Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

**про виконання лабораторних та практичних робіт блоку № 3**

На тему:  «Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції зі змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.»

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

до:

Практичних Робіт до блоку № 3

**Виконала:**

Студентка групи ШІ-12

Лазаревич Юлія Дмитрівна

Львів 2024

**Тема роботи:**

Цикли. Вкладені цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функції. Функції зі змінною кількістю параметрів. Рекурсія. Вбудовані функції.

**Мета роботи:**

Ознайомитись з циклами, вкладеними циклами, функціями, простором імен, перевантаженням функції, функціями зі змінною кількістю параметрів, рекурсією, вбудованими функціями в мовах С та С++.

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

Тема №1: Введення в Цикли та їх Види в С++

Тема №2: Управління Виконанням Циклів

Тема №3: Вкладені Цикли

Тема №4: Основи Функцій у С++

Тема №5: Перевантаження Функцій та Простір Імен

Тема №6: Розширені Можливості Функцій

Тема №7: Вбудовані Функції в С++

1. Індивідуальний план опрацювання теорії:

* Тема №1: Введення в Цикли та їх Види в С++.
* Джерела Інформації:
* [Цикли. Оператори циклу for, while, do…while](https://www.bestprog.net/uk/2017/09/04/cycles-operators-of-the-cycle-for-while-do-while_ua/)
* [Урок №6 - Цикли та оператори в них (For, While, Do While)](https://itproger.com/ua/course/cpp/6)
* Що опрацьовано:
* Оператори циклу у мові C++:
* for (ініціалізація; вираз; приріст){//}
* while (вираз){//}
* do{//} while (вираз);
* Також, існують такі цикли:
* Вкладені цикли(далі детальніше)
* Нескінчений цикл – це цикл, який ніколи не закінчується. Його часто можна написати випадково/помилково.
* Статус: ознайомлена.
* Початок опрацювання теми: 20.09.24
* Завершення опрацювання теми 25.09.24
* Тема №2: Управління Виконанням Циклів.
* Джерела Інформації:
* [Урок №73. Оператори break і continue](https://acode.com.ua/urok-73-operatory-break-i-continue/)
* Що опрацьовано:
* Оператор break - використовується для негайного припинення виконання циклу. Коли цикл зустрічає *break*, виконання виходить з циклу, і програма продовжує виконання коду, що йде після циклу.
* Оператор continue - використовується для пропуску поточної ітерації циклу та переходу до наступної ітерації. Це корисно, коли необхідно пропустити частину коду в циклі за певної умови.
* Статус: ознайомлена.
* Початок опрацювання теми: 30.09.24
* Завершення опрацювання теми 05.10.24
* Тема №3: Вкладені Цикли
* Джерела Інформації:
* [Цикли. Оператори циклу for, while, do…while](https://www.bestprog.net/uk/2017/09/04/cycles-operators-of-the-cycle-for-while-do-while_ua/)
* Що опрацьовано:
* Вкладені цикли - цикли, які знаходяться всередині інших циклів. Можуть використовуватися, при роботі з багатовимірними структурами даних, таких як матриці. Вони дозволяють виконувати операції з кожним елементом такої структури, проходячи через кожен рівень вкладеності.
* Статус: ознайомлена.
* Початок опрацювання теми: 20.09.24
* Завершення опрацювання теми 25.09.24
* Тема №4: Основи Функцій у С++
* Джерела Інформації:
* [Урок №15. Функції](https://acode.com.ua/urok-15-funktsiyi-i-operator-return/)
* [Функції. Частина 1.](https://www.bestprog.net/uk/2017/04/15/%d1%84%d1%83%d0%bd%d0%ba%d1%86%d1%96%d1%97-%d1%87%d0%b0%d1%81%d1%82%d0%b8%d0%bd%d0%b0-1-%d0%be%d0%bf%d0%b8%d1%81-%d1%84%d1%83%d0%bd%d0%ba%d1%86%d1%96%d1%97-%d1%84%d0%b0%d0%ba%d1%82%d0%b8%d1%87/#:~:text=%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F%20%E2%80%93%20%D1%86%D0%B5%20%D1%87%D0%B0%D1%81%D1%82%D0%B8%D0%BD%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%2C%20%D1%8F%D0%BA%D0%B0%20%D0%BC%D0%B0%D1%94%20%D1%82%D0%B0%D0%BA%D1%96,%D0%BF%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80%D1%96%D0%B2%2C%20%D1%8F%D0%BA%D1%96%20%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%8E%D1%82%D1%8C%D1%81%D1%8F%20%D1%97%D0%B9%20%D0%B4%D0%BB%D1%8F%20%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B8%20%D0%B0%D0%B1%D0%BE%20%D0%B2%D0%B8%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D0%BD%D1%8F.)
* Що опрацьовано:
* Функція - це блок коду, який виконує певне завдання і може бути викликаний з різних частин програми. Функція має ім'я, тип повертаємого значення, список параметрів (аргументів) і тіло.
* Загальна форма запису:

тип ім’я\_функції(список\_параметрів або void)

{

тіло\_функції

[return] (вираз);

