Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт №5**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Файли. Системи числення. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек»

***Виконав:***

студент групи ШІ-14

Зімнов Олександр Сергійович

# **Тема роботи:**

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

# **Мета роботи:**

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Робота з [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ами
* Тема №2: Динамічні [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)труктури даних
* Тема №3: Бібліотеки

1. Індивідуальний план опрацювання теорії:

* Тема №1: Робота з [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ами
  + Джерела Інформації
    - Лекції
    - Лекційний матеріал на ВНС
  + Що опрацьовано:
    - О[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)нови роботи з [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ами.
    - Ввід і вивід [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)труктурованих [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ів.
    - Потоковий ввід-вивід. [Відкриття й закриття потоку](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210920&displayformat=dictionary). Блоковий ввід-вивід.
    - Робота з тек[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)товими [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ами. О[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)новні функції [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ової [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)и[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)теми.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 26.11.2023
  + Звершення опрацювання теми: 27.11.2023
* Тема №2: Динамічні [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)труктури даних
  + Джерела Інформації:
    - Лекції
    - Лекційний матеріал на ВНС
  + Що опрацьовано:
    - [Лінійний однонаправлений список](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210912&displayformat=dictionary).
    - Робота із двонаправленими [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)пи[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)ками.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 28.11.2023
* Тема №3: Бібліотеки
  + Джерела Інформації:
    - Лекції
    - Лекційний матеріал на ВНС
  + Що опрацьовано:
    - Стандартна бібліотека та деталі/методи роботи з файлами.
    - Створення й використання бібліотек.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 28.11.2023
  + Звершення опрацювання теми: 29.11.2023

# **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 3-9)

* Створення діаграм для програм в draw io

Завдання №3 - Lab# programming: VNS Lab 6

* Варіант № 4
* 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. Текст містить не більше 255 символів. Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом. Надрукувати всі слова, які співпадають з її першим словом.
* Важливі деталі для врахування в імплементації програми: Кожне слово відокремлюється пробілом, що допоможе легше порівнювати слова, також в кінці ставиться крапка, що допоможе зрозуміти коли стрічка закінчиться.

Завдання №4 - Lab# programming: VNS Lab 8

* Варіант № 4
* Деталі завдання: Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вмістиме, виконати знищення й додавання елементів у відповідності зі своїм варіантом, використовуючи для пошуку елементів що знищуються чи додаються, функцію. Формування, друк, додавання й знищення елементів оформити у вигляді функцій. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.
* Структура "Людина":
* прізвище, ім'я, по батькові;
* домашня адреса;
* номер телефону;
* вік.

Знищити усі елементи із заданим віком, додати елемент після елемента із заданим номером.

* Важливі деталі для врахування в імплементації програми: Всі дії над структурою потрібно здійснювати через відповідну функцію, також при помилці відкриття файлу.

Завдання №5 - Lab# programming: VNS Lab 9

* Варіант № 4
* Деталі завдання: Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію. Скопіювати з файлу F1 у файл F2 рядки, починаючи з 4. Підрахувати кількість символів в останньому слові F2.
* Важливі деталі для врахування в імплементації програми: Рядки можна ввести власноруч, а не очікувати їх від користувача.

Завдання №6 - Lab# programming: Algotester Lab 4

* Варіант № 2
* 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. Потрібно перевірити чи K не більше N, якщо так, то можна найти остачу від ділення K на N

Завдання №7 - Lab# programming: Algotester Lab 6

* Варіант № 2
* Деталі завдання: У вас є шахова дошка розміром 8×8 та дуже багато фігур. Кожна клітинка може мати таке значення:
* Пуста клітинка O
* Пішак P
* Тура R
* Кінь N
* Слон B
* Король K
* Королева Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1) Далі йдуть Q запитів з координатами клітинки {x,y}. На кожен запит ви маєте вивести стрічку si - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз). У випадку, якщо на клітинці стоїть якась фігура - виведіть символ X. У випадку, якщо клітинку не атакують - виведіть O. Наявніть фігури у певній клітинці не блокує атаку для іншої фігури. Тобто якщо між турою та клітинкою стоїть інша фігура - вважається що тура атакує цю клітинку.

* Важливі деталі для врахування в імплементації програми: |rowi|=N, rowi ∈ {O,P,R,N,B,K,Q}, 1≤Q≤64, 1≤x,y≤8

Завдання №8 - Practice# programming: Class Practice Task

* Деталі завдання: У вас Реалізувати функцію створення файлу і запису в нього даних: 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 файлу.

* Важливі деталі для врахування в імплементації програми: Статус операції має проходити через enum. При створенні файлу з ім`ям вже існуючого файлу потрібно переписати його з новим, вписаним користувачем, змістом.

Завдання №9 - Practice# programming: Self Practice Task

* Деталі завдання: У вас є карта гори розміром N×M. Також ви знаєте координати {x,y}, у яких знаходиться вершина гори. Ваше завдання - розмалювати карту таким чином, щоб найнижча точка мала число 0, а пік гори мав найбільше число. Клітинки які мають суміжну сторону з вершиною мають висоту на один меншу, суміжні з ними і не розфарбовані мають ще на 1 меншу висоту і так далі.
* Важливі деталі для врахування в імплементації програми: 1≤N,M≤103, 1≤x≤N, 1≤y≤M, спочатку варто знайти число найвищої точки, після цього згенерувати гору буде простіше

Завдання №10 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

* Створення звіту і завантажити його з програмами на GitHub

Завдання №11 - Results Evaluation and Release

* Захист звіту.

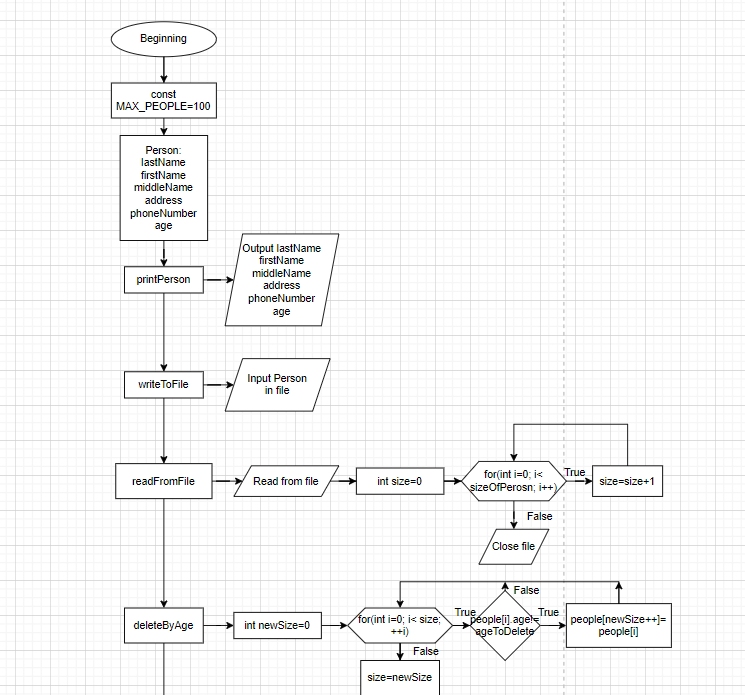
## **2. Дизайн та планована оцінка часу виконання завдань:**

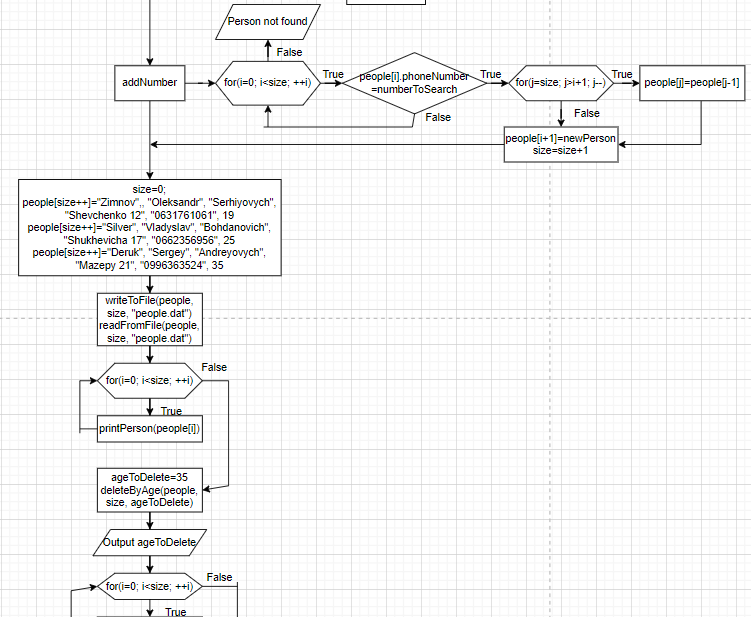
Програма №1 VNS Lab 6

* Час на виконання: 2 години
* Важливі деталі для врахування в імплементації програми: Були використані функції strtok\_s для розбиття слів на токени для більш зручної перевірки, а також strcmp для самої перевірки

Програма №2 VNS Lab 8

* Блок-схема





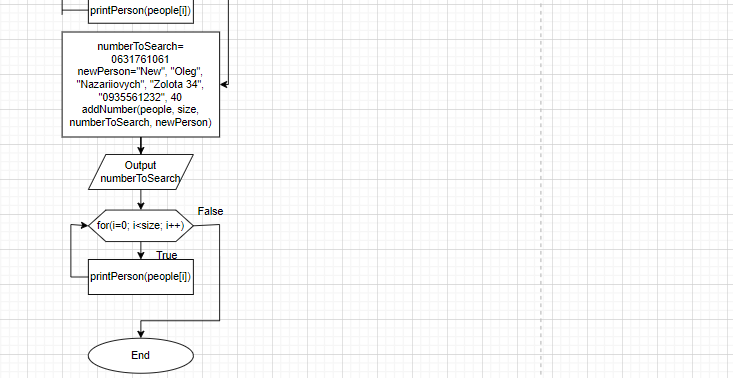


Рисунок 1: Блок схема до програми №2

* Час на виконання: 3 години
* Важливі деталі для врахування в імплементації програми: В програмі досить багато функцій, важливо коректно передавати дані через них. А також потрібно збільшувати кількість персон лише при їх добавленні

Програма №3 VNS Lab 9

* Час на виконання: 1 година
* Важливі деталі для врахування в імплементації програми: sizeof повертає кількість бітів, що добре підходить для виведення рядка через getline, коли ж для підрахування кількості букв краще підходить length()

Програма №4 Algotester Lab 4

* Час на виконання: 1 година і 30 хвилин
* Важливі деталі для врахування в імплементації програми: sizeof відображає кількість байтів, це не допоможе в вимірі довжини рядка. В довжині рядку приймає участь і його кінець, тобто місце під новий рядок.

