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++

2. Індивідуальний план опрацювання теорії:
   - Тема №1: Введення в Цикли та їх Види в С++.
     - Джерела Інформації:
       - Цикли. Оператори циклу for, while, do…while
       - Урок №6 - Цикли та оператори в них (For, While, Do While)
     - Що опрацьовано:
       - Оператори циклу у мові С++:
         - 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
  - 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
  - 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 C++
  - Джерела Інформації:
    - Урок №15. Функції
    - Функції. Частина 1.
  - 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.
  - Простори імен. Ключові слова namespace, using.
- 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 замовчуванням дозволяють 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 C++
    - Джерела Інформації:
        - Урок №107. Вбудовані функції
    - 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

$$a_n = \frac{n!}{(2n)!}$$

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

Деталі завдання:

- Для x, 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	n	Сума
$e^{cosx}\cos(sinx)$	$0.1 \le x \le 1$	20	$S = 1 + \frac{cosx}{1!} + \cdots + \frac{cosnx}{n!}$

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.

а) для віднімання десяткових дробів;

б) для віднімання звичайних дробів.

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.

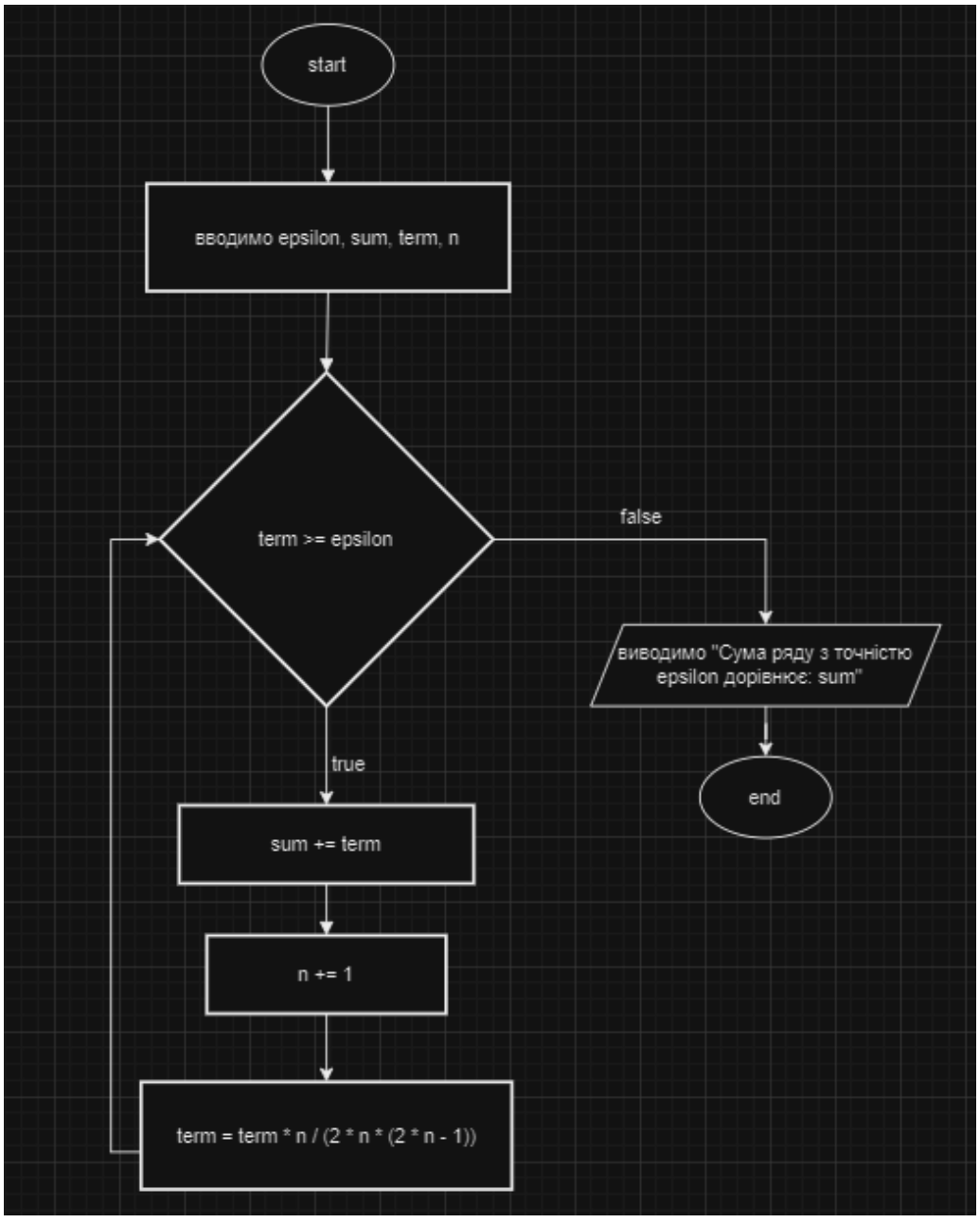
Вихідні дані

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.

