Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

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

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

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

до:

ВНС Лабораторної Роботи № 2

ВНС Лабораторної Роботи № 3

ВНС Лабораторної Роботи № 7

Практичних Робіт до блоку № 3

**Виконала:**

Студентка групи ШІ-13

Козак Наталія Володимирівна

Львів - 2024

**Тема роботи:** Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції

**Мета Роботи:** Ознайомитись з циклами, функціями, простором імен. Реалізувати перевантаження функцій, функції з змінною кількістю параметрів, рекурсію.

**Теоретичні відомості:**

1. Введення в Цикли та їх Види в С++:

○ Значення та роль циклів у програмуванні.

○ Огляд видів циклів: for, while, do-while.

○ Синтаксис та основи використання кожного типу циклу.

○ Приклади базових циклів для різних задач.

1. Управління Виконанням Циклів:

○ Застосування операторів break та continue.

○ Умови завершення циклів.

○ Передчасне завершення виконання циклу.

○ Приклади та вправи з управлінням циклами.

1. Вкладені Цикли:

○ Поняття та важливість вкладених циклів.

○ Реалізація вкладених циклів: приклади для різних сценаріїв.

○ Практичні завдання на вкладені цикли.

1. Основи Функцій у С++:

○ Визначення та оголошення функцій.

○ Параметри функцій: передача за значенням і за посиланням.

○ Параметри за замовчуванням.

○ Повернення значень з функцій.

○ Приклади створення та використання функцій.

1. Перевантаження Функцій та Простір Імен:

○ Концепція перевантаження функцій.

○ Правила та приклади перевантаження функцій.

○ Поняття та використання просторів імен.

○ Вкладені простори імен (C++ 17)

○ Роль просторів імен у організації коду.

1. Розширені Можливості Функцій:

○ Функції зі змінною кількістю параметрів (еліпсис): синтаксис та приклади.

○ Область видимості функції – static, extern.

○ Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.

○ Передача масивів та об'єктів як параметрів.

○ Повернення масивів та об'єктів з функцій.

1. Вбудовані Функції в С++:

○ Огляд вбудованих функцій у С++.

○ Приклади використання стандартних функцій у програмуванні.

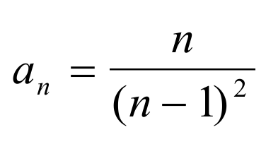
○ Роль вбудованих функцій у спрощенні коду.

○ Практичні завдання для розуміння вбудованих функцій.

Виконання роботи:

1. Опрацювання завдання та вимог до програм:

**Завдання №1 VNS Lab2 – Task 1 - 19**

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

Використовувати рекурентну формулу для отримання наступного члена ряду.

При складанні програми вважати, що точність досягнута, якщо аn < ε

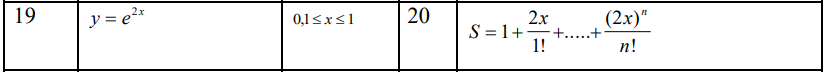
**Завдання №2 VNS Lab3 - Task1-19**

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.

**Завдання №3 VNS Lab7 - Task1-19**

Розв’язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів.

Написати функцію (або макровизначення), що визначає чи належить точка з координатами (х,у) кругу з заданим радіусом R. Написати функцію belong зі змінною кількістю параметрів, що визначає скільки точок з координатами (х,у) належать заданому кругу. Написати викликаючу функцію main, що звертається до функції belong не менше трьох разів з кількістю параметрів 3, 9, 11.

**Завдання №4 VNS Lab7 – Task2-19**

Написати перевантажені функції й основну програму, що їх викликає.

а) для масиву цілих чисел знаходить мінімальний елемент;

б) для рядка знаходить довжину найкоротшого слова .

**Завдання №5 Class Practice Work**

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

Програма повинна вміти

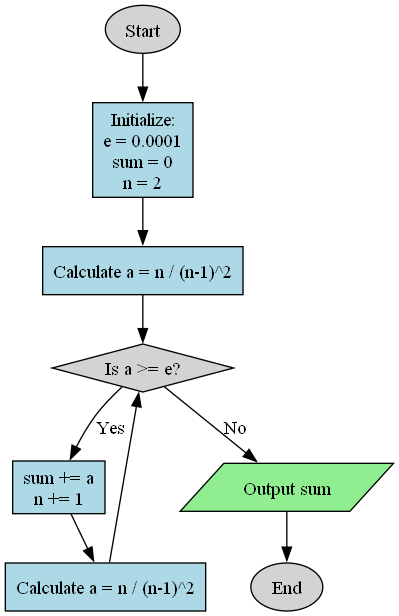
* Перерахувати всі книги.
* Дозволити взяти книгу (за наявності).
* Дозволити повернення книги.