Програма №5 Algotester Lab 6

* Час на виконання: 6 годин
* Важливі деталі для врахування в імплементації програми: Якщо координати за межами масиву, то можна одразу вивести «О», так само якщо координати вказані на якійсь фігурі, то можна одразу вивести «Х». Якщо багато фігур стають в одну незайняту клітинку, то пітрібно вивести їх в алфавітному порядку і без повторень

Програма №6 Class Practice Task

* Час на виконання: 1 година
* Важливі деталі для врахування в імплементації програми: Для зчитування рядка використовується функція getline(), а для перевірки чи файл відкрився is\_open().

Програма №7 Self Practice Task

* Час на виконання: 2 години
* Важливі деталі для врахування в імплементації програми: Щоб виміряти цифру якоїсь частини гори потрібно спершу знайти яке число буде на найбільшій вершині гори

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

Завдання №1: VNS Lab 6

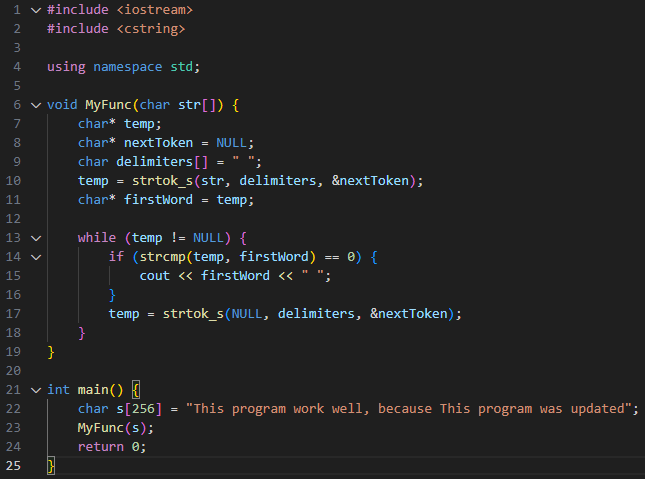
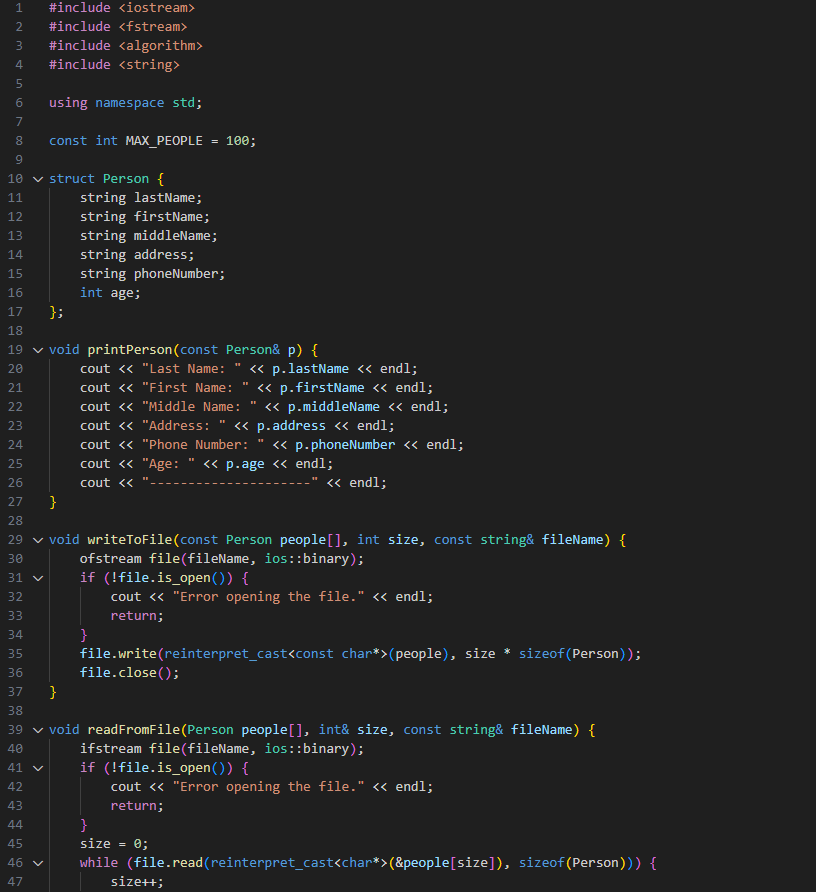


Рисунок 2: Код завдання з VNS №6

Завдання №2: VNS Lab 8



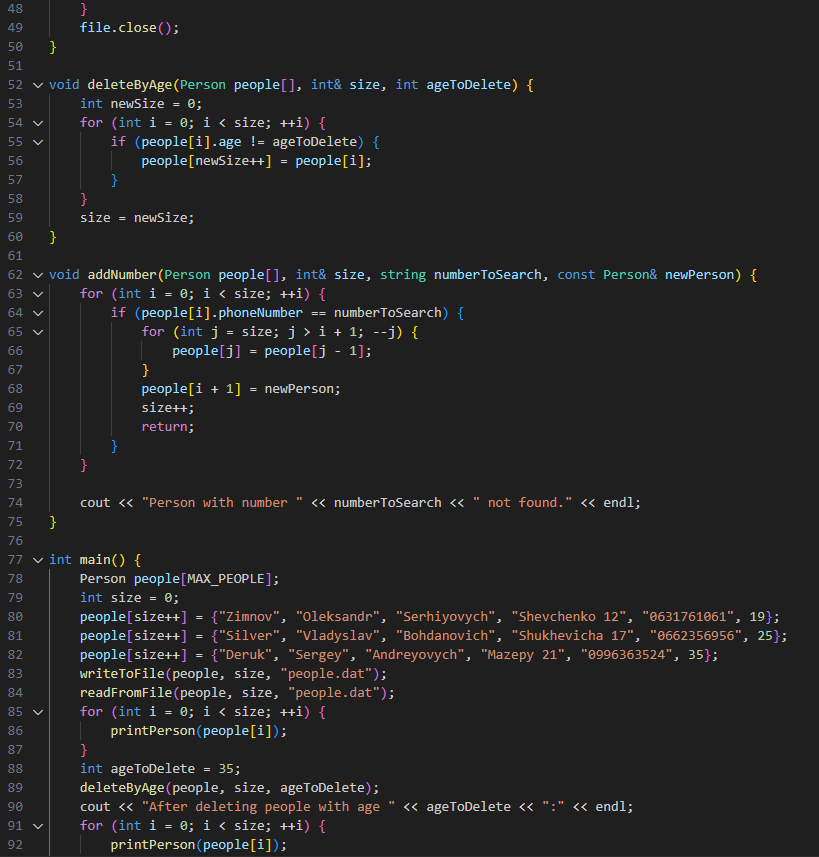
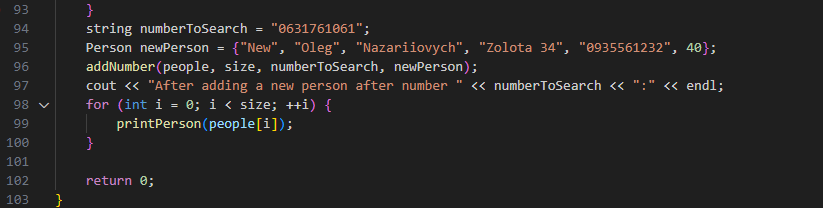
 

