Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт 3**

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

***з розділу***: «Епік 3: Цикли. Вкладені цикли. Функції. Перевантаження функцій. Рекурсія»

***Виконав:***

студент групи ШІ-13

Юнко Дмитро Богданович

# **Тема роботи:**

# Робота з циклами та вкладеними циклами в мові програмування C++. Функції в мові C++. Перевантажені та рекурсивні функції, функції багатьох змінних

# **Мета роботи:**

# Мета полягає в освоєнні ітераційних та арифметичних циклів, використанні функцій, перевантажених функцій та рекурсії для створення програм. Особливий акцент зроблено на обчислення функцій за допомогою їх розкладу в степеневий ряд (ряд Маклорена).

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

- Тема №1: Цикли.

- Тема №2: Оператори переходу.

- 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
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
    - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Визначення та властивості лінійних та розгалужених алгоритмів
    - Визначення та властивості ітераційних алгоритмів
    - Приклади ітераційних алгоритмів
    - Практичне застосування ітераційних алгоритмів у програмуванні
    - Використання циклів для ефективного виконання повторюваних завдань
    - Розуміння різниці між різними типами циклів (for, while, do-while) та їх використання
  + Статус: Ознайомлений
  + Початок опрацювання теми: 16.11.2023
  + Звершення опрацювання теми: 16.11.2023
* Тема №2: Оператори переходу.
  + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
    - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Як оператори переходу використовуються для контролю потоку виконання програми.
    - Різні типи операторів переходу та їх використання в різних контекстах.
    - Приклади використання умовних операторів у програмуванні
    - Практичне застосування умовних операторів у програмуванні
  + Статус: Ознайомлений
  + Початок опрацювання теми: 16.11.2023
  + Звершення опрацювання теми: 16.11.2023
* Тема №3: Функції.
  + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
    - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - як функції допомагають структурувати код та робити його більш читабельним та повторно використовуваним
    - як визначати та викликати функції, а також як вони взаємодіють з областями видимості.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 17.11.2023
  + Звершення опрацювання теми: 17.11.2023
* Тема №4: Перевантаження функцій.
  + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
    - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Як можна використовувати одну назву функції для виконання різних завдань на основі вхідних параметрів.
    - Як визначати та використовувати перевантажені функції.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 17.11.2023
  + Звершення опрацювання теми: 17.11.2023
* Тема №5: Рекурсія.
  + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
    - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Концепцію рекурсії, яка дозволяє функції викликати саму себе.
    - Як рекурсія може бути використана для розв’язання складних проблем, які можна розбити на менші підзадачі.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 17.11.2023
  + Звершення опрацювання теми: 17.11.2023
* Тема №6: Функції зі змінною кількістю параметрів.
  + Джерела Інформації:
    - [Functions with Variable Number of Arguments in C++ - Scaler Topics](https://www.scaler.com/topics/cpp/functions-with-variable-number-of-arguments-in-cpp/)
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
    - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Як визначати та використовувати функції, які можуть приймати різну кількість аргументів.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 17.11.2023
  + Звершення опрацювання теми: 17.11.2023

# **Виконання роботи:**

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

**Завдання №1 Опрацювання теорії**

* Деталі завдання: опрацювання різноманітних матеріалів, аналіз та дослідження відео, статей, книг на задані теми. Організація та структурування отриманих даних для можливості ефективніше засвоювати отримані знання на практиці. Вивчення найважливіших моментів.

**Завдання №2 Проектування за допомогою Draw.io(складання блок-схем)**

* Деталі завдання:

Побудова блок-схем до завдань 3-9.

