

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 3

На тему: «Програмування: алгоритм, програма, код. Системи числення.
Двійкова система числення. Розробка та середовище розробки програми.»

з дисципліни: «Основи програмування»

до:

Практичних Робіт до блоку № 3

Виконав:
Студент групи ШІ-11
Силіч Богдан

Львів 2024

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

Мета: ознайомлення з основами роботи з циклами та вкладеними циклами, а також способами завершення їх виконання. Розглянути функції, простір імен, перевантаження функцій, функції з еліпсисом, рекурсія та вбудовані функції для поглиблення розуміння структури і поведінки програм.

Теоретичні відомості:

1. Введення в Цикли та їх Види в C++:

- 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.

2. Управління Виконанням Циклів:

- Застосування операторів break та continue.
- Умови завершення циклів.
- Передчасне завершення виконання циклу.
- Приклади та вправи з управлінням циклами.

3. Вкладені Цикли:

- Поняття та важливість вкладених циклів.
- Реалізація вкладених циклів: приклади для різних сценаріїв.
- Практичні завдання на вкладені цикли.

4. Основи Функцій у C++:

- Визначення та оголошення функцій.
- Параметри функцій: передача за значенням і за посиланням.
- Параметри за замовчуванням.
- Повернення значень з функцій.
- Приклади створення та використання функцій.

5. 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.

6. 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.

7. Вбудовані Функції в C++:

- Огляд вбудованих функцій у C++.
- Приклади використання стандартних функцій у програмуванні.
- Роль вбудованих функцій у спрощенні коду.
- Практичні завдання для розуміння вбудованих функцій.

Індивідуальний план опрацювання теорії:

Введення в Цикли та їх Види в C++

Управління Виконанням Циклів

Вкладені Цикл

Основи Функцій у C++

Перевантаження Функцій та Простір Імен

Розширені Можливості Функцій

Вбудовані Функції в C++

Джерела:

- Chat gpt
- Список відтворення на YouTube (<https://youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&si=sXvmPdnGkwvJLXUi>)
- Власний досвід

Виконання роботи:

VNS Lab 2 Task 1 variant 6

Використовуючи оператор циклу, знайти суму елементів, зазначених у конкретному варіанті. Результат надрукувати, надавши відповідний заголовок.

б) Знайти суму ряду з точністю $\varepsilon=0.0001$, загальний член якого

$$a_n = \frac{1}{2^n} + \frac{1}{3^n}$$

VNS Lab 3 Task 1 variant 6

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 ($\varepsilon=0.0001$).

Для порівняння знайти точне значення функції.

№	функція	діапазон зміни аргументу	n	сума
6	$y = e^{x \cos \frac{\pi}{4}} \cdot \cos(x \sin \frac{\pi}{4})$	$0,1 \leq x \leq 1$	25	$S = 1 + \frac{\cos \frac{\pi}{4}}{1!} x + \dots + \frac{\cos n \frac{\pi}{4}}{n!} x^n$

VNS Lab 7 Task 1 & 2 variant 6

1. Написати функцію `min` зі змінною кількістю параметрів, що знаходить мінімальне із чисел типу `int`. Написати викликаючу функцію `main`, що звертається до функції `min` не менше трьох разів з кількістю параметрів 5, 10, 12.

2. Написати перевантажені функції й основну програму, що їх викликає.

а) для віднімання дійсних чисел;

б) для віднімання комплексних чисел.

