Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 3 (лабораторні № 2, 3, 7)**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія.»

***Виконав:***

студент групи **ШІ-14**

**Нападайло Микита Олександрович**

# **Тема роботи:**

Знайомство з циклами while, do while, for, for each, вкладеними циклами, функціями зі змінною кількістю параметрів, рекурсивними функціями, перевантаженими функціями.

# **Мета роботи:**

# Навчитися користуватися різними видами циклів, писати перевантажені, рекурсивні функції, а також функції зі змінною кількістю параметрів для обчислення виразів та суми рядів.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли.
* Тема №2: Вкладені цикли.
* Тема №3: Функції.
* Тема №4: Перевантаження функцій.
* Тема №5: Рекурсія.
* Тема №6: Функції зі змінною кількістю параметрів.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли.
  + Джерела Інформації
    - Курс CS50 2023, відео <https://youtu.be/ywg7cW0Txs4>.
  + Що опрацьовано:
    - Які існують цикли, відмінність між циклами while, do while, for, for each
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.2023
  + Завершення опрацювання теми: 07.2023
* Тема №2: Вкладені цикли.
  + - Курс CS50 2023, відео <https://youtu.be/ywg7cW0Txs4>
  + Що опрацьовано:
    - Що таке вкладений цикл
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.2023
  + Завершення опрацювання теми: 07.2023
* Тема №3: Функції.
  + Джерела Інформації:
    - Курс CS50 2023 <https://cs50.harvard.edu/x/2023/> .
  + Що опрацьовано:
    - Що таке функція, як вона задається
  + Статус: Ознайомлений
  + Початок опрацювання теми: 30.10.2023
  + Завершення опрацювання теми: 30.10.2023
* Тема №4: Перевантаження функцій.
  + Джерела Інформації:
    - Книжка.
    - Відео <https://youtu.be/8jLOx1hD3_o?t=59556> .
  + Що опрацьовано:
    - Що таке перевантаження функції, як воно реалізовується і з якою метою
  + Статус: Ознайомлений
  + Початок опрацювання теми: 30.10.2023
  + Завершення опрацювання теми: 30.10.2023
* Тема №5: Рекурсія.
  + Джерела Інформації:
    - Курс CS50 2023, відео <https://youtu.be/4oqjcKenCH8> .
  + Що опрацьовано:
    - Що таке рекурсія у програмуванні, які бувають рекурсії
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.2023
  + Завершення опрацювання теми: 07.2023
* Тема №5: Функції зі змінною кількістю параметрів.
  + Джерела Інформації:
    - Відео <https://www.youtube.com/watch?v=oDC208zvsdg> .
  + Що опрацьовано:
    - Як працюють функції зі змінною кількістю параметрів
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Завершення опрацювання теми: 25.11.2023

# **Виконання роботи:**

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

Завдання №1 **VNS Lab 2 - Task 1-N**

* Варіант завдання – 18
* Деталі завдання: Знайти суму ряду з точністю ε=0.0001.
* Важливі деталі для врахування в імплементації програми: Вважати, що точність досягнута при .

Завдання №2 **VNS Lab 3 - Task 1-N**

* Варіант завдання - 18
* Деталі завдання: Обчислити функцію , використовуючи її розклад в степеневий ряд для заданого n та для заданої точності ε=0.0001 .
* Важливі деталі для врахування в імплементації програми: Для порівняння знайти точне значення функції.

Завдання №3 **VNS Lab 7 - Task 1-N**

* Варіант завдання – 18
* Деталі завдання: Написати функцію NSK зі змінною кількістю параметрів, що знаходить найменше спільне кратне для декількох чисел.
* Важливі деталі для врахування в імплементації програми: Функція main має звертатися до NSK не менше трьох разів з кількістю параметрів 3, 5, 6.

Завдання №4 **VNS Lab 7 - Task 2-N**

* Варіант завдання – 18
* Деталі завдання: Написати перевантажені функції, які для масиву цілих чисел знаходять максимальний елемент, а для рядка знаходять довжину найдовшого слова.

Завдання №5 **Class Practice Work**

* Деталі завдання: Створити просту програму керування бібліотекою. Користувач має мати можливість взяти книгу або повернути її.
* Важливі деталі для врахування в імплементації програми: В програмі необхідно використати цикли while, do while, for, for each та оператор переходу goto.

Завдання №6 **Self-Practice Work**

* Варіант завдання 0021
* Деталі завдання: Визначити яка мінімальна кількість купюр знадобиться депутату, щоб заплатити за покупку на суму n.