#### Завдання №3 VNS Lab 2

* Варіант завдання: **20**
* Деталі завдання: Знайти суму ряду з точністю ε=0.0001, загальний член якого

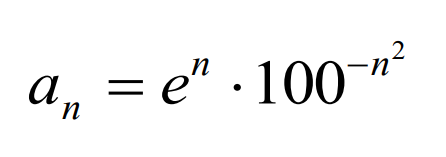


Рисунок 1 Загальний член до VNS lab 2

#### Завдання №4 VNS Lab 3

* Варіант завдання: **20**
* Деталі завдання: 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

#### Завдання №5 VNS Lab 7 Task 1

Варіант завдання: **20**

* Деталі завдання: Написати функцію triangle зі змінною кількістю параметрів, яка визначає, скільки трійок чисел можуть бути довжинами сторін трикутника кількістю параметрів 3, 9, 11.
* Важливі деталі: Використання бібліотеки stdarg.h чи vector

#### Завдання №6 VNS Lab 7 Task 2

* Варіант завдання: **20**
* Деталі завдання:

Написати перезавантажену функції до:

а) для масиву цілих чисел знаходить кількість парних елементів;

б) для рядка знаходить кількість слів, що починаються на букву «а» .

#### Завдання №7 Class Practice

* Варіант завдання: **-**
* Деталі завдання: Написати програму для простого менеджменту бібліотеки.
* Важливі деталі: Використання циклів for, do…while та while та вкладених циклів