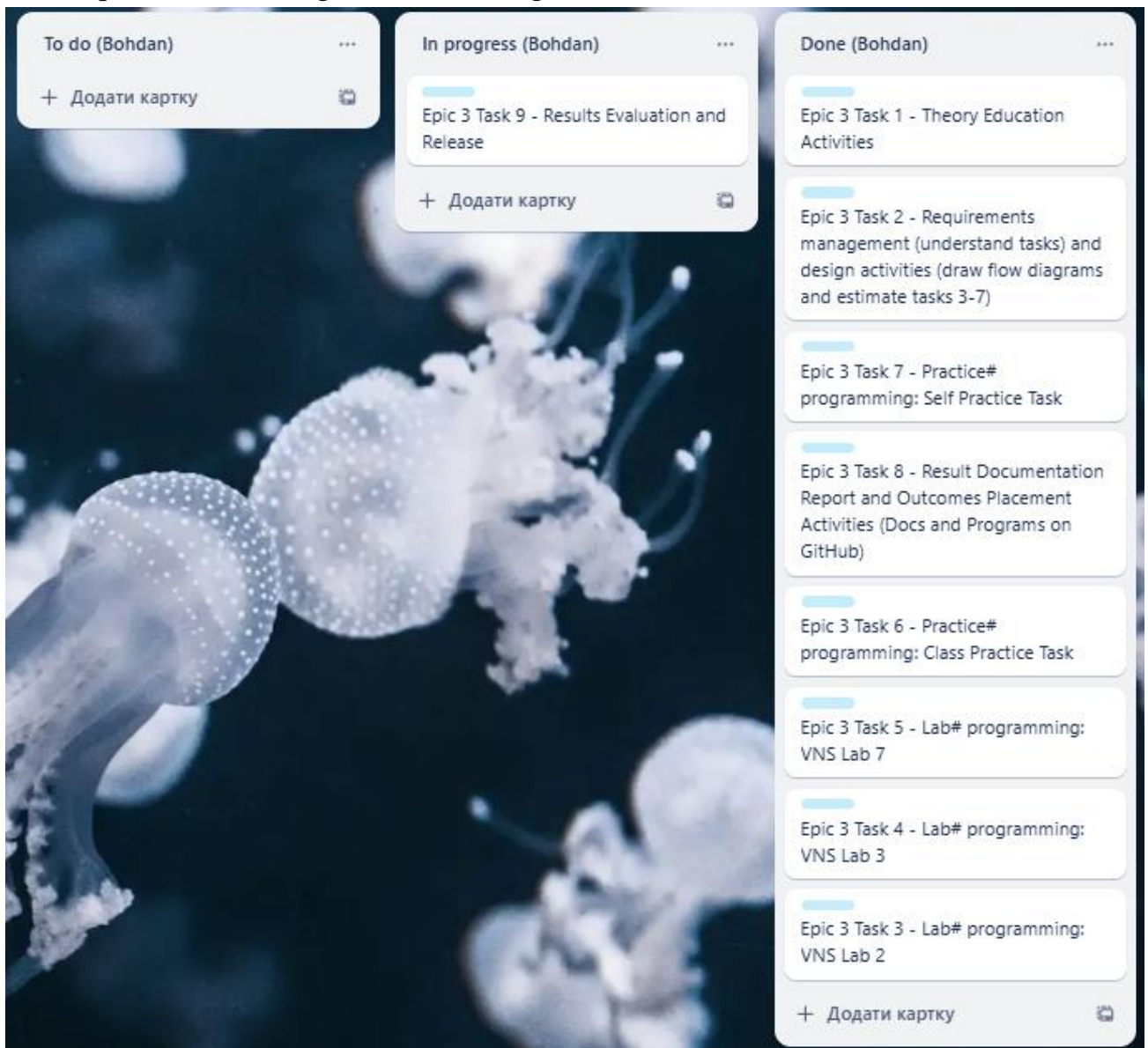
Class Practice Task:

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

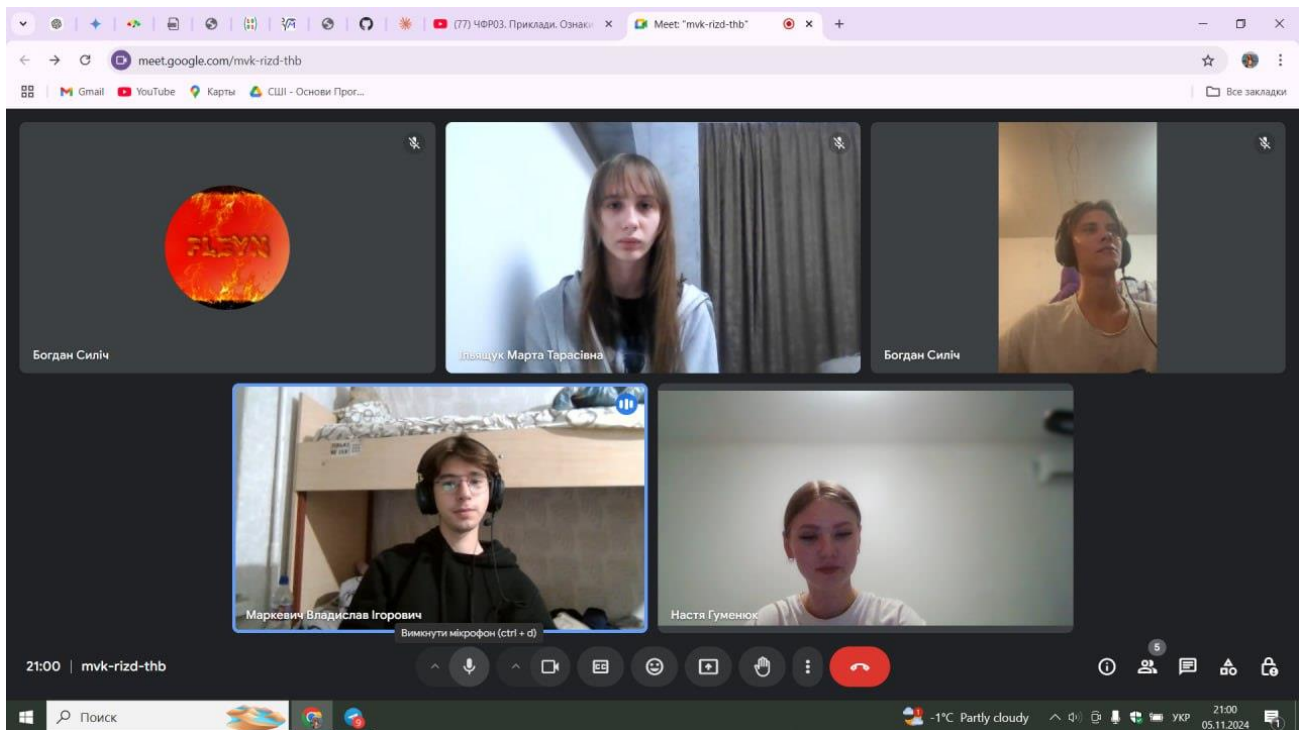
Self Practice Task:

Вам задано послідовність із n цілих чисел a_i . Ваша задача — знайти довжину найбільшої зростаючої підпослідовності заданої послідовності.