}

* Область видимості формальних параметрів функції визначається межами тіла функції, в якій вони описані.
* Статус: ознайомлена.
* Початок опрацювання теми: 27.09.24
* Завершення опрацювання теми 05.10.24
* Тема №5: Перевантаження Функцій та Простір Імен
* Джерела Інформації:
* [Перевантаження функцій.](https://www.bestprog.net/uk/2018/08/07/overloaded-functions-overloaded-functions-in-classes-overloading-of-class-constructors-access-to-the-overloaded-function-using-a-pointer-examples_ua/#:~:text=%D0%A9%D0%BE%20%D0%BE%D0%B7%D0%BD%D0%B0%D1%87%D0%B0%D1%94%20%D1%82%D0%B5%D1%80%D0%BC%D1%96%D0%BD%20%E2%80%9C%D0%BF%D0%B5%D1%80%D0%B5%D0%B2%D0%B0%D0%BD%D1%82%D0%B0%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F%E2%80%9D%3F%20%E2%80%9C%D0%9F%D0%B5%D1%80%D0%B5%D0%B2%D0%B0%D0%BD%D1%82%D0%B0%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F%E2%80%9D%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%97%20%E2%80%93%20%D1%86%D0%B5,%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%96%20%D0%B2%D0%B8%D0%B4%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%96%20%D1%96%D0%BC%E2%80%99%D1%8F%20%E2%80%9C%D0%BF%D0%B5%D1%80%D0%B5%D0%B2%D0%B0%D0%BD%D1%82%D0%B0%D0%B6%D0%B5%D0%BD%D0%B0%E2%80%9D%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F%20%D0%BE%D0%B3%D0%BE%D0%BB%D0%BE%D1%88%D1%83%D1%94%D1%82%D1%8C%D1%81%D1%8F%20%D0%B4%D0%B5%D0%BA%D1%96%D0%BB%D1%8C%D0%BA%D0%B0%20%D1%80%D0%B0%D0%B7%D1%96%D0%B2.)
* [Простори імен. Ключові слова namespace, using.](https://www.bestprog.net/uk/2022/04/17/c-namespaces-keywords-namespace-using-ua/#:~:text=%D0%A9%D0%BE%D0%B1%20%D1%81%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%B8%D1%82%D0%B8%20%D0%B2%D0%B8%D0%BA%D0%BE%D1%80%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D0%BD%D1%8F%20%D1%96%D0%BC%D0%B5%D0%BD%20%D1%83%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0%D1%85%20%D0%BD%D0%B0%20C%2B%2B,%D1%8F%D0%BA%D0%B0%20%D0%BE%D0%B1%E2%80%99%D1%94%D0%B4%D0%BD%D0%B0%D0%BD%D0%B0%20%D0%B2%20%D1%94%D0%B4%D0%B8%D0%BD%D0%B8%D0%B9%20%D1%96%D0%BC%D0%B5%D0%BD%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B9%20%D0%B0%D0%B1%D0%BE%20%D0%BD%D0%B5%D1%96%D0%BC%D0%B5%D0%BD%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B9%20%D0%B1%D0%BB%D0%BE%D0%BA.)
* Що опрацьовано:
* Перевантаження функції – це оголошення функції з тим же іменем декілька разів. Щоб компілятор міг їх відрізняти, ці функції повинні відрізнятися між собою списком вхідних параметрів.
* Відрізняються за:
* кількістю параметрів;
* якщо кількість параметрів однакова, то їх типами.
* Простір імен – це область визначення змінних, типів, констант та функцій, яка об’єднана в єдиний іменований або неіменований блок. Використовується для організації коду і запобігання конфліктів імен у великих проектах. Дозволяє групувати пов'язані класи, функції і змінні під спільним ідентифікатором.
* Статус: ознайомлена.
* Початок опрацювання теми: 10.10.24
* Завершення опрацювання теми 17.10.24
* Тема №6: Розширені Можливості Функцій
* Джерела Інформації:
* Окремі відео на ютуб
* Допомога копілота
* Що опрацьовано:
* Функції в C++ можуть мати розширені можливості, які роблять код більш гнучким та ефективним. До таких можливостей належать аргументи за замовчуванням, шаблони функцій, рекурсія та функції як параметри.
* Аргументи за замовчуваннямдозволяють задавати значення параметрів функції, які будуть використовуватись, якщо аргументи не передані при виклику функції. Це зменшує кількість перевантажень функцій і покращує читабельність коду.
* Шаблони функцій дозволяють створювати функції, які можуть працювати з різними типами даних, забезпечуючи повторне використання коду та зменшуючи кількість перевантажених функцій.
* Рекурсія дозволяє функції викликати саму себе, що особливо корисно для вирішення завдань, які можуть бути розбиті на менші підзадачі. Важливо уникати нескінченних рекурсій.
* Статус: ознайомлена.
* Початок опрацювання теми: 18.10.24
* Завершення опрацювання теми 25.10.24
* Тема №7: Вбудовані Функції в С++
* Джерела Інформації:

* [Урок №107. Вбудовані функції](https://acode.com.ua/urok-107-vbudovani-funktsiyi/)
* Що опрацьовано:
* Вбудована функція - це функція, яку компілятор замінює її викликами на сам код функції під час компіляції. Щоб оголосити функцію як вбудовану, використовується ключове слово inline перед визначенням функції.
* Переваги:
* Вбудовані функції дозволяють зменшити витрати на виклик функції, що може значно підвищити продуктивність в програмі з частими викликами невеликих функцій;
* Уникаючи витрат на збереження і відновлення стеку викликів і переходів до функції, програма виконується швидше.
* Недоліки:
* Оскільки код вбудованої функції копіюється в кожне місце її виклику, це може збільшити розмір виконуваного файлу, що може бути проблематичним для великих програм;
* Вбудовані функції повинні бути невеликими і простими, інакше вигоди від їх використання можуть бути знижені.
* 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 2 - Task 1- 10.

*Деталі завдання:*

* Використовуючи оператор циклу, знайти суму елементів, зазначених у

конкретному варіанті. Результат надрукувати, надавши відповідний

заголовком.

* Знайти суму ряду з точністю ε=0.0001, загальний член якого

Завдання №2**-** VNS Lab 3 - Task 1 - 10.

*Деталі завдання:*

* + - Для х, що змінюється від a до b з кроком (b-a)/k, де (k=10), обчислити функцію f(x), використовуючи її розклад в степеневий ряд у двох випадках:

а) для заданого n;

б) для заданої точності ε (ε=0.0001).

* + - Для порівняння знайти точне значення функції.

|  |  |  |  |
| --- | --- | --- | --- |
| Функція | Діапазон зміни аргументу |  | Сума |
|  |  | 20 |  |

Завдання №3 – VNS Lab 7 - Task 1 - 10.

*Деталі завдання:*

* Розв’язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів:
* Написати функцію kvadr зі змінною кількістю параметрів, що визначає

кількість чисел, що є точними квадратами (2, 4, 9, 16,. . . ) типу int. Написати викликаючу функцію main, що звертається до функції kvadr не менше трьох разів з кількістю параметрів 3, 7, 11.

Завдання №4**-** VNS Lab 7 - Task 2 - 10.

*Деталі завдання:*

* + - Написати перевантажені функції й основну програму, що їх викликає.

Завдання №5**–** Class Practice Work - Менеджмент бібліотеки.

*Деталі завдання:*

* Ви створюєте просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

*Програма повинна вміти*

* Перерахувати всі книги.
* Дозволити взяти книгу (за наявності).
* Дозволити повернення книги.

*Структури даних*

* Використовуйте масив або вектор для зберігання назв книг.
* Використовуйте інший масив або вектор для збереження стану доступності кожної книги.