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

Програма №1 **VNS Lab 2 - Task 1-N**

* Блок-схема

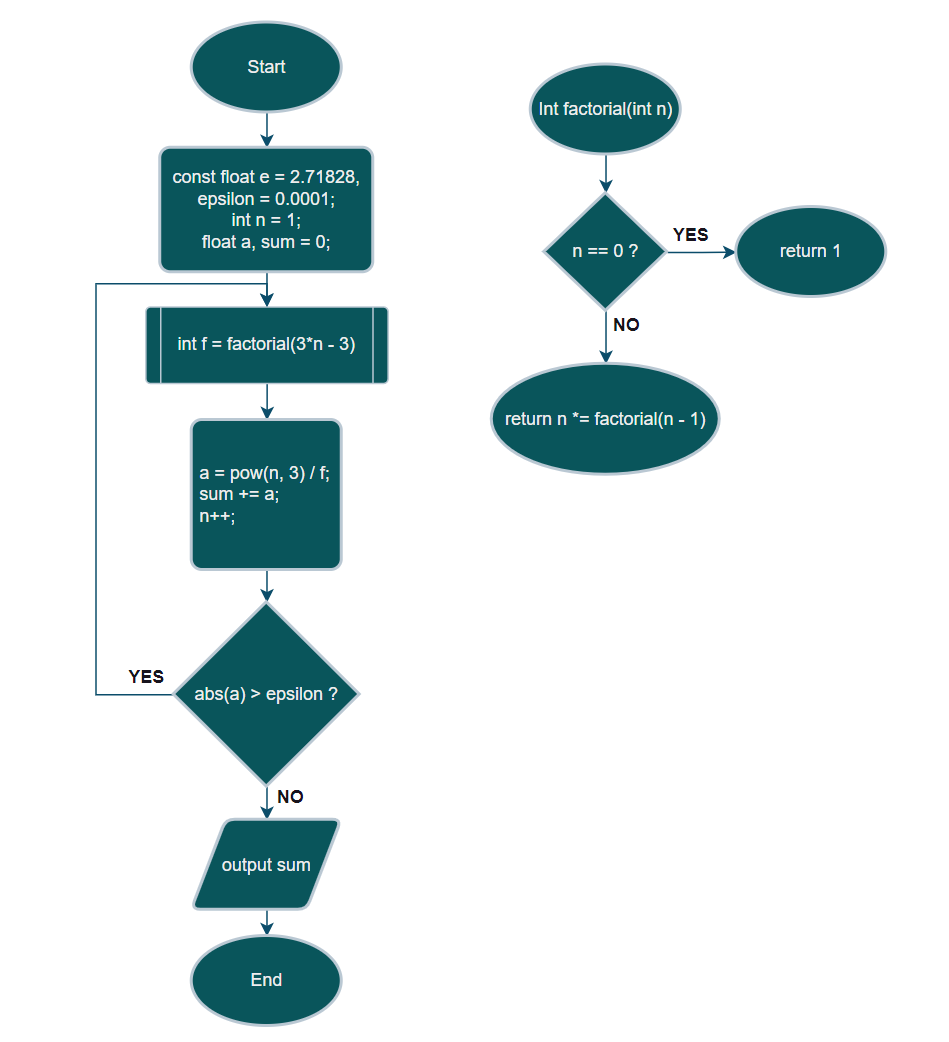


Рисунок : блок-схема до програми VNS Lab 2 - Task 1-N

* Планований час на реалізацію – 30 хвилин
* Важливі деталі для врахування в імплементації: Вважати, що точність досягнута при .

