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

***Виконав(ла):***

студент групи Ші-13

Мельников Дмитро Володимирович

# **Тема роботи:**

Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія.

# **Мета роботи:**

Навчитись добре використовувати цикли , функції , знайти різницю між звичайними ти перевантаженими функціями.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

**Тема №1: Цикли**

**Тема №2: Функції**

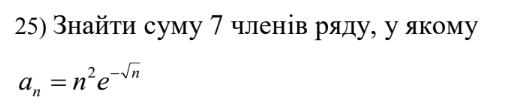
* **Тема №1:Цикли.**
  + Джерела Інформації:
  + <https://www.youtube.com/watch?v=zPi163Xo4m8&list=PLQOaTSbfxUtCrKs0nicOg2npJQYSPGO9r&index=19>
  + <https://www.youtube.com/watch?v=j1OrjcZyrhg&list=PLQOaTSbfxUtCrKs0nicOg2npJQYSPGO9r&index=26>
  + Що опрацьовано:
    - Цикл for , while , do while , to do.(**Вставив перше та останнє відео автора про цикли)**
  + Статус: Ознайомлений
  + Початок опрацювання теми: Дата 15.11.2023
  + Звершення опрацювання теми: Дата 28.11.2023
* **Тема №3**: **Функції.**
  + Джерела Інформації:
  + <https://www.youtube.com/watch?v=HFH65-sNQmI&list=PLQOaTSbfxUtCrKs0nicOg2npJQYSPGO9r&index=43>
  + https://www.youtube.com/watch?v=7btUwxD4V5s&list=PLQOaTSbfxUtCrKs0nicOg2npJQYSPGO9r&index=63
  + Що опрацьовано:
    - Функції…
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.11.2023
  + Звершення опрацювання теми: 28.11.2023

# **Виконання роботи:**

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

**Lab# programing: VNS lab 2**

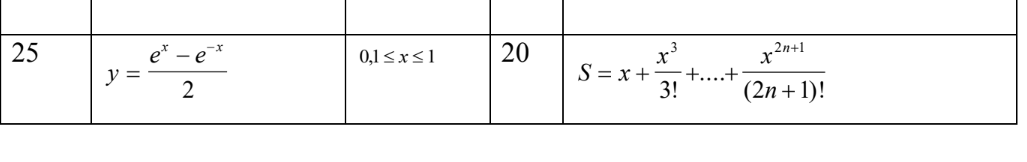
* Варіант 25



1 Lab# programing: VNS lab 2

**Lab# programing: VNS lab 3**

**-Варіант 25**

****

**2** Lab# Programing:VNS lab3

**Lab# programing: VNS lab 7**

**-Варіант 25**

**Написати функцію (або макровизначення), що знаходить довжину сторони**

**по координатах його точок.. Написати функцію square, що обчислює площу**

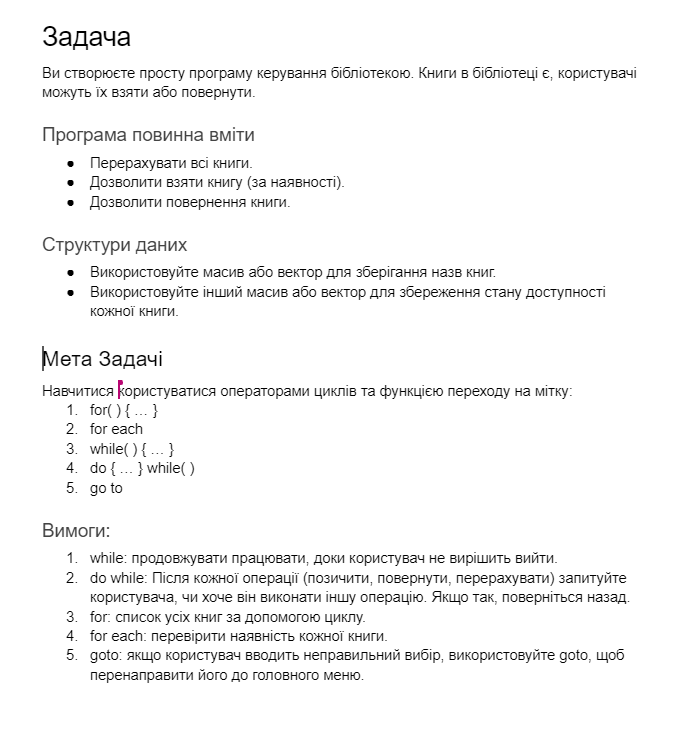
**трикутника, заданого координатами вершин. Написати функцію square1 з**

