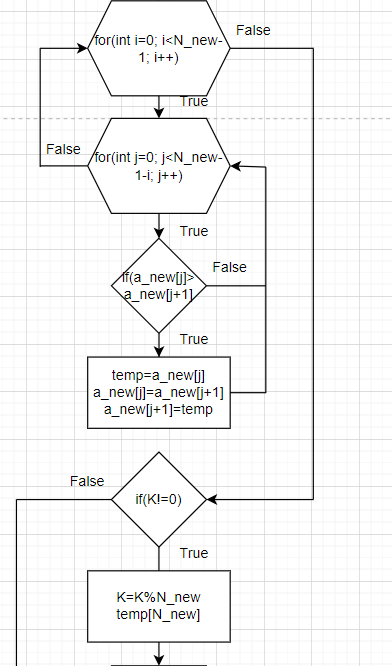
* Час на виконання: 3 години
* Важливі деталі для врахування в імплементації програми: Зсув K потрібно здійснити необхідну кількість разів, тобто якщо K довше ніж символів рядку, то K потрібно зменшити.

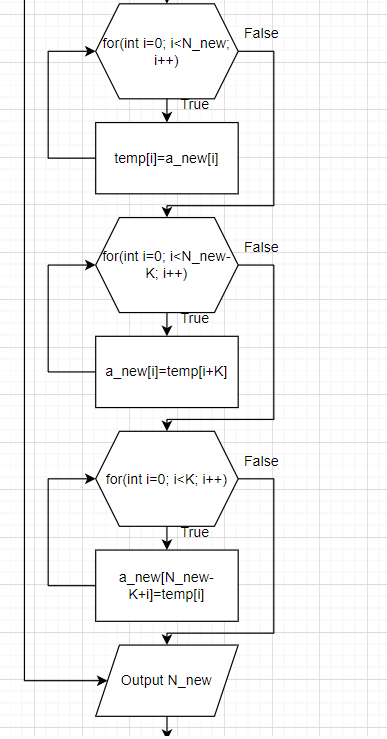
Програма №6 Self Practice Task

* Блок-схема









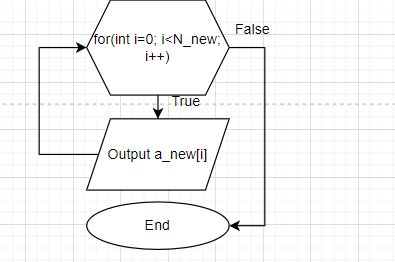


Рисунок 6: Блок схема до програми №6

* Час на виконання: 2 години
* Важливі деталі для врахування в імплементації програми: При викликанні рекурсії функції необхідно задати початкове значення на один більше, а кінцевий на один менший і при тому перевірити чи вони не рівні, це зроблено для того щоб перевірити кожен елемент.