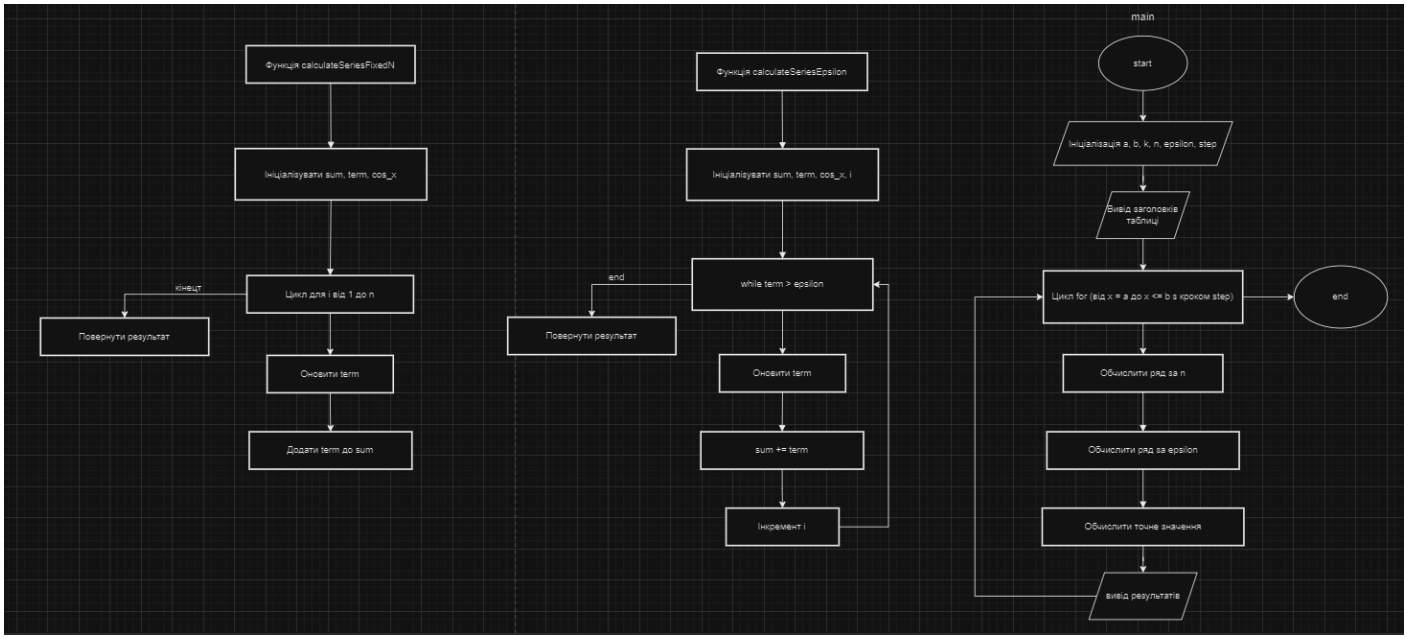
2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 - VNS Lab 2 - Task 1- 10.