**Завдання №6 Self Practice Work**

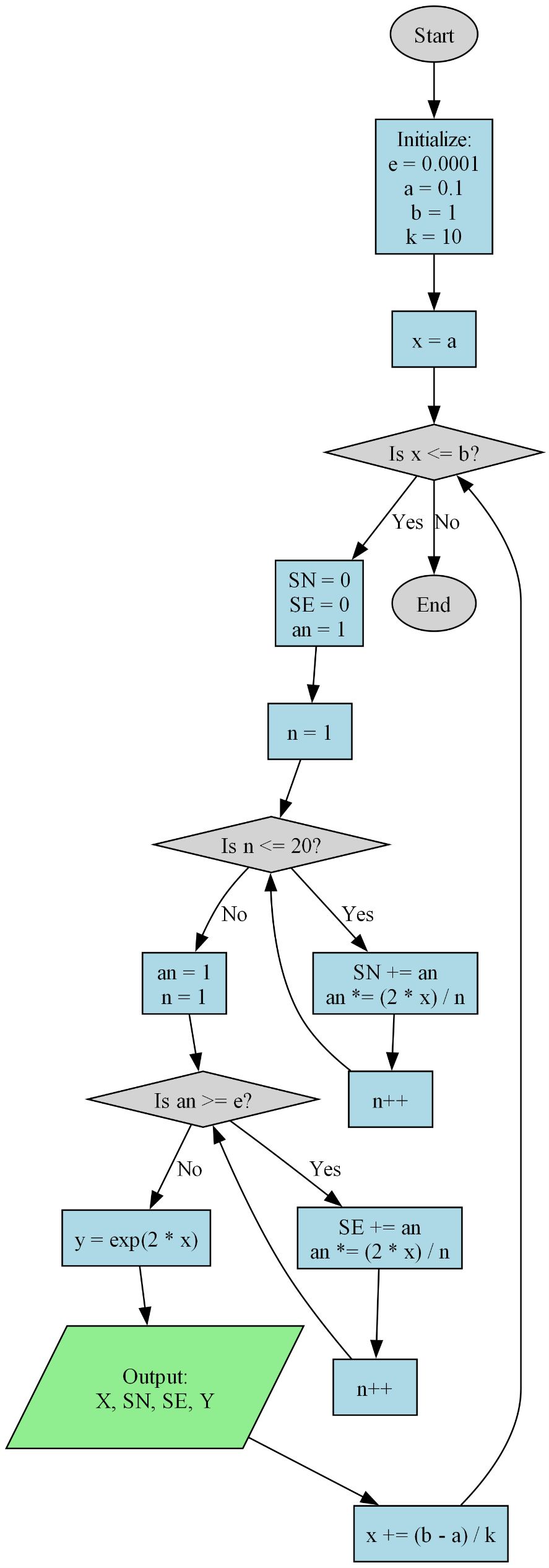
Little Imp has **N** teeth. The i-th tooth has a sharp coeficient Ai.

Little Imp calls a tooth “cool” if its sharp coeficient is at least **K**. Your task is to find the maximal number of continuous “cool” teeth Little Imp has.

2. Дизайн та планована оцінка часу виконання завдань:

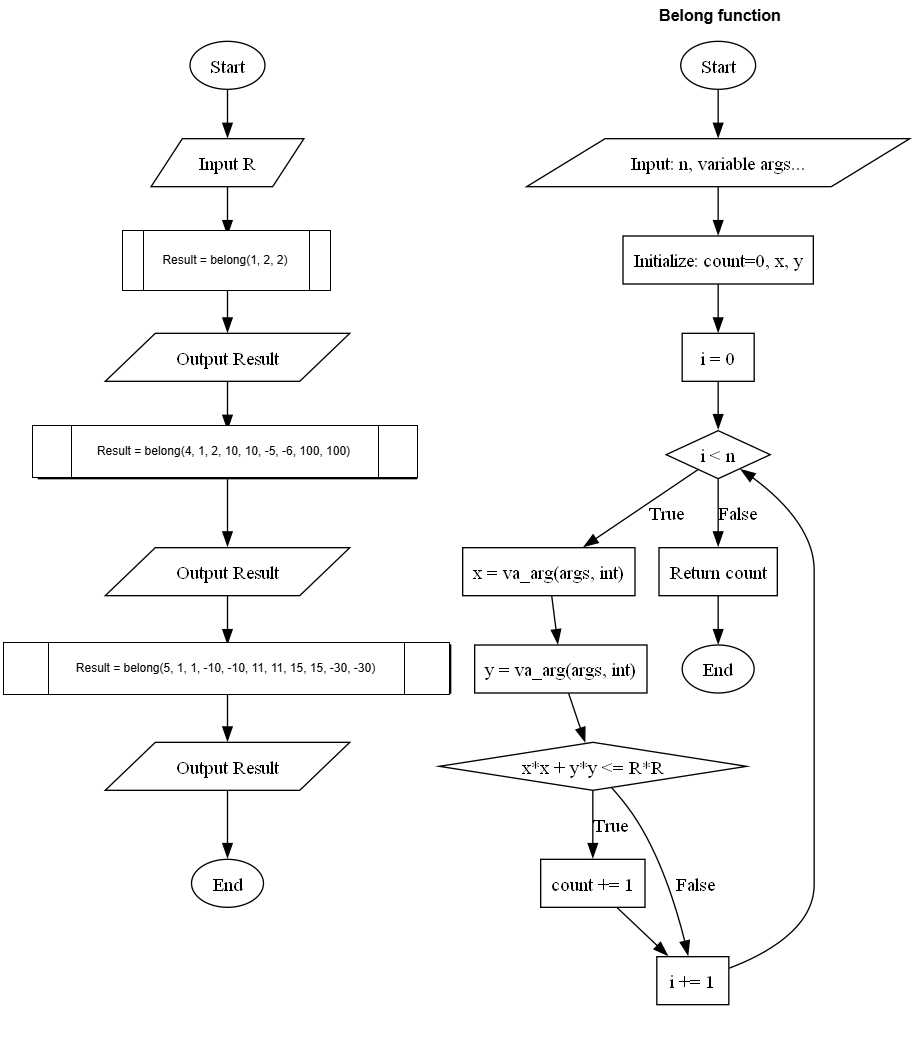
**Програма №1 VNS Lab2 – Task 1 - 19**

Планований час на реалізацію – 15хв

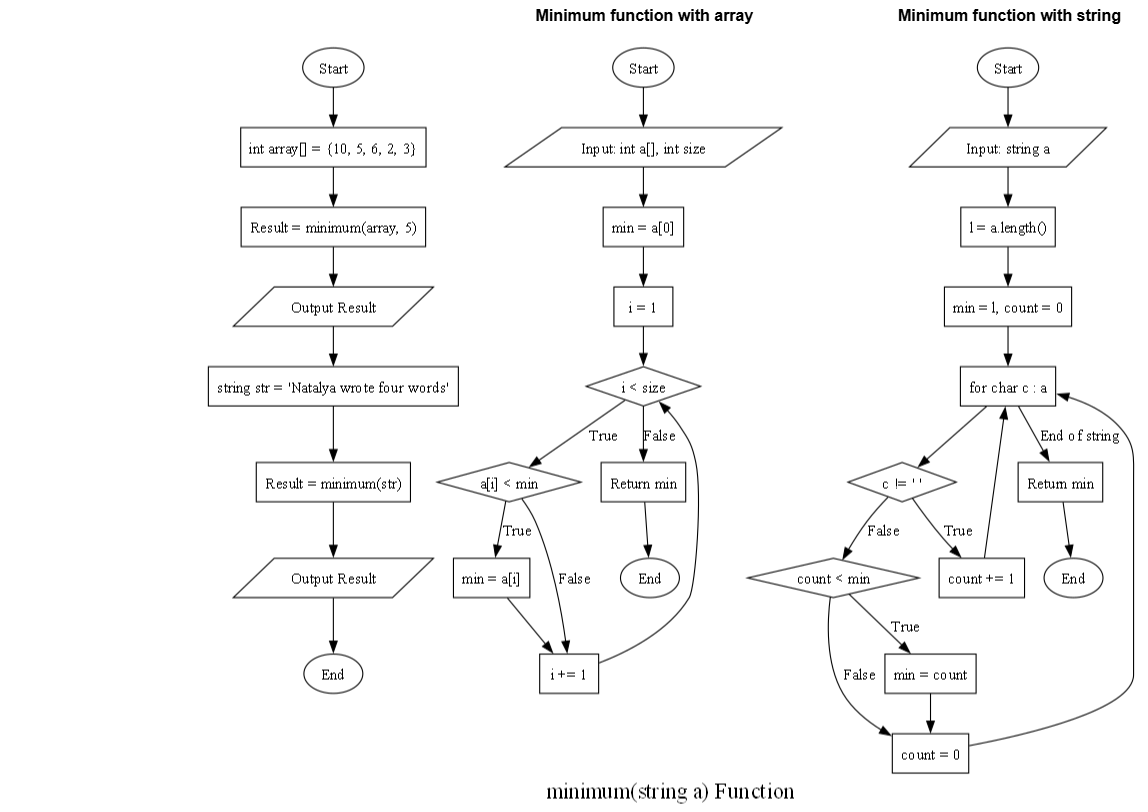
**Програма №2 VNS Lab3 - Task1-19**

Планований час на реалізацію – 20хв

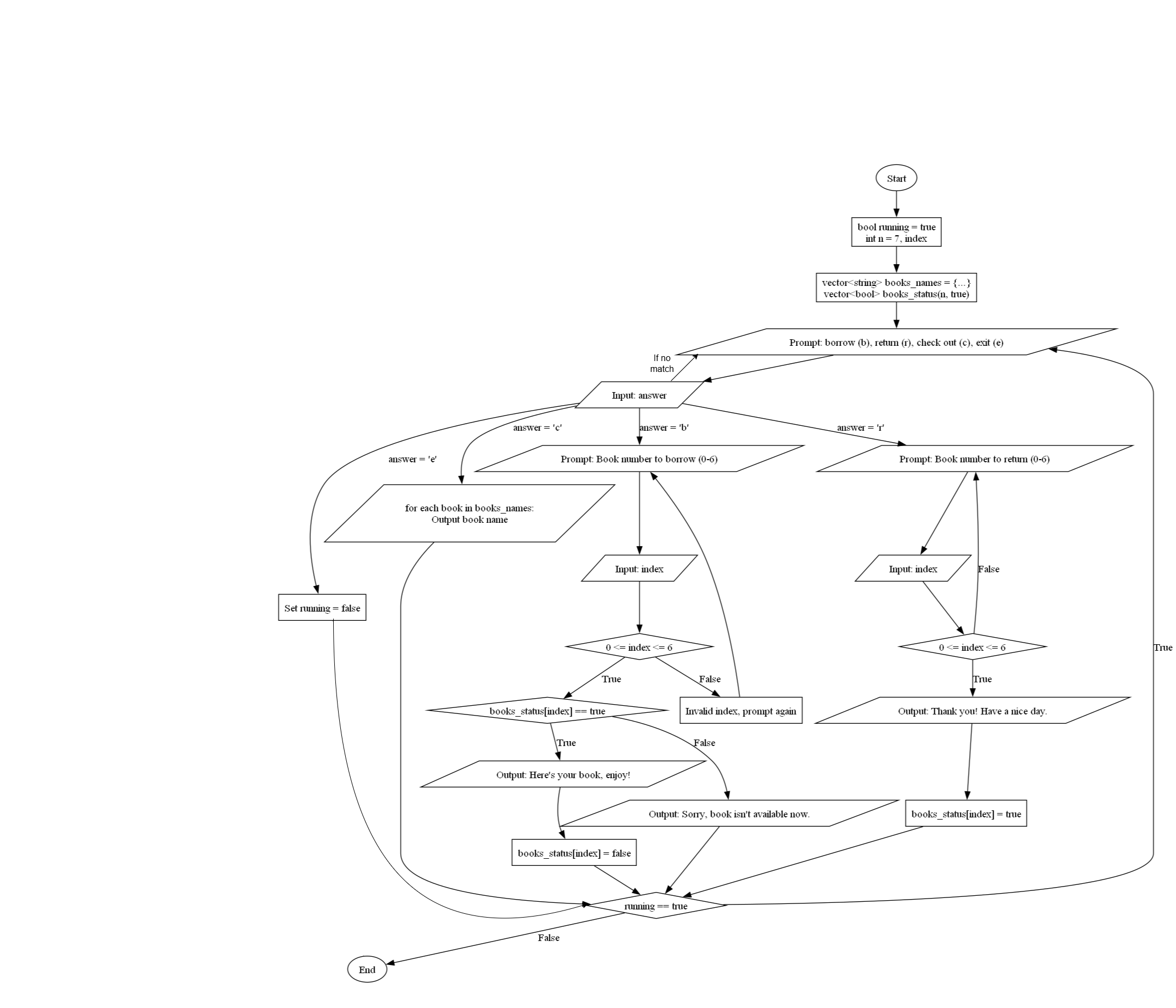
**Програма №3 VNS Lab7 - Task1-19**



Планований час на реалізацію – 25хв

**Програма №4 VNS Lab7 – Task2-19**

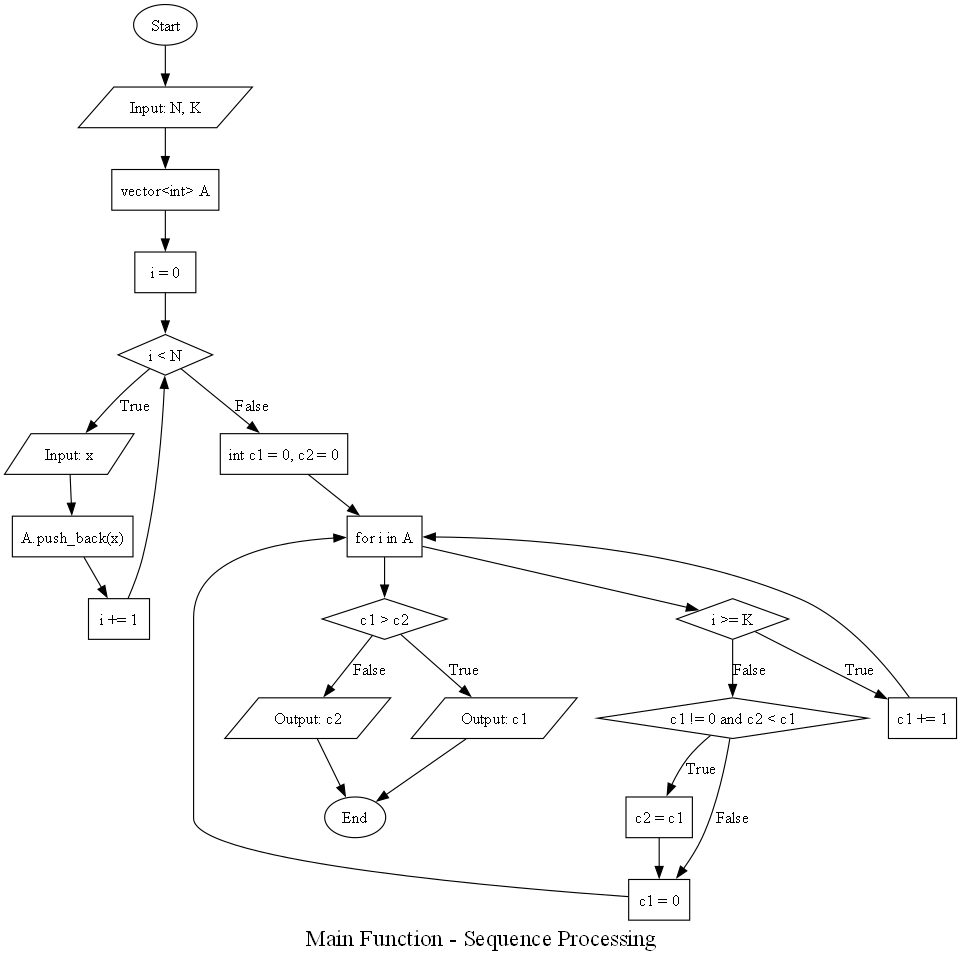
Планований час на реалізацію – 20хв

**Програма №5 Class Practice Work**

For each book in books\_names

Output book name

Планований час на реалізацію – 30хв

**Програма №6 Self Practice Work**

after the last element

Планований час на реалізацію – 15хв

3. Код програм і фактично затрачений час