Рисунок 3: Код завдання з VNS №8

Завдання №3: VNS Lab 9

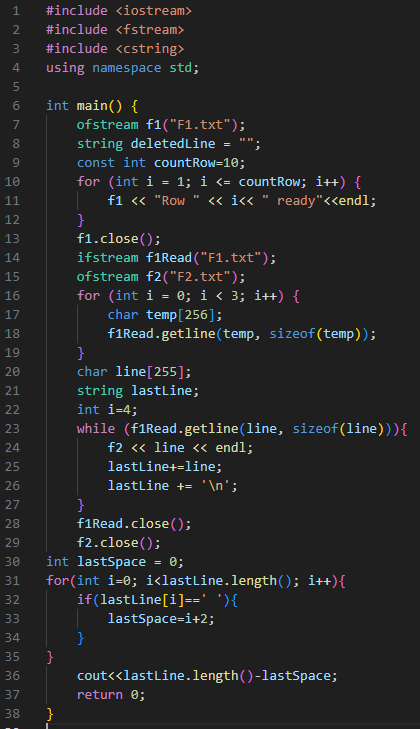
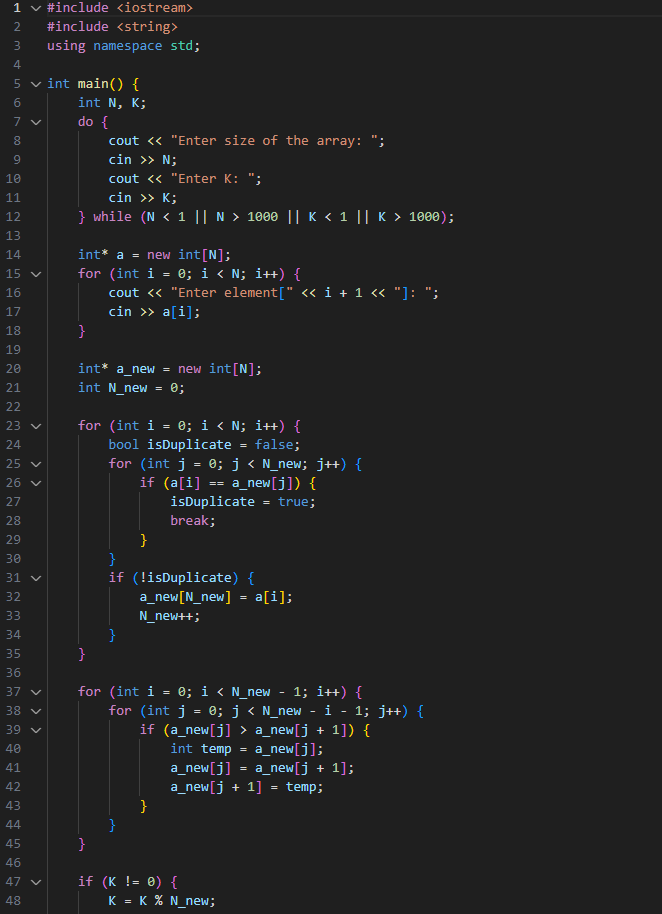