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

Завдання №1: VNS Lab 4

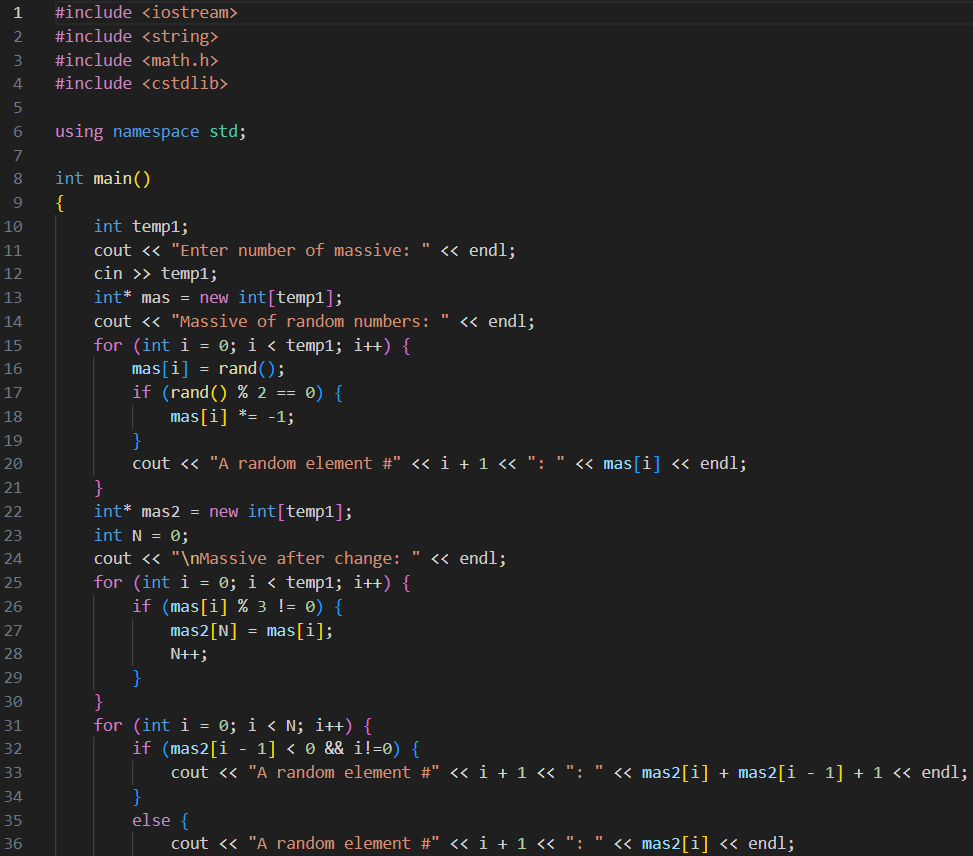




Рисунок 7: Код завдання з VNS №4

Завдання №2: VNS Lab 5

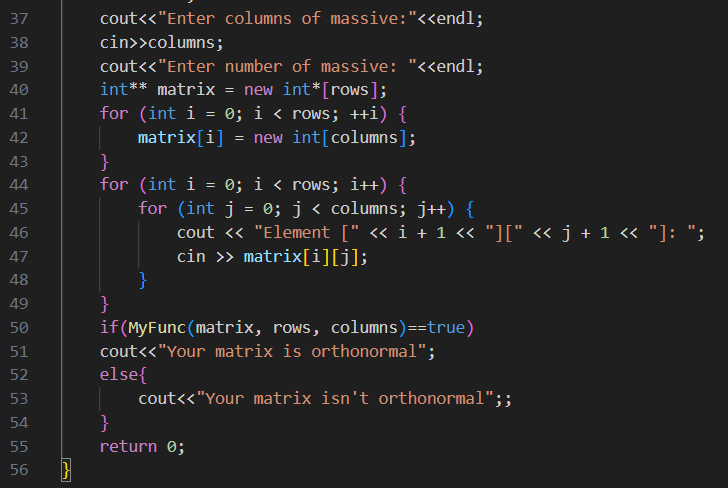
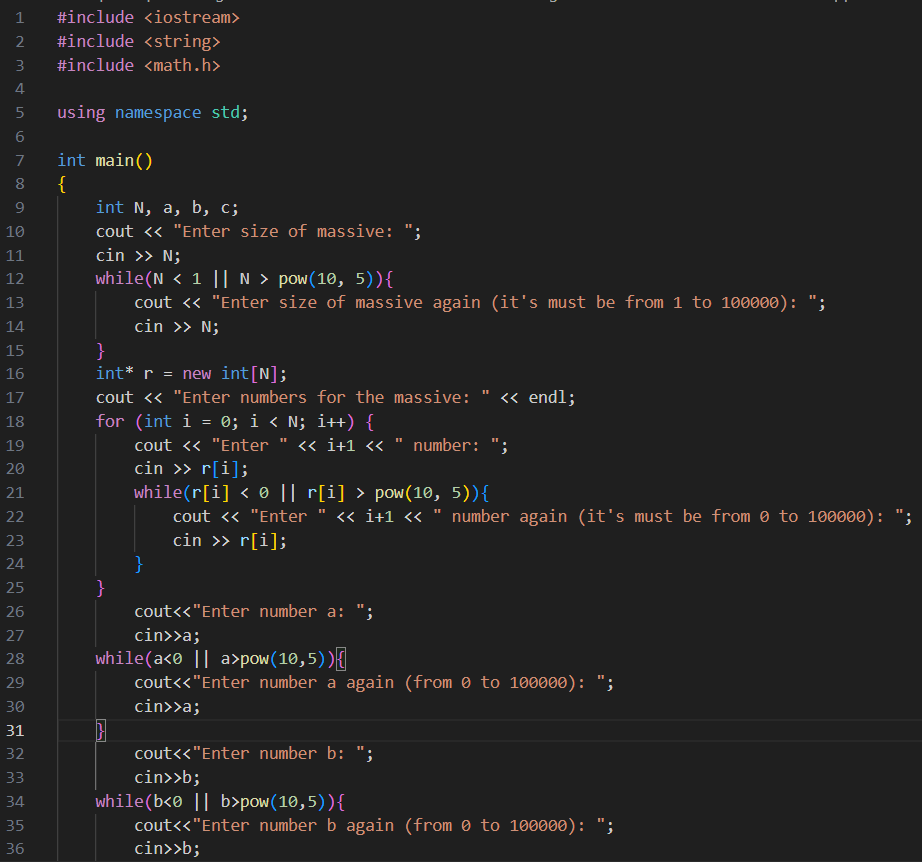
 