я1. Requirements management and design activities

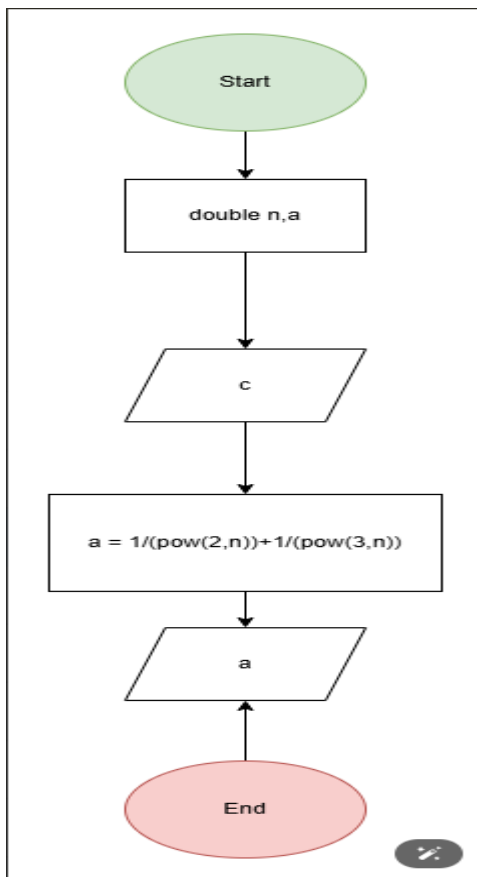


Team meeting



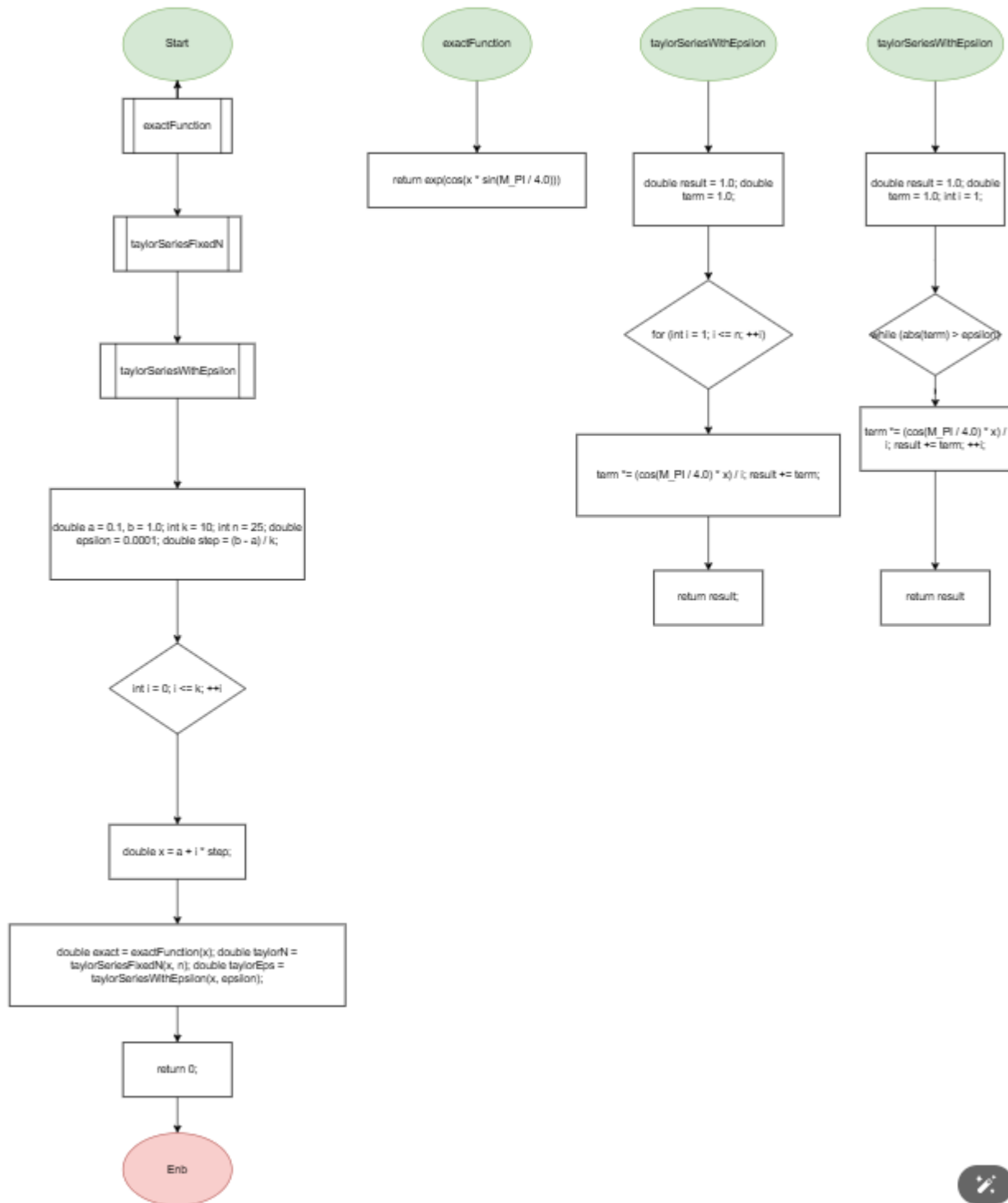
UML-diagram block-scheme for each task

