

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 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 ІІІ-13  
Осінна Єлизавета Сергіївна

Львів 2024

## Тема роботи:

"Використання основних операторів мови С. Обчислення функцій з використанням їхнього розкладу в степеневий ряд. Функції зі змінною кількістю параметрів та перевантаження функцій в С++"

## Мета роботи:

Отримання навичок у виборі й використанні операторів С; знайомство з ітераційними процесами. Практика в організації ітераційних й арифметичних циклів. Знайомство з організацією функцій зі змінною кількістю параметрів та перевантаженням функцій.

## 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.
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 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.
5. Перевантаження Функцій та Простір Імен:
  - Концепція перевантаження функцій.
  - Правила та приклади перевантаження функцій.
  - Поняття та використання просторів імен.
  - Вкладені простори імен (С++ 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 С++:
  - Огляд вбудованих функцій у С++.
  - Приклади використання стандартних функцій у програмуванні.
  - Роль вбудованих функцій у спрощенні коду.
  - Практичні завдання для розуміння вбудованих функцій.

1) Індивідуальний план опрацювання теорії:

- **C Variadic function - Sort a variable number of integers**
- Оператори break і continue в C++ / aCode
- **goto Statement in C++**
- **continue Statement in C++ - GeeksforGeeks**
- How to create functions with a variable number of arguments using stdarg.h | C Programming Tutorial  
[https://www.youtube.com/watch?v=3iX9a\\_l9W9Y&t=528s](https://www.youtube.com/watch?v=3iX9a_l9W9Y&t=528s)
- **Уроки C++ для початківців / #11 – Перевантаження функцій**

## Виконання роботи:

### 1. Опрацювання завдання та вимог до програм та середовища:

Завдання № 1 VNS Lab 2 – Task 1

- Варіант 5
  - Деталі завдання
- 5) Знайти суму ряду з точністю  $\epsilon=0.0001$ , загальний член якого

$$a_n = \frac{(-1)^{n-1}}{n^n}$$

Завдання № 2 VNS Lab 3 – Task 1

- Варіант 5
- 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  $\epsilon$  ( $\epsilon=0.0001$ ).
- Для порівняння знайти точне значення функції.

5	$y = e^x$	$1 \leq x \leq 2$
---	-----------	-------------------

Завдання № 3 VNS Lab 7 – Task 1

- Варіант 5
- Написати функцію sum зі змінною кількістю параметрів, що знаходить суму чисел типу int за формулою:
- $S=a1*a2-a2*a3+a3*a4- \dots$
- Написати викликаючу функцію main, що звертається до функції sum не менше трьох разів з кількістю параметрів 5, 10, 12.

#### Завдання № 4 VNS Lab 7 – Task 2

- Варіант 5
- Деталі завдання
- Написати перевантажені функції й основну програму, що їх викликає.
- а) для множення дійсних чисел;
- б) для множення комплексних чисел.

#### Завдання № 5 Class Practice work

- 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.

#### Вимоги:

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

#### Завдання № 6 Self Practice Work

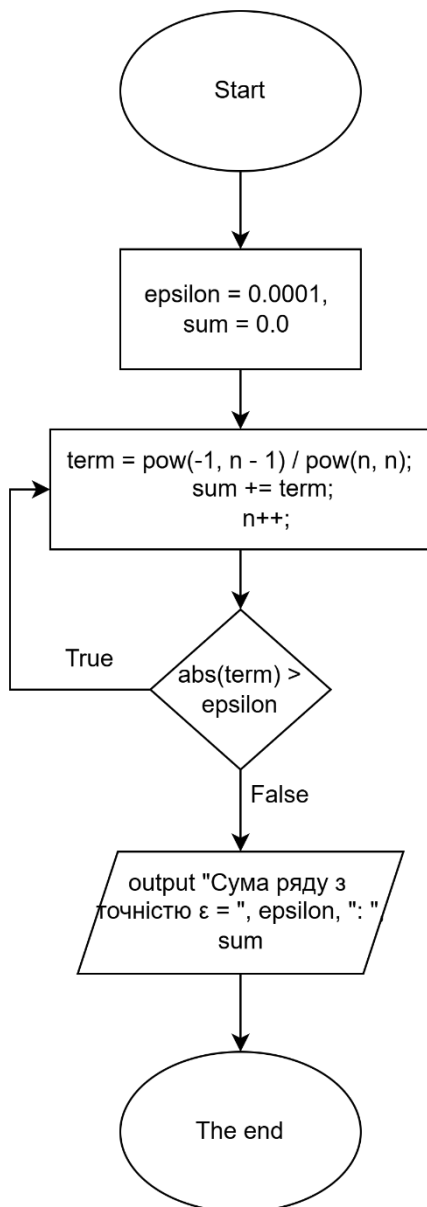
- Деталі завдання
- Одного разу до Ужгорода на літню школу з алгоритмічного програмування приїхали nn студентів, що сформували kk команд. Відомо, що кожна команда складається з одного, двох або трьох студентів.
- Вам необхідно визначити, скільки студентів було в кожній із команд.
- Вхідні дані: Єдиний рядок містить два цілих числа n та k — кількості студентів та команд.
- Вихідні дані: У єдиному рядку виведіть kk цілих чисел а<sub>і</sub> через пробіл. Тут а<sub>і</sub> — кількість студентів у і-тій команді.
- Якщо існує більше одного розв'язку — виведіть будь-який.
- Якщо розв'язку не існує — виведіть **Impossible**.