Рисунок 8: Код завдання з VNS №5

Завдання №3: Algotester Lab 2



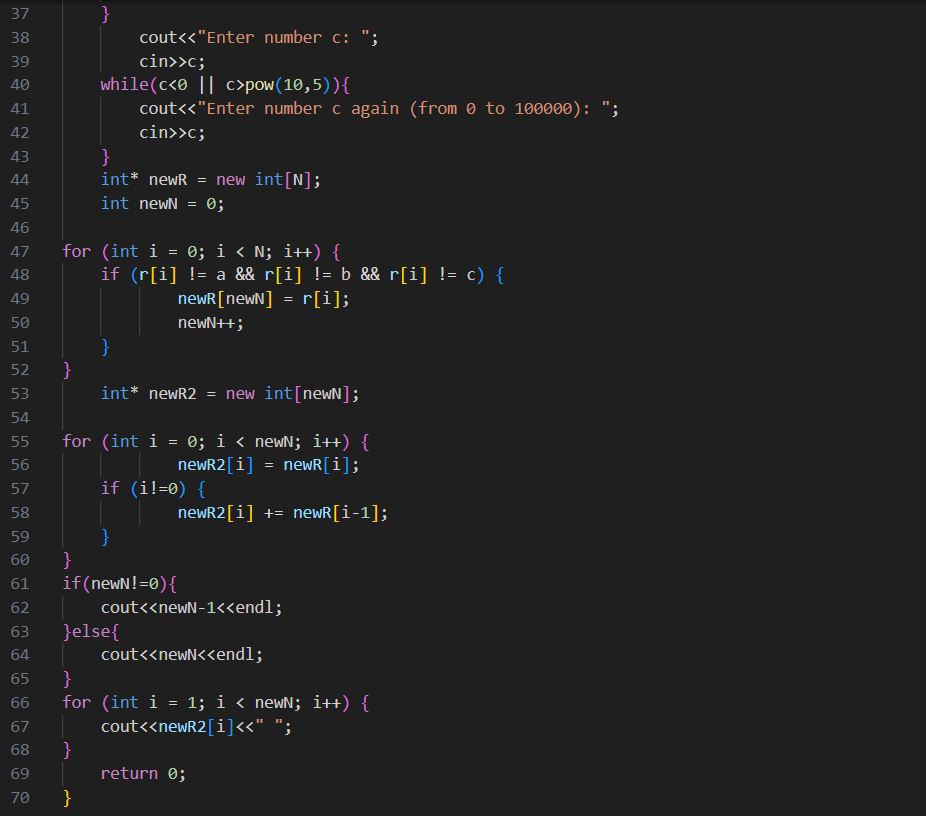
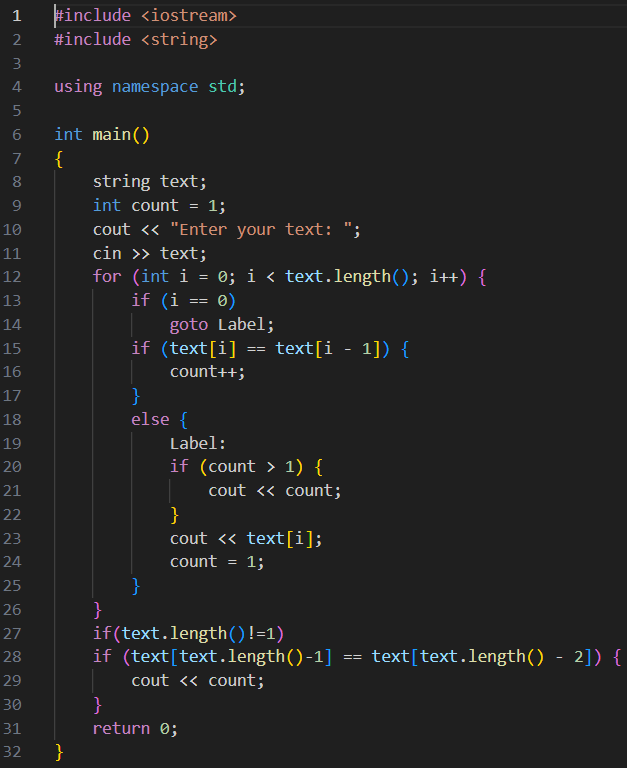
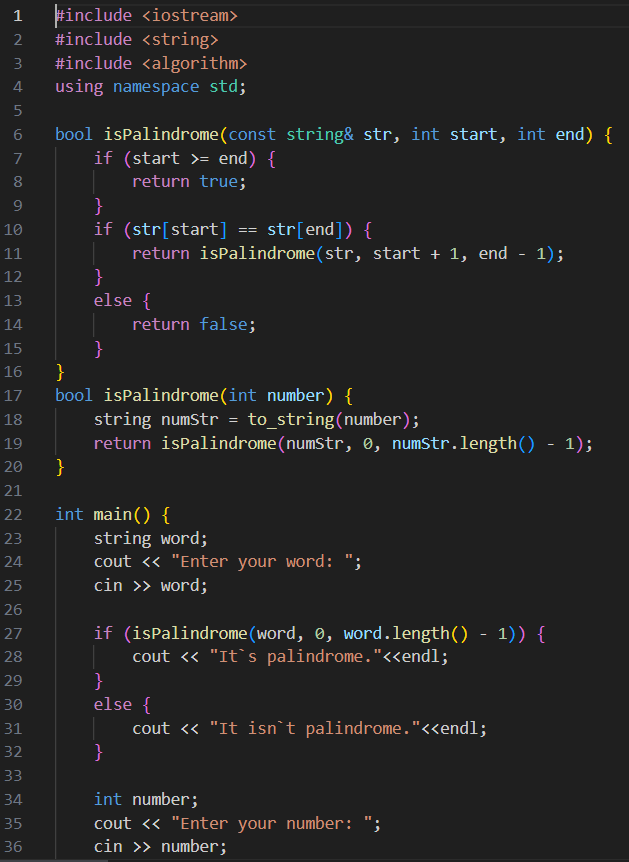