**змінною кількістю параметрів, що визначає площу трикутника, що містить**

**діагональ найбільшої довжини опуклого багатокутника, заданого**

**координатами своїх вершин**

**Class practice: Library.**

****

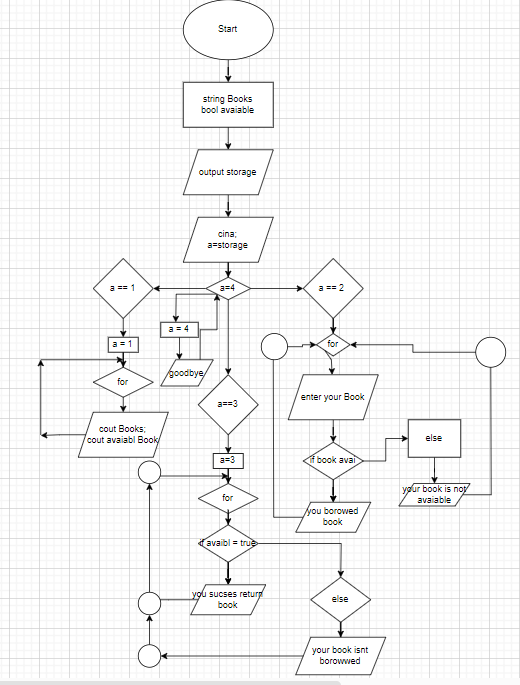
**3** Class practice:Library

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

**Class practice: Library.**

-Плановий час на реалізацію 1 година 20 хвилин

- створити бібліотеку з поверненям книжок та можливістю взяти книжку



4 class practice:library diagram

Програма №13 Заголовок задачі

* Блок-схема
* Планований час на реалізацію
* Важливі деталі для врахування в імплементації

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

**Class practice: Library code.**

**Git link:**

#include<iostream>

#include<string>

using namespace std;

int main()

{

    string Books[4] = {"Sherlock Holms", "The Theory of Lie", "Alice in Wonderland", "1984"};

    bool available[4] = {true, true, true, true};

    int a;

    string NameBook;

    storage:

    cout << "Library Management" << endl;

    cout << "1. List all books" << endl;

    cout << "2. Borrow a book" << endl;

    cout << "3. Return a book" << endl;

    cout << "4. Exit" << endl;

    cout << "Enter your choice: ";

    cin >> a;

    if (a == 1)

    {

        for (int i = 0; i < 4; i++)

        {

            cout << i + 1 << ". " << Books[i] << (available[i] ? " (Available)" : " (Borrowed)") << endl;

        }

        cout << endl;

    }

    else if (a == 2)

    {

        cout << "Enter your book name:";

        cin >> NameBook;

        for (int i = 0; i < 4; i++)

        {

            if (Books[i] == NameBook)

            {

                if (available[i])

                {

                    available[i] = false;

                    cout << "You successfully borrowed your book" << endl;

                }

                else

                {

                    cout << "Your book is already borrowed" << endl;

                }

                break;

            }

        }

    }

    else if (a == 3)

    {

        cout << "Enter your book name:";

        cin >> NameBook;

        for (int i = 0; i < 4; i++)

        {

            if (Books[i] == NameBook)