- Обмеження
- $1 \leq n \leq 103$   $1 \leq n \leq 103$ ,
- $1 \leq k \leq 100$   $1 \leq k \leq 100$ ,
- $1 \leq a_j \leq 3$ .

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

Програма № 1 VNS Lab 2 – Task 1

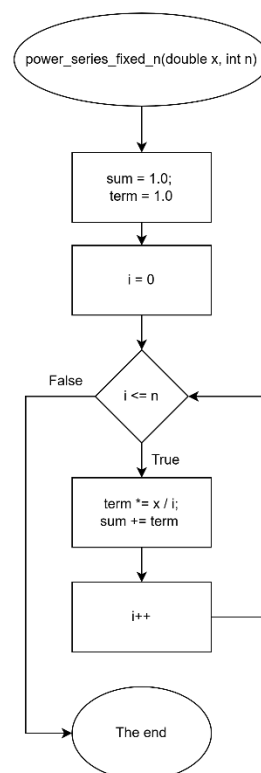
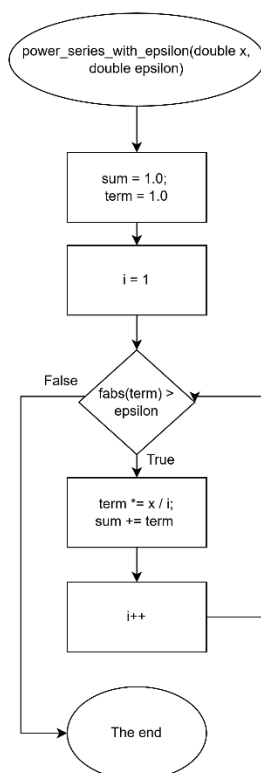
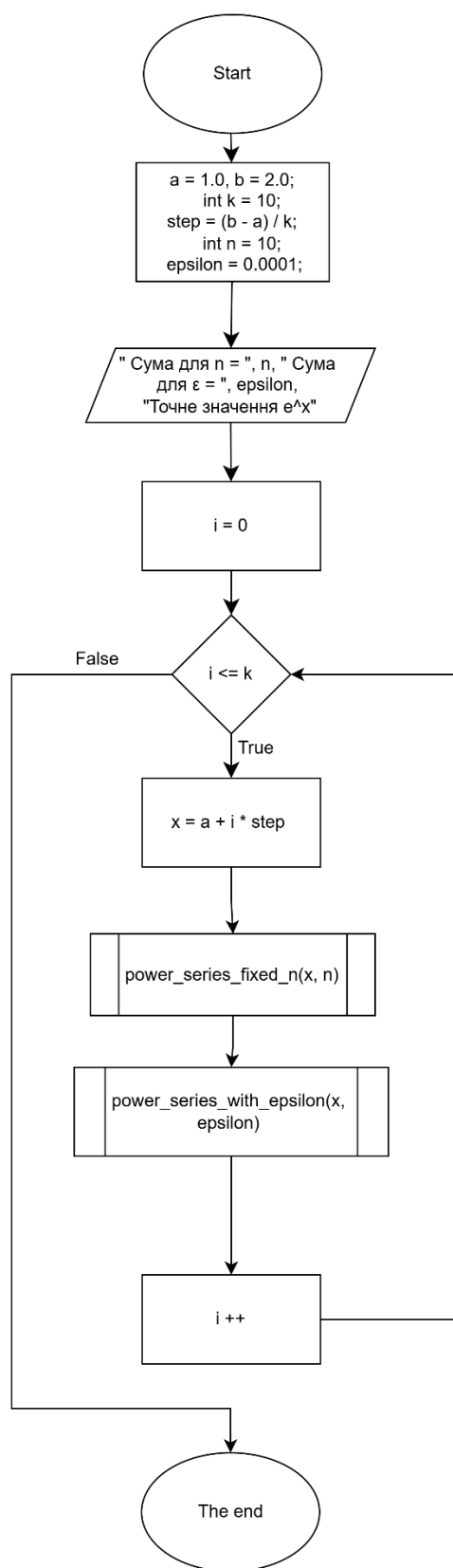
- Блок-схема