VNS Lab 2 Task 1



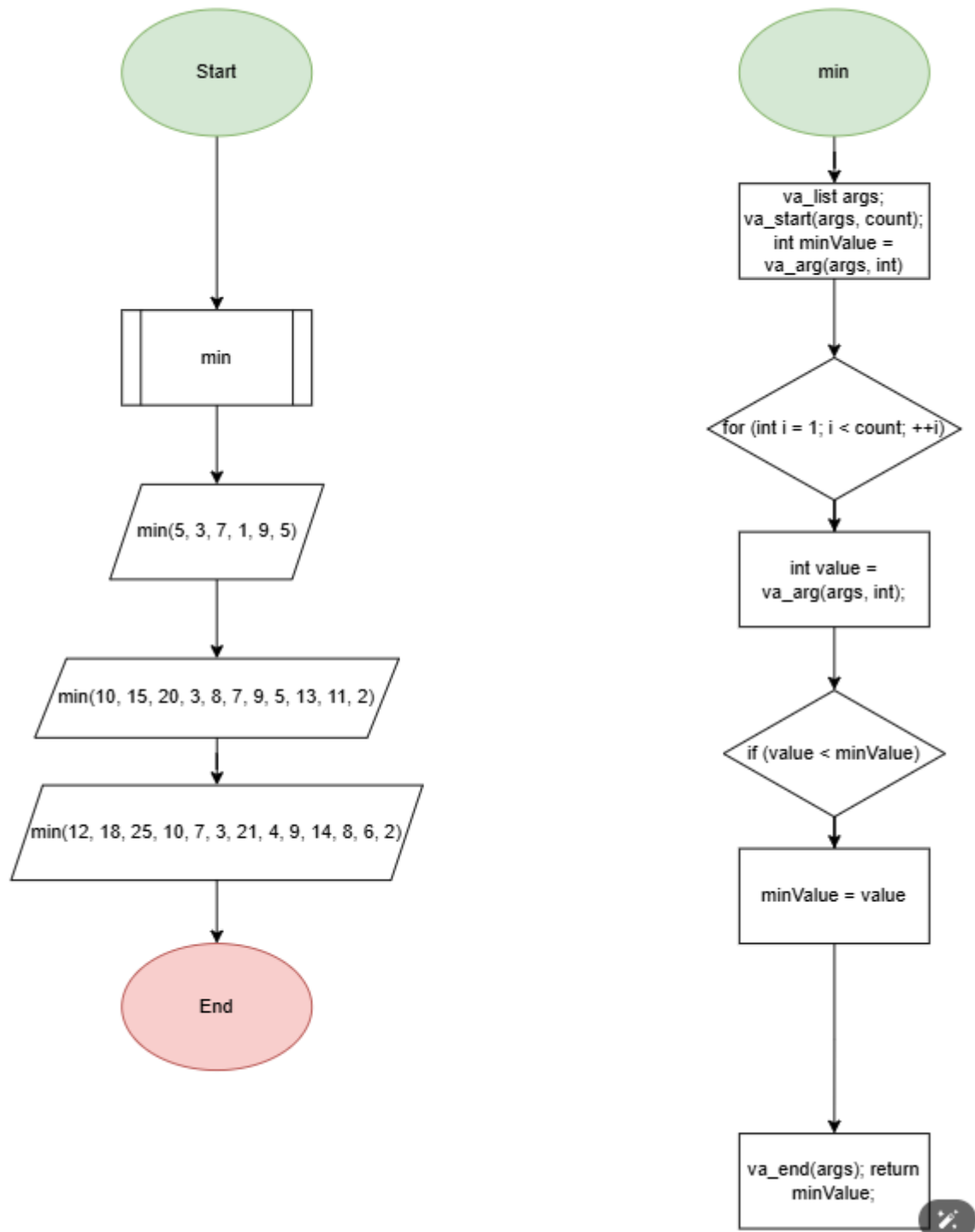
20 XB +-

VNS Lab 3 Task 1

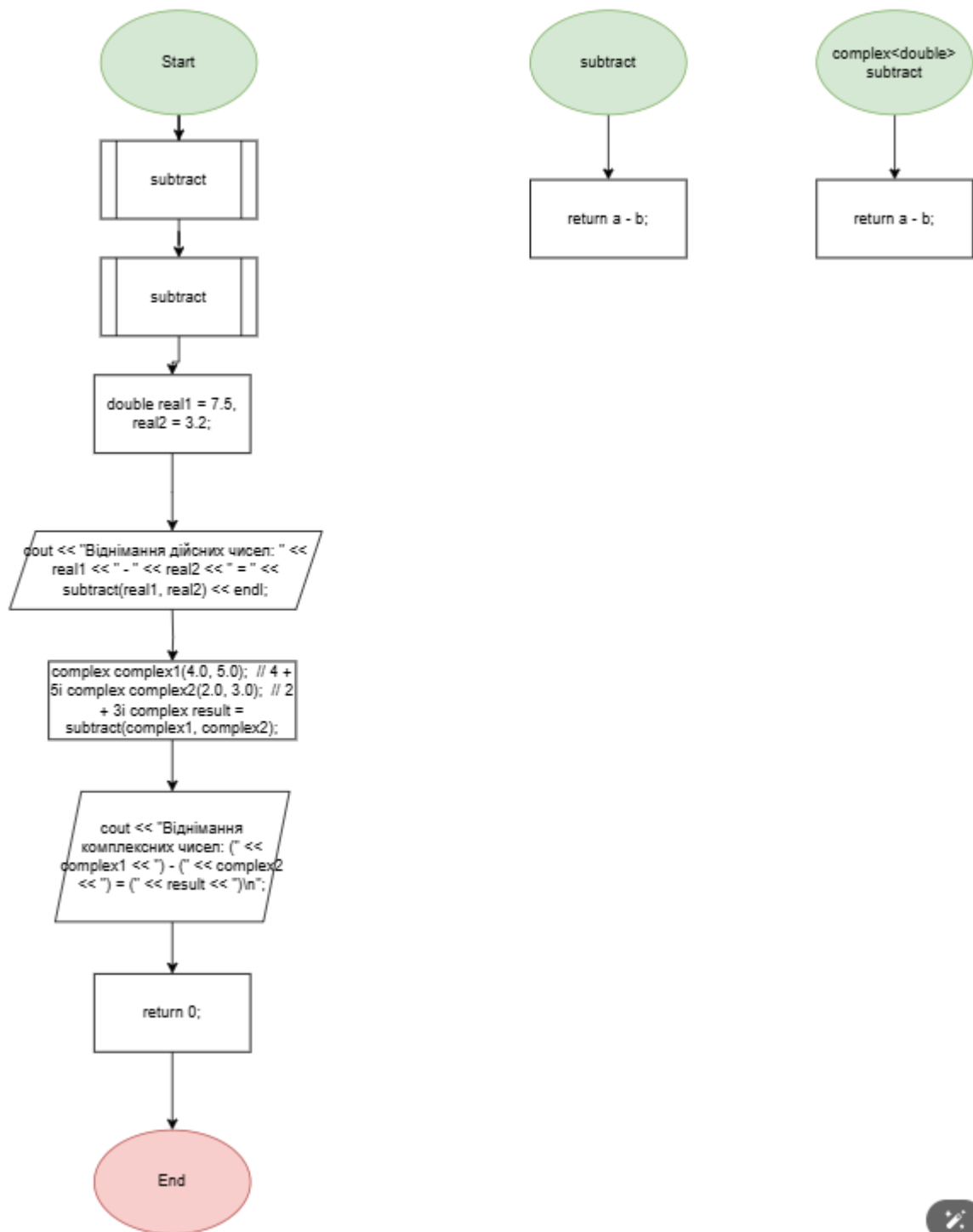


30 XB+-