Програма №2 **VNS Lab 3 - Task 1-N**

* Блок-схема

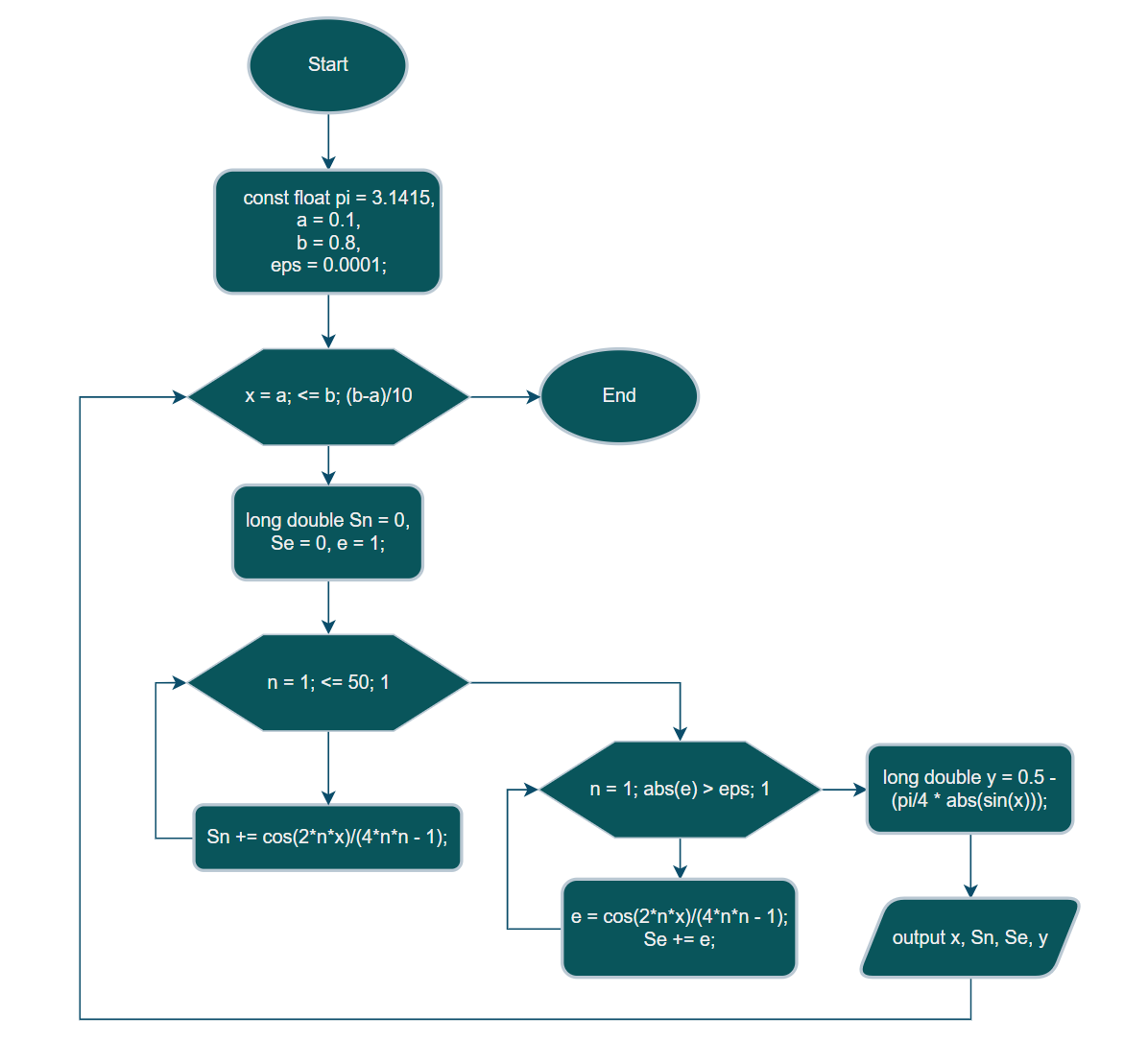
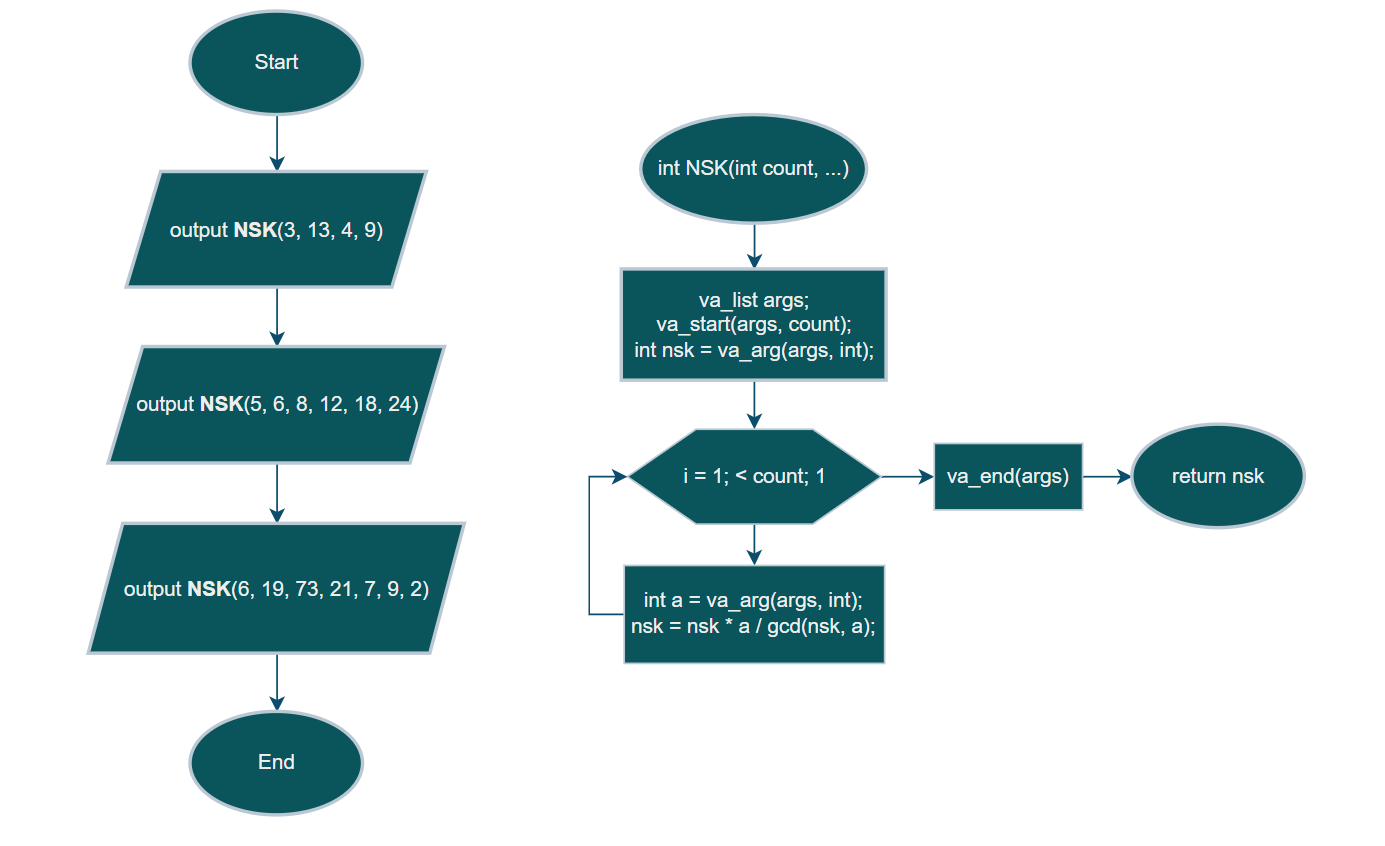