Рисунок 4: Код завдання з VNS №9

Завдання №4: Algotester Lab 4



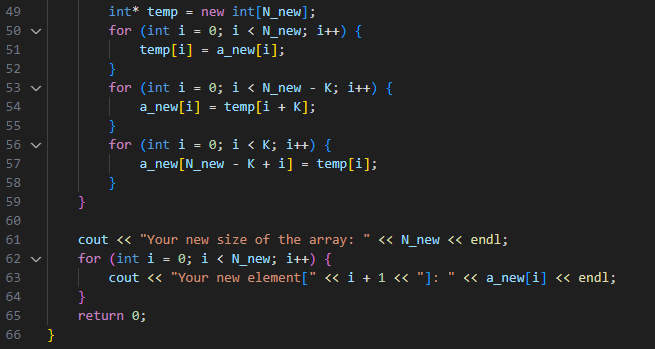
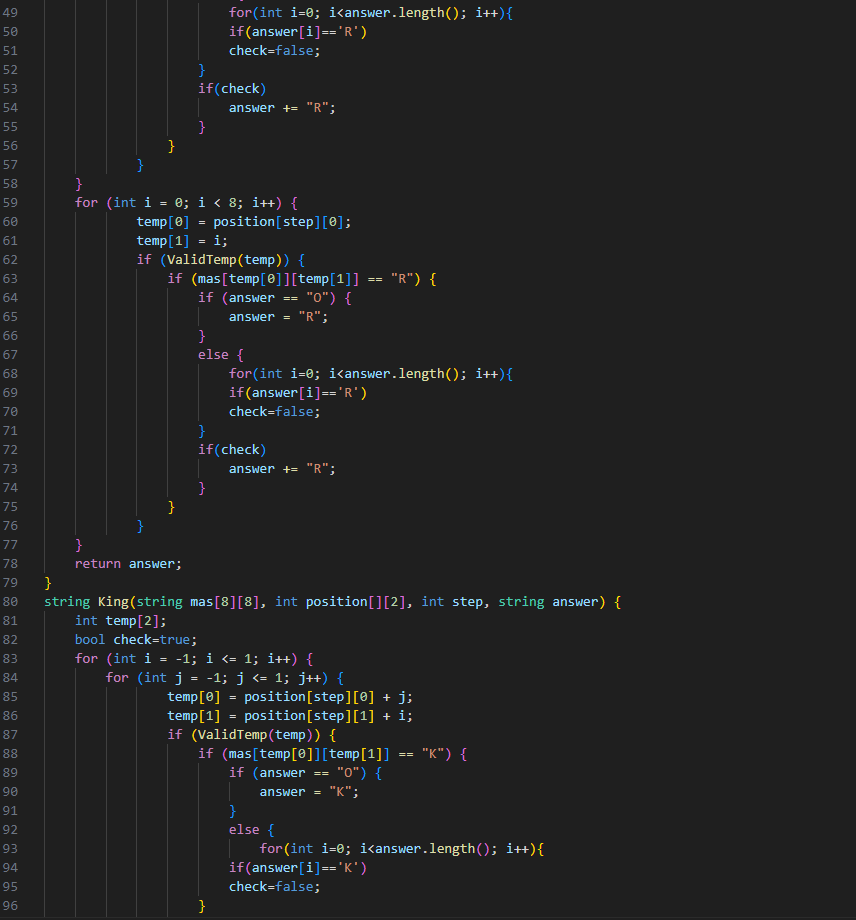
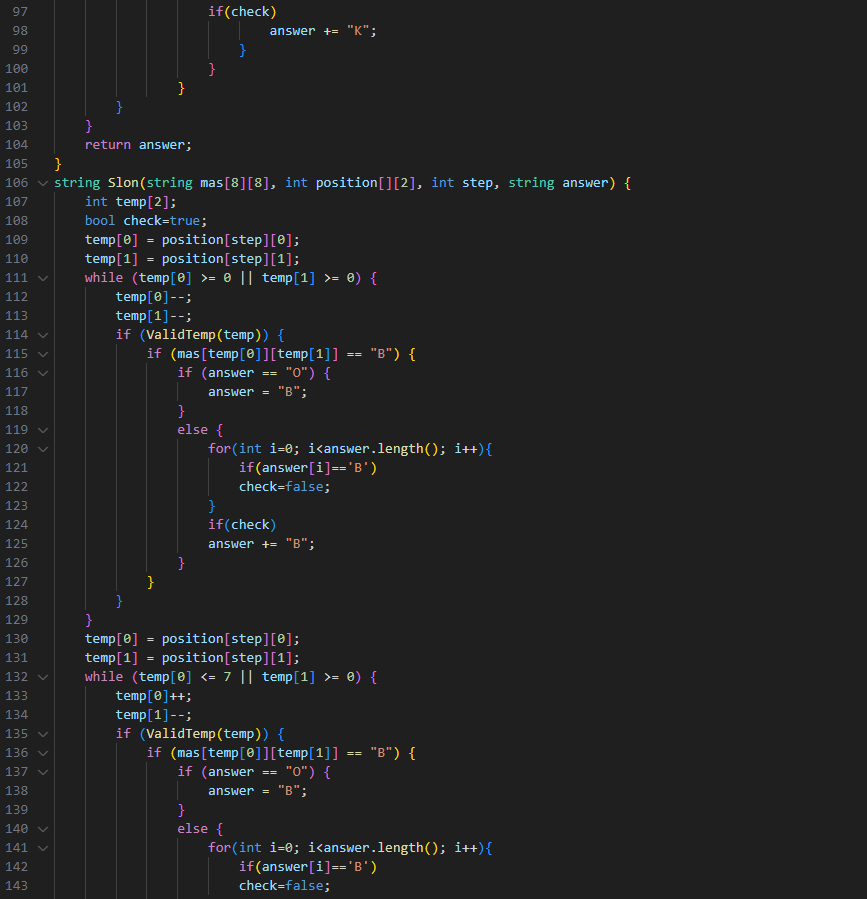
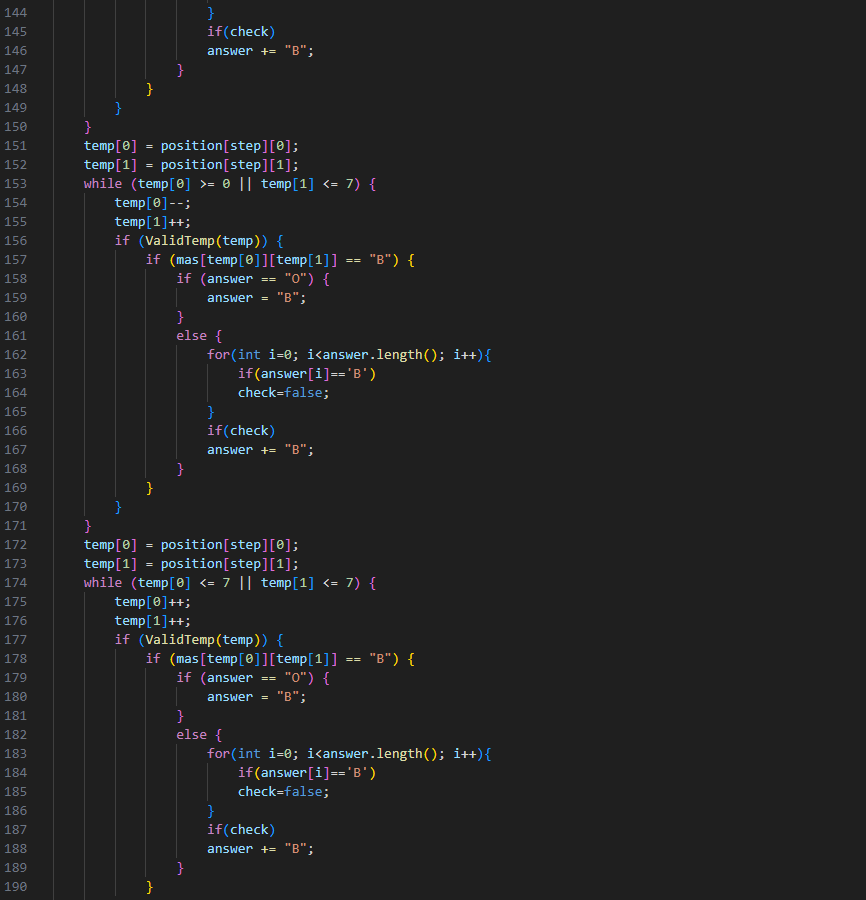
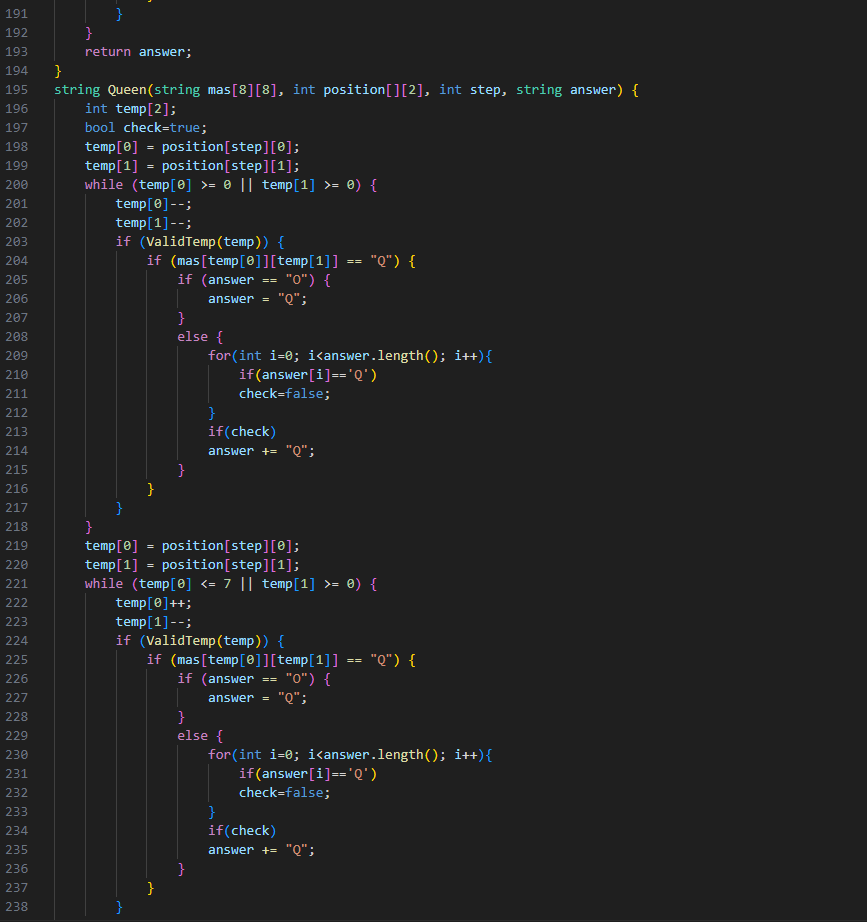
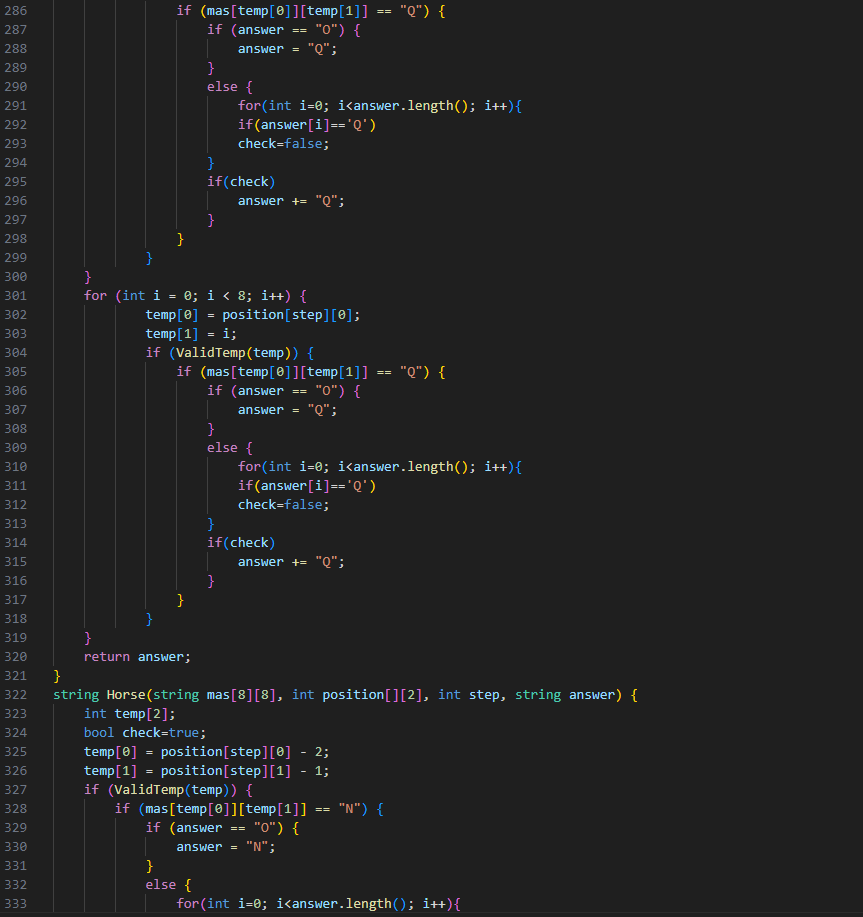
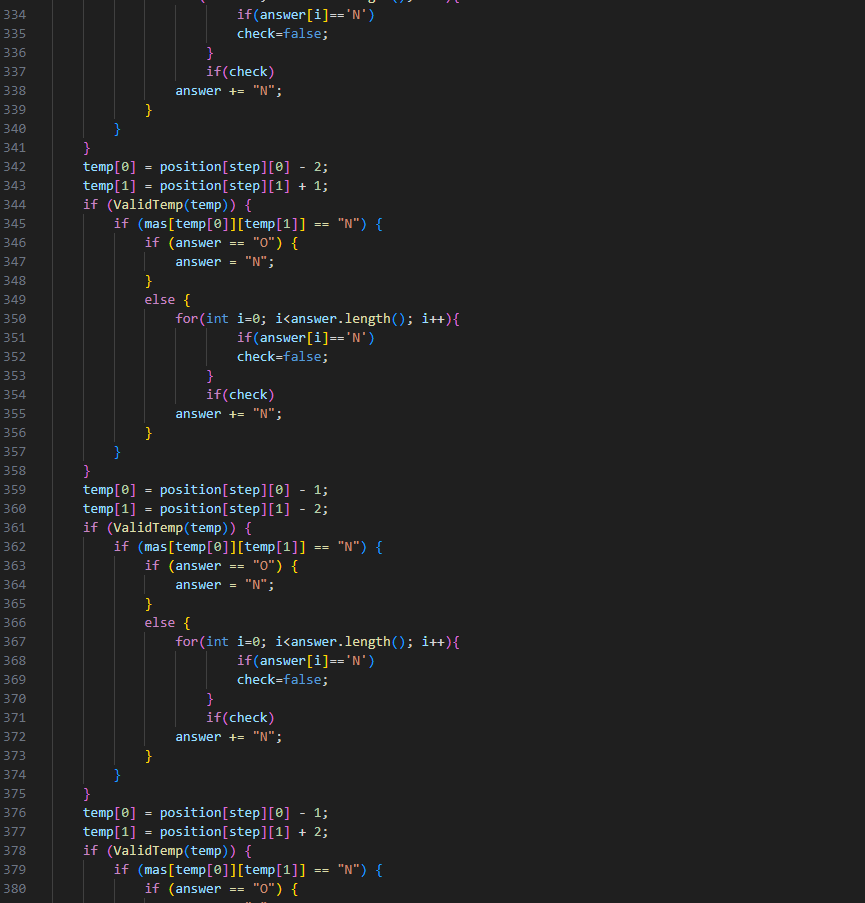
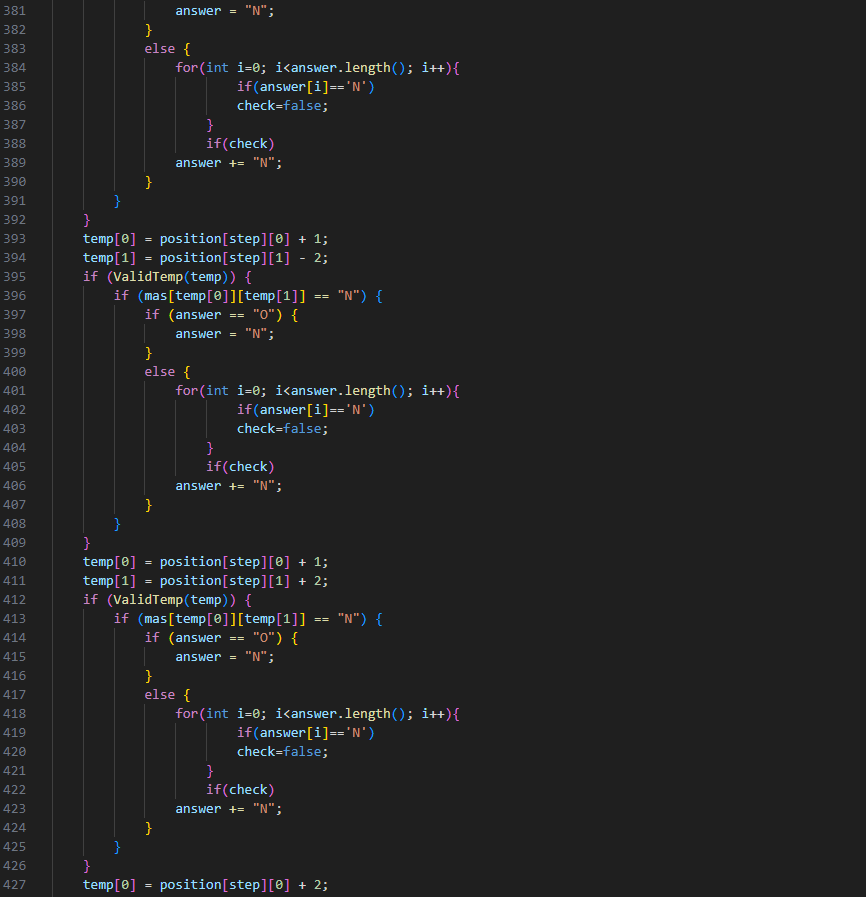
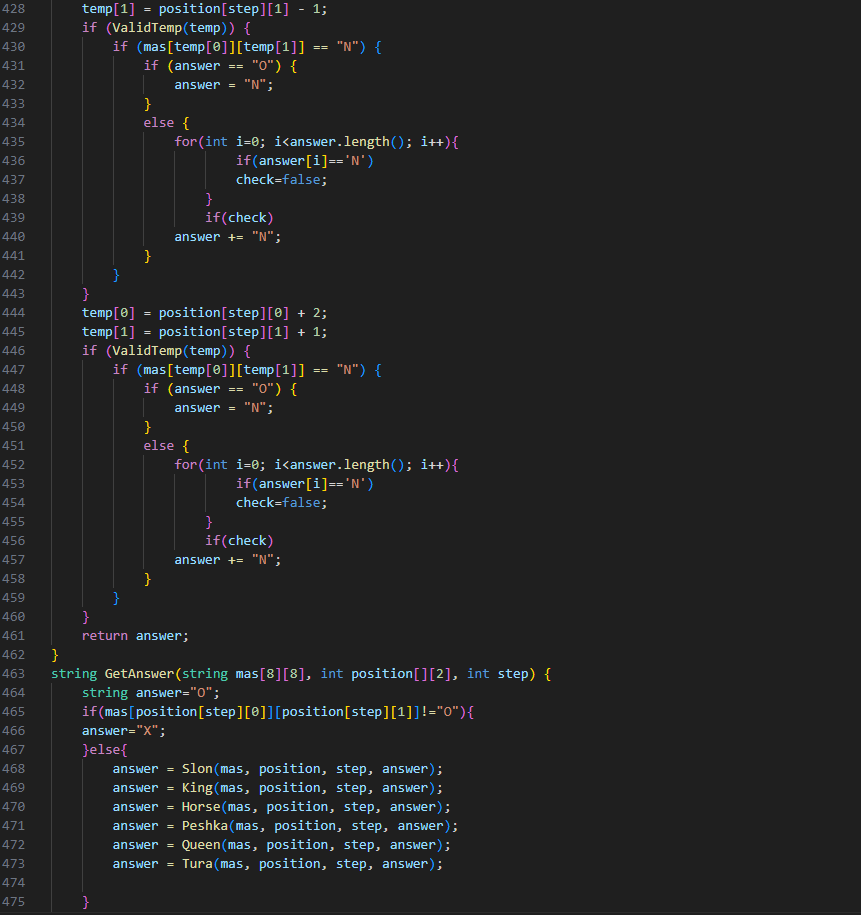
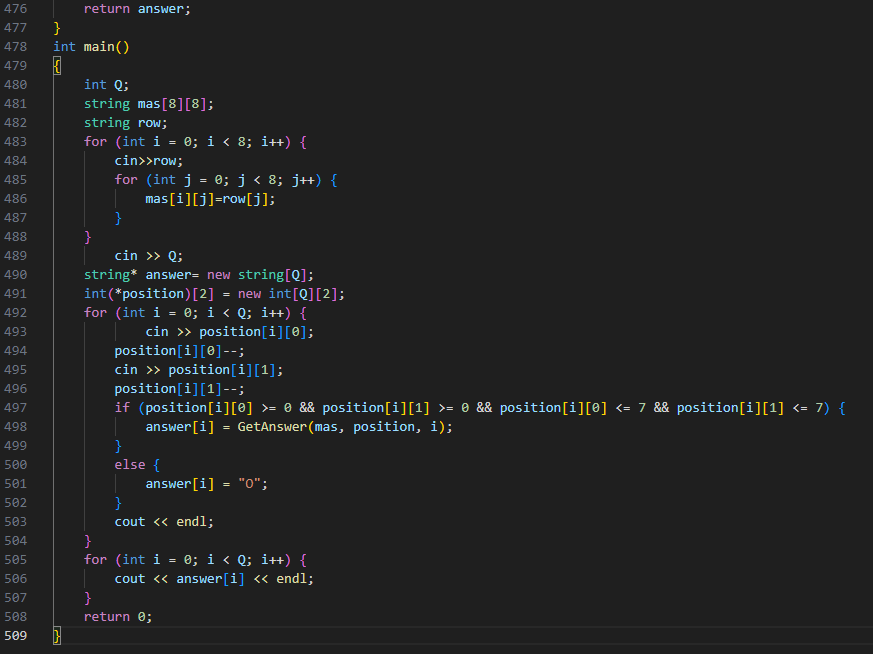