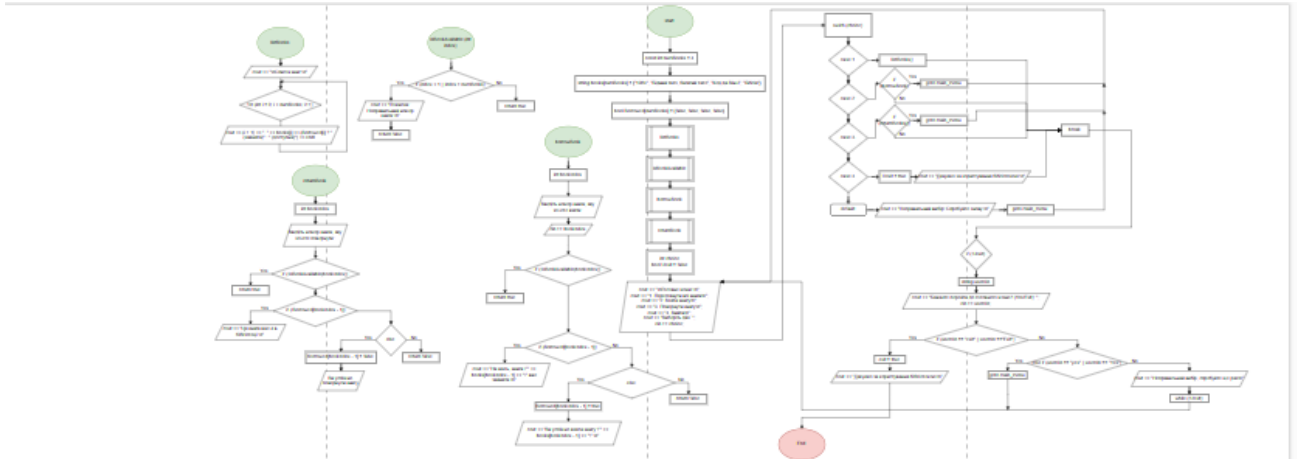
VNS Lab 7 Task 1



VNS Lab 7 Task 2

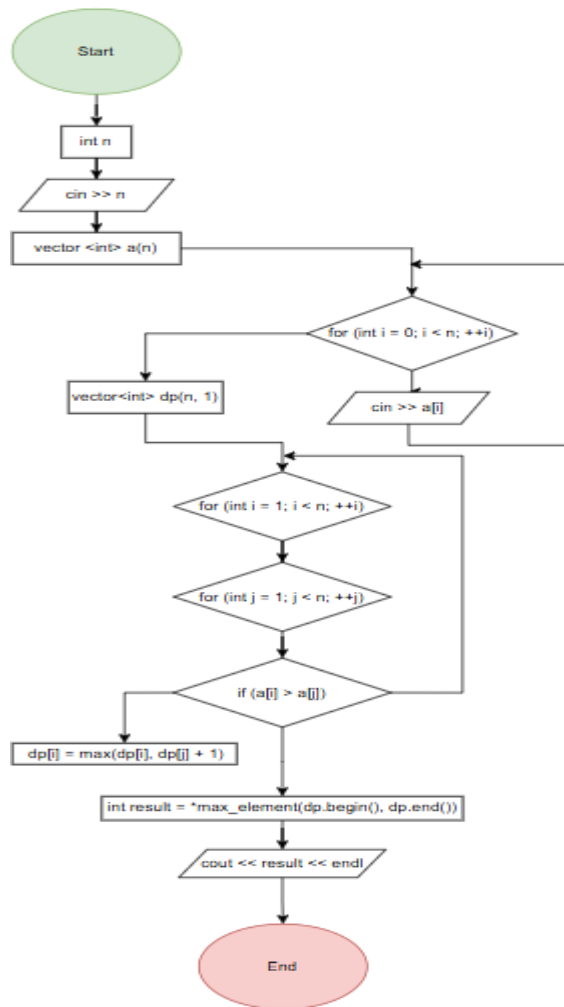


Class Practice Task



50 XB+-

Self Practice Task



20 XB+-

Код програми VNS Lab 2 Task 1

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

using namespace std;

int main(){
    double n, a;
    cin >> n;

    a = 1/(pow(2,n))+1/(pow(3,n));