Рисунок 9: Код завдання з Algotester №2

Завдання №4: Algotester Lab 3

  
Рисунок 10: Код завдання з Algotester №3

Завдання №5: Class Practice Task



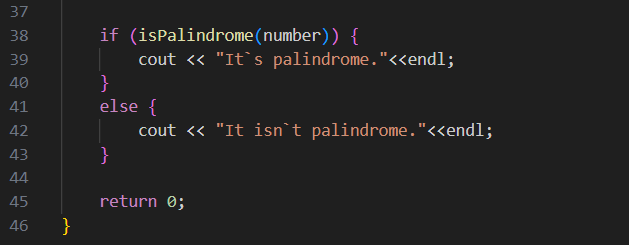
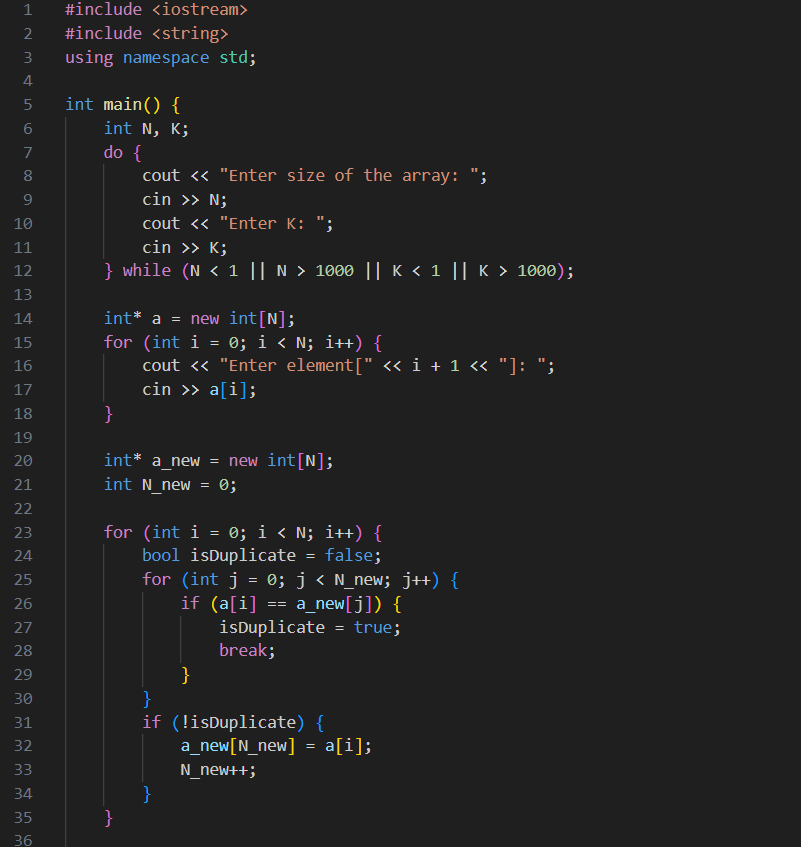