Рисунок 5: Код завдання з Algotester №4

Завдання №5: Algotester Lab 6



           
Рисунок 6: Код завдання з Algotester №6

Завдання №6: Class Practice Task

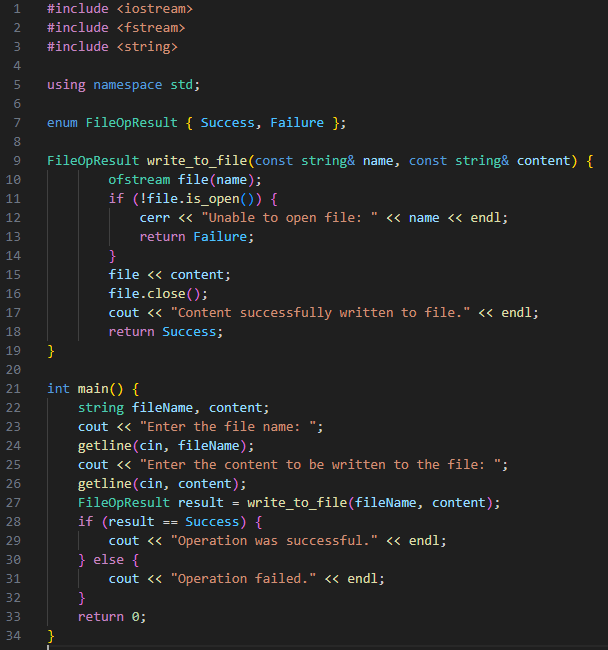


Рисунок 7: Код програми з практичної роботи

Завдання №7: Self Practice Task

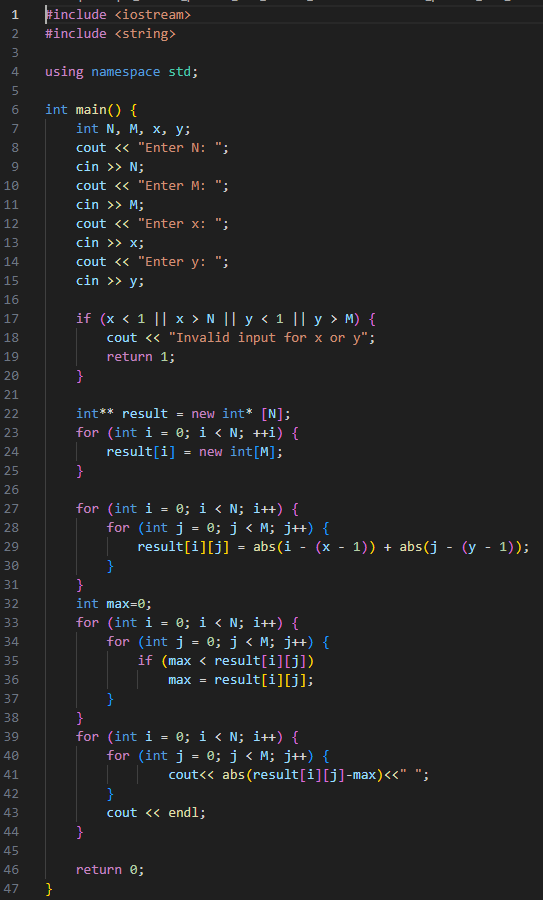


Рисунок 8: Код програми для самостійного опрацювання

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 6

Результати виконання програми:



Рисунок 9: Результати виконання програми з VNS №6

Деталі по виконанню:

* Функція strcmp повертає значення 0, якщо порівнювані значення рівні
* Слова одинакові лише при урахуванні регістру

Час затрачений на виконання завдання: 1 година

Завдання №2: VNS Lab 8

Результати виконання програми:

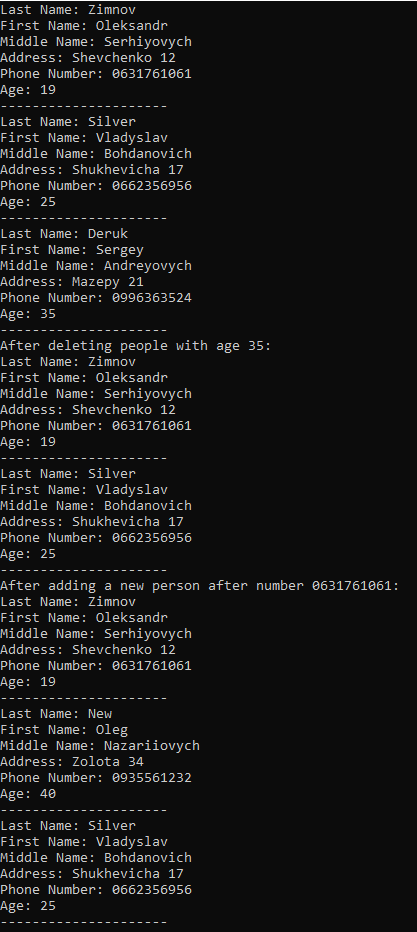


Рисунок 10: Результати виконання програми з VNS №8

Деталі по виконанню:

* Було використано ios::binary для створення двійкового файлу
* Для записування в файл використовується ofstream, а для зчитування ifstream

Час затрачений на виконання завдання: 2 години

Завдання №3: VNS Lab 9



Рисунок 11: Результати виконання програми з VNS №9

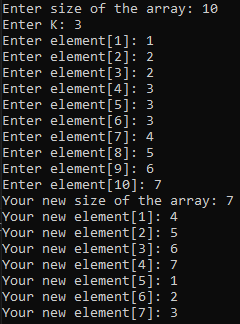
Деталі по виконанню:

* Створено змінну lastLine, яка копіює всі символи після внесення змін. Це використано для того, щоб виміряти кількість букв в останньому слові, для цього lastLine було віднято від координати останнього пробілу

Час затрачений на виконання завдання: 1 година

Завдання №4: Algotester Lab 4

Результати виконання програми:



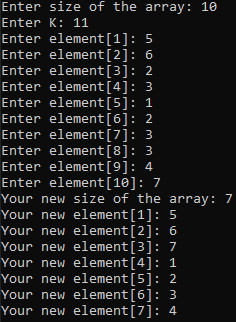


Рисунок 12: Результати виконання програми з Algotester №4

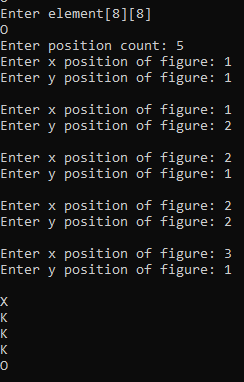
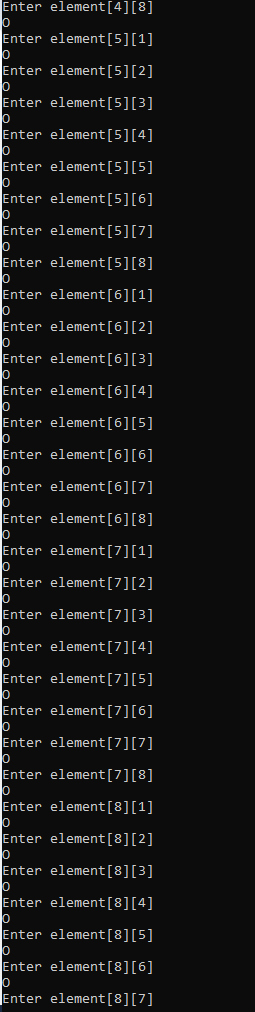
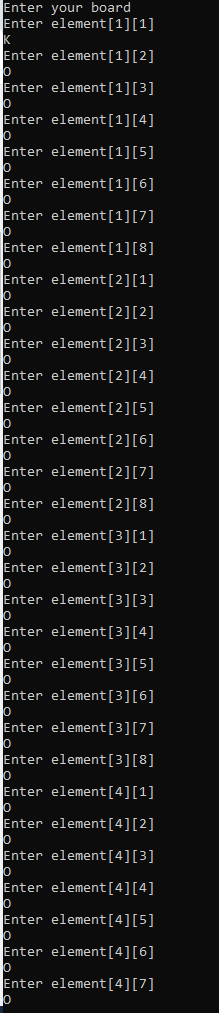
Деталі по виконанню:

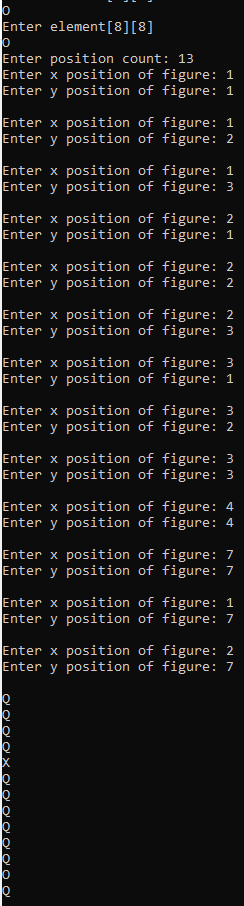
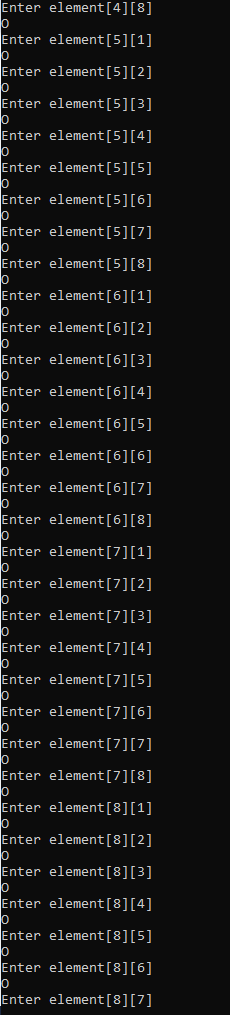
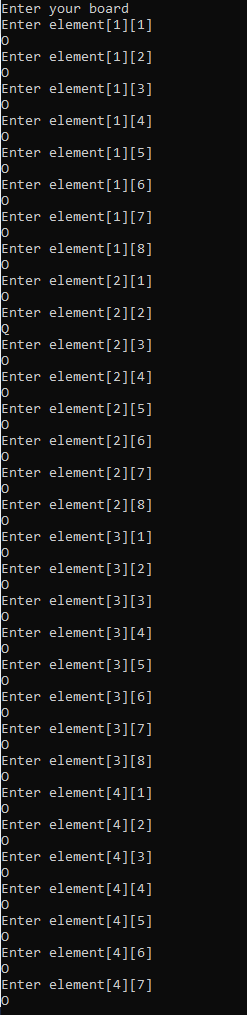
* Для перевірки на повторюваність використовувалась зміна isDuplicate, щоб уникнути проблем з видаленням всіх однакових елементів
* Частину масиву, яку потрібно вписати в кінець копіюється в temp, щоб не загубити значення масиву, які потрібно вписати на початок

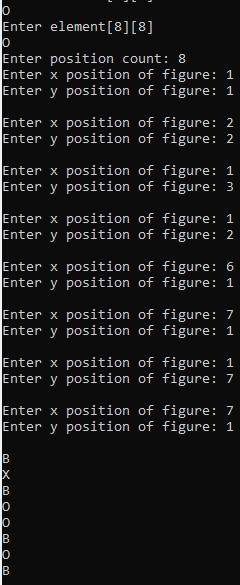
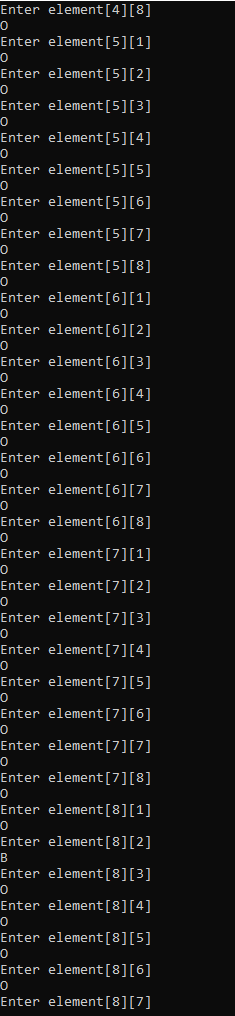
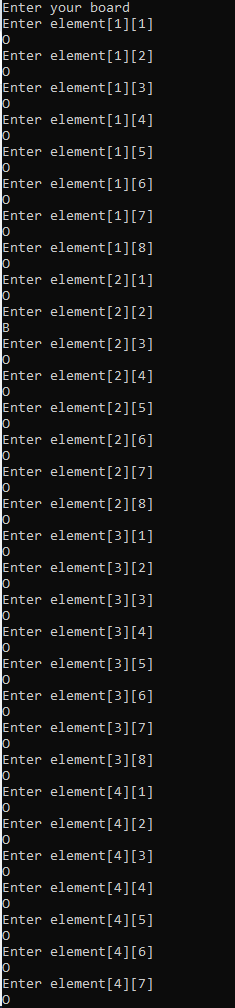
Час затрачений на виконання завдання: 1 година