**Завдання №1 VNS Lab2 – Task 1 – 19**

#include <iostream>

#include <cmath>

using namespace std;

int main(){

    float e = 0.0001, a;

    double sum = 0;

    int n = 2;

    a = n/pow(n-1, 2);

    while(a >= e){

        sum += a;

        n += 1;

        a = n/pow(n-1, 2);

    }

    cout << endl << sum;

    return 0;

}

Фактичний час виконання – 15хв

**Завдання №2 VNS Lab3 - Task1-19**

#include <stdio.h>

#include <cmath>

using namespace std;

int main(){

    float e = 0.0001, a = 0.1;

    int b = 1, k = 10;

    double SN = 0, SE = 0, y;

    for(float x = a; x <= b; x+=(b-a)/k){

        double an = 1;

        SN = 0;

        SE = 0;

        for(int n = 1; n <= 20; n++){

            SN += an;

            an \*= 2\*x/n;

        }

        an = 1;

        int n = 1;

        while(an >= e){

            SE += an;

            an \*= 2\*x/n;

            n++;

        }

        y = exp(2\*x);

        printf("\nX=%f SN=%f SE=%f Y=%f", x, SN, SE, y);

return 0;

    }

Фактичний час виконання – 20хв

**Завдання №3 VNS Lab7 - Task1-19**

#include <iostream>

#include <cstdarg>

using namespace std;

int R;

int belong(int n, ...){

        va\_list args;

        va\_start(args, n);

        int count = 0, x, y;

        for(int i = 0; i < n ; i+=1){

            x = va\_arg(args, int);

            y = va\_arg(args, int);

            if(x\*x + y\*y <= R\*R){

                count += 1;