*Мета Задачі*

Навчитися користуватися операторами циклів та функцією переходу на мітку:

* for( ) { … }
* for each
* while( ) { … }
* do { … } while( )
* go to

*Вимоги:*

* while: продовжувати працювати, доки користувач не вирішить вийти.
* do while: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
* for: список усіх книг за допомогою циклу.
* for each: перевірити наявність кожної книги.
* goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

Завдання №6- Self Practice Work - Офісна Вулиця.

Зустрілися якось працівники великих компаній і почали... Обговорювати план вулиці. Виявляється, всі приміщення, які орендуватимуть ці компанії, збудують вздовж однієї вулиці. i-та компанія орендуватиме офіс довжиною li метрів. Офіси будуватимуть один за одним, починаючи з точки 0. Всі працівники приїжджатимуть на стоянку, яку побудують в точці 0, та будуть йти до офісів своїх компаній.

Тобто, якщо офіси будуть збудовані в порядку p1,p2,...,pn, то перший офіс почнеться в точці 0 і закінчиться в точці lp1, другий почнеться в lp1 і закінчиться в lp1+lp2 і т.д. Двері кожного офісу завжди є в кінці будинку, який є ближчим до стоянки.

Ваше завдання — **допомогти розмістити офіси компаній на цій вулиці в такому порядку, щоб сумарна відстань від точки 0 до усіх офісів була мінімальною.**

***Вхідні дані***

У першому рядку задане ціле число n — кількість компаній.

У наступному рядку задано n цілих чисел li через пробіл — довжини офісів усіх компаній.

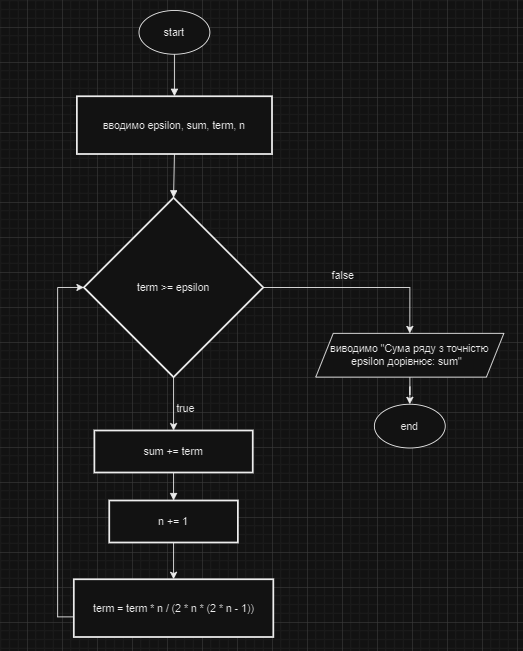
***Вихідні дані***

У єдиному рядку виведіть n чисел від 1 до n — порядок компаній, в якому варто будувати офіси.

Якщо існує декілька оптимальних порядків — виведіть будь-який із них.

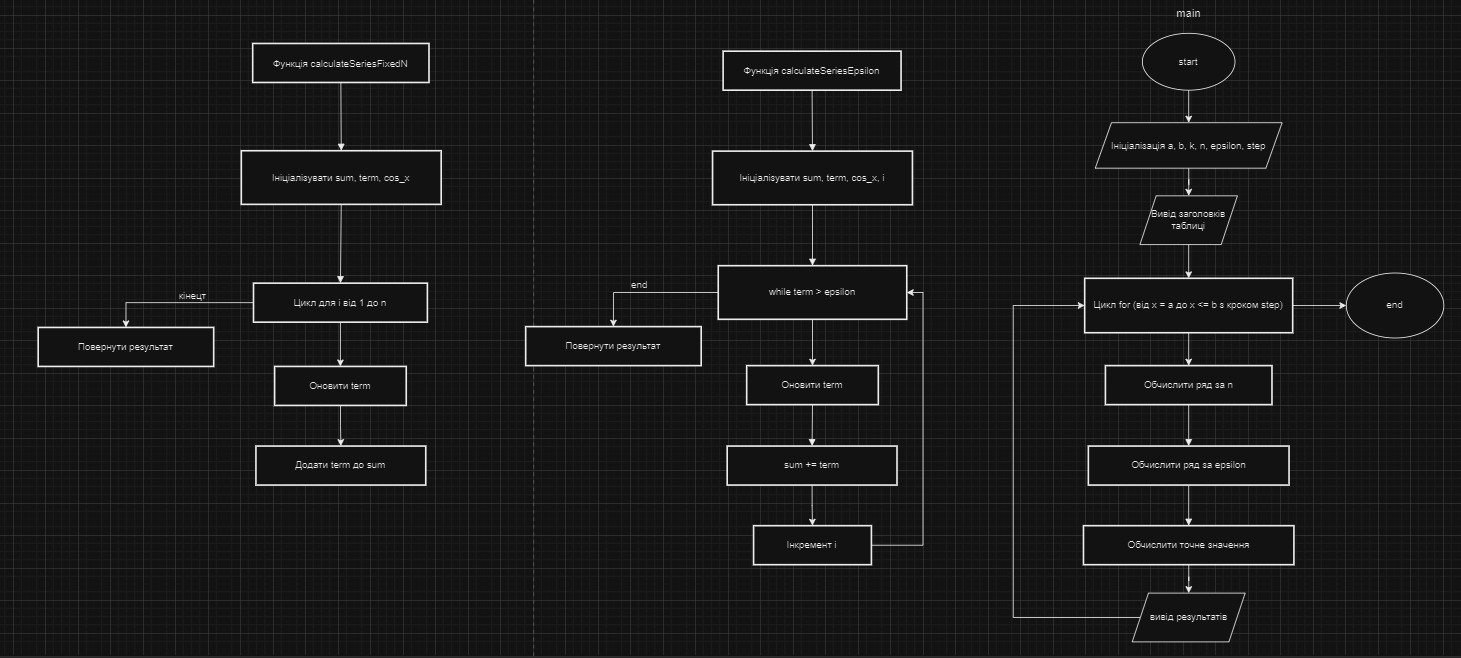
1. **Дизайн та планована оцінка часу виконання завдань:**

Програма №1 *-*VNS Lab 2 - Task 1- 10.