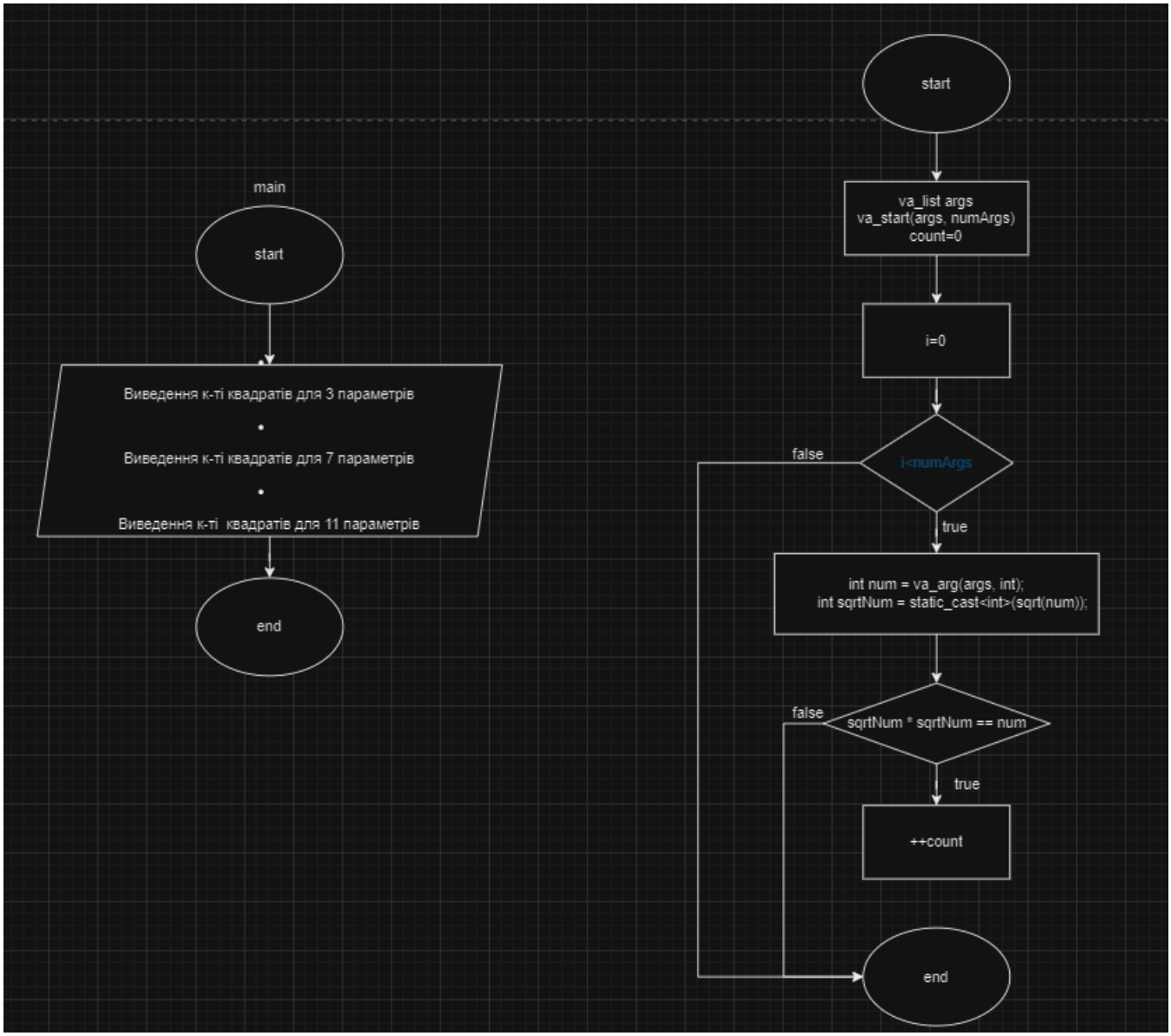
Плановий час на реалізацію: 30 хвилин

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



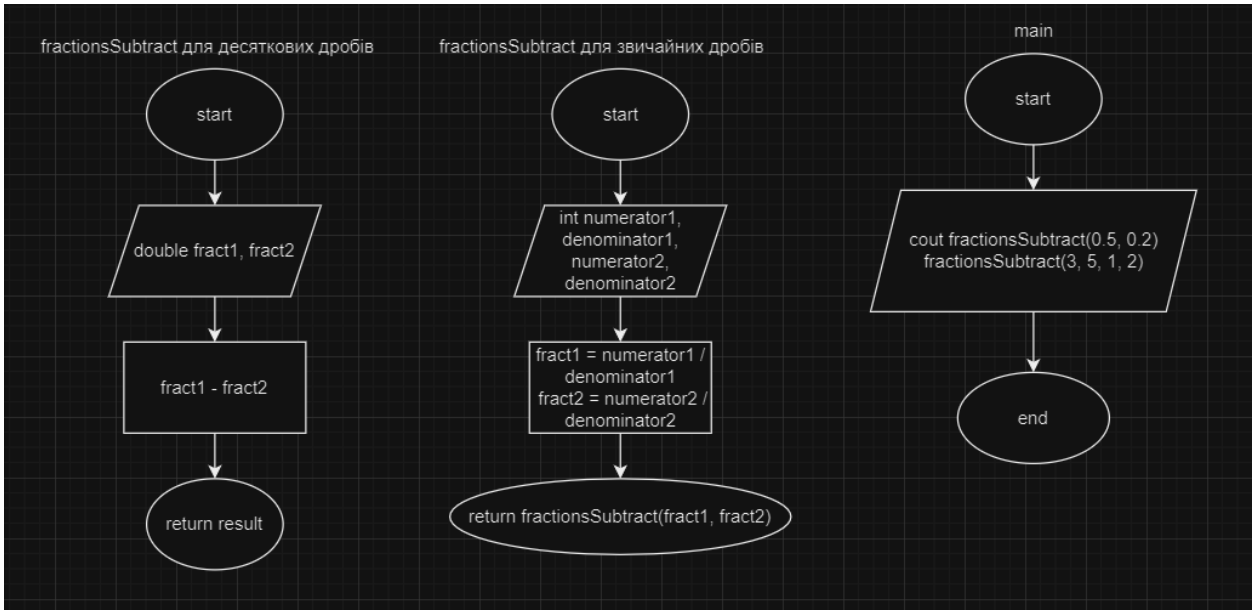
Плановий час на реалізацію: 2 години