Рисунок 2: блок-схема до програми VNS Lab 3 - Task 1-N

* Планований час на реалізацію – 1 година

Програма №3 **VNS Lab 7 - Task 1-N**

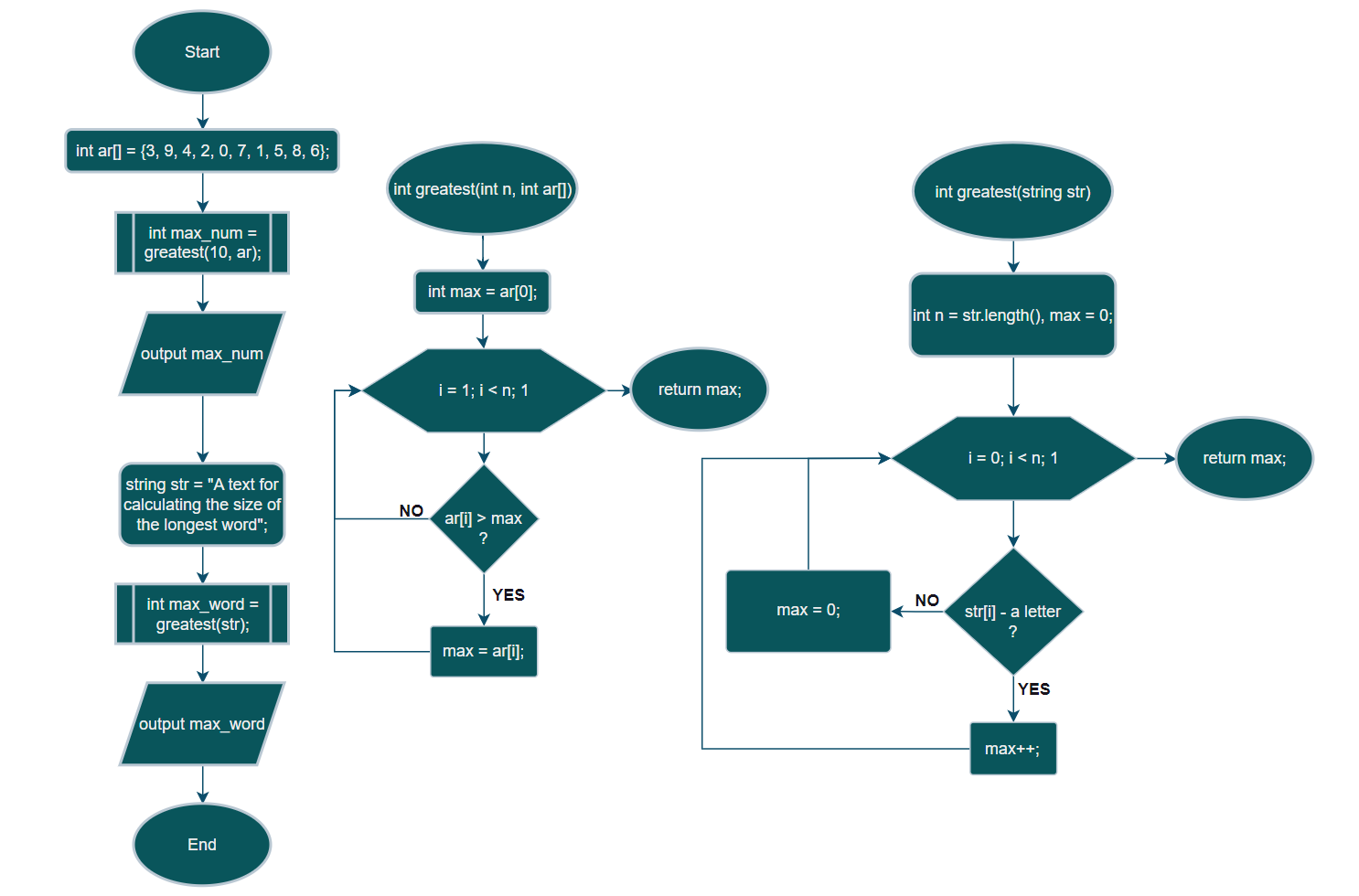
* Блок-схема



* Рисунок 3: блок-схема до програми VNS Lab 7 - Task 1-N
* Планований час на реалізацію – 1 година
* Важливі деталі для врахування в імплементації програми: Функція main має звертатися до NSK не менше трьох разів з кількістю параметрів 3, 5, 6.