            }

        }

        va\_end(args);

        return count;

    }

int main(){

    cout << endl << "R = ";

    cin >> R;

    cout << belong(1, 2, 2) << endl;

    cout << belong(4, 1, 2, 10, 10, -5, -6, 100, 100) << endl;

    cout << belong(5, 1, 1 -10, -10, 11, 11, 15, 15, -30, -30) << endl;

    return 0;

}

Фактичний час виконання – 30хв

**Завдання №4 VNS Lab7 – Task2-19**

#include <iostream>

using namespace std;

int minimum(int a[], int size){

    int min = a[0];

    for(int i = 1; i < size; i++){

        if(a[i] < min){

            min = a[i];

        }

    }

    return min;

}

int minimum(string a){

    int l = a.length();

    int min = l, count = 0;

    for(char c : a){

        if(c != ' '){

            count += 1;

        }

        else{

            if(count < min) {

                min = count;

            }

            count = 0;

        }

    }

    return min;

}

int main(){

    int array[] = {10,5,6,2,3};

    cout << endl << minimum(array, 5);

    string str = "Natalya wrote four words";

    cout << endl << minimum(str);

    return 0;

}

Фактичний час виконання – 25хв

**Завдання №5 Class Practice Work**

#include <iostream>

#include <vector>

using namespace std;

int main(){

    bool running = true;

    int n = 7, index;

    vector<string> books\_names = {"Anna Karenina", "To Kill a Mockingbird", "The Great Gatsby", "One Hundred Years of Solitude", "A Passage to India", "Invisible Man", "Don Quixote"};

    vector<bool> books\_status(n, true);

    char answer;

    do{

        cout << endl << endl << "Welcome to the library, would you like to borrow (b), return (r) or check out (c) books? To exit - type (e)" << endl;

        cin >> answer;

        switch(answer){

            case 'b':

                borrowing:

                    cout << "Number of book you want to borrow(0-6)"<< endl;

                    cin >> index;

                    if (index >= 0 && index <=6){

                        if (books\_status[index] == true){

                            cout << "Here's your book, enjoy!"<< endl;

                            books\_status[index] = false;

                        }

                        else{

                            cout << "Sorry, book isn't available now." << endl;

                        }

                    }

                    else{

                        goto borrowing;

                    }

                    break;

            case 'r':

                returning:

                    cout << "Number of book you want to return(0-6)"<< endl;

                    cin >> index;

                    if (index >= 0 && index <=6){

                        cout << "Thank you! Have a nice day."<< endl;

                        books\_status[index] = true;

                    }

                    else{

                        goto returning;

                    }

                break;

            case 'c':

                for(string book : books\_names){

                    cout << book << endl;

                }

                break;

            case 'e':

                running = false;

                break;

        }

    }

    while(running);

    return 0;

Фактичний час виконання – 40хв

**Завдання №6 Self Practice Work**

#include <iostream>

#include <vector>

using namespace std;

int main(){

    int N, K, x;

    cin >> N >> K;

    vector<int> A;

    for(int i = 0; i < N; i++){

        cin >> x;

        A.push\_back(x);

    }

    int c1 = 0, c2 = 0;

    for (int i : A){

        if(i >= K){

            c1 += 1;

        }

        else{

            if (c1 != 0 && c2 < c1){

                c2 = c1;

            }

            c1 = 0;

        }

    }

    if(c1 > c2){

        cout << c1;

    }

    else{

        cout << c2;

    }

    return 0;

}

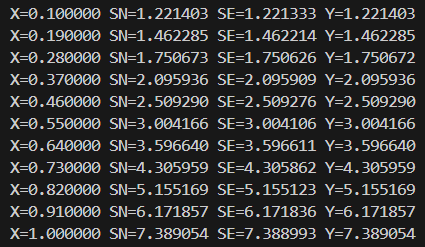
Фактичний час виконання – 15хв

4. Результати виконання завдань і тестування:

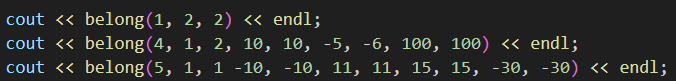
**Завдання №1 VNS Lab2 – Task 1 – 19**

****

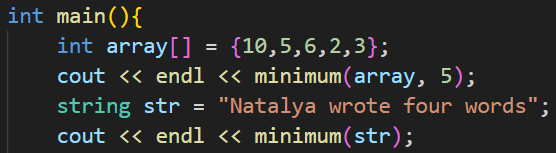
**Завдання №2 VNS Lab3 - Task1-19**

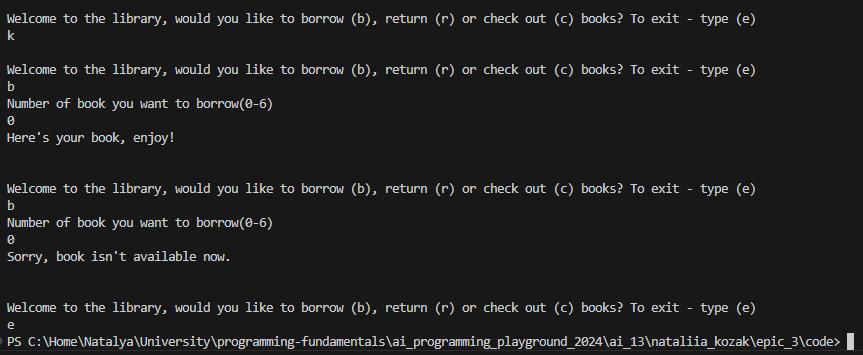
****

**Завдання №3 VNS Lab7 - Task1-19**

****

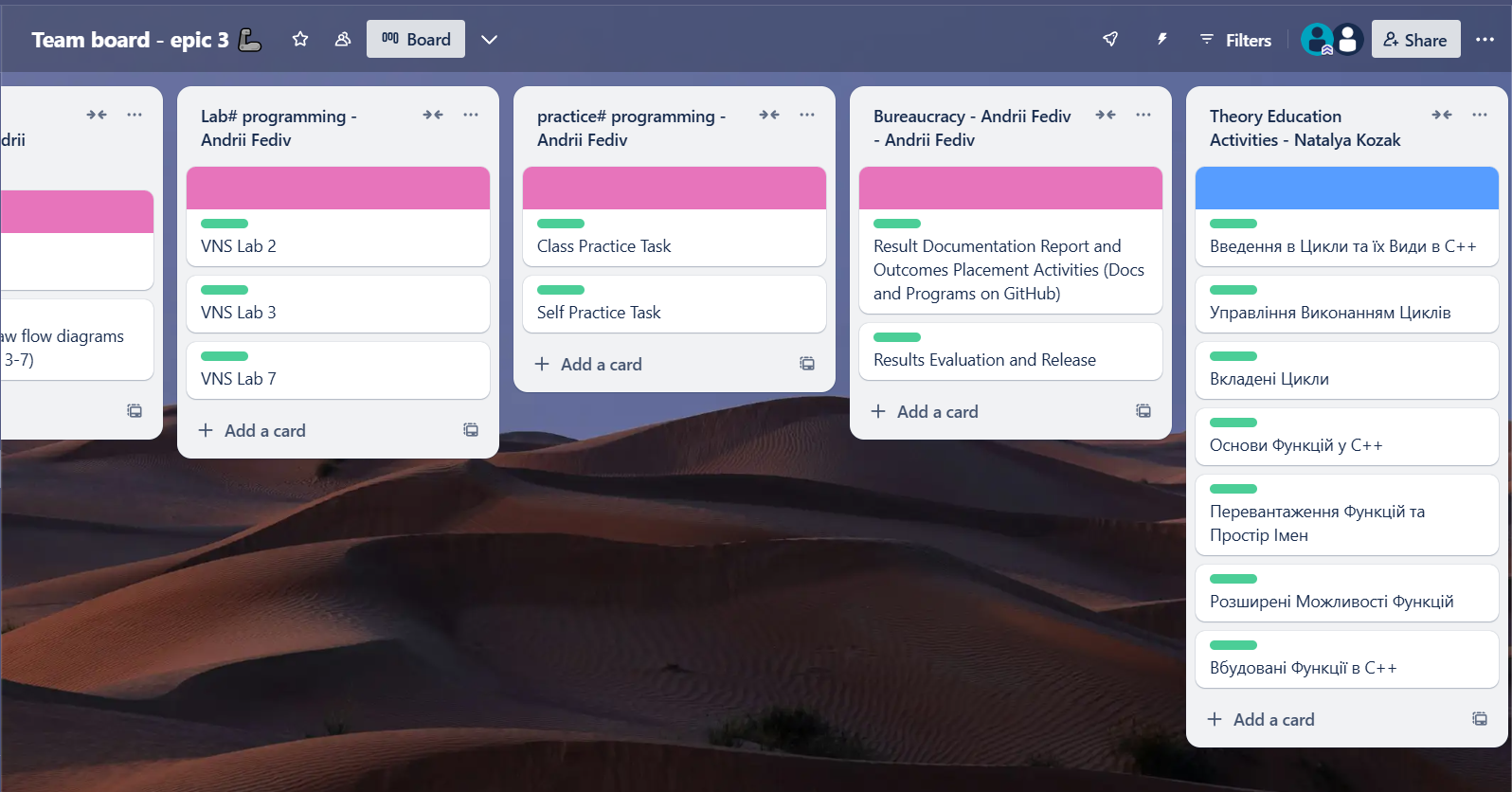
**Завдання №4 VNS Lab7 – Task2-19**

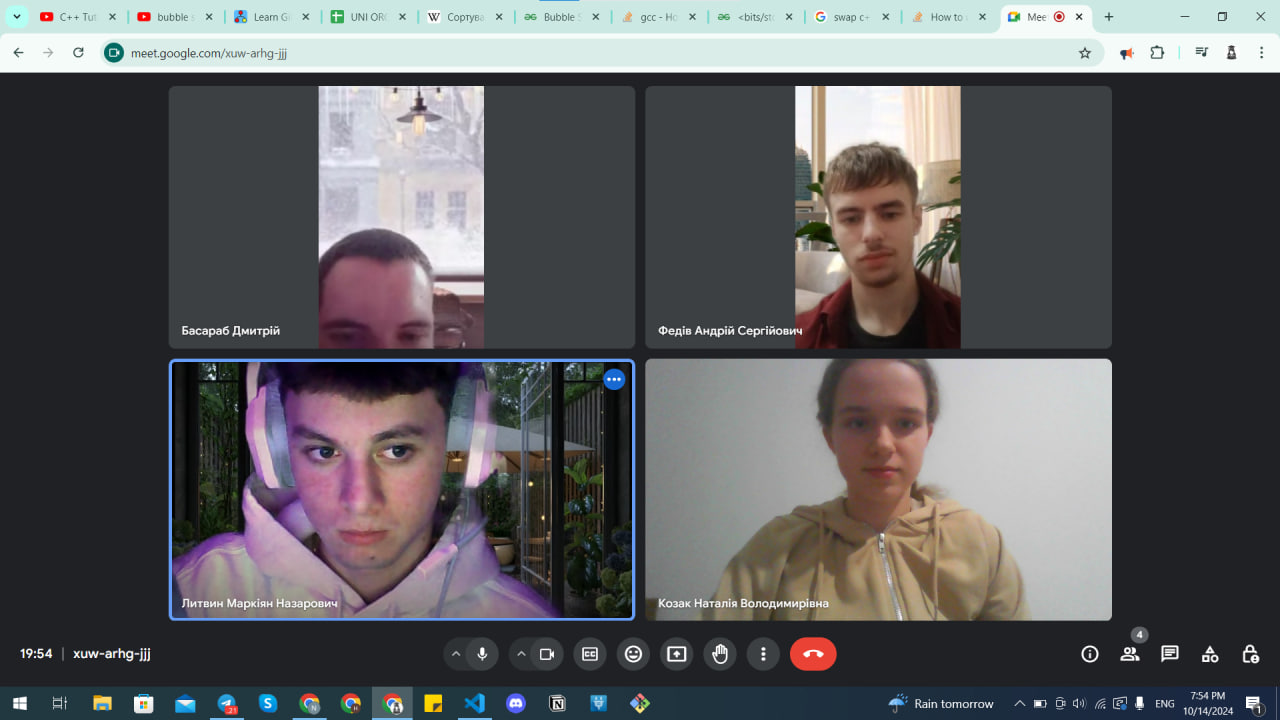
****

**Завдання №5 Class Practice Work**

**Завдання №6 Self Practice Work**

5. Кооперація з командою





Висновки:

У результаті виконання цієї роботи я отримала глибше розуміння основних концепцій програмування в C++, зокрема циклів, функцій, перевантаження функцій, рекурсії та роботи з просторами імен. Робота над циклами дозволила зрозуміти, як ефективно використовувати різні типи циклів для виконання задач, а також як керувати виконанням циклів за допомогою операторів `break` та `continue`.

Застосування вкладених циклів виявилося корисним для вирішення складніших задач, а функції з змінною кількістю параметрів і рекурсія дозволили розширити можливості створюваних програм. Перевантаження функцій та робота з просторами імен допомогли краще організувати код і уникнути конфліктів імен. Застосування вбудованих функцій стало важливим кроком до оптимізації програм і зменшення їх складності.

Ця робота сприяла глибшому розумінню важливих аспектів програмування на C++ та допомогла набути практичних навичок для створення ефективного та організованого коду.