- Планований час на реалізацію 15 хв

Програма № 2 VNS Lab 3 – Task 1

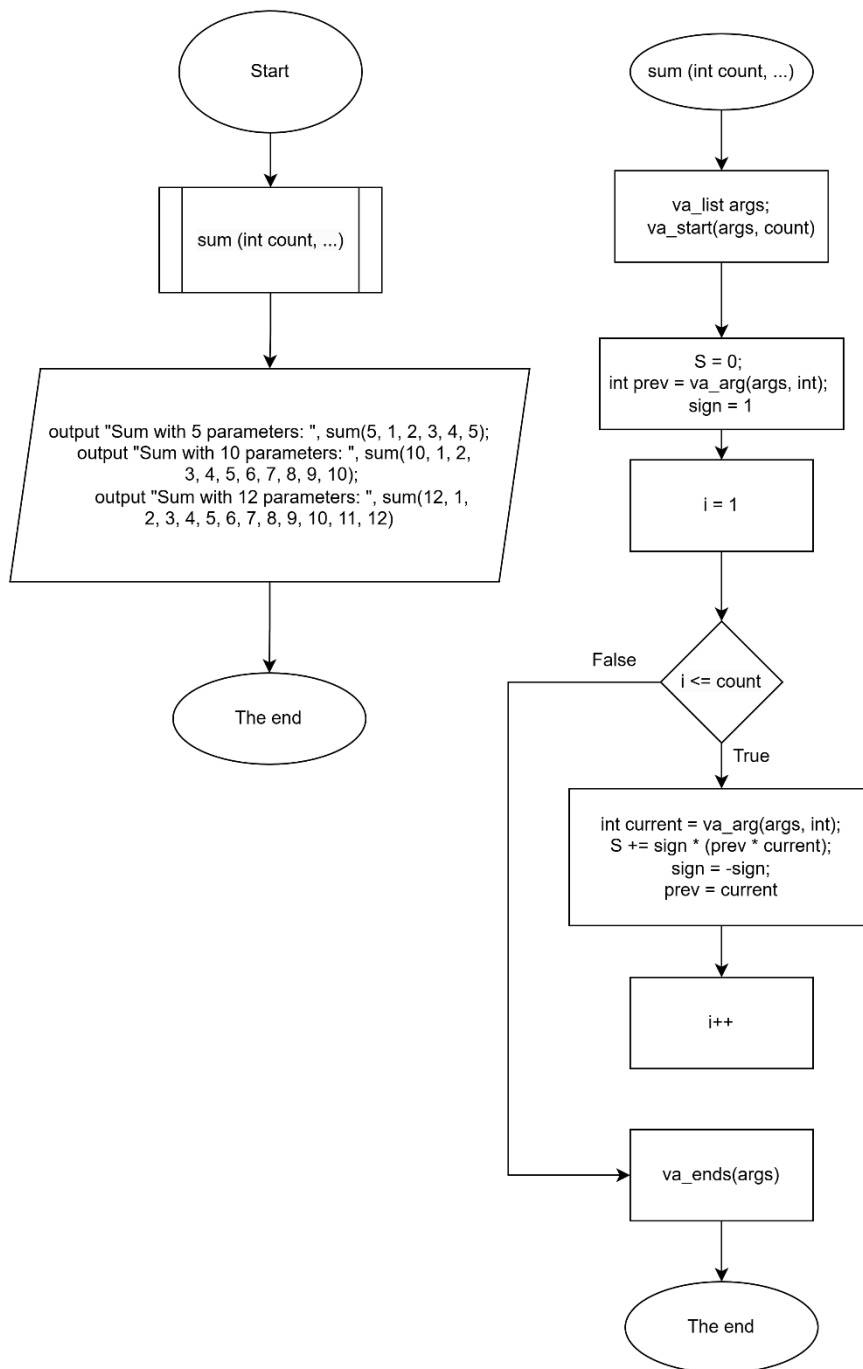
- Блок-схема



- Планований час на реалізацію: 25 хв

Програма № 3 VNS Lab 7 – Task 1

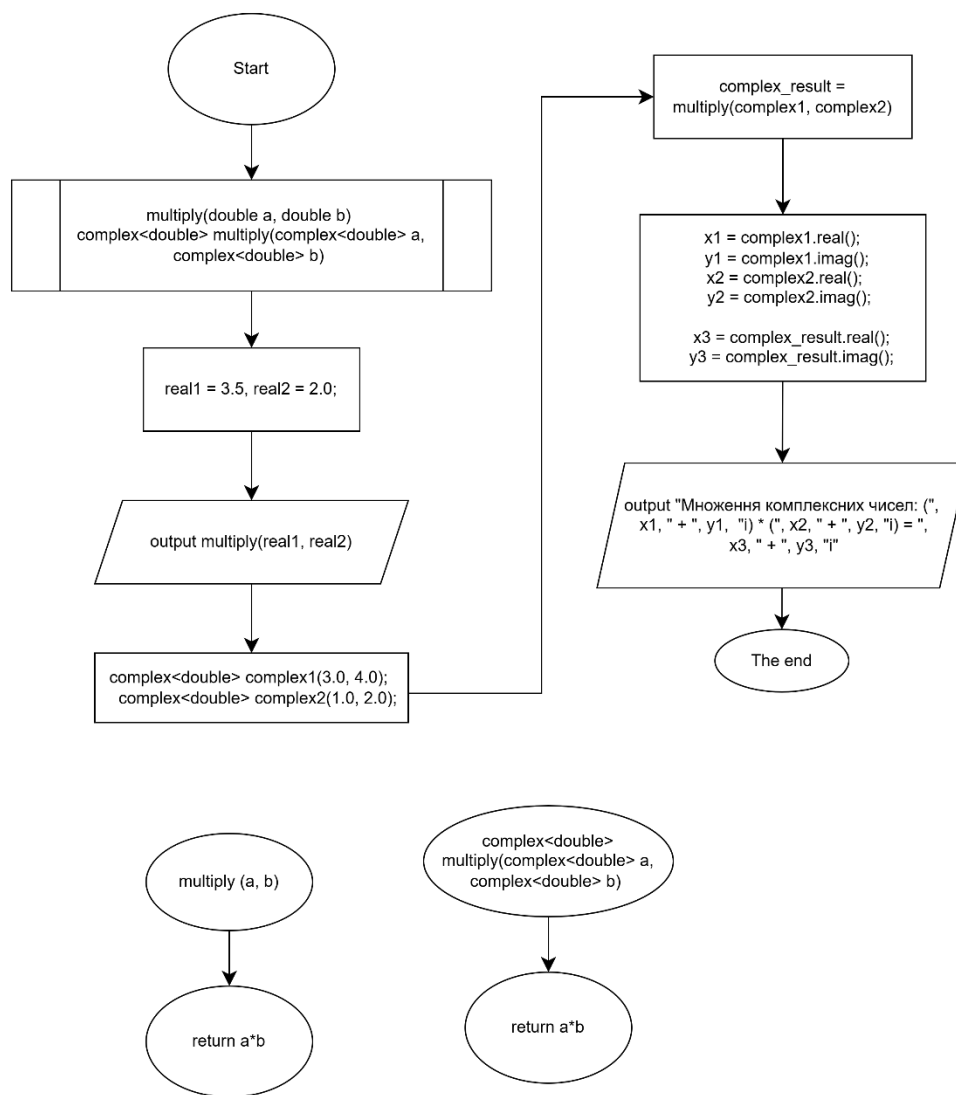
- Блок-схема



- Планований час на реалізацію: 30 хв

Програма № 4 VNS Lab 7 – Task 2

- Блок-схема

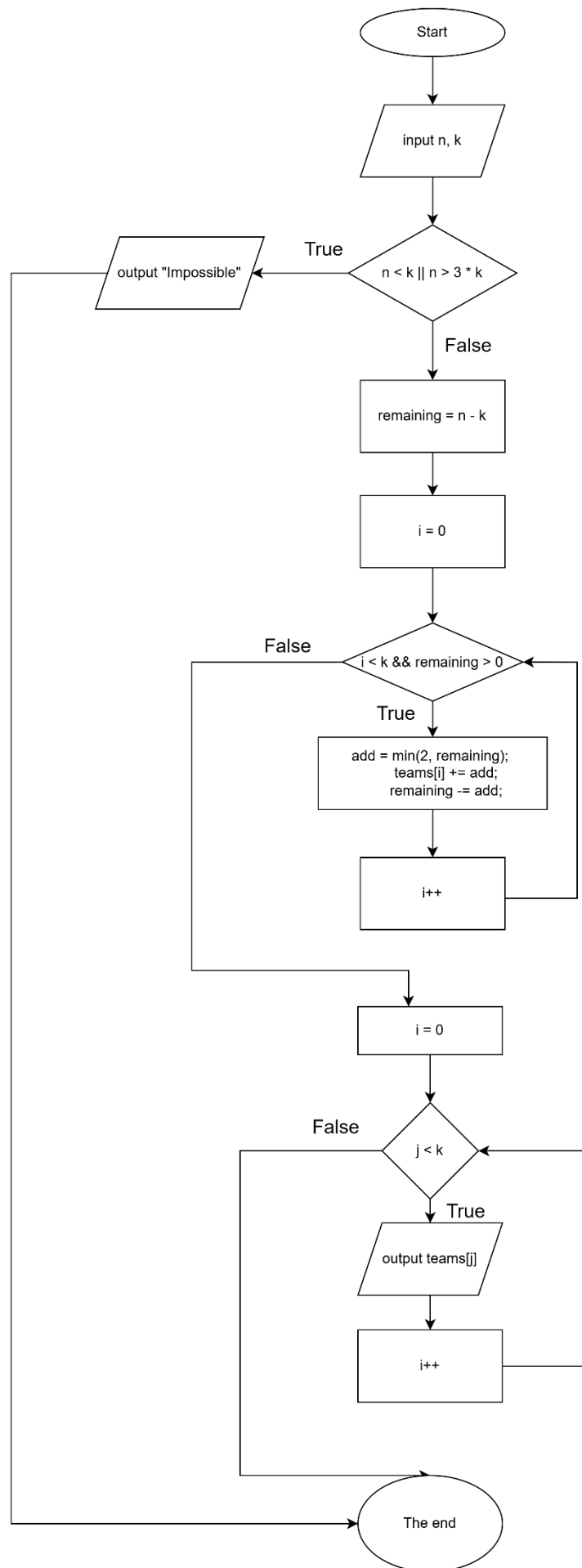