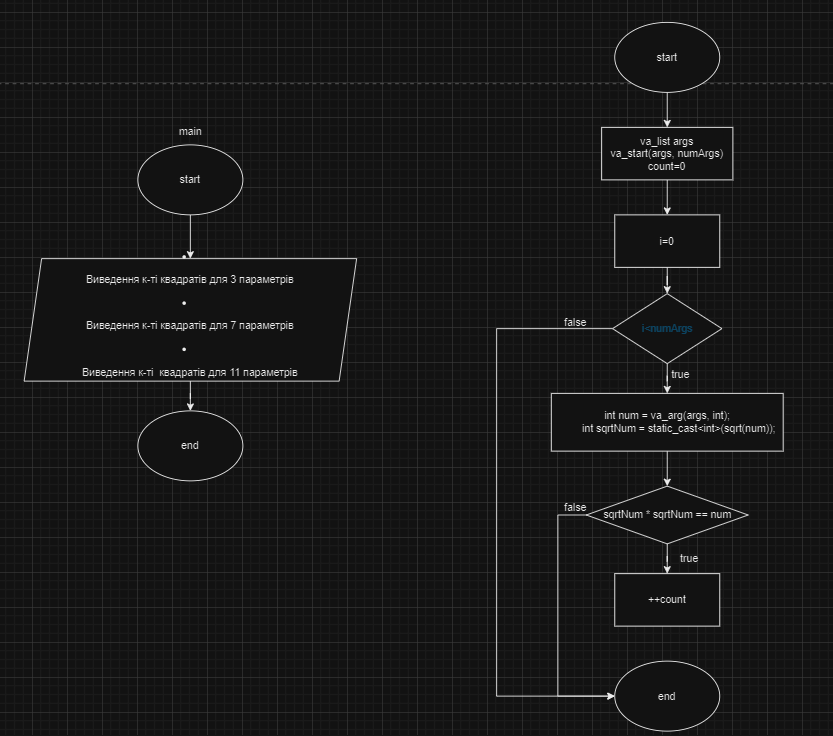
Плановий час на реалізацію: 30 хвилин

Програма №2 *-*VNS Lab 3 - Task 1 - 10.



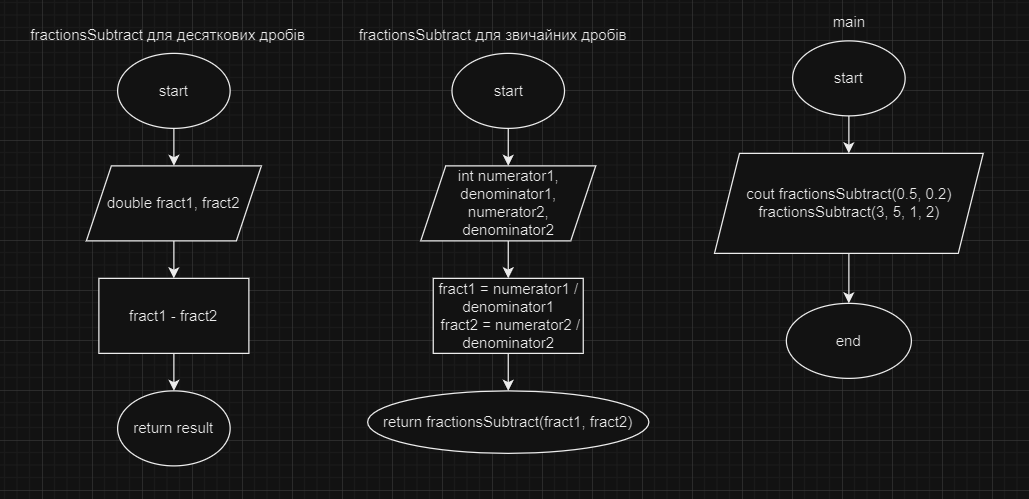
Плановий час на реалізацію: 2 години

Програма №3 – VNS Lab 7 - Task 1 - 10.



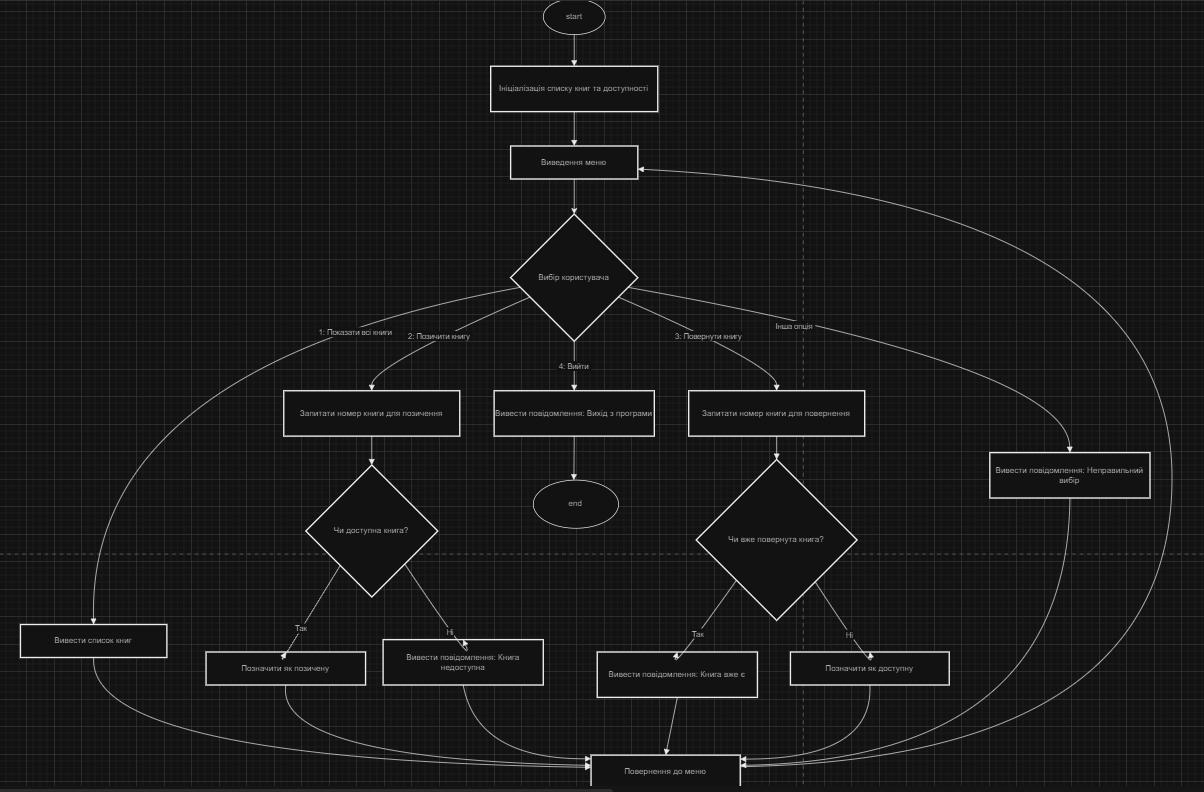
Плановий час на реалізацію: 30 хвилин

Програма №4- VNS Lab 7 - Task 2 - 10.