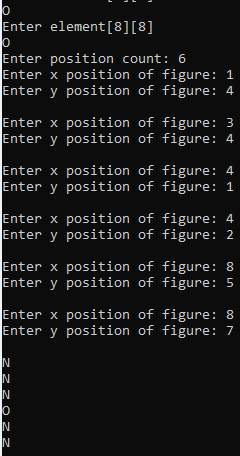
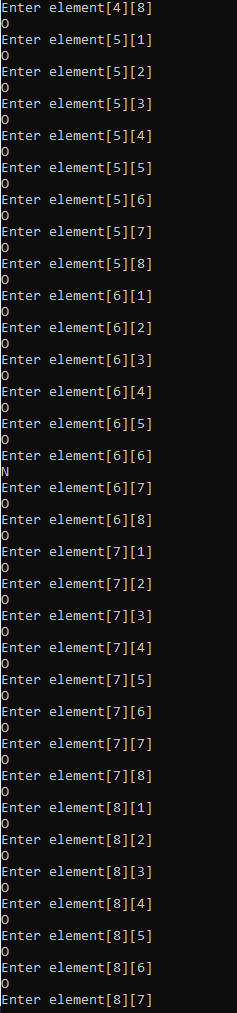
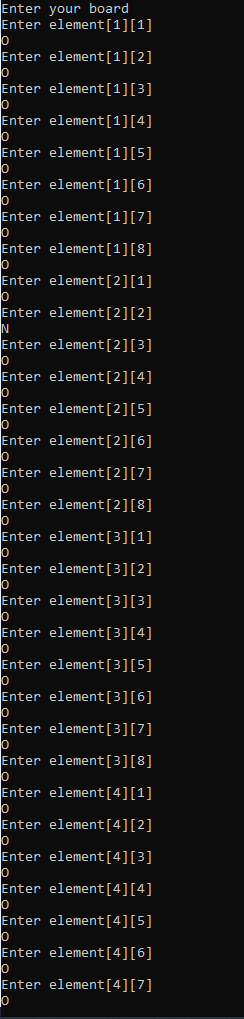
Завдання №5: Algotester Lab 6

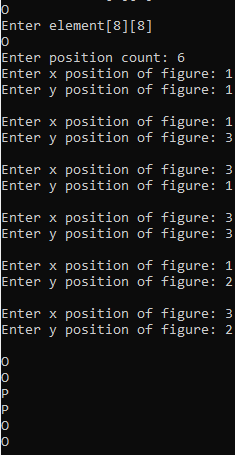
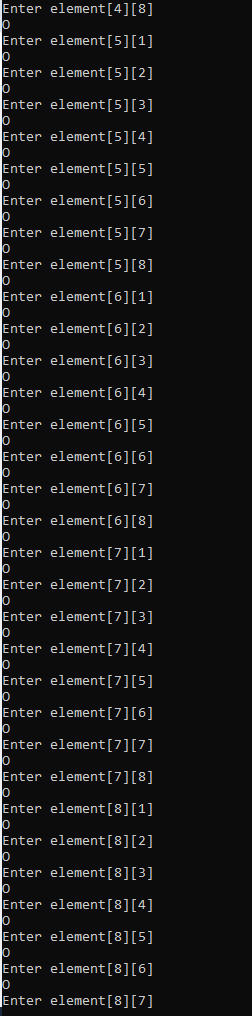
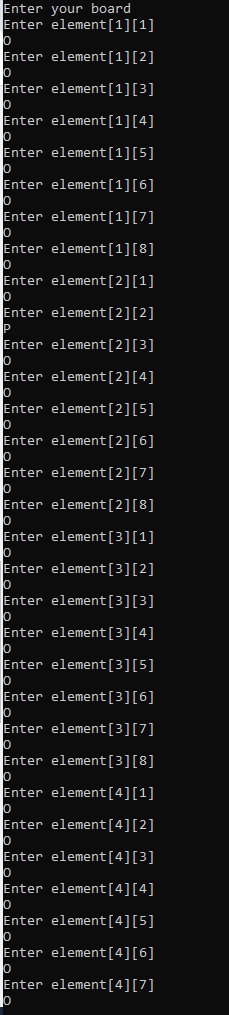
Результати виконання програми:











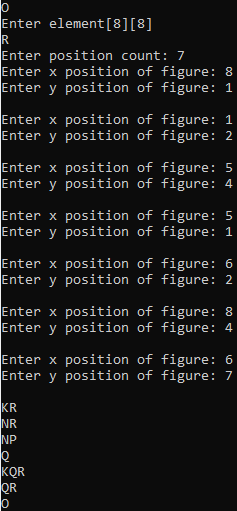
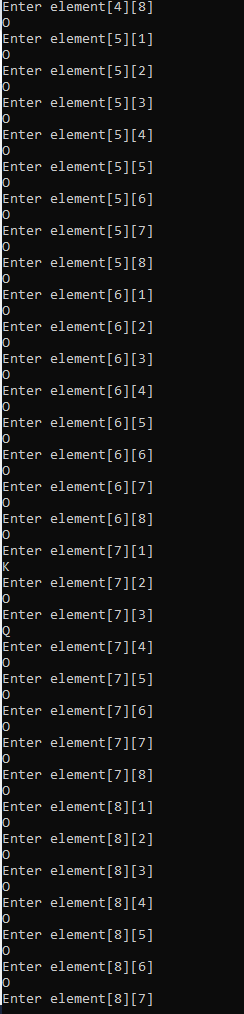
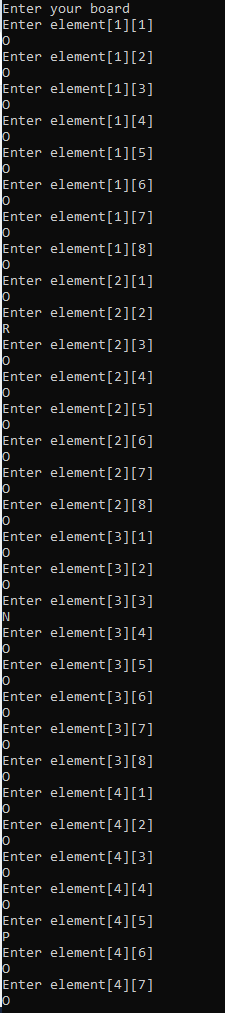


Рисунок 13: Результати виконання програми з Algotester №6

Деталі по виконанню:

* Створено 7 функцій, 6 з них провіряють чи є фігури які можуть встати в задану позицію, для кожної фігури окрема функція. А сьома функція провіряє чи після змін позиція не стає за межами масиву

Час затрачений на виконання завдання: 5 годин

Завдання №6: Class Practice Task

Результати виконання програми:

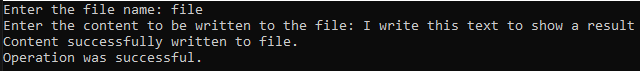


Рисунок 14: Результати виконання практичної роботи

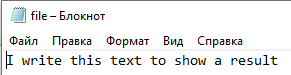


Рисунок 15: Результати виконання практичної роботи в текстовому документі

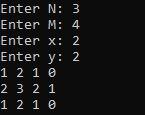
Деталі по виконанню:

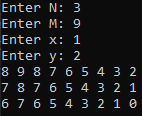
* Було використано enum, який вийшов альтернативою bool для даною функцію, тому що фактично заставляє повернути функцію “true” або “false”.

Час затрачений на виконання завдання: 30 хвилин

Завдання №7: Self Practice Task

Результати виконання програми:





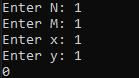


Рисунок 16: Результати виконання роботи для самостійного опрацювання

Деталі по виконанню:

* Перше завдяки модулю числа було знайдено число максимальної вершини і виставлено кожній вершині протилежне значення (максимальна вершина = 0), після чого кожну вершину відняв від максимального значення по модулю, що зробило значення коректними

Час затрачений на виконання завдання: 1 година

## **5. Кооперація з командою:**

Знімок конференції:

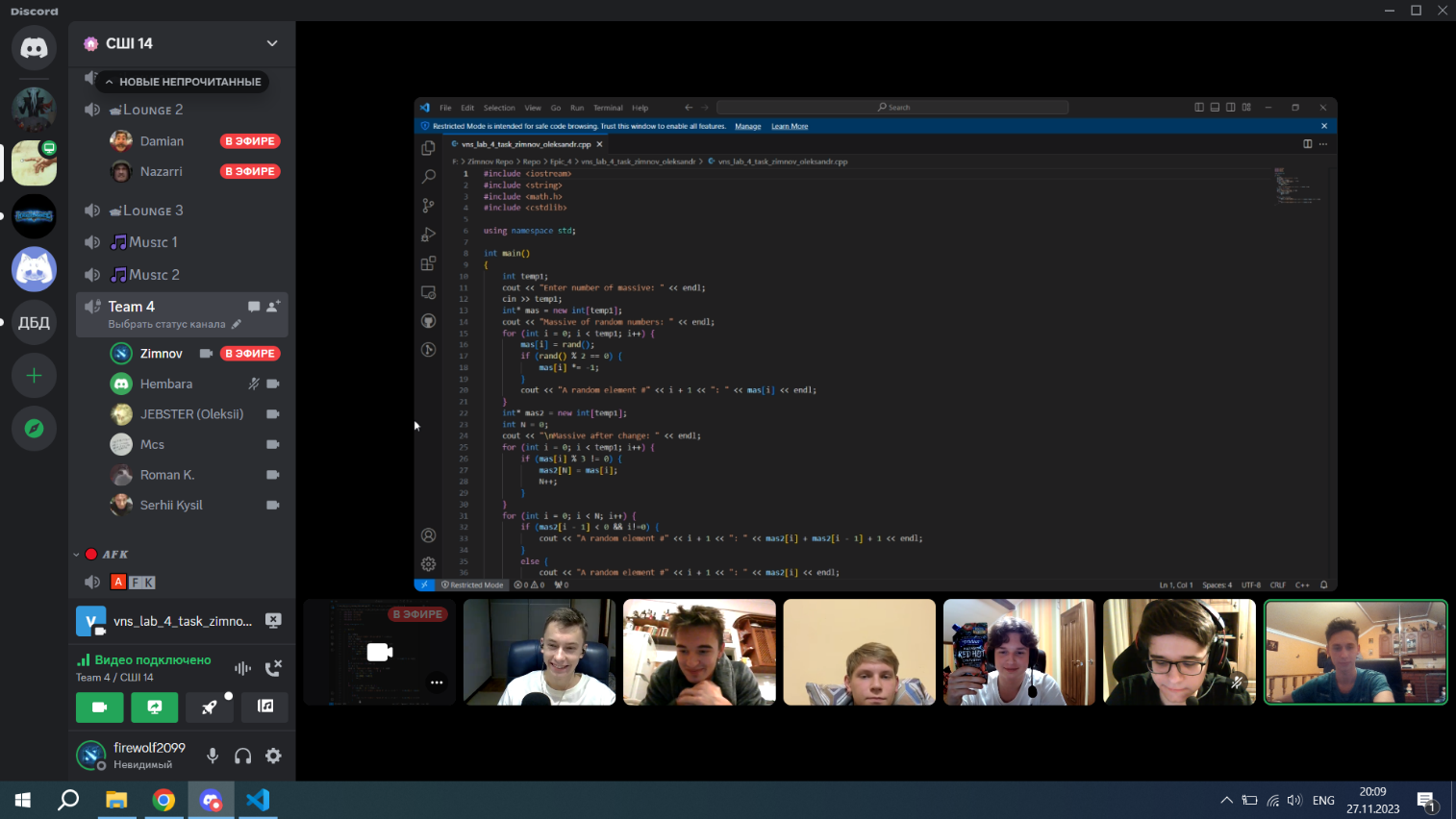


Рисунок 17: Знімок зустрічі з командою

Знімок Trello:

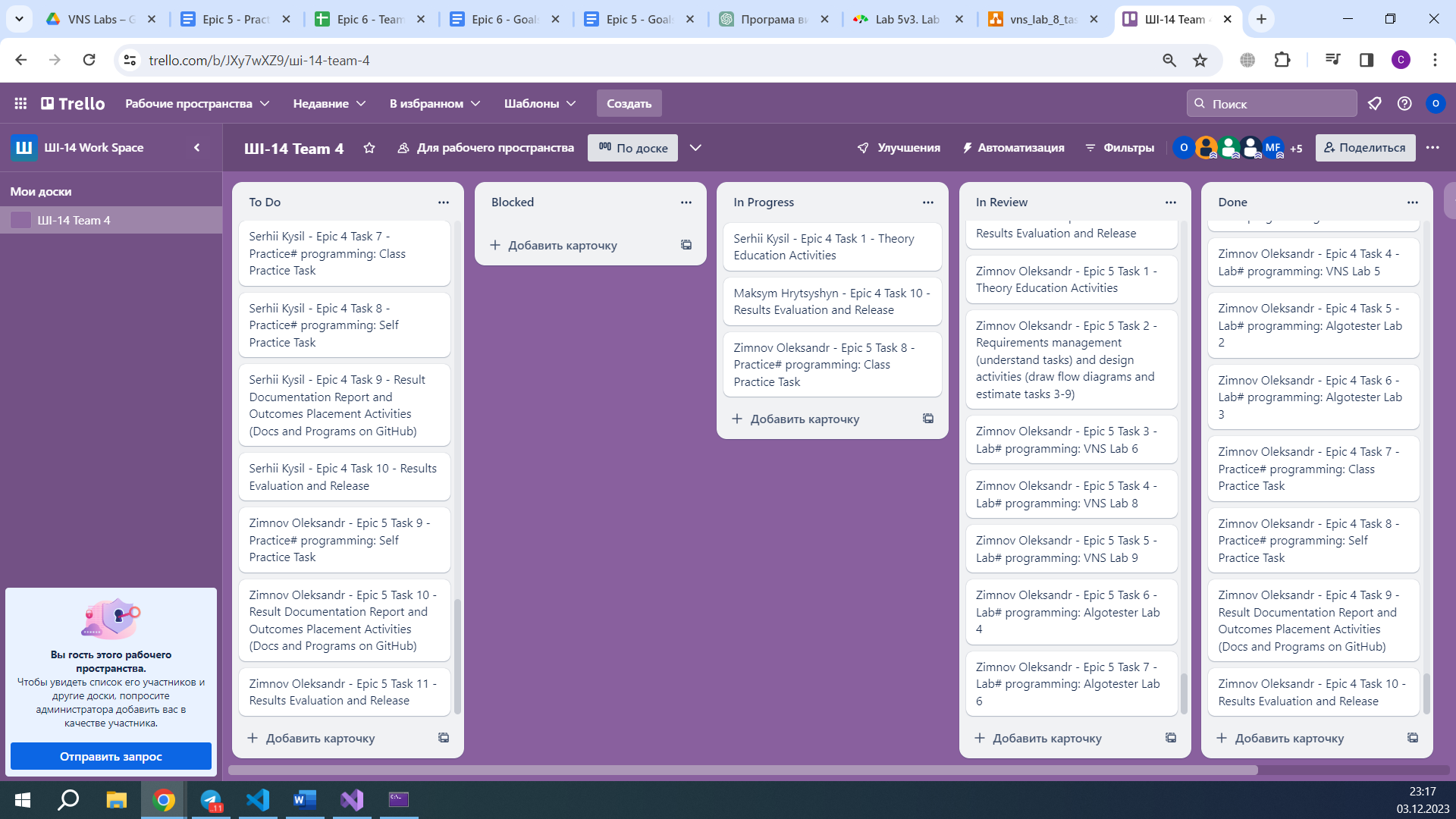


Рисунок 18: Завдання в Trello

Знімок коментарів від команди:

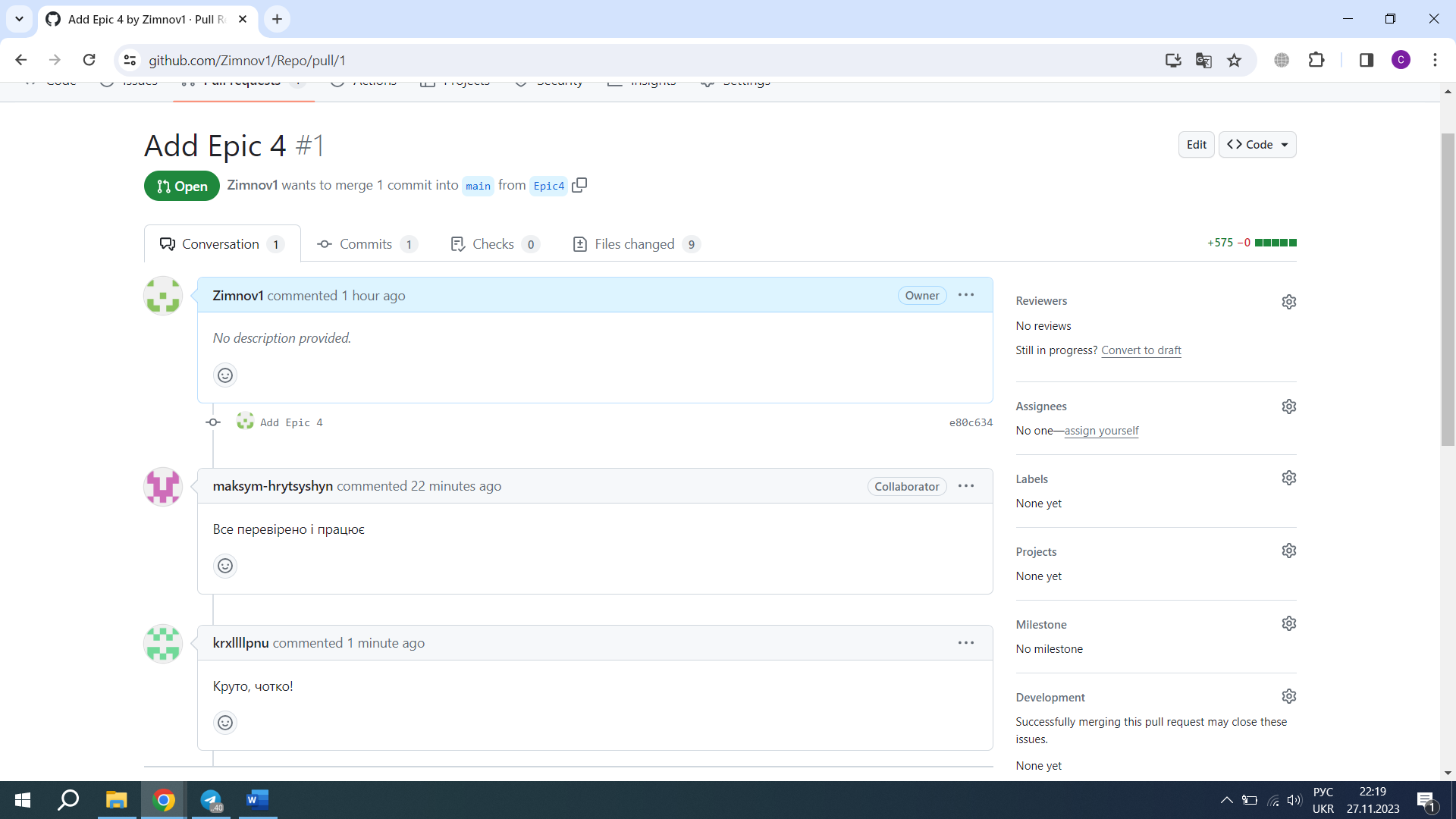


Рисунок 19: Коментарі від команди

# **Висновки:**

# Протягом лабораторних та практичних робіт №5було вивчено як створювати файли, повторено різні системи числення, також було вивчено як вписувати і зчитувати інформацію файлу, створено й використано бібліотеки.

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/507