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



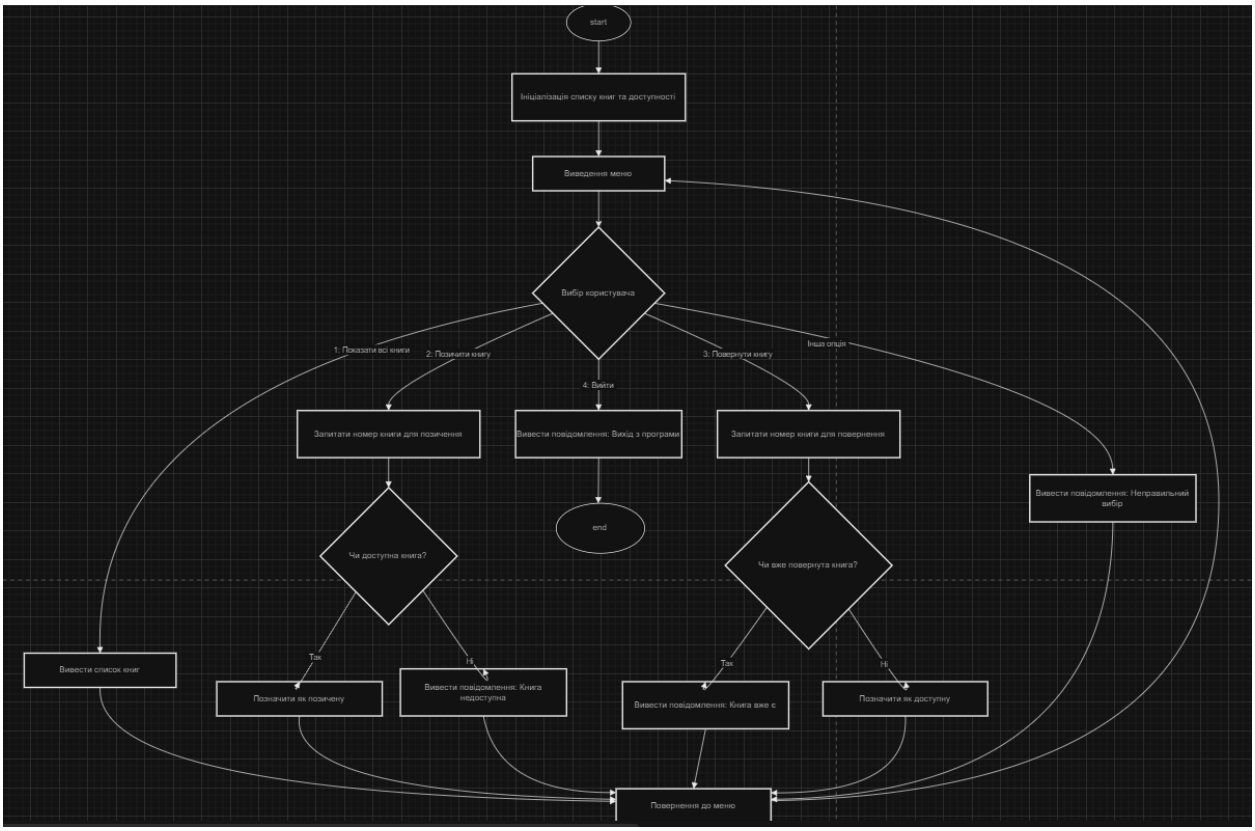
Плановий час на реалізацію: 30 хвилин

Програма №4 - VNS Lab 7 - Task 2 - 10.