    cout << setprecision(4) << fixed << "a=" << a << endl;
}
```

```
5
a= 0.0354
PS E:\Projekts\c++\Epic 3> |
```

10 хВ +-

VNS Lab 3 Task 1

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

using namespace std;

double exactFunction(double x) {
    return exp(cos(x * sin(M_PI / 4.0)));
}

double taylorSeriesFixedN(double x, int n) {
    double result = 1.0;
    double term = 1.0;
    for (int i = 1; i <= n; ++i) {
        term *= (cos(M_PI / 4.0) * x) / i;
        result += term;
    }

    return result;
}

double taylorSeriesWithEpsilon(double x, double epsilon) {
    double result = 1.0;
    double term = 1.0;
    int i = 1;

    while (abs(term) > epsilon) {
        term *= (cos(M_PI / 4.0) * x) / i;
        result += term;
        ++i;
    }

    return result;
}

int main() {
    double a = 0.1, b = 1.0;
    int k = 10;
    int n = 25;
    double epsilon = 0.0001;

    double step = (b - a) / k;

    cout << fixed << setprecision(8);
    cout << "x\tExact\tTaylor(n=25)\tTaylor(epsilon=0.0001)\n";

    for (int i = 0; i <= k; ++i) {
        double x = a + i * step;
        double exact = exactFunction(x);
        double taylorN = taylorSeriesFixedN(x, n);
        double taylorEps = taylorSeriesWithEpsilon(x, epsilon);

        cout << fixed << setprecision(4) << x << "\t" << exact << "\t" << taylorN << "\t" << taylorEps << "\n";
    }

    return 0;
}
```

x	Exact	Taylor(n=25)	Taylor(epsilon=0.0001)
0.1000	2.7115	1.0733	1.0733
0.1900	2.6939	1.1438	1.1438
0.2800	2.6657	1.2190	1.2189
0.3700	2.6273	1.2990	1.2990
0.4600	2.5794	1.3844	1.3844
0.5500	2.5227	1.4754	1.4754
0.6400	2.4580	1.5723	1.5723
0.7300	2.3862	1.6756	1.6756
0.8200	2.3084	1.7857	1.7857
0.9100	2.2256	1.9031	1.9031
1.0000	2.1388	2.0281	2.0281

PS E:\Projekts\c++\Epic 3>

VNS Lab 7 Task 1

```
#include <iostream>
#include <cstdlib>

using namespace std;

int min(int count, ...) {
    va_list args;
    va_start(args, count);

    int minValue = va_arg(args, int);
    for (int i = 1; i < count; ++i) {
        int value = va_arg(args, int);
        if (value < minValue) {
            minValue = value;
        }
    }

    va_end(args);
    return minValue;
}

int main () {
    cout << "Мінімум серед (3, 7, 1, 9, 5): " << min(5, 3, 7, 1, 9, 5) << endl;
    cout << "Мінімум серед (15, 20, 3, 8, 7, 9, 5, 13, 11, 2): " << min(10, 15, 20, 3, 8, 7, 9, 5, 13, 11, 2) << endl;
    cout << "Мінімум серед (18, 25, 10, 7, 3, 21, 4, 9, 14, 8, 6, 2): " << min(12, 18, 25, 10, 7, 3, 21, 4, 9, 14, 8, 6, 2) << endl;
}
```

```
Мінімум серед (3, 7, 1, 9, 5): 1
Мінімум серед (15, 20, 3, 8, 7, 9, 5, 13, 11, 2): 2
Мінімум серед (18, 25, 10, 7, 3, 21, 4, 9, 14, 8, 6, 2): 2
PS E:\Projekts\c++\Epic 3>
```

15 хв +-

VNS Lab 7 Task 2

```
#include <iostream>
#include <complex>

using namespace std;

double subtract(double a, double b) {
    return a - b;
}

complex<double> subtract(complex<double> a, complex<double> b) {
    return a - b;
}

int main() {
    double real1 = 7.5, real2 = 3.2;
    cout << "Віднімання дійсних чисел: " << real1 << " - " << real2 << " = " << subtract(real1, real2) << endl;

    complex<double> complex1(4.0, 5.0);
    complex<double> complex2(2.0, 3.0);
    complex<double> result = subtract(complex1, complex2);

    cout << "Віднімання комплексних чисел: (" << complex1 << ") - (" << complex2 << ") = (" << result << ")\n";

    return 0;
}
```

```
Віднімання дійсних чисел: 7.5 - 3.2 = 4.3
Віднімання комплексних чисел: ((4,5)) - ((2,3)) = ((2,2))
PS E:\Projekts\c++\Epic 3>
```

20 хв +-

Висновки:

Виконуючи цей епік я поглибив свої знання з циклів, що допоможе мені оптимізувати виконання програм. Розуміння функцій, перевантаження дозволить мені створювати гнучкі й потужні рішення для різних завдань. Знання вбудованих функцій і роботи з простором імен допомагало уникати конфліктів та забезпечити більш чітку організацію коду.