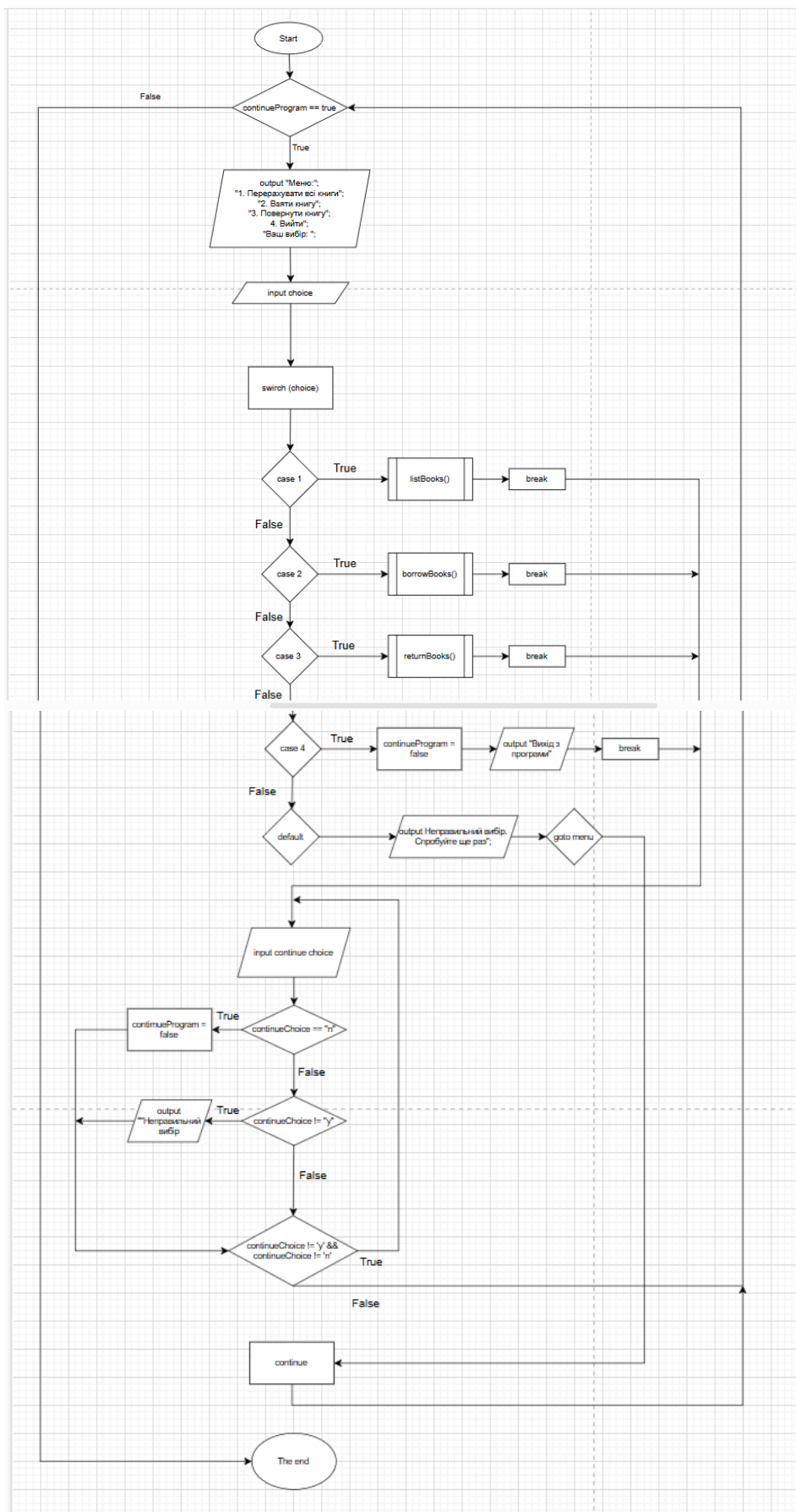
- Планований час на реалізацію

## Програма № 5 Class Practice work

- Блок-схема







- Планований час на реалізацію 1,5 год

## Програма № 6 Self Practice Work

- Блок-схема
- Планований час на реалізацію 30 хв

### 4. Код програм:

#### Завдання № 1 VNS Lab 2 – Task 1

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

using namespace std;

int main() {

    double epsilon = 0.0001;
    double sum = 0.0;
    double term;
    int n = 1;

    do {
        term = pow(-1, n - 1) / pow(n, n);
        sum += term;
        n++;
    } while (abs(term) > epsilon);

    cout << "Сума ряду з точністю  $\epsilon$  = " << epsilon << ": " << setprecision(4) <<
    fixed << sum << endl;

    return 0;
}
```

Результат:

```
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_3> &
'c:\Users\serhe\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-
x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-
dbwk010i.wu5' '--stdout=Microsoft-MIEngine-Out-50jthskc.24v' '--stderr=Microsoft-MIEngine-
Error-rsxzkdcp.wqi' '--pid=Microsoft-MIEngine-Pid-gwvpyo0w.qzc' '--
dbgExe=C:\msys64\mingw64\bin\gdb.exe'
'--interpreter=mi'
Сума ряду з точністю  $\epsilon$  = 0.0001: 0.7834
```

Час затрачений на виконання завдання: 30 хв

#### Завдання № 2 VNS Lab 3 – Task 1

```
#include <iostream>
#include <cmath> // Потрібно для функцій exp та fabs
```

```

#include <iomanip>

using namespace std;

// Функція для обчислення  $e^x$  за розкладом у ряд з фіксованим n
double power_series_fixed_n(double x, int n) {
    double sum = 1.0;
    double term = 1.0;
    for (int i = 1; i <= n; i++) {
        term *= x / i;
        sum += term;
    }
    return sum;
}

// Функція для обчислення  $e^x$  за розкладом у ряд з точністю epsilon
double power_series_with_epsilon(double x, double epsilon) {
    double sum = 1.0;
    double term = 1.0;
    int i = 1;
    while (fabs(term) > epsilon) {
        term *= x / i;
        sum += term;
        i++;
    }
    return sum;
}

int main() {
    double a = 1.0, b = 2.0;
    int k = 10;
    double step = (b - a) / k;
    int n = 10;
    double epsilon = 0.0001;

    cout << "|x| |Сума для n = " << n << "|" << setw(20) << " |Сума для  $\epsilon$  = " <<
epsilon << "|" << setw(20) << " |Точне значення  $e^x$ |\n";
    cout << "-----\n";

    for (int i = 0; i <= k; i++) {
        double x = a + i * step;

        double sum_fixed_n = power_series_fixed_n(x, n);

        double sum_with_epsilon = power_series_with_epsilon(x, epsilon);

        double exact_value = exp(x);

        cout << x << "\t" << sum_fixed_n << "\t\t" << sum_with_epsilon << "\t\t" <<
exact_value << endl;
    }
}

```

```

    }

    return 0;
}

```

Результат:

```

PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_3> &
'c:\Users\serhe\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-
x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-
unvzbfl.cwq' '--stdout=Microsoft-MIEngine-Out-5gsacph1.o34' '--stderr=Microsoft-MIEngine-
Error-i4cvz12v.goo' '--pid=Microsoft-MIEngine-Pid-as3ihrb.cgd' '--
dbgExe=C:\msys64\mingw64\bin\gdb.exe'
'--interpreter=mi'
|x| |Сума для n = 10| |Сума для ε = 0.0001| |Точне значення e^x|
-----
1      2.71828      2.71828      2.71828
1.1    3.00417      3.00416      3.00417
1.2    3.32012      3.32012      3.32012
1.3    3.6693       3.66929      3.6693
1.4    4.0552       4.05519      4.0552
1.5    4.48169      4.48169      4.48169
1.6    4.95303      4.95303      4.95303
1.7    5.47394      5.47394      5.47395
1.8    6.04963      6.04963      6.04965
1.9    6.68586      6.68589      6.68589
2      7.38899      7.38905      7.38906

```

Час затрачений на виконання завдання: 40 хв

Завдання № 3 VNS Lab 7 – Task 1

```

#include <iostream>
#include <cstdlib> // Для використання va_list та пов'язаних макросів

using namespace std;

int sum(int count, ...) {
    va_list args;
    va_start(args, count); // Ініціалізуємо список аргументів

    int S = 0; // Результируюча сума
    int prev = va_arg(args, int); // Отримуємо перший аргумент як початковий

    int sign = 1;

    for (int i = 1; i < count; i++) {
        int current = va_arg(args, int); // Отримуємо наступний аргумент
        S += sign * (prev * current); // Додаємо або віднімаємо згідно з чергуванням
        sign = -sign;
        prev = current; // Оновлюємо попереднє значення
    }
}

```

```

    va_end(args);
    return S;
}

int main() {
    cout << "Sum with 5 parameters: " << sum(5, 1, 2, 3, 4, 5) << endl;
    cout << "Sum with 10 parameters: " << sum(10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10) <<
endl;
    cout << "Sum with 12 parameters: " << sum(12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12) << endl;

    return 0;
}

```

Результат:

```

PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_3> &
'c:\Users\serhe\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-
x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-
wwglpkrn.d0n' '--stdout=Microsoft-MIEngine-Out-g2djidh0.tga' '--stderr=Microsoft-MIEngine-
Error-qsw5jlui.bag' '--pid=Microsoft-MIEngine-Pid-p4dngww0.5df' '--
dbgExe=C:\msys64\mingw64\bin\gdb.exe'
'--interpreter=mi'
Sum with 5 parameters: -12
Sum with 10 parameters: 50
Sum with 12 parameters: 72

```

Завдання № 4 VNS Lab 7 – Task 2

```

#include <iostream>
#include <complex>

using namespace std;

// Перевантажена функція для множення дійсних чисел
double multiply(double a, double b) {
    return a * b;
}