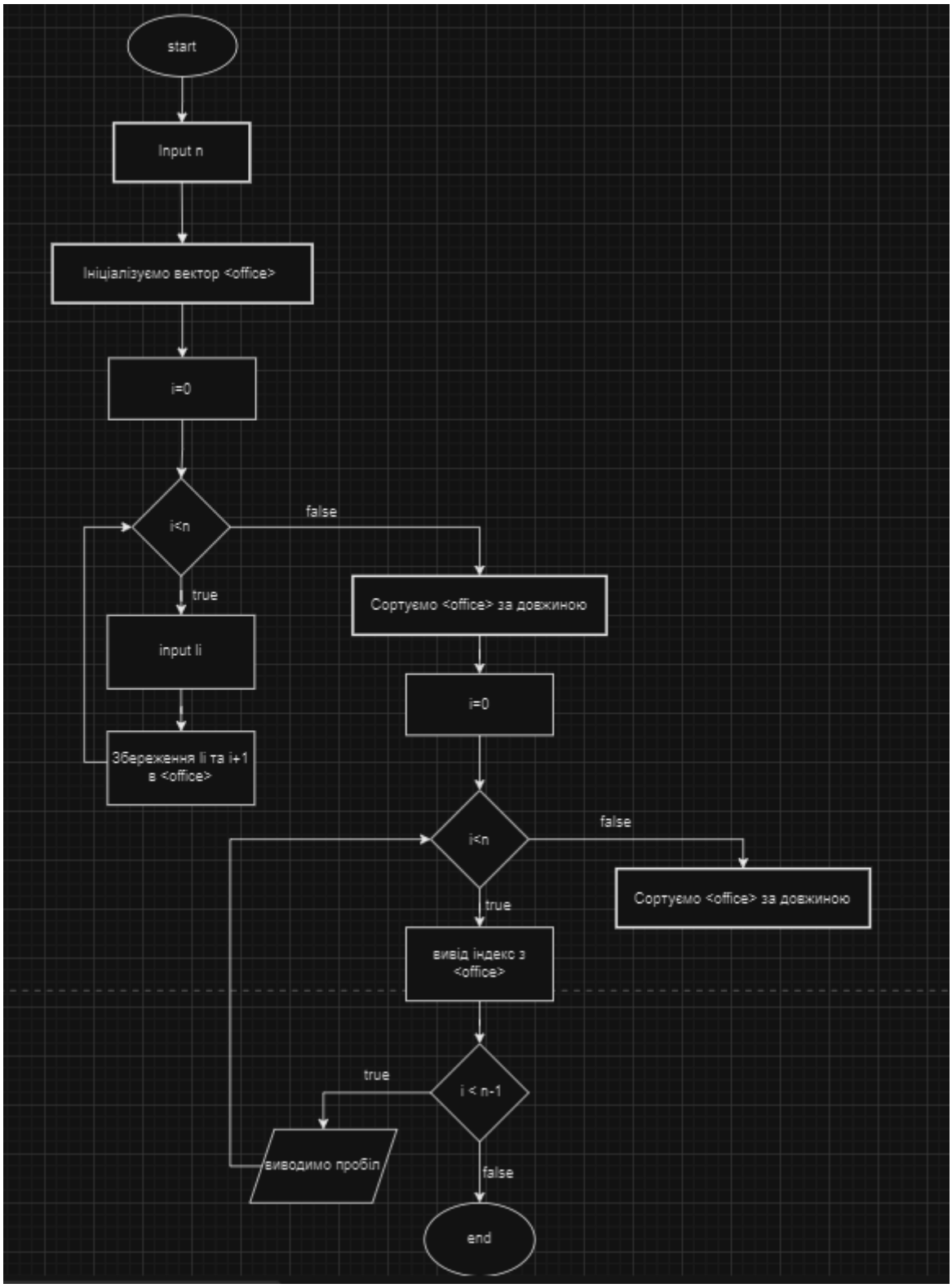
Плановий час на реалізацію: 30 хвилин

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

3. Код програм з посиланням на зовнішні ресурси:

Завдання №1 - VNS Lab 2 - Task 1- 10.

vns_lab_2_task_1_variant_1_yuliia_lazarevych.cpp > ...

```cpp
#include <iostream>
using namespace std;

int main() {
    const double epsilon = 0.0001;
    double sum = 0.0;
    double term = 1.0;
    int n = 0;

    while (term >= epsilon) {
        sum += term;
        ++n;
        term = term * static_cast<double>(n) / (2 * n * (2 * n - 1));
    }

    cout << "Сума ряду з точністю " << epsilon << " дорівнює: " << sum << endl;
    return 0;
}
```