Програма №4 **VNS Lab 7 - Task 2-N**

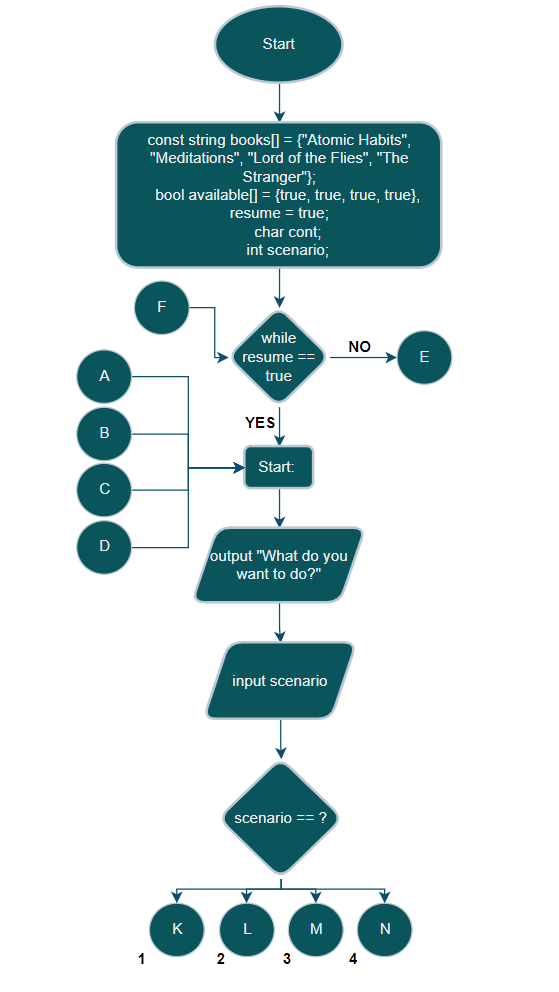
* Блок-схема



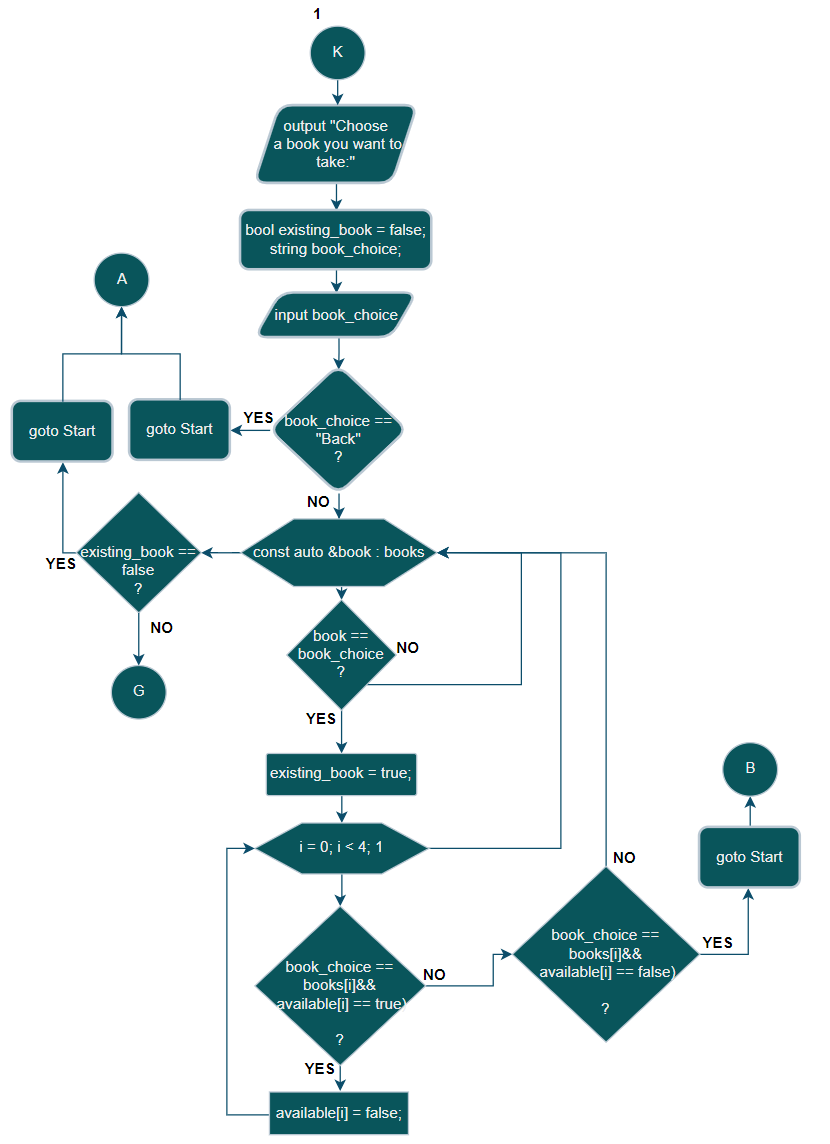
* Рисунок 4: блок-схема до програми VNS Lab 7 - Task 2-N
* Планований час на реалізацію – 30 хвилин

Програма №5 **Class Practice Work**

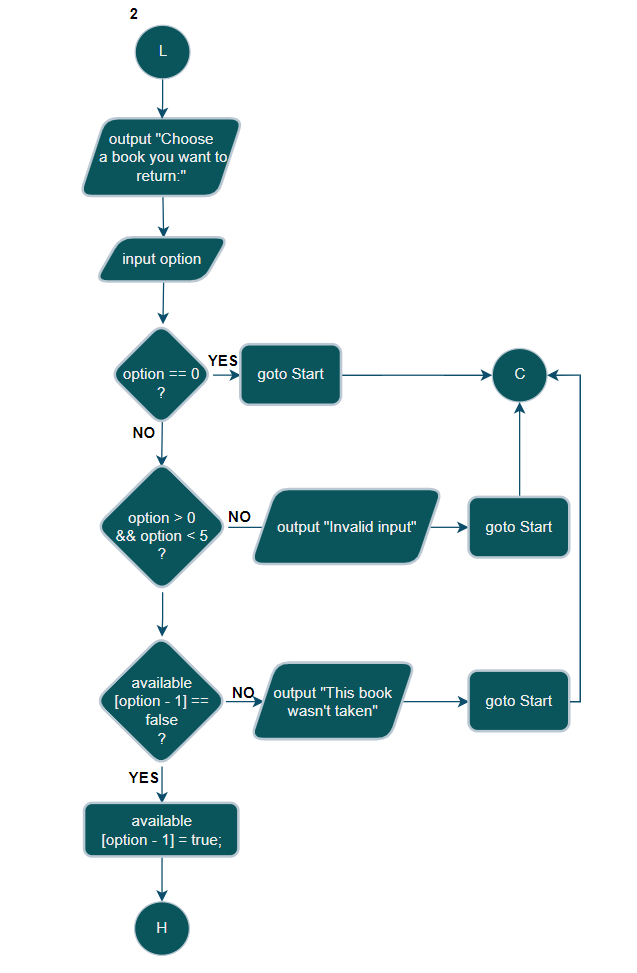
* Блок-схема



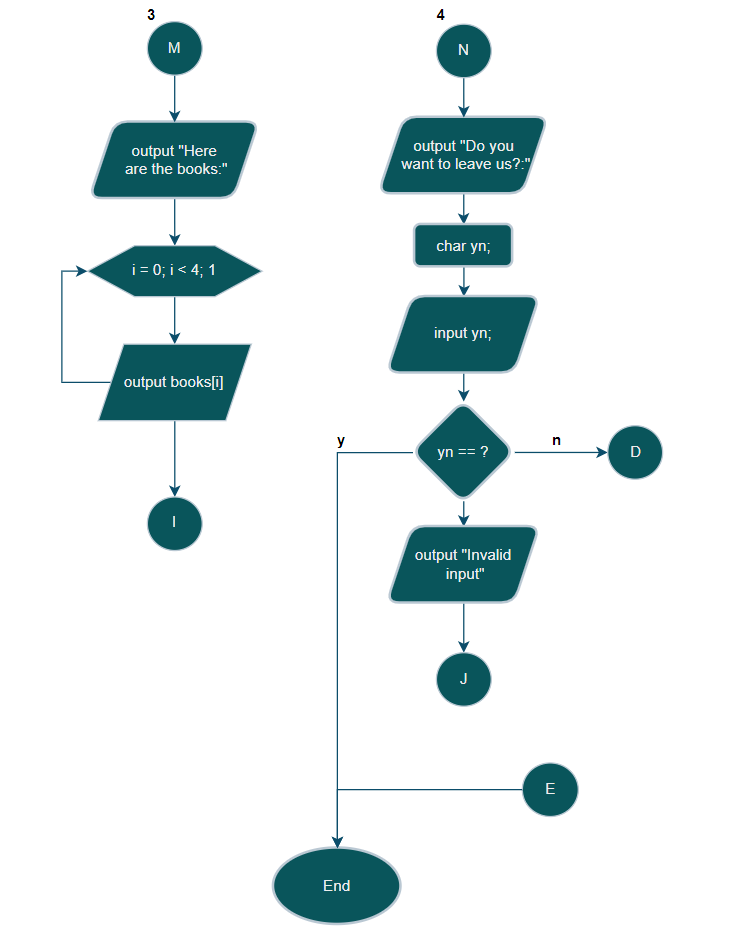
* Рисунок 5: блок-схема до програми Class Practice Work (1)