#### Завдання №8 Self-Practice | Class Practice using do-while

* Деталі завдання: Вирішення Class Practice використовуючи do-while

#### Завдання №9 Self-Practice | VNS Lab 7 Task 1 using library vector

* Деталі завдання:  вирішення VNS Lab 7 Task 1 використовуючи бібліотеку vector

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

#### Програма № 3 VNS Lab 2

* Блок-схема

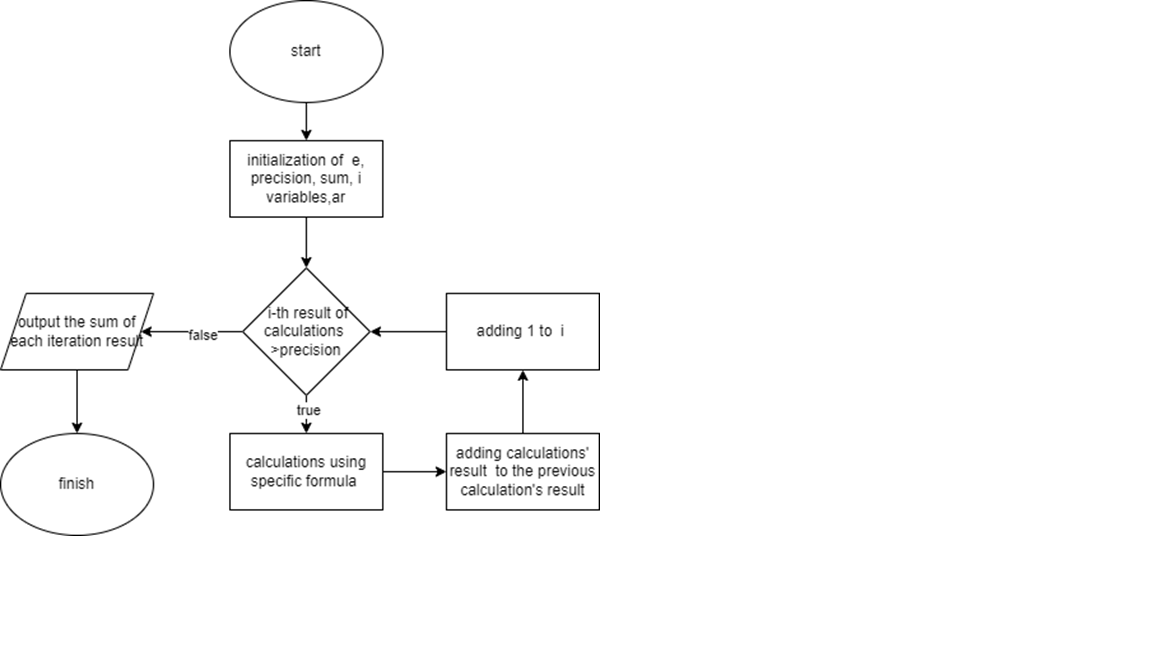


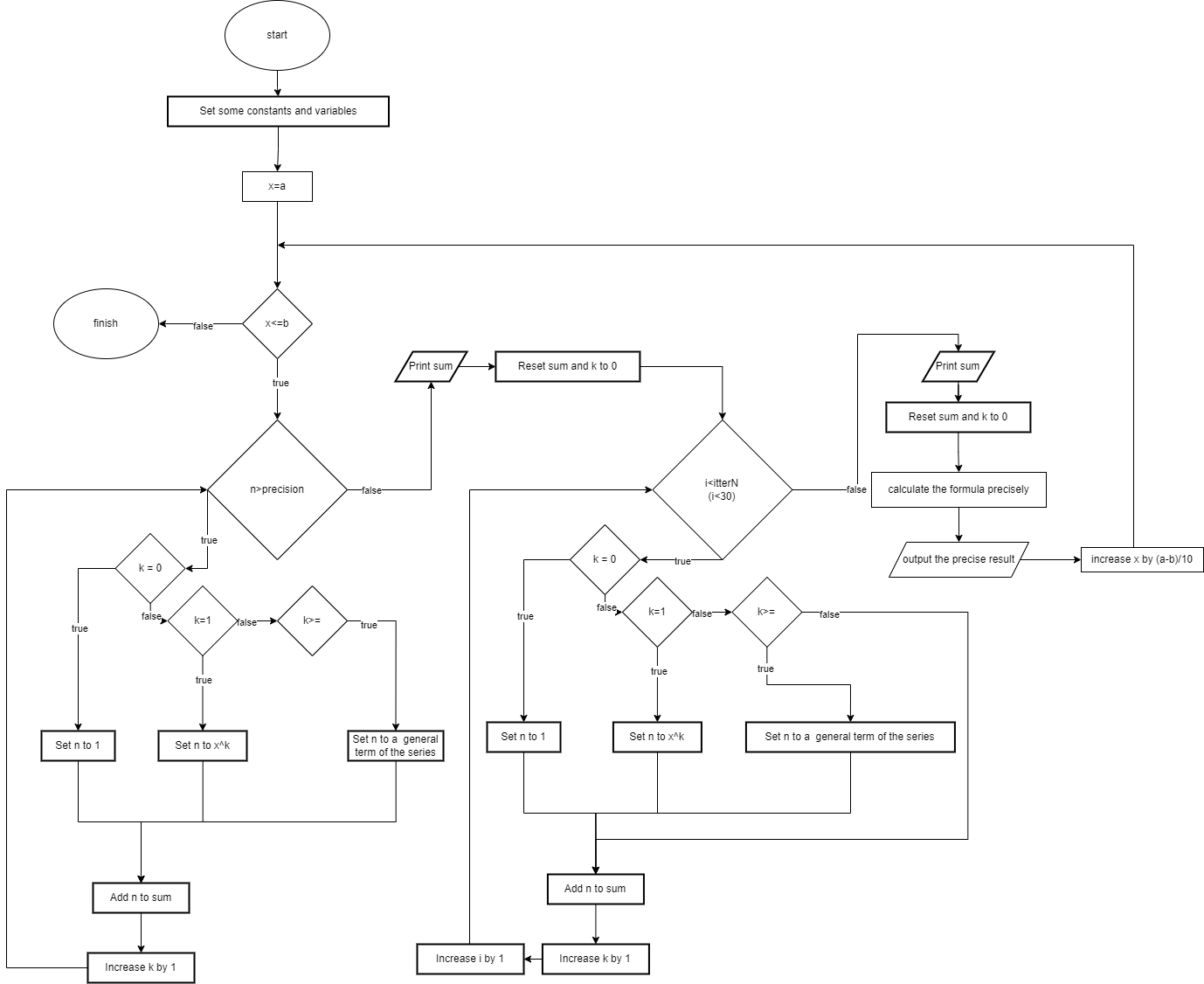
Рисунок 2 VNS Lab 2 flow-chart

* Планований час на реалізацію: 15хв

використання бібліотеки cmath

#### Програма №4 VNS Lab 3

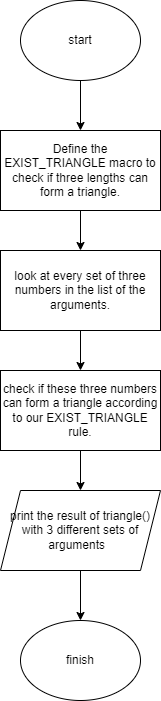
* Блок-схема



* Планований час на реалізацію: 60хв

#### Програма №5 VNS Lab 7 Task 1

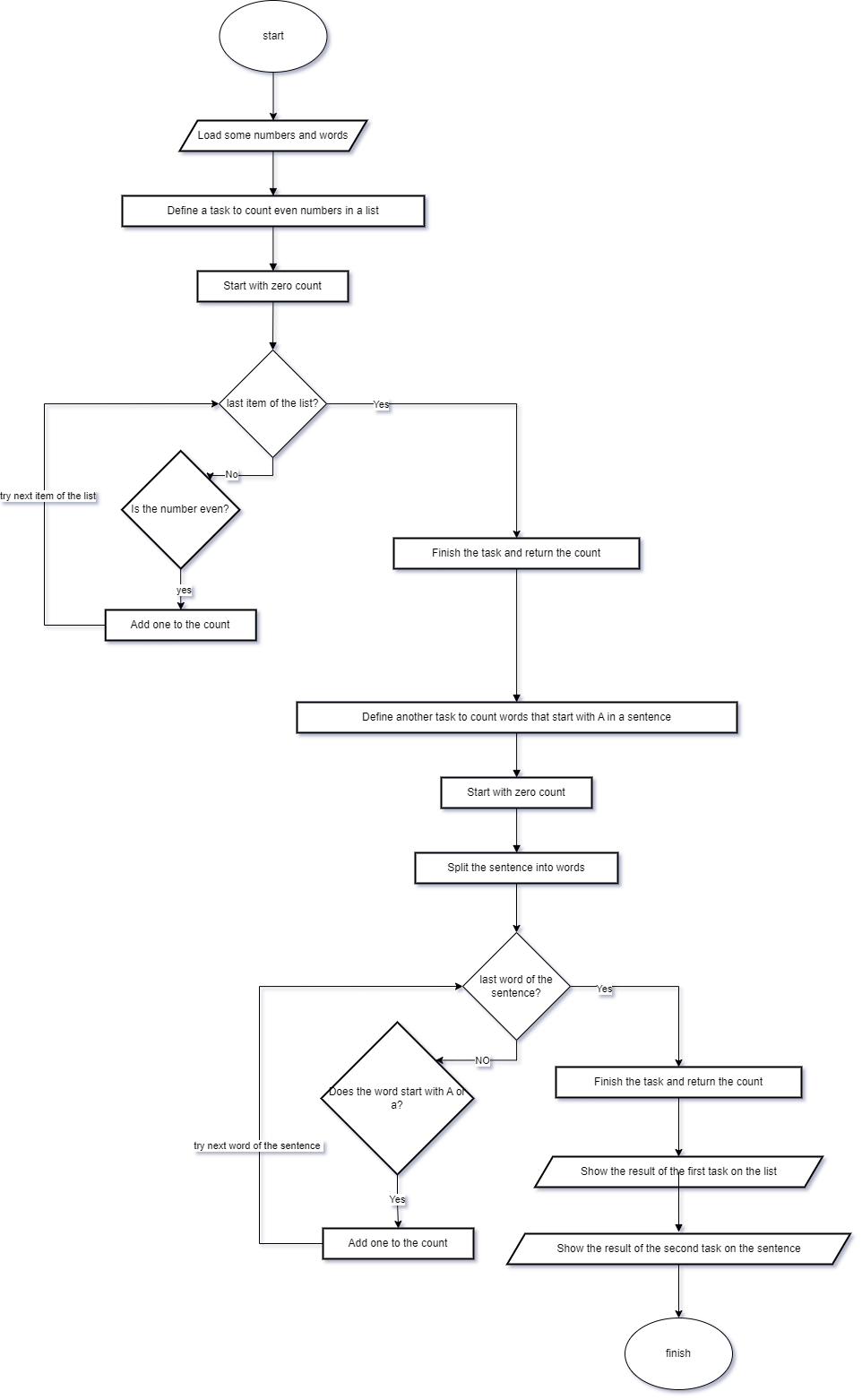
* Блок-схема



* Планований час на реалізацію 15хв

#### Програма №6 VNS Lab 7 Task 2

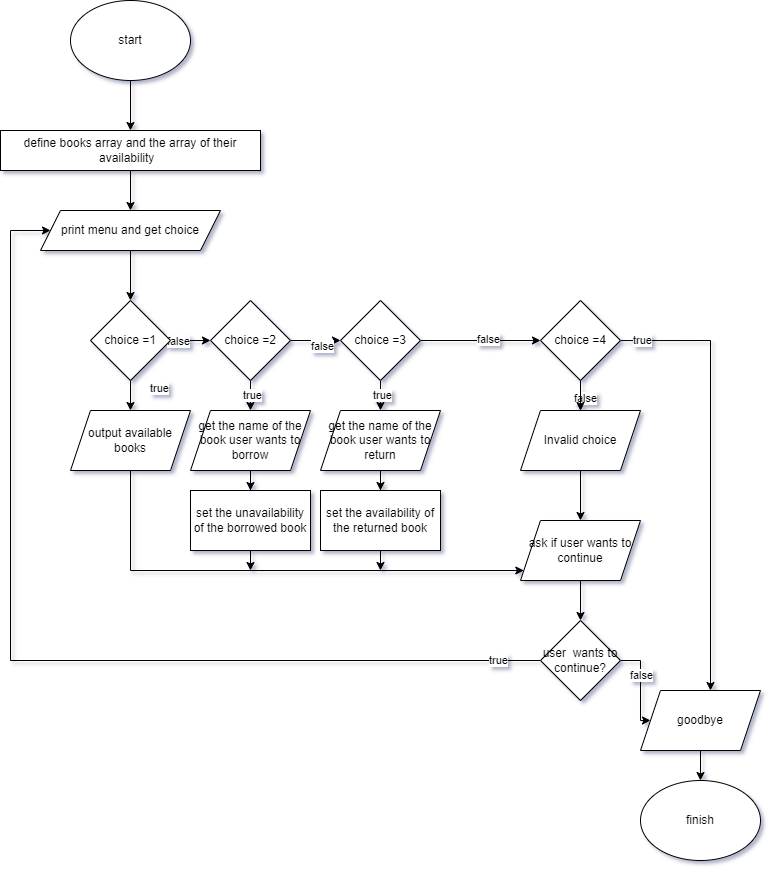