****

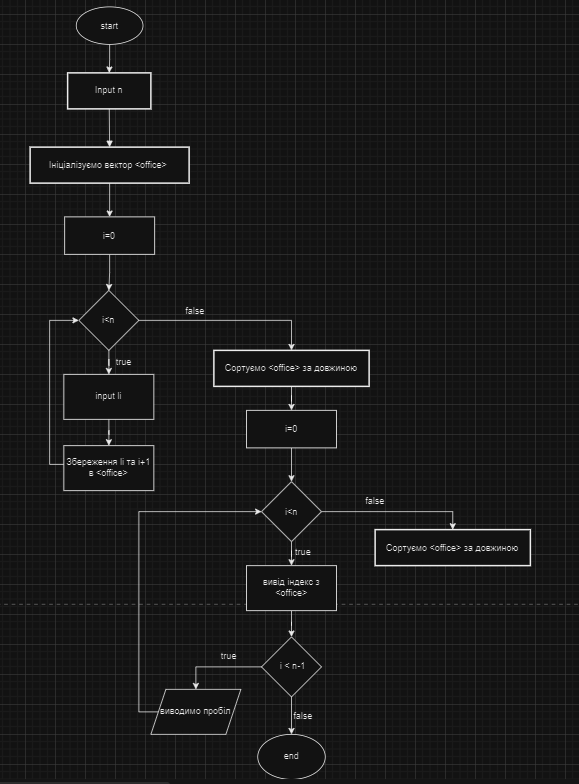
Плановий час на реалізацію: 30 хвилин

Програма №5- Class Practice Work – Менеджмент бібліотеки

****

Плановий час на реалізацію: 1 година

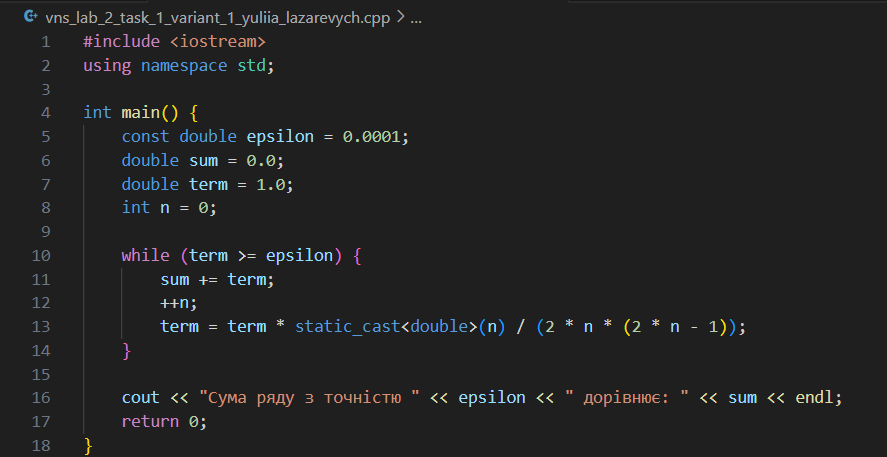
Програма №6- Self Practice Work – Офісна Вулиця.

****

Плановий час на реалізацію: 2 години

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

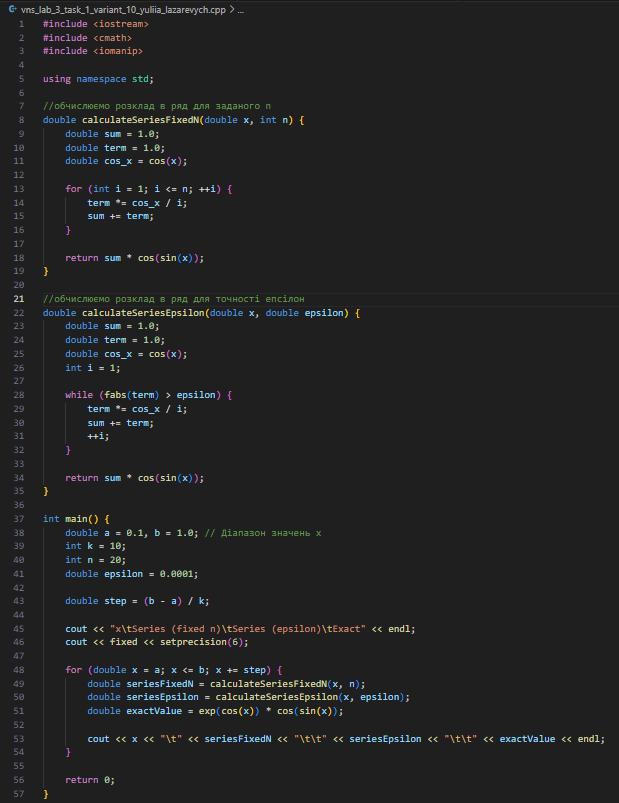
Завдання №1 *-*VNS Lab 2 - Task 1- 10.