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-

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

vns_lab_3_task_1_variant_10_yuliia_lazarevych.cpp > ...

```cpp
#include <iostream>
#include <cmath>
#include <iomanip>

using namespace std;

//обчислюємо розклад в ряд для заданого n
double calculateSeriesFixedN(double x, int n) {
    double sum = 1.0;
    double term = 1.0;
    double cos_x = cos(x);

    for (int i = 1; i <= n; ++i) {
        term *= cos_x / i;
        sum += term;
    }

    return sum * cos(sin(x));
}

//обчислюємо розклад в ряд для точності епсілон
double calculateSeriesEpsilon(double x, double epsilon) {
    double sum = 1.0;
    double term = 1.0;
    double cos_x = cos(x);
    int i = 1;

    while (fabs(term) > epsilon) {
        term *= cos_x / i;
        sum += term;
        ++i;
    }

    return sum * cos(sin(x));
}

int main() {
    double a = 0.1, b = 1.0; // Діапазон значень x
    int k = 10;
    int n = 20;
    double epsilon = 0.0001;

    double step = (b - a) / k;

    cout << "x\tSeries (fixed n)\tSeries (epsilon)\tExact" << endl;
    cout << fixed << setprecision(6);

    for (double x = a; x <= b; x += step) {
        double seriesFixedN = calculateSeriesFixedN(x, n);
        double seriesEpsilon = calculateSeriesEpsilon(x, epsilon);
        double exactValue = exp(cos(x)) * cos(sin(x));

        cout << x << "\t" << seriesFixedN << "\t\t" << seriesEpsilon << "\t\t" << exactValue << endl;
    }

    return 0;
}
```

Завдання №3 – VNS Lab 7 - Task 1- 10.

vns_lab_7_task_1_variant_10_yuliia_lazarevych.cpp > ...

```cpp
#include <iostream>
#include <cmath>
#include <cstdarg>

using namespace std;

int kvadr(int numArgs, ...) {
    va_list args;
    va_start(args, numArgs);

    int count = 0;
    for (int i = 0; i < numArgs; ++i) {
        int num = va_arg(args, int);
        int sqrtNum = static_cast<int>(sqrt(num));
        if (sqrtNum * sqrtNum == num) {
            ++count;
        }
    }

    va_end(args);
    return count;
}

int main() {
    cout << "Кількість точних квадратів (3 параметри): " << kvadr(3, 2, 4, 9) << endl;

    cout << "Кількість точних квадратів (7 параметрів): " << kvadr(7, 1, 4, 8, 9, 16, 23, 25) << endl;

    cout << "Кількість точних квадратів (11 параметрів): " << kvadr(11, 3, 6, 9, 12, 16, 25, 36, 48, 49, 64, 100) << endl;

    return 0;
}
```

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

Завдання №4 - VNS Lab 7 - Task 2- 10.

vns_lab_7_task_2_variant_10_yuliia_lazarevych.cpp > ...

```cpp
#include <iostream>
#include <cmath>

using namespace std;

double fractionsSubtract(double fract1, double fract2)
{
    return fract1 - fract2;
}

double fractionsSubtract(int numerator1, int denominator1, int numerator2, int denominator2)
{
    double fract1 = (double)numerator1 / denominator1;
    double fract2 = (double)numerator2 / denominator2;
    return fractionsSubtract(fract1, fract2);
}

int main()
{
    cout << "Віднімання десяткових дробів: " << fractionsSubtract(0.5, 0.2) << endl;
    cout << "Віднімання звичайних дробів: " << fractionsSubtract(3,5,1,2) << endl;
    return 0;
}
```

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

Завдання №5 – Class Practice Work

practice_work_task_1_yuliia_lazarevych.cpp > ...

```cpp
#include <iostream>
#include <vector>
#include <string>

using namespace std;

void listBooks(const vector<string>& books, const vector<bool>& available) {
    cout << "Список книг:" << endl;
    for (size_t i = 0; i < books.size(); ++i) {
        cout << i + 1 << ". " << books[i] << " - " << (available[i] ? "доступна" : "позичена") << endl;
    }
}

void borrowBook(vector<bool>& available) {
    int index;
    cout << "Введіть номер книги, яку хочете позичити: ";
    cin >> index;
    if (index < 1 || index > available.size() || !available[index - 1]) {
        cout << "Книга недоступна для позичення" << endl;
    } else {
        available[index - 1] = false;
        cout << "Ви позичили книгу" << endl;
    }
}

void returnBook(vector<bool>& available) {
    int index;
    cout << "Введіть номер книги, яку хочете повернути: ";
    cin >> index;
    if (index < 1 || index > available.size() || available[index - 1]) {
        cout << "Ця книга вже є в бібліотеці" << endl;
    } else {
        available[index - 1] = true;
        cout << "Ви повернули книгу" << endl;
    }
}

int main() {
    vector<string> books = {"Книга 1", "Книга 2", "Книга 3"};
    vector<bool> available(books.size(), true);
    int choice;

    do {
    menu:
        cout << "\nМеню:\n";
        cout << "1. Показати всі книги\n";
        cout << "2. Позичити книгу\n";
        cout << "3. Повернути книгу\n";
        cout << "4. Вийти\n";
        cout << "Виберіть опцію: ";
        cin >> choice;

        switch (choice) {
            case 1:
                listBooks(books, available);
                break;
            case 2:
                borrowBook(available);
                break;
            case 3:
                returnBook(available);
                break;
            case 4:
                cout << "Вихід з програми" << endl;
                break;
            default:
                cout << "Неправильний вибір." << endl;
                goto menu;
        }

        char again;
        cout << "Хочете виконати іншу операцію? (y/n): ";
        cin >> again;
        if (again == 'n' || again == 'N') break;

    } while (true);

    return 0;
}
```