* Блок-схема



* Планований час на реалізацію 45 хв

#### Програма №7 Class Practice

* Блок-схема



* Планований час на реалізацію: 50хв

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

#include <math.h>

#include <iostream>

using namespace std;

int main()

{

    const double e {2.71828};

    double precision {0.0001};

    double sum {0};

    int i {1};

    double ar {pow(e,i)\*pow(100, -pow(i, 2))};

    while (abs(ar)  > precision)

    {

        ar = pow(e,i)\*pow(100, -pow(i, 2));

        sum += ar;

        i++;

    }

    cout << sum <<endl;

    return 0;

}

###### Код до програми № 3 VNS Lab 2 – Task 1-6

#include <iostream>

#include <cmath>

using namespace std;

int main(int argc, char const \*argv[])

{

    const double precision {0.0001}, e {2.71828},a{0.1},b{1};

    double sum{0};

    double n{0};

    int itterN {30};

    int k {0};

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

    {

        do{

        if(k==0){

            // n= 7\*sqrt(e)/4;

            n=1;

        }

        if(k==1){

            // n= 15\*sqrt(e)/8;

            n=pow(x,k);

        }

        if(k>=2){

            n= pow(2,-k)\*(1/tgamma(-2+k)+1/tgamma(-1+k)+1/tgamma(k))\*pow(x,k);

        }

        sum += n  ;

        k +=1;

    }

    while (n>=precision);

     cout<<"X" << x<<" ";

    cout<<"SE" << sum<<" ";

    sum = 0;

    k=0;

    for (size\_t i = 0; i < itterN; i++)

        {

        if(k==0){

            // n= 7\*sqrt(e)/4;

            n=1;

        }

        if(k==1){

            // n= 15\*sqrt(e)/8;

            n=pow(x,k);

        }

        if(k>=2){

            n= pow(2,-k)\*(1/tgamma(-2+k)+1/tgamma(-1+k)+1/tgamma(k))\*pow(x,k);

        }

        sum += n;

        k +=1;