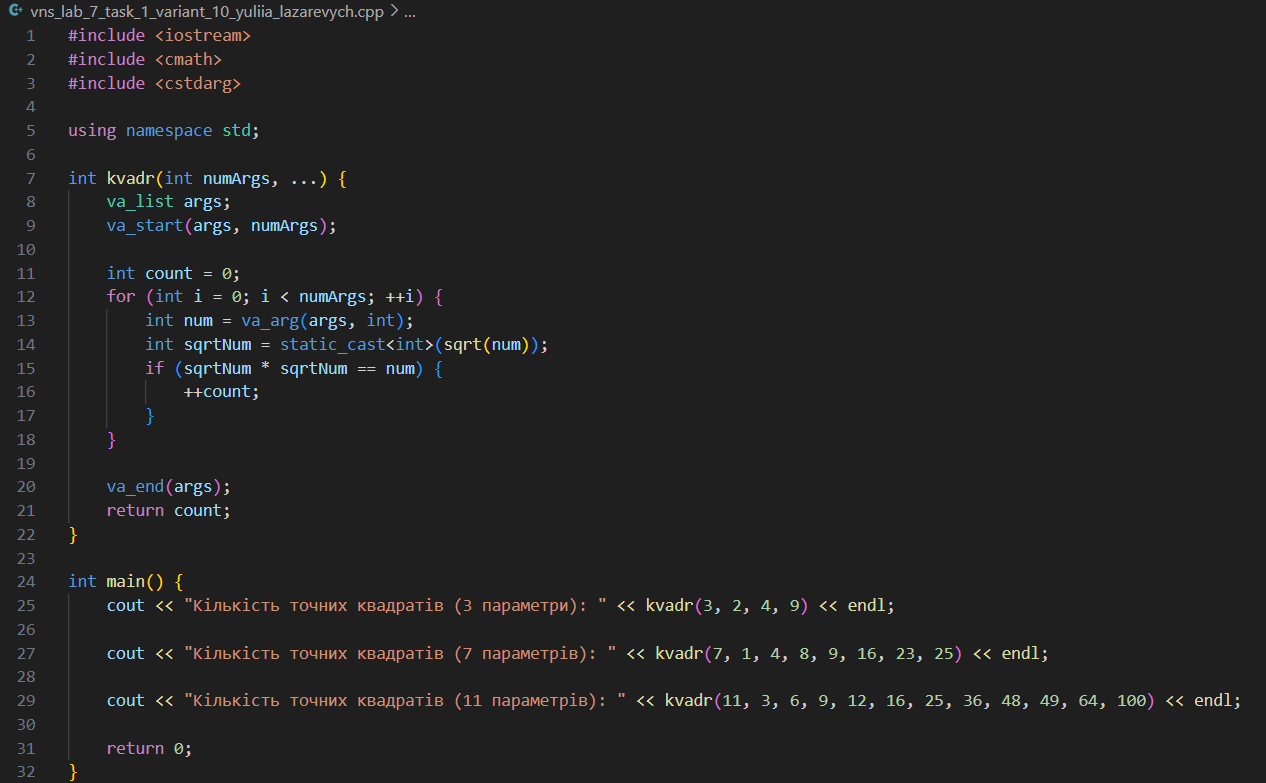
[ai\_programming\_playground\_2024/ai\_12/yuliia\_lazarevych/epic\_3/codes/vns\_lab\_2\_task\_1\_variant\_1\_yuliia\_lazarevych.cpp at epic\_3\_practice\_and\_labs\_yuliia\_lazarevych · artificial-intelligence-department/ai\_programming\_playground\_2024](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_yuliia_lazarevych/ai_12/yuliia_lazarevych/epic_3/codes/vns_lab_2_task_1_variant_1_yuliia_lazarevych.cpp)

Завдання №2 *-*VNS Lab 3 - Task 1- 10.

[ai\_programming\_playground\_2024/ai\_12/yuliia\_lazarevych/epic\_3/codes/vns\_lab\_3\_task\_1\_variant\_10\_yuliia\_lazarevych.cpp at epic\_3\_practice\_and\_labs\_yuliia\_lazarevych · artificial-intelligence-department/ai\_programming\_playground\_2024](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_yuliia_lazarevych/ai_12/yuliia_lazarevych/epic_3/codes/vns_lab_3_task_1_variant_10_yuliia_lazarevych.cpp)

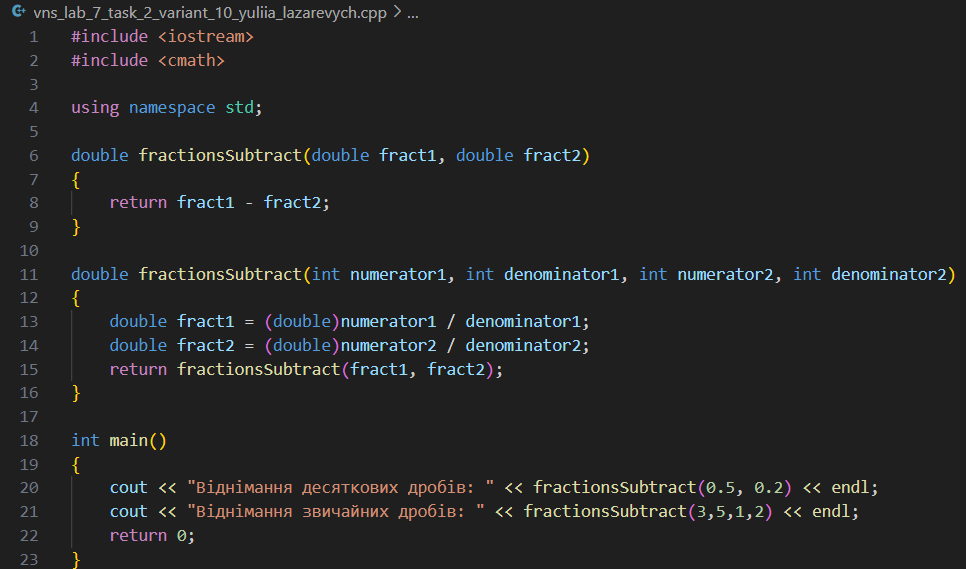


Завдання №3 – VNS Lab 7 - Task 1- 10.