// Перевантажена функція для множення комплексних чисел
complex<double> multiply(complex<double> a, complex<double> b) {
    return a * b;
}

int main() {
    // Множення дійсних чисел
    double real1 = 3.5, real2 = 2.0;
    cout << "Множення дійсних чисел: " << real1 << " * " << real2 << " = " <<
multiply(real1, real2) << endl;

    // Множення комплексних чисел
    complex<double> complex1(3.0, 4.0);
}

```

```

    complex<double> complex2(1.0, 2.0);

    complex<double> complex_result = multiply(complex1, complex2);

    double x1 = complex1.real();
    double y1 = complex1.imag();
    double x2 = complex2.real();
    double y2 = complex2.imag();

    double x3 = complex_result.real();
    double y3 = complex_result.imag();

    cout << "Множення комплексних чисел: (" << x1 << " + " << y1 << "i) * ("
          << x2 << " + " << y2 << "i) = " << x3 << " + " << y3 << "i" << endl;

    return 0;
}

```

Результат:

```

PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_3> &
'c:\Users\serhe\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-
x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-
m2s1m10m.2wk' '--stdout=Microsoft-MIEngine-Out-nmtqp0re.jzt' '--stderr=Microsoft-MIEngine-
Error-ipqopgey.lnd' '--pid=Microsoft-MIEngine-Pid-ugt1mbna.dcx' '--
dbgExe=C:\msys64\mingw64\bin\gdb.exe'
'--interpreter=mi'
Множення дійсних чисел: 3.5 * 2 = 7
Множення комплексних чисел: (3 + 4i) * (1 + 2i) = -5 + 10i

```

Завдання № 5 Class Practice work

```

#include <iostream>
#include <vector>
#include <string>

using namespace std;

vector<string> books = {"Старий і море", "Гаррі Поттер", "Тіні забутих предків",
"1984", "Кобзар"};
vector<bool> available = {true, true, true, true, true};

// Функція для перерахунку всіх книг
void listBooks() {
    cout << "\nСписок всіх книг:\n";
    for (int i = 0; i < books.size(); i++) {
        cout << i + 1 << ". " << books[i] << " - " << (available[i] ? "Доступна" :
"Недоступна") << endl;
    }
}

```

```

// Функція для перевірки наявності книги та взяття її користувачем
void borrowBook() {
    int bookNumber;
    listBooks();
    cout << "\nВведіть номер книги, яку бажаєте взяти: ";
    cin >> bookNumber;

    // Перевірка коректності номера
    if (bookNumber < 1 || bookNumber > books.size()) {
        cout << "Неправильний номер книги!\n";
        goto menu;

        // Вибір книги
    } if (available[bookNumber - 1]) {
        available[bookNumber - 1] = false;
        cout << "Ви взяли книгу \"" << books[bookNumber - 1] << "\".\n";
    } else {
        cout << "Ця книга вже недоступна!\n";
    }
}

menu:
    return;
}

// Функція для повернення книги
void returnBook() {
    int bookNumber;
    listBooks();
    cout << "\nВведіть номер книги, яку бажаєте повернути: ";
    cin >> bookNumber;

    // Перевірка коректності номера
    if (bookNumber < 1 || bookNumber > books.size()) {
        cout << "Неправильний номер книги!\n";
        goto menu;
    }

    // Повернення книги
    if (!available[bookNumber - 1]) {
        available[bookNumber - 1] = true;
        cout << "Ви повернули книгу \"" << books[bookNumber - 1] << "\".\n";
    } else {
        cout << "Ця книга вже була доступна!\n";
    }
}

menu:
    return;
}

int main() {

```



```

int choice;
bool continueProgram = true;

while (continueProgram) { // Продовжувати працювати, поки користувач не вирішить
вийти
    cout << "\nМеню:\n";
    cout << "1. Перерахувати всі книги\n";
    cout << "2. Взяти книгу\n";
    cout << "3. Повернути книгу\n";
    cout << "4. Вийти\n";
    cout << "Ваш вибір: ";
    cin >> choice;

    switch (choice) {
        case 1:
            listBooks();
            break;
        case 2:
            borrowBook();
            break;
        case 3:
            returnBook();
            break;
        case 4:
            continueProgram = false;
            cout << "Вихід з програми.\n";
            break;
        default:
            cout << "Неправильний вибір. Спробуйте ще раз.\n";
            goto menu;
    }