    }

    cout<<"SN" << sum<<" ";

    sum = 0;

    k=0;

    cout<<"Y:" << (pow(x,2)/4+x/2+1)\*pow(e,x/2)<<endl;

    }

    return 0;

}

###### Код до програми № 4 VNS Lab 3 [https://github.com/artificial-intelligence department/ai\_programming\_playground/pull/494](https://github.com/artificial-intelligence%20department/ai_programming_playground/pull/494)

#include <iostream>

#include <cstdarg>

using namespace std;

#define EXIST\_TRIANGLE(x, y, z) ((x + y > z) && (x + z > y) && (y + z > x))

int triangle(int n, ...){

    va\_list args;

    va\_start(args, n);

    int triangleAm {0};

for (int i = 0; i < n-2 ; i+=3) {

        int x = va\_arg(args, int);

        int y = va\_arg(args, int);

        int z = va\_arg(args, int);

        if (EXIST\_TRIANGLE(x, y, z)) {

            triangleAm++;

            cout<< x<< " "<<y<<" "<<z<<endl;

        }

}

return triangleAm;

}

int main(int argc, char const \*argv[])

{

    cout<< triangle(3,3,4,5)<<endl;

    cout<< triangle(9,9,1, 3, 4, 5,6,8,7,13)<<endl;

    cout<< triangle(11,9,1, 3, 2, 5,6,8,2,13, 15,16)<<endl;

    return 0;

}

###### Код до програми № 5 VNS Lab 7 Task 1 [https://github.com/artificial-intelligence department/ai\_programming\_playground/pull/494](https://github.com/artificial-intelligence%20department/ai_programming_playground/pull/494)

#include <iostream>

#include <vector>

#include <string>

#include <sstream>

using namespace std;

int process(const vector<int>& arr) {

    int count = 0;

    for (int num : arr) {

        if (num % 2 == 0) {

            count++;

        }

    }

    return count;

}

int process(const string& str) {

    int count = 0;

    istringstream iss(str);

    string word;

    while (iss >> word) {

        if (word[0] == 'a' || word[0] == 'A') {

            count++;

        }

    }

    return count;

}

int main() {

    vector<int> arr = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

    string str = "apple banana aprf avoerr asrfsd besdrg";

    cout << "Amount of even numbers: " << process(arr) << endl;

    cout << "Amount of words which start from a: " << process(str) << endl;

    return 0;

}

###### Код до програми № 6 VNS Lab 7 Task 2 [https://github.com/artificial-intelligence department/ai\_programming\_playground/pull/494](https://github.com/artificial-intelligence%20department/ai_programming_playground/pull/494)

#include <cstdio>

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main() {

    vector<string> titles = {"Harry Potter", "The Hobbit", "Pride and Prejudice", "1984"};

    vector<bool> status = {true, true, true, true};

    int selection;

    string titleName;

    libraryMenu:

    cout << "Library Management\n";

    cout << "1. List all books\n";

    cout << "2. Borrow a book\n";

    cout << "3. Return a book\n";

    cout << "4. Exit\n";

    cout << "Enter your selection: ";

    cin >> selection;

    switch(selection) {

        case 1:

            for (int i = 0; i < titles.size(); i++) {

                cout << i + 1 << ". " << titles[i] << (status[i] ? " (Available)" : " (Borrowed)") << "\n";

            }

            break;

        case 2:

            cin.ignore();

            cout << "Enter book name to borrow: ";

            getline(cin, titleName);

            for (const auto &title : titles) {

                if (title == titleName) {

                    int index = &title - &titles[0];

                    if (status[index]) {

                        status[index] = false;

                        cout << "Book borrowed successfully!\n";

                    } else {

                        cout << "Book is already borrowed.\n";

                    }

                    break;

                }

            }

            break;

        case 3:

            cin.ignore();

            cout << "Enter book name to return: ";

            getline(cin, titleName);

            for (const auto &title : titles) {

                if (title == titleName) {

                    int index = &title - &titles[0];

                    if (!status[index]) {

                        status[index] = true;

                        cout << "Book returned successfully!\n";

                    } else {

                        cout << "Book wasn't borrowed.\n";

                    }

                    break;