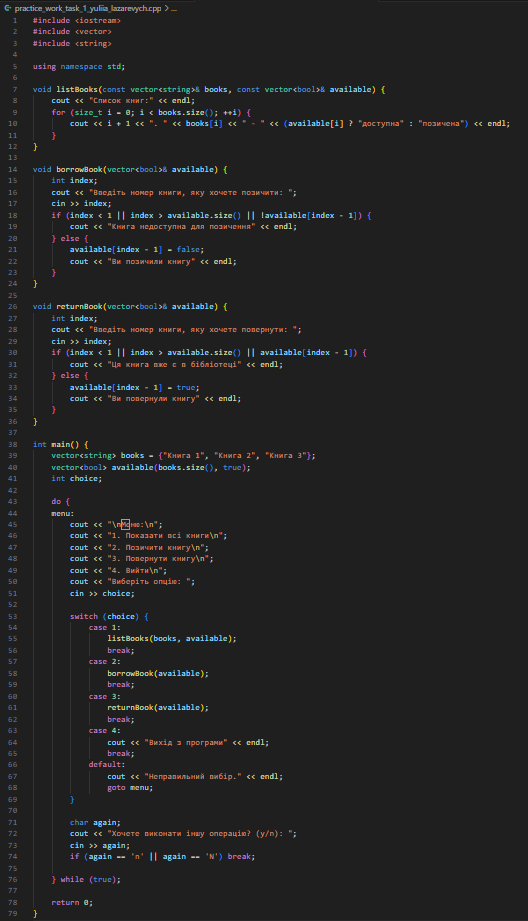
[ai\_programming\_playground\_2024/ai\_12/yuliia\_lazarevych/epic\_3/codes/vns\_lab\_7\_task\_1\_variant\_10\_yuliia\_lazarevych.cpp at epic\_3\_practice\_and\_labs\_yuliia\_lazarevych · artificial-intelligence-department/ai\_programming\_playground\_2024](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_yuliia_lazarevych/ai_12/yuliia_lazarevych/epic_3/codes/vns_lab_7_task_1_variant_10_yuliia_lazarevych.cpp)

Завдання №4- VNS Lab 7 - Task 2- 10.



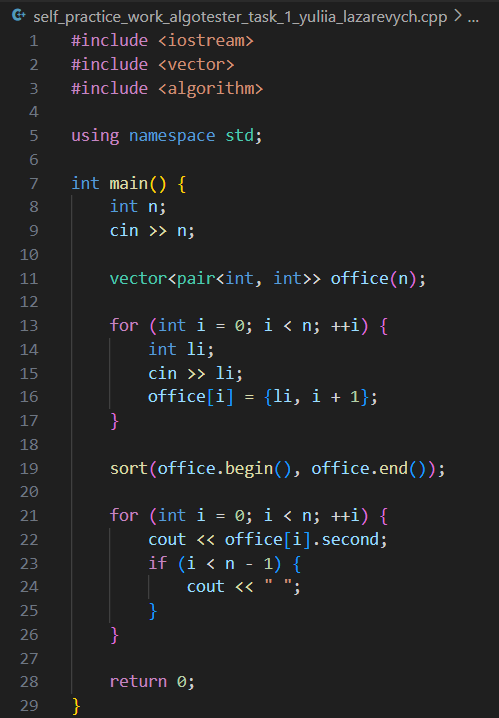
[ai\_programming\_playground\_2024/ai\_12/yuliia\_lazarevych/epic\_3/codes/vns\_lab\_7\_task\_2\_variant\_10\_yuliia\_lazarevych.cpp at epic\_3\_practice\_and\_labs\_yuliia\_lazarevych · artificial-intelligence-department/ai\_programming\_playground\_2024](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_yuliia_lazarevych/ai_12/yuliia_lazarevych/epic_3/codes/vns_lab_7_task_2_variant_10_yuliia_lazarevych.cpp)

Завдання №5– Class Practice Work

****

[ai\_programming\_playground\_2024/ai\_12/yuliia\_lazarevych/epic\_3/codes/practice\_work\_task\_1\_yuliia\_lazarevych.cpp at epic\_3\_practice\_and\_labs\_yuliia\_lazarevych · artificial-intelligence-department/ai\_programming\_playground\_2024](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_yuliia_lazarevych/ai_12/yuliia_lazarevych/epic_3/codes/practice_work_task_1_yuliia_lazarevych.cpp)

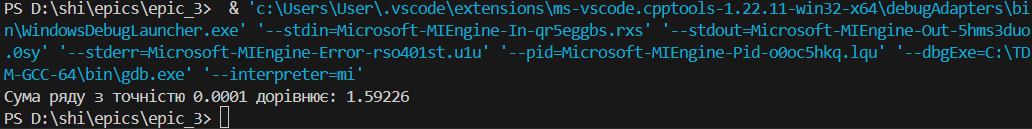
Завдання №6- Self Practice Work – Офісна Вулиця.

****

[ai\_programming\_playground\_2024/ai\_12/yuliia\_lazarevych/epic\_3/codes/self\_practice\_work\_algotester\_task\_1\_yuliia\_lazarevych.cpp at epic\_3\_practice\_and\_labs\_yuliia\_lazarevych · artificial-intelligence-department/ai\_programming\_playground\_2024](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_3_practice_and_labs_yuliia_lazarevych/ai_12/yuliia_lazarevych/epic_3/codes/self_practice_work_algotester_task_1_yuliia_lazarevych.cpp)

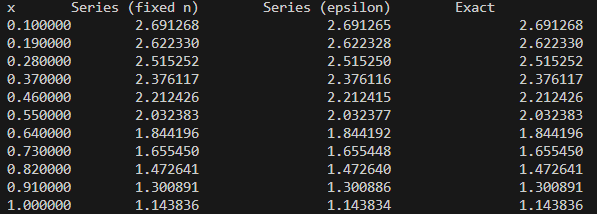
1. **Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 *-*VNS Lab 2 - Task 1- 10.



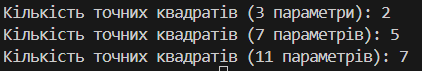
Витрачений час: 40 хвилин

Завдання №2 *-*VNS Lab 3 - Task 1- 10.



Витрачений час: 2 години

Завдання №3 – VNS Lab 7 - Task 1- 10.



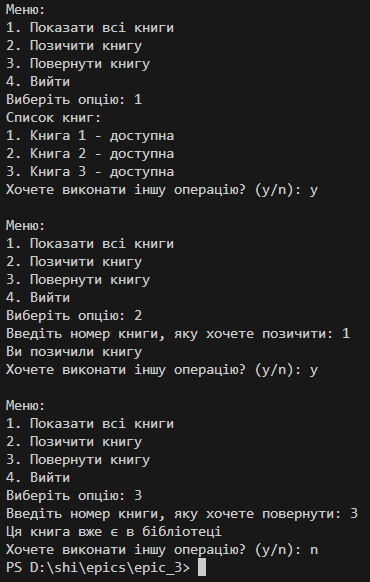
Витрачений час: 30 хвилин

Завдання №4 – VNS Lab 7 - Task 2- 10.



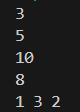
Витрачений час: 30 хвилин

Завдання №5- Class Practice Work - Особистий порадник.