Рисунок 11: Код програми з практичної роботи

Завдання №6: Self Practice Task



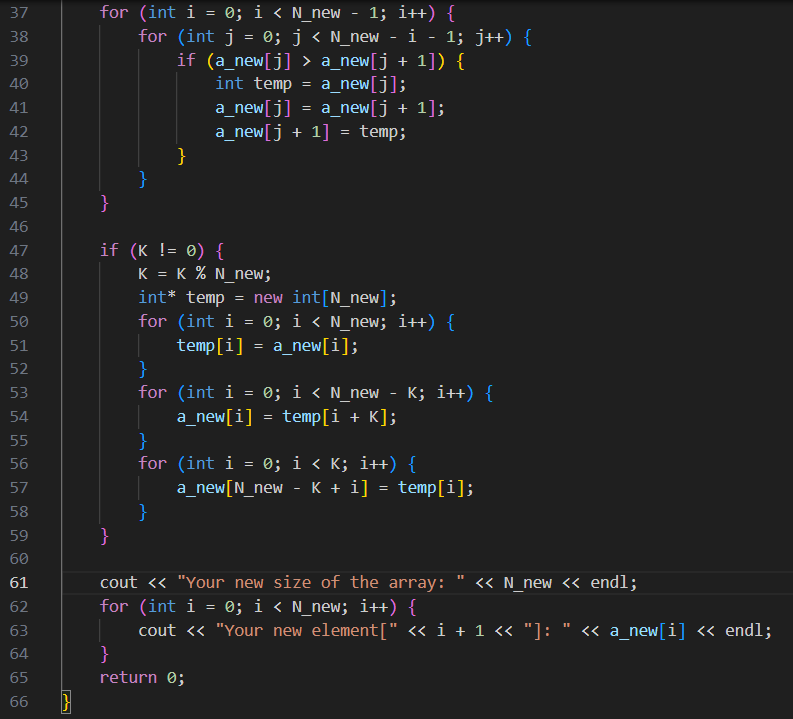


Рисунок 12: Код програми для самостійного опрацювання

Посилання на програми в GitHub: <https://github.com/Zimnov1/Repo/tree/main>

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

Завдання №1: VNS Lab 4

Результати виконання програми:

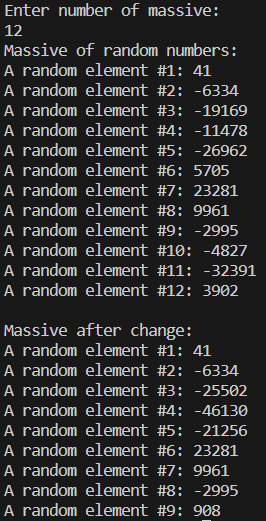


Рисунок 13: Результати виконання програми з VNS №4

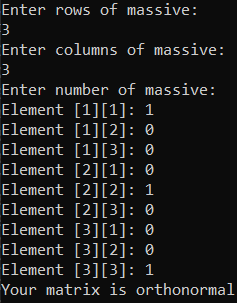
Деталі по виконанню:

* Потрібно врахувати, що масив після змін стає меншим

Час затрачений на виконання завдання: 40 хвилин

Завдання №2: VNS Lab 5

Результати виконання програми:



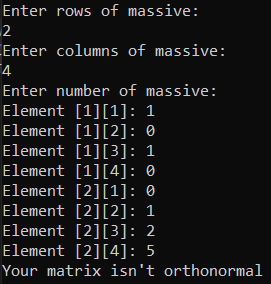


Рисунок 14: Результати виконання програми з VNS №5

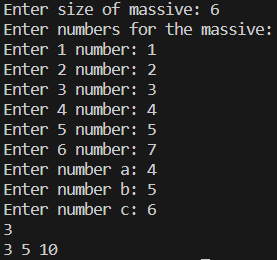
Деталі по виконанню:

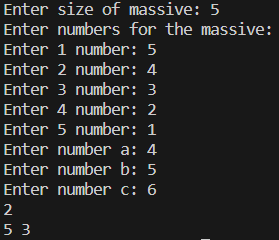
* Рядків і стовпців має бути більше за 1. Для задавання розміру масиву, потрібно це робити два рази і окремо, так як використано два вказівники

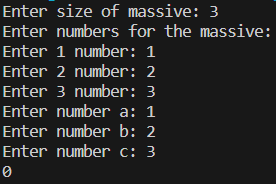
Час затрачений на виконання завдання: 1 година

Завдання №3: Algotester Lab 2

Результати виконання програми:







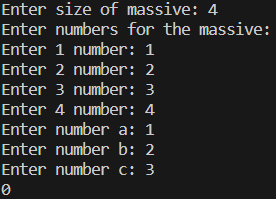


Рисунок 15: Результати виконання програми з Algotester №2

Деталі по виконанню:

* Важливо не віднімати попередній елемент від першого елементу

Час затрачений на виконання завдання: 1 година

Завдання №4: Algotester Lab 3

Результати виконання програми:







Рисунок 16: Результати виконання програми з Algotester №3

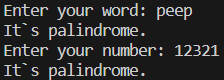
Деталі по виконанню:

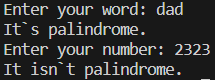
* Потрібно врахувати, що розмір тексту може бути один

Час затрачений на виконання завдання: 1 година і 30 хвилин

Завдання №5: Class Practice Task

Результати виконання програми:





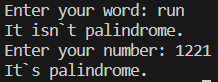


Рисунок 17: Результати виконання практичної роботи

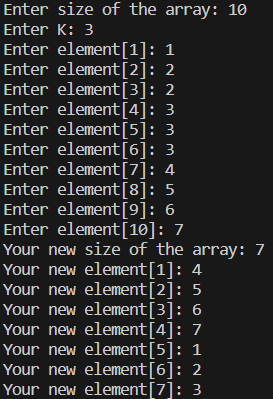
Деталі по виконанню:

* Використав функцію to\_string() для перетворення числа в тип string для подальшого порівнювання. Для цієї функції була підключена бібліотека algorithm

Час затрачений на виконання завдання: 2 години

Завдання №6: Self Practice Task

Результати виконання програми:



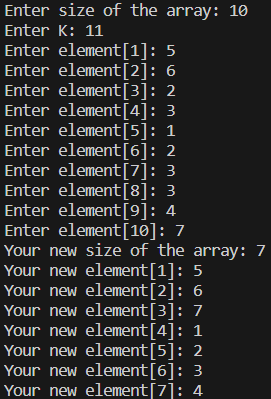


Рисунок 18: Результати виконання роботи для самостійного опрацювання

Деталі по виконанню:

* Було зроблено альтернативну версію програми використовуючи unique, sort та rotate.

Час затрачений на виконання завдання: 2 години і 30 хвилин

## **5. Кооперація з командою:**

Знімок конференції:

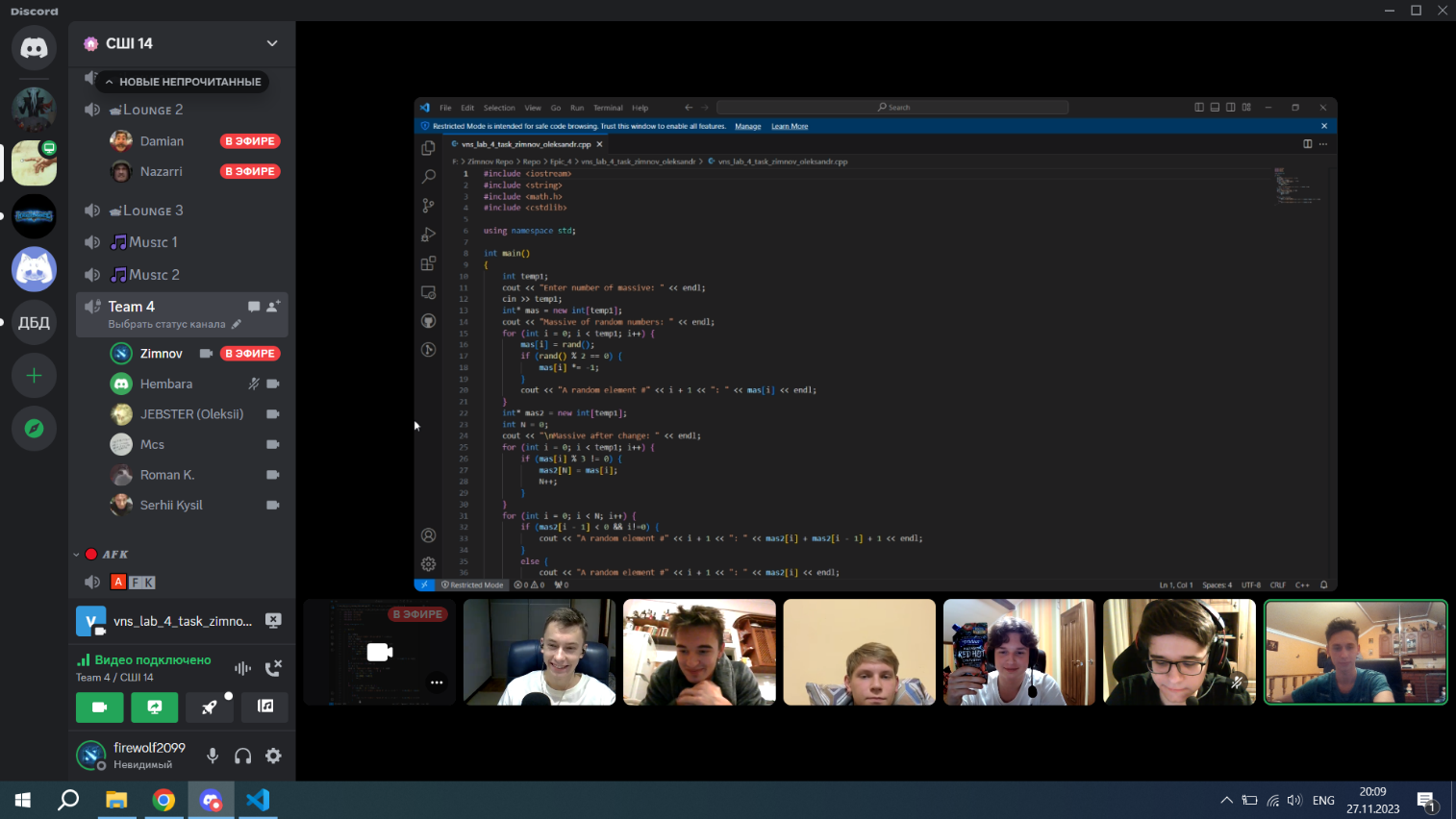


Рисунок 19: Знімок зустрічі з командою

Знімок Trello:

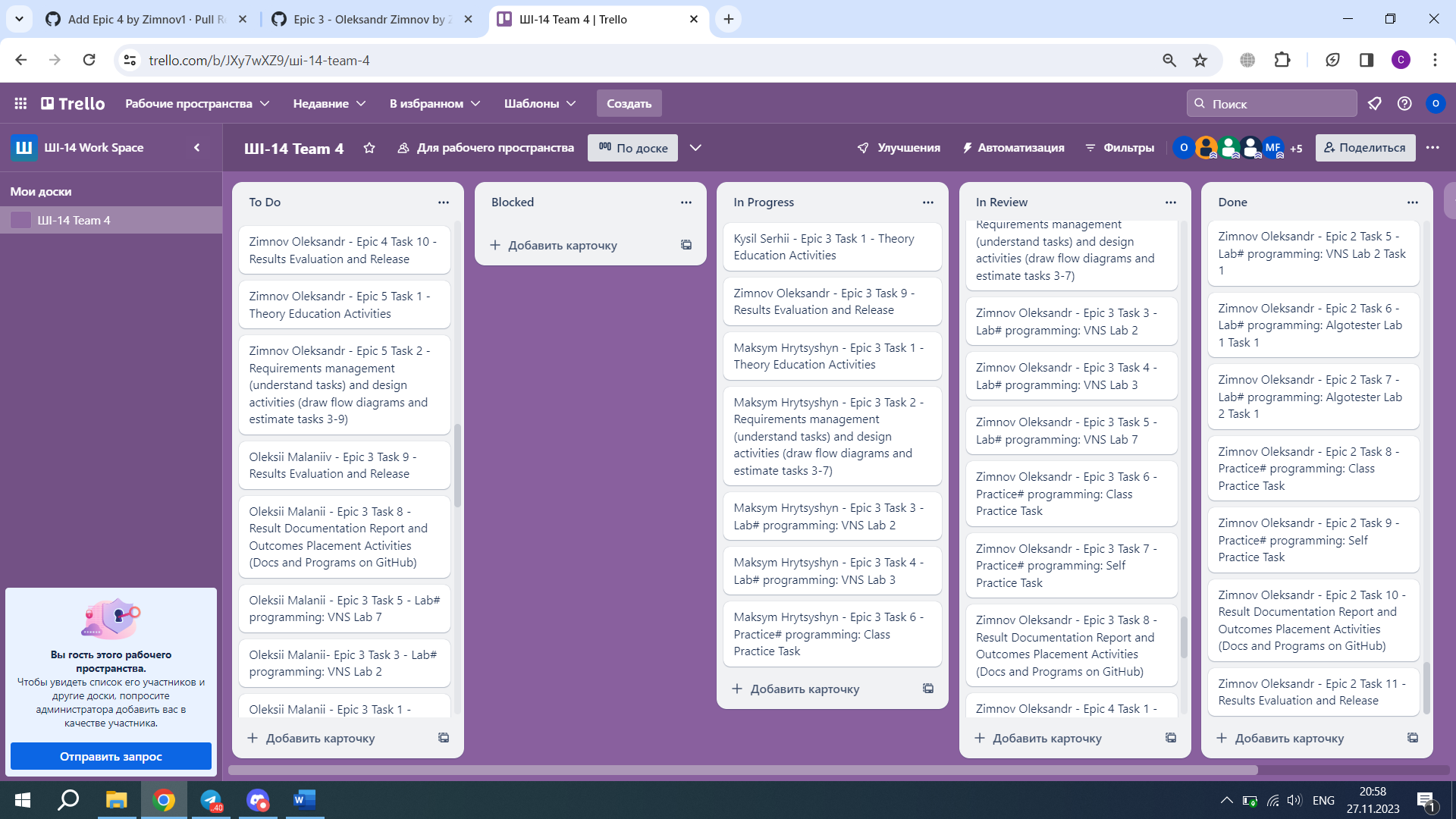


Рисунок 20: Завдання в Trello

Знімок коментарів від команди:

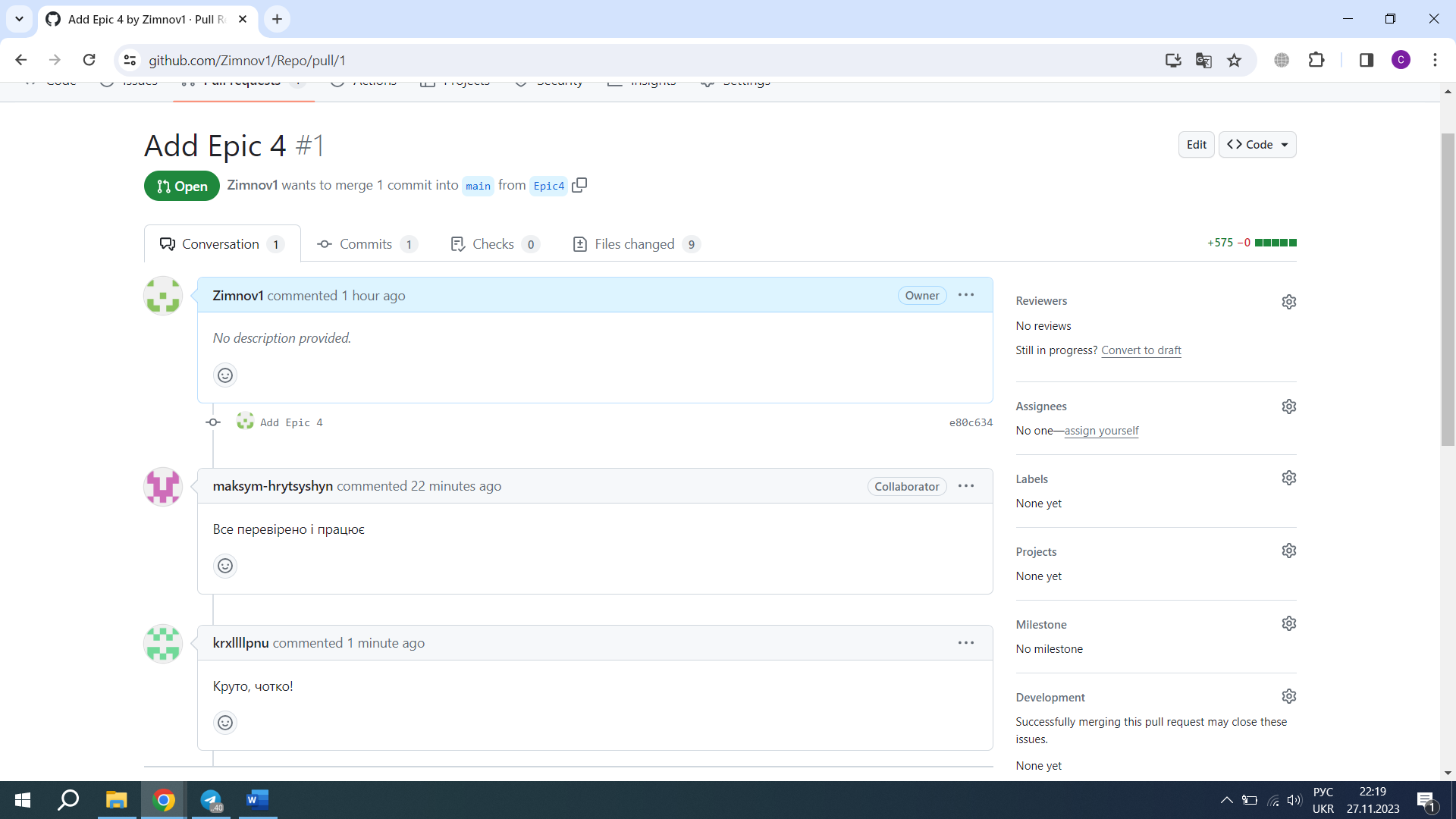


Рисунок 21: Коментарі від команди

# **Висновки:**

Протягом лабораторних та практичних робіт №4було вивчено прості структури даних, такі як: Array, Structure, Vector та List, також було вивчено одновимірні масиви, двовимірні масиви, а також алгоритми обробки даних: сортування, пошук, хешування та дерева

Всі завдання були відслідковані в дошці на Trello. Були онлайн зустрічі з командою, в яких ми владнали всі непорозуміння.

Посилання на Pull request: https://github.com/artificial-intelligence department/ai\_programming\_playground/pull/506