    // Запит на продовження після кожної операції
    char continueChoice;
    do {
        cout << "\nБажаєте виконати іншу операцію? (y/n): ";
        cin >> continueChoice;
        if (continueChoice == 'n' || continueChoice == 'N') {
            continueProgram = false;
        } else if (continueChoice != 'y' && continueChoice != 'Y') {
            cout << "Неправильний вибір. ";
        }
    } while (continueChoice != 'y' && continueChoice != 'Y' && continueChoice !=
'n' && continueChoice != 'N');

menu:
    continue;
}

return 0;

```

```
}
```

Результат:

Меню:

1. Перерахувати всі книги
2. Взяти книгу
3. Повернути книгу
4. Вийти

Ваш вибір: 2

Список всіх книг:

1. Старий і море - Доступна
2. Гаррі Поттер - Доступна
3. Тіні забутих предків - Доступна
4. 1984 - Доступна
5. Кобзар - Доступна

Введіть номер книги, яку бажаєте взяти: 5

Ви взяли книгу "Кобзар".

Бажаєте виконати іншу операцію? (y/n): n

Час затрачений на виконання завдання: 2 год

#### Завдання № 6 Self Practice Work

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

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

    // Перевірка можливості розподілу студентів
    if (n < k || n > 3 * k) {
        cout << "Impossible" << endl;
        return 0;
    }

    vector<int> teams(k, 1);
    int remaining = n - k;

    for (int i = 0; i < k && remaining > 0; i++) {
        int add = min(2, remaining);
        teams[i] += add;
        remaining -= add;
    }

    for (int j = 0; j < k; j++) {
```

```

        cout << teams[j] << " ";
    }
    cout << endl;

    return 0;
}

```

Результат:

```

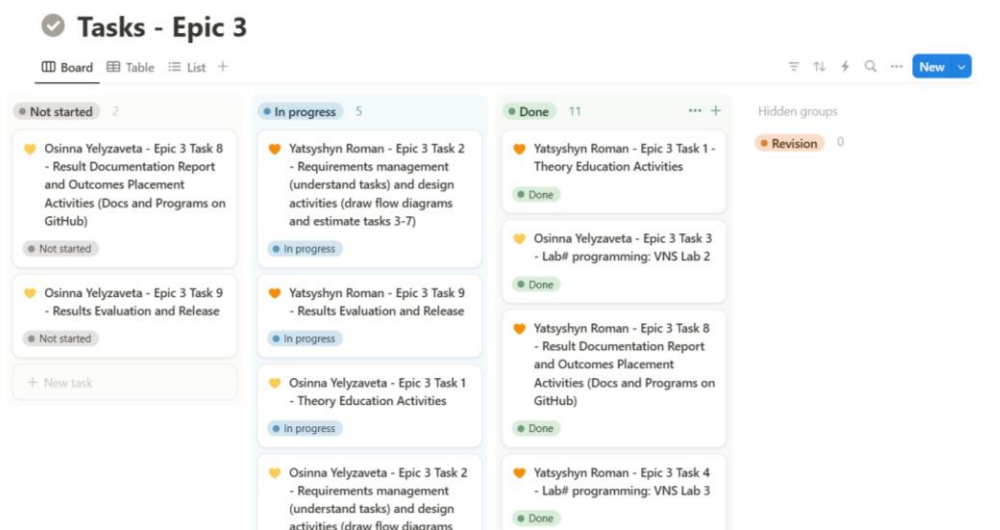
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\epic_3> &
'c:\Users\serhe\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-
x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-
b4vtjnjt.0j5' '--stdout=Microsoft-MIEngine-Out-ga4ljsh2.43i' '--stderr=Microsoft-MIEngine-
Error-iop5rq3i.y1n' '--pid=Microsoft-MIEngine-Pid-ownjvixg.swj' '--
dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
9 5
3 3 1 1 1

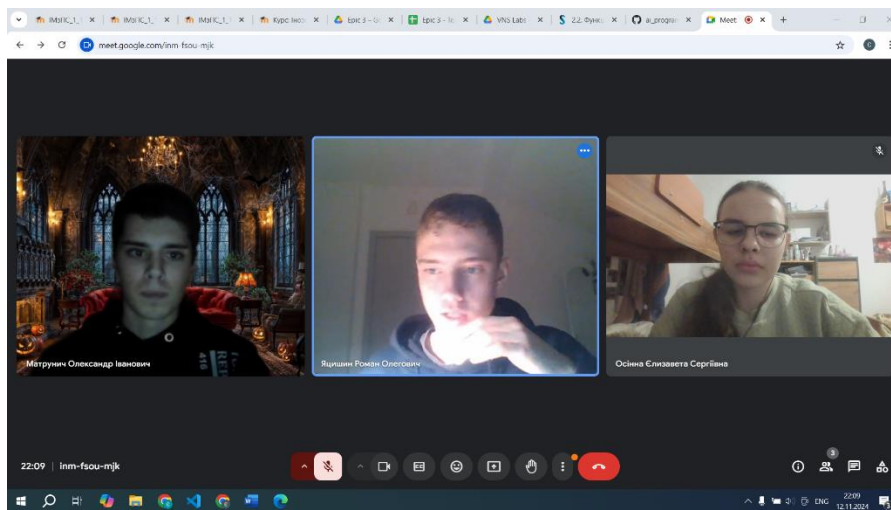
```

Час затрачений на виконання завдання: 40 хв

## 6. Кооперація з командою:

- Notion





## Висновки:

Виконавши цю роботу, я отримала навички у виборі й використанні операторів C; ознайомила з ітераційними процесами, попрактикувалася в організації ітераційних й арифметичних циклів та ознайомила з організацією функцій зі змінною кількістю параметрів та перевантаженням функцій.