****

Витрачений час: 2 години

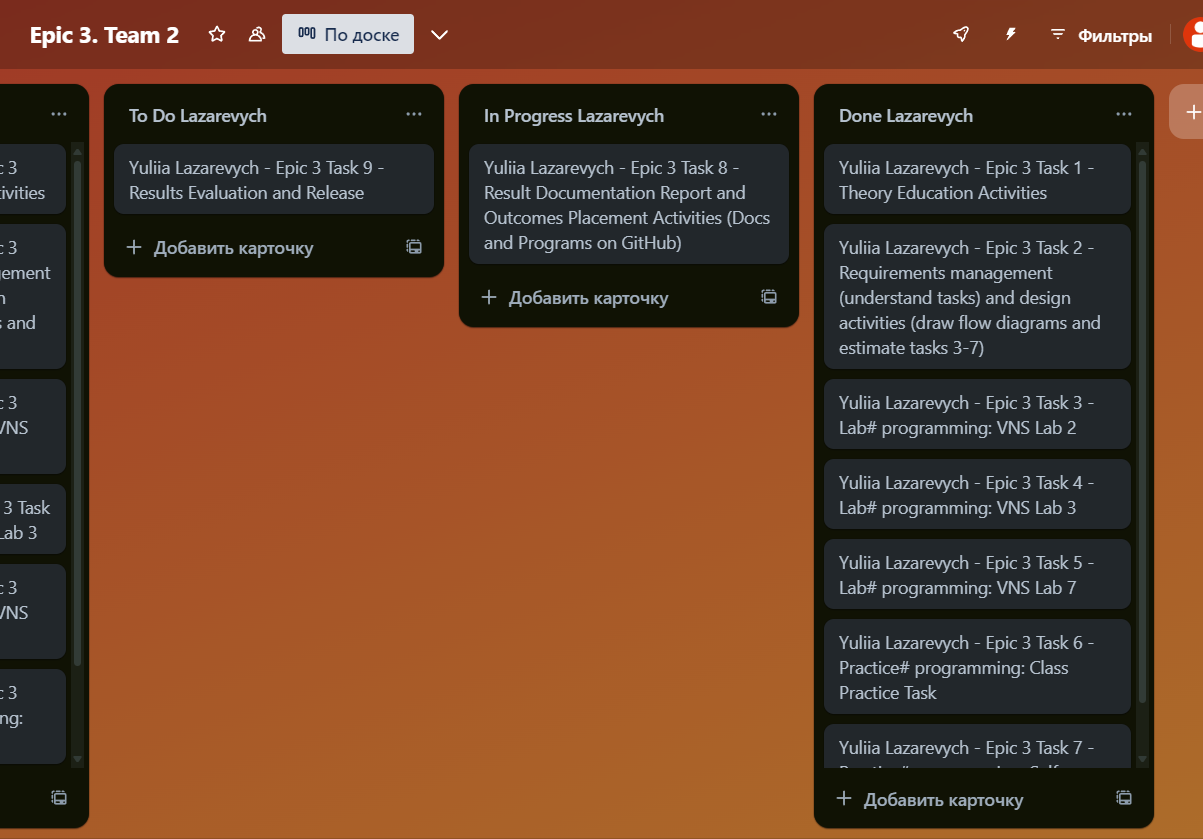
Завдання №6- Self Practice Work –Щасливий результат



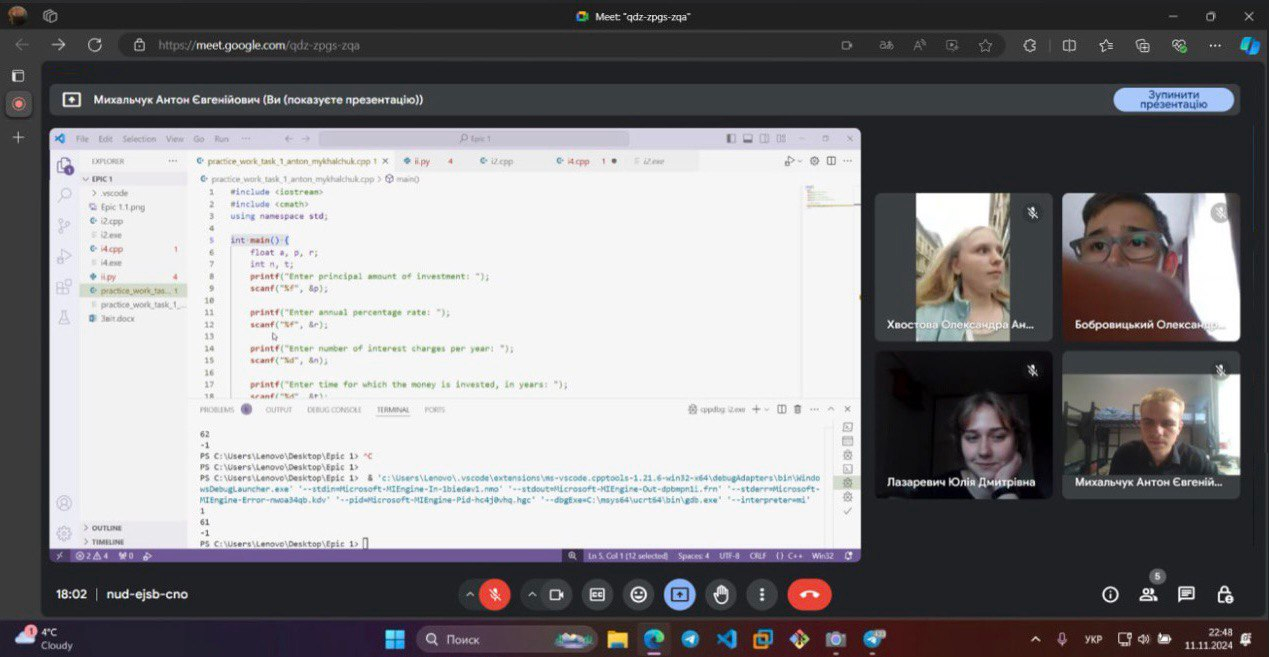
Витрачений час: 30 хвилин

**6. Кооперація з командою:**

* Скрін прогресу по Трелло



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



**Висновки:** Виконуючи третій епік я ознайомилась з циклами, вкладеними циклами, функціями, простором імен, перевантаженням функції, функціями зі змінною кількістю параметрів, рекурсією, вбудованими функціями в мовах С та С++. Також я написала шість кодів де застосувала нові знання на практиці.