                }

            }

            break;

        case 4:

            cout << "Goodbye!";

            return 0;

        default:

            cout << "Invalid selection.\n";

            goto libraryMenu;

    }

    char continueSelection;

    do {

        cout << "Do you want to perform another operation? (Y/N): ";

        cin >> continueSelection;

    } while (continueSelection != 'Y' && continueSelection != 'N' && continueSelection != 'y' && continueSelection != 'n');

    if (continueSelection == 'Y' || continueSelection == 'y') {

        goto libraryMenu;

    } else {

        cout << "Goodbye!";

    }

    return 0;

}

###### Код до програми № 7 Class Practice [https://github.com/artificial-intelligence department/ai\_programming\_playground/pull/494](https://github.com/artificial-intelligence%20department/ai_programming_playground/pull/494)

#include <cstdio>

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main() {

    vector<string> titles = {"Harry Potter", "The Hobbit", "Pride and Prejudice", "1984"};

    vector<bool> status = {true, true, true, true};

    int selection;

    string titleName;

    char continueSelection;

    do {

        cout << "Library Management\n";

        cout << "1. List all books\n";

        cout << "2. Borrow a book\n";

        cout << "3. Return a book\n";

        cout << "4. Exit\n";

        cout << "Enter your selection: ";

        cin >> selection;

        switch(selection) {

            case 1:

                for (int i = 0; i < titles.size(); i++) {

                    cout << i + 1 << ". " << titles[i] << (status[i] ? " (Available)" : " (Borrowed)") << "\n";

                }

                break;

            case 2:

                cin.ignore();

                cout << "Enter book name to borrow: ";

                getline(cin, titleName);

                for (const auto &title : titles) {

                    if (title == titleName) {

                        int index = &title - &titles[0];

                        if (status[index]) {

                            status[index] = false;

                            cout << "Book borrowed successfully!\n";

                        } else {

                            cout << "Book is already borrowed.\n";

                        }

                        break;

                    }

                }

                break;

            case 3:

                cin.ignore();

                cout << "Enter book name to return: ";

                getline(cin, titleName);

                for (const auto &title : titles) {

                    if (title == titleName) {

                        int index = &title - &titles[0];

                        if (!status[index]) {

                            status[index] = true;

                            cout << "Book returned successfully!\n";

                        } else {

                            cout << "Book wasn't borrowed.\n";

                        }

                        break;

                    }

                }

                break;

            case 4:

                cout << "Goodbye!";

                return 0;

            default:

                cout << "Invalid selection.\n";

        }

        do {

            cout << "Do you want to perform another operation? (Y/N): ";

            cin >> continueSelection;

        } while (continueSelection != 'Y' && continueSelection != 'N' && continueSelection != 'y' && continueSelection != 'n');

    } while (continueSelection == 'Y' || continueSelection == 'y');

    cout << "Goodbye!";

    return 0;

}

###### Код до програми № 8 Self-Practice Class Practice using do-while [https://github.com/artificial-intelligence department/ai\_programming\_playground/pull/494](https://github.com/artificial-intelligence%20department/ai_programming_playground/pull/494)

#include <iostream>

#include <vector>

using namespace std;

#define EXIST\_TRIANGLE(x, y, z) ((x + y > z) && (x + z > y) && (y + z > x))

int triangle(vector<int>& n){

    int triangleAm {0};

for (int i = 0; i < n.size()-2 ; i+=3) {

        int x = n[i];

        int y = n[i+1];

        int z = n[i+2];

        if (EXIST\_TRIANGLE(x, y, z)) {

            triangleAm++;

        }

}

return triangleAm;

}

int main(int argc, char const \*argv[])

{

    cout<< triangle({3,4,5})<<endl;

    cout<< triangle({9,1, 3, 4, 5,6,8,2,13})<<endl;

    cout<< triangle({9,1, 3, 2, 5,6,8,2,13, 15,16})<<endl;

    return 0;

}

###### Код до програми № 9 Self-Practice VNS Lab 7 Task 1 (vector) - [https://github.com/artificial-intelligence department/ai\_programming\_playground/pull/494](https://github.com/artificial-intelligence%20department/ai_programming_playground/pull/494)