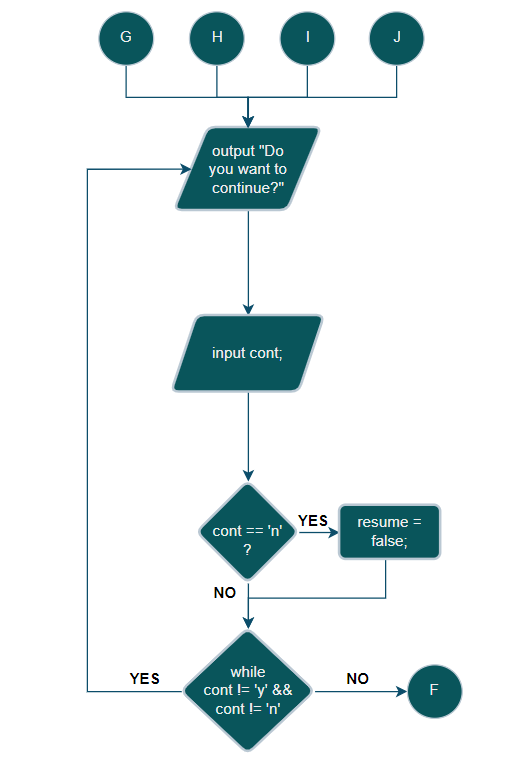
* Рисунок 6: блок-схема до програми Class Practice Work (2)



* Рисунок 7: блок-схема до програми Class Practice Work (3)



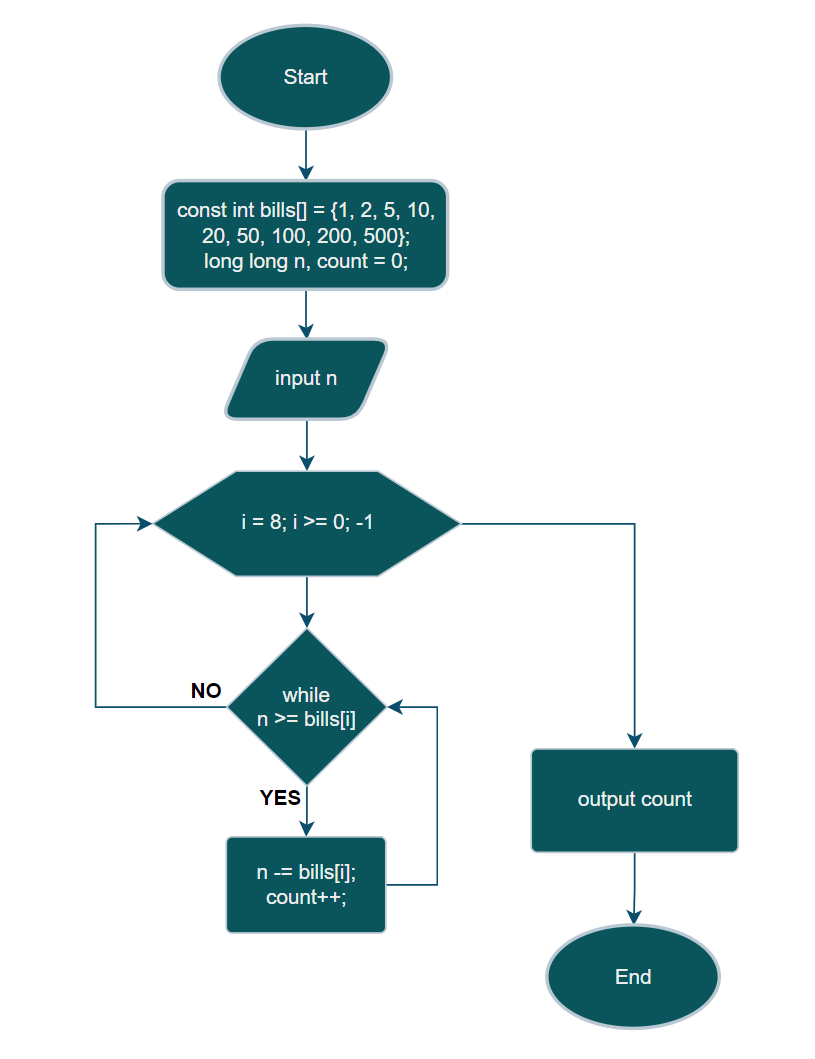
* Рисунок 8: блок-схема до програми Class Practice Work (4)



* Рисунок 9: блок-схема до програми Class Practice Work (5)
* Планований час на реалізацію 1 година 20 хвилин
* Важливі деталі для врахування в імплементації програми: В програмі необхідно використати цикли while, do while, for, for each та оператор переходу goto.

Програма №6 **Self-Practice Work**

* Блок-схема



* Рисунок 10: блок-схема до програми Self-Practice Work (3)
* Планований час на реалізацію: 5 хвилин

## **3. Конфігурація середовища до виконання завдань:**

Використав конфігурацію з Епіку 1.