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

Завдання №6 - Self Practice Work – Офісна Вулиця.

self_practice_work_algotester_task_1_yuliia_lazarevych.cpp > ...

```cpp
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main() {
    int n;
    cin >> n;

    vector<pair<int, int>> office(n);

    for (int i = 0; i < n; ++i) {
        int li;
        cin >> li;
        office[i] = {li, i + 1};
    }

    sort(office.begin(), office.end());

    for (int i = 0; i < n; ++i) {
        cout << office[i].second;
        if (i < n - 1) {
            cout << " ";
        }
    }

    return 0;
}
```

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

5. Результати виконання завдань, тестування та фактично затрачений час:

Завдання №1 - VNS Lab 2 - Task 1- 10.

```
Сума ряду з точністю 0.0001 дорівнює: 1.59226
PS D:\shi\epics\epic_3>
```

Витрачений час: 40 хвилин

Завдання №2 - VNS Lab 3 - Task 1- 10.

```
x       Series (fixed n)        Series (epsilon)        Exact
0.100000        2.691268                2.691265                2.691268
0.190000        2.622330                2.622328                2.622330
0.280000        2.515252                2.515250                2.515252
0.370000        2.376117                2.376116                2.376117
0.460000        2.212426                2.212415                2.212426
0.550000        2.032383                2.032377                2.032383
0.640000        1.844196                1.844192                1.844196
0.730000        1.655450                1.655448                1.655450
0.820000        1.472641                1.472640                1.472641
0.910000        1.300891                1.300886                1.300891
1.000000        1.143836                1.143834                1.143836
```

Витрачений час: 2 години

Завдання №3 – VNS Lab 7 - Task 1- 10.

```
Кількість точних квадратів (3 параметри): 2
Кількість точних квадратів (7 параметрів): 5
Кількість точних квадратів (11 параметрів): 7
```

Витрачений час: 30 хвилин

Завдання №4 – VNS Lab 7 - Task 2- 10.

```
Віднімання десяткових дробів: 0.3
Віднімання звичайних дробів: 0.1
```

Витрачений час: 30 хвилин

Завдання №5 - Class Practice Work - Особистий порадник.

```
Меню:
1. Показати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію: 1
Список книг:
1. Книга 1 - доступна
2. Книга 2 - доступна
3. Книга 3 - доступна
Хочете виконати іншу операцію? (y/n): y

Меню:
1. Показати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію: 2
Введіть номер книги, яку хочете позичити: 1
Ви позичили книгу
Хочете виконати іншу операцію? (y/n): y

Меню:
1. Показати всі книги
2. Позичити книгу
3. Повернути книгу
4. Вийти
Виберіть опцію: 3
Введіть номер книги, яку хочете повернути: 3
Ця книга вже є в бібліотеці
Хочете виконати іншу операцію? (y/n): n
PS D:\shi\epics\epic_3>
```