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

#### Завдання №3 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_03\output> & .\'vns\_lab\_02.exe'

0.0271829

VNS Lab 2

#### Завдання №4 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_03\output> & .\'vns\_lab\_03.exe'

X0.1 SE1.10532 SN1.10532 Y:1.10646

X0.19 SE1.21033 SN1.21033 Y:1.21405

X0.28 SE1.32674 SN1.32674 Y:1.33386

X0.37 SE1.45639 SN1.45639 Y:1.46699

X0.46 SE1.60137 SN1.60137 Y:1.61466

X0.55 SE1.76398 SN1.76399 Y:1.77814

X0.64 SE1.94682 SN1.94682 Y:1.95883

X0.73 SE2.15274 SN2.15275 Y:2.15821

X0.82 SE2.38494 SN2.38494 Y:2.37791

X0.91 SE2.64693 SN2.64694 Y:2.61964

X1 SE2.94263 SN2.94263 Y:2.88526

VNS Lab 3

#### Завдання №5 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_03\output> & .\'vns\_lab\_07.exe'

3 4 5

1

4 5 6

1

2 5 6

1

VNS Lab 7 Task1

#### Завдання №6 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_03\output> & .\'vns\_lab\_07\_task\_02.exe'

5

1 2 3 4 5

asd ss f fgh a a dvxcvx

Amount of even numbers: 2

Amount of words which start from a: 3

VNS lab 7 Task 2

#### Завдання №7 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_03\output> & .\'class\_practice\_goto.exe'

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 1

1. Harry Potter (Available)

2. The Hobbit (Available)

3. Pride and Prejudice (Available)

4. 1984 (Available)

Do you want to perform another operation? (Y/N): y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 2

Enter book name to borrow: Harry Potter

Book borrowed successfully!

Do you want to perform another operation? (Y/N): y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 1

1. Harry Potter (Borrowed)

2. The Hobbit (Available)

3. Pride and Prejudice (Available)

4. 1984 (Available)

Do you want to perform another operation? (Y/N): y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 3

Enter book name to return: Harry Potter

Book returned successfully!

Do you want to perform another operation? (Y/N): y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 1

1. Harry Potter (Available)

2. The Hobbit (Available)

3. Pride and Prejudice (Available)

4. 1984 (Available)

Do you want to perform another operation? (Y/N): n

Goodbye!

Class Practice

#### Завдання №8 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_03\output> & .\'class\_practice\_goto.exe'

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 1

1. Harry Potter (Available)

2. The Hobbit (Available)

3. Pride and Prejudice (Available)

4. 1984 (Available)

Do you want to perform another operation? (Y/N): y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 2

Enter book name to borrow: Harry Potter

Book borrowed successfully!

Do you want to perform another operation? (Y/N): y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 1

1. Harry Potter (Borrowed)

2. The Hobbit (Available)

3. Pride and Prejudice (Available)

4. 1984 (Available)

Do you want to perform another operation? (Y/N): y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 3

Enter book name to return: Harry Potter

Book returned successfully!

Do you want to perform another operation? (Y/N): y

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your selection: 1

1. Harry Potter (Available)

2. The Hobbit (Available)

3. Pride and Prejudice (Available)

4. 1984 (Available)

Do you want to perform another operation? (Y/N): n

Goodbye!

Self-Practice Class Practice using do-while

#### Завдання №8 Деталі по виконанню і тестуванню програми

PS D:\ED\University\C++\epic\_03\output> & .\'vns\_lab\_07.exe'

3 4 5

1

4 5 6

1

2 5 6

1

Self-Practice VNS Lab 7 Task 1 (vector)

Час затрачений на виконання завдання: 3дн

# **Висновки:**

У процесі вивчення теоретичного матеріалу та виконання завдань розділу я оволодів навичками роботи з Я навчився, як використовувати цикли, оператори переходу, функції, перевантаження функцій, рекурсію та функції зі змінною кількістю параметрів для розв’язання різних задач.Тема “Функції з змінною кількістю аргументів”виділилася на фоні інших, але я успішно подолав усі труднощі. Я закріпив свої знання на практиці, виконавши лабораторні та практичні роботи, а також самостійно практикуючись. Я впевнений, що я якісно опрацював теорію та готовий до практичного застосування своїх знань.