## **4. Код програм з посиланням на зовнішні ресурси:**

Завдання №1 **VNS Lab 2 - Task 1-N**

#include <iostream>

#include <cmath>

using namespace std;

int factorial(int n);

int main ()

{

    const float e = 2.71828, epsilon = 0.0001;

    int n = 1;

    float a, sum = 0;

    do

    {

        int f = factorial(3\*n - 3);

        a = pow(n, 3) / f;

        sum += a;

        n++;

    }

    while (abs(a) > epsilon);

    cout << "The sum of series is equal to " << sum;

    return 0;

}

int factorial(int n)

{

    if (n == 0)

    {

        return 1;

    }

    else return n \*= factorial(n - 1);

}

Завдання №2 **VNS Lab 3 - Task 1-N**

#include <iostream>

#include <cmath>

using namespace std;

int main ()

{

    const float pi = 3.1415, a = 0.1, b = 0.8, eps = 0.0001;

    for (float x = a; x <= b; x+=(b-a)/10)

    {

        long double Sn = 0, Se = 0, e = 1;

        for (int n = 1; n <= 50; n++)

        {

            Sn += cos(2\*n\*x)/(4\*n\*n - 1);

        }

        for (int n = 1; abs(e) > eps; n++)

        {

            e = cos(2\*n\*x)/(4\*n\*n - 1);

            Se += e;

        }

        long double y = 0.5 - (pi/4 \* abs(sin(x)));

        cout << "X=" << x << "    Sn=" << Sn << "   Se=" << Se << "   Y=" << y << endl;

    }

return 0;

}

Завдання №3 **Lab 7 - Task 1-N**

#include <iostream>

#include <numeric>

#include <cstdarg>

using namespace std;

int NSK(int count, ...);

int main()

{

    cout << "The least common multiple for 13, 4, 9 is: " << NSK(3, 13, 4, 9) << endl;

    cout << "The least common multiple for 6, 8, 12, 18, 24 is: " << NSK(5, 6, 8, 12, 18, 24) << endl;

    cout << "The least common multiple for 19, 73, 21, 7, 9, 2 is: " << NSK(6, 19, 73, 21, 7, 9, 2) << endl;

    return 0;

}

int NSK(int count, ...)

{

    va\_list args;

    va\_start(args, count);

    int nsk = va\_arg(args, int);

    for (int i = 1; i < count; i++)

    {

        int a = va\_arg(args, int);

        nsk = nsk \* a / gcd(nsk, a);

    }

    va\_end(args);

    return nsk;

}

Завдання №4 **VNS Lab 7 - Task 2-N**

#include <iostream>

using namespace std;

int greatest(int n, int ar[]);

int greatest(string str);

int main()

{

    int ar[] = {3, 9, 4, 2, 0, 7, 1, 5, 8, 6};

    int max\_num = greatest(10, ar);

    cout << "The greatest number is: " << max\_num << endl;

    string str = "A text for calculating the size of the longest word";

    int max\_word = greatest(str);

    cout << "The longest word`s length is: "<< max\_word;

    return 0;

}

int greatest(int n, int ar[])

{

    int max = ar[0];

    for (int i = 1; i < n; i++)

    {

        if (ar[i] > max) max = ar[i];

    }

    return max;

}

int greatest(string str)

{

    int n = str.length(), tmp = 0, max = 0;

    for (int i = 0; i < n; i++)

    {

        if ((str[i] >= 'A' && str[i <= 'Z']) || (str[i] >= 'a' && str[i <= 'z'])) tmp++;

        else tmp = 0;

        if (tmp > max) max = tmp;

    }

    return max;

}

Завдання №5 **Class Practice Work**

#include <iostream>

using namespace std;

int main()

{

    const string books[] = {"Atomic Habits", "Meditations", "Lord of the Flies", "The Stranger"};

    bool available[] = {true, true, true, true}, resume = true;

    char cont;

    int scenario;

    while(resume == true)

    {

        Start:

            cout << "What do you want to do?\n1 - Take a book\n2 - Return a book\n3 - View all books\n4 - Exit\n";

            cin >> scenario;

            switch (scenario)

            {

                case 1:

                {

                    cout << "Choose a book you want to take:\n -Atomic Habits\n -Meditations\n -Lord of the Flies\n -The Stranger\nBack - Go to the main menu\n";

                    bool existing\_book = false;

                    string book\_choice;

                    cin.ignore();

                    getline(cin, book\_choice);

                    cout << "================================\n";

                    if (book\_choice == "Back") goto Start;

                    for (const auto book : books)

                    {

                        if (book == book\_choice)

                        {

                            existing\_book = true;

                            for (int i = 0; i < 4; i++)

                            {

                                if (book\_choice == books[i] && available[i] == true)

                                {

                                    cout << "You took the book " << books[i] << "\nDon`t forget to return it!\n";

                                    cout << "================================\n";

                                    available[i] = false;

                                    break;

                                }

                                else if (book\_choice == books[i] && available[i] == false)

                                {

                                    cout <<  "This book is already taken:(\n";

                                    cout << "================================\n";

                                    goto Start;

                                }

                            }

                        }

                    }

                    if (existing\_book == false)

                    {

                        cout << "This book doesn`t exist\n";

                        cout << "================================\n";

                        goto Start;

                    }

                }

                break;

                case 2:

                {

                    cout << "Choose a book you want to return:\n 1 - Atomic Habits\n 2 - Meditations\n 3 - Lord of the Flies\n 4 - The Stranger\n 0 - Go to the main menu\n";

                    cout << "================================\n";

                    int option;

                    cin >> option;

                    if (option == 0) goto Start;

                    else if (option > 0 && option < 5)

                    {

                        if (available[option - 1] == false)

                        {

                            available[option - 1] = true;

                            cout << "Thanks for returning the book!\n";

                            cout << "================================\n";

                        }

                        else

                        {

                            cout << "This book wasn't taken\n";

                            cout << "================================\n";

                            goto Start;

                        }

                    }

                    else

                    {

                        cout << "Invalid input\n";