Витрачений час: 2 години

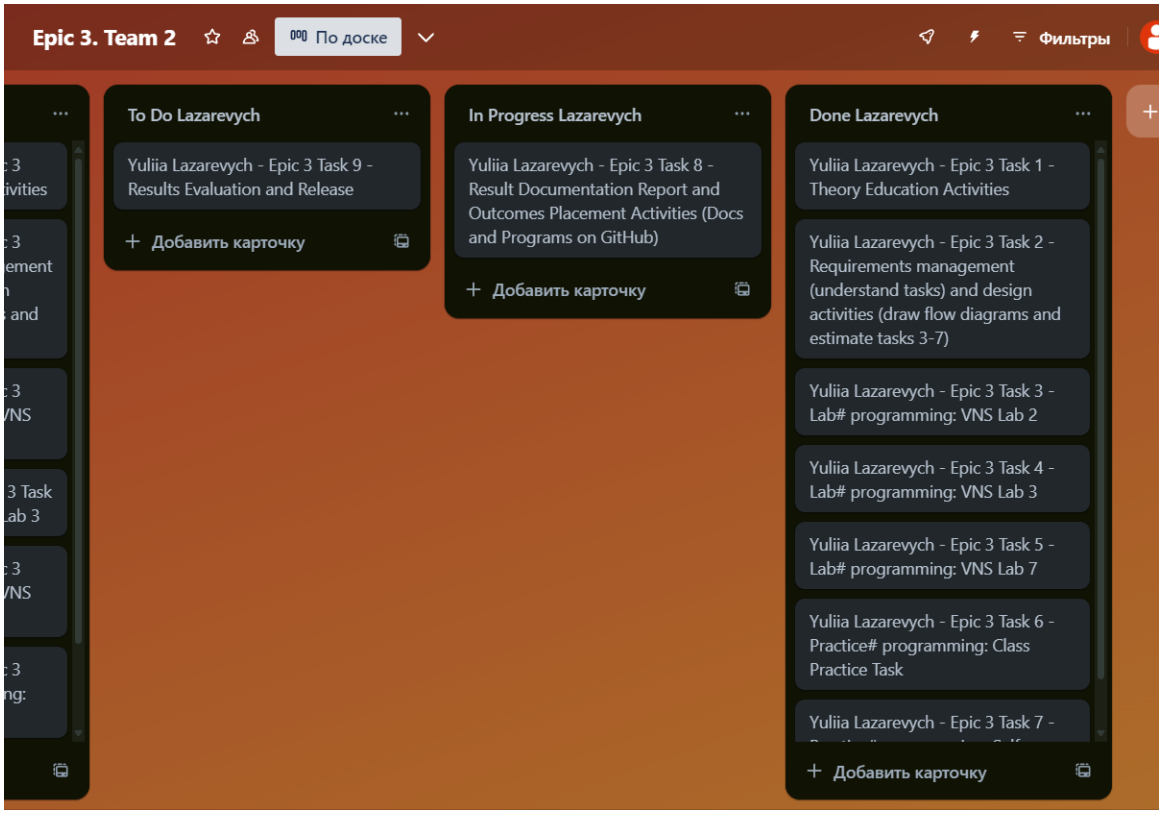
Завдання №6 - Self Practice Work –Щасливий результат

```
3
5
10
8
1 3 2
```

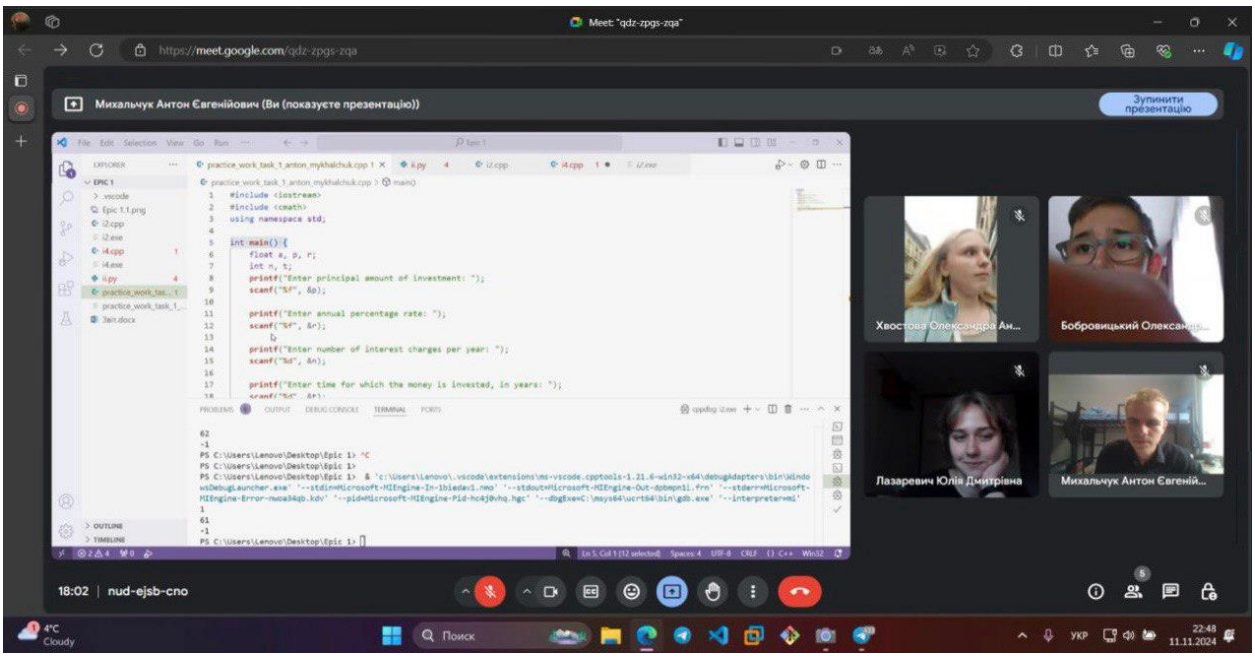
Витрачений час: 30 хвилин

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 C та C++. Також я написала шість кодів де застосувала нові знання на практиці.