                        cout << "================================\n";

                        goto Start;

                    }

                }

                break;

                case 3:

                {

                    cout << "Here are the books:\n ";

                    for (int i = 0; i < 4; i++) cout << books[i] << endl;

                    cout << "================================\n";

                }

                break;

                case 4:

                {

                    cout << "Are you sure you want to leave us?\n y - yes, n - no\n";

                    char yn;

                    cin >> yn;

                    if (yn == 'y') return 0;

                    else if (yn == 'n') goto Start;

                    else cout << "Invalid input\n";

                }

                break;

            }

        do

        {

            cout << "Do you want to continue?\ny - yes, n - no\n";

            cin >> cont;

            if (cont == 'n') resume = false;

        } while (cont != 'y' && cont != 'n');

    }

    return 0;

}

Завдання №6 **Self-Practice Work**

#include <iostream>

using namespace std;

int main()

{

    const int bills[] = {1, 2, 5, 10, 20, 50, 100, 200, 500};

    long long n, count = 0;

    cin >> n;

    for (int i = 8; i >= 0; i--)

    {

        while (n >= bills[i])

        {

            n -= bills[i];

            count++;

        }

    }

    cout << count;

    return 0;

}

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

Завдання №1 **VNS Lab 2 - Task 1-N**

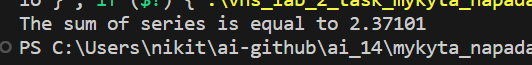


Рисунок 11: результат виконання програми №1

Час затрачений на виконання завдання: 30 хвилин

Завдання №2 **VNS Lab 3 - Task 1-N**

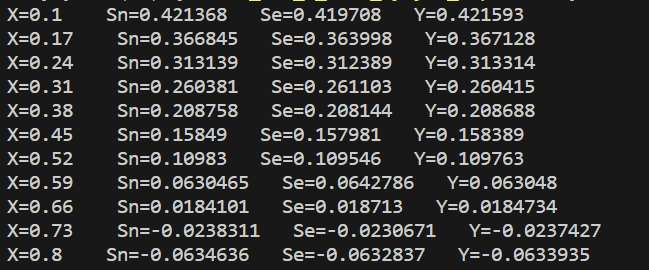


Рисунок 12: результат виконання програми №2

Час затрачений на виконання завдання: 40 хвилин

Завдання №3 **VNS Lab 7 - Task 1-N**

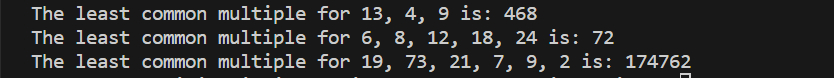


Рисунок 13: результат виконання програми №3

Час затрачений на виконання завдання: 1 година 20 хвилин

Завдання №4 **VNS Lab 7 - Task 2-N**



Рисунок 14: результат виконання програми №4

Час затрачений на виконання завдання: 30 хвилин

Завдання №5 **Class Practice Work**

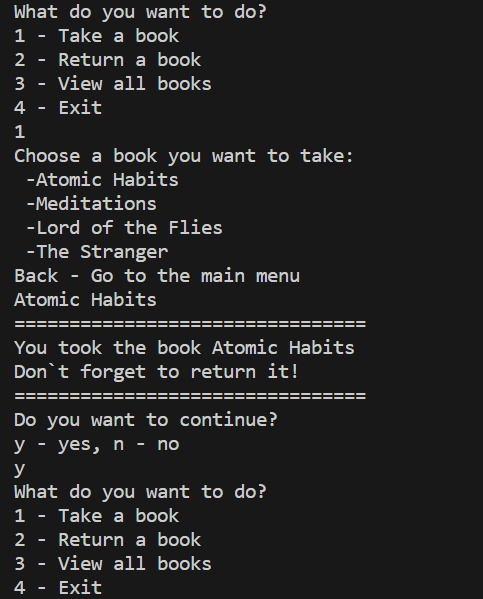


Рисунок 15: результат виконання програми №5(1)

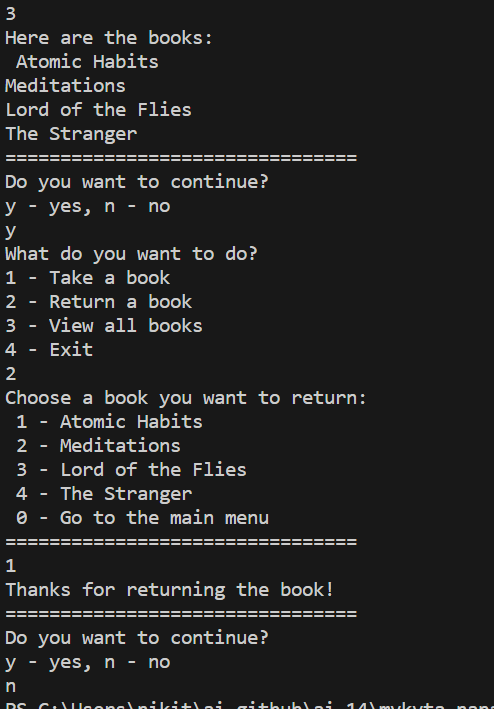


Рисунок 16: результат виконання програми №5(2)

Час затрачений на виконання завдання: 2 години 30 хвилин

Завдання №6 **Self-Practice Work**

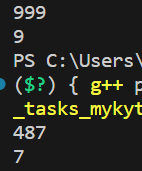


Рисунок 17: результат виконання програми №6

Час затрачений на виконання завдання: 5 хвилин

# **Висновки:**

В ході виконання лабораторних та практичних робіт в цьому епіку, я попрактикувався в використанні різних циклів, в тому числі розібрався як працює цикл for each; спробував реалізувати рекурсивну функцію, перевантажену функцію та функцію зі змінною кількістю параметрів для обчислення різних виразів та виконання інших задач.