            {

                if (!available[i])

                {

                    available[i] = true;

                    cout << "You successfully returned your book" << endl;

                }

                else

                {

                    cout << "Your book isn't borrowed" << endl;

                }

                break;

            }

        }

    }

    else if (a == 4)

    {

        cout << "Goodbye";

        return 0;

    }

    goto storage;

}

5 Class practice: Library code

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

**Class practice: Library code result.**

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your choice: 1

1. Sherlock Holms (Available)

2. The Theory of Lie (Available)

3. Alice in Wonderland (Available)

4. 1984 (Available)

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your choice: 2

Enter your book name:1984

You successfully borrowed your book

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your choice: 3

Enter your book name:1984

You successfully returned your book

Library Management

1. List all books

2. Borrow a book

3. Return a book

4. Exit

Enter your choice: 4

Goodbye

PS J:\Epic3>

6 class practice: Library result.

**Lab# programing: VNS lab 2**

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

    float sum=0;

    for(int n = 1 ; n < 8; n++)

    {

        sum += n\*n\*exp(-sqrt(n));

        cout << sum << "\n";

    }

    return 0;

}

**7** Lab# programing: VNS lab 2 code.

**Lab# programing: VNS lab 2**

**Code result.**

0.367879

1.34035

2.93264

5.098

7.76995

10.8781

14.3548

PS J:\Epic3>

**8** Lab# programing: VNS lab 2 code result

**Lab# programing: VNS lab 3**

#include <iostream>

#include <cmath>

using namespace std;

unsigned long long Factorial(int n)

{

unsigned long long factorial = 1;

    for (int i = 1; i <= n; ++i) {

        factorial \*= i;

    }

    return factorial;

}

double func\_for\_y(double x) {

    double y = exp(x) - exp(-x) / 2.0;

    return y;

}

double func\_for\_sum(int n, double x) {

    double firstnumber = 1.0;

    double sum = 1.0;

    for (int i = 1; i <= 20; i++) {

       double sum = pow(x,2\*i+1)/Factorial(2\*i+1);

        firstnumber += sum;

    }

    return firstnumber;

}

double SEpsilon(double epsilon, double x) {

    double result = 1.0;

    double term = 1.0;

    int i = 0;

    while (fabs(term) > epsilon) {

        term \*= -1;

        term \*= (2 \* i + 1) \* (2 \* i + 2);

        term /= (2 \* i + 1) \* (2 \* i + 2);

        term \*= x \* x;

        term /= (2 \* i + 2);

        result += term;

        ++i;

    }

    return result;

}

int main() {

    for (double x = 0.1; x <= 1.0; x += 0.045) {

        cout << "x = " << x;

        double epsilon = 0.0001;

        cout << " SN = ";

        for (int i = 0; i < 20; i++) {

            double sum = func\_for\_sum(i, x);

            printf("%0.4f ", sum);

            break;

        }

         double resultEps = SEpsilon(epsilon, x);

         cout << "SE = " << resultEps;

        cout << " y = ";

        double y = func\_for\_y(x);

        printf("%0.4f\n", y);

    }

    return 0;

}

**9** Lab# programing: VNS lab 3 code

**Lab# programing: VNS lab 3 code result**

x = 0.1 SN = 1.0002 SE = 0.995012 y = 0.6528

x = 0.145 SN = 1.0005 SE = 0.989543 y = 0.7235

x = 0.19 SN = 1.0011 SE = 0.982112 y = 0.7958

x = 0.235 SN = 1.0022 SE = 0.972765 y = 0.8696

x = 0.28 SN = 1.0037 SE = 0.961558 y = 0.9452

x = 0.325 SN = 1.0058 SE = 0.948558 y = 1.0228

x = 0.37 SN = 1.0085 SE = 0.933839 y = 1.1024

x = 0.415 SN = 1.0120 SE = 0.917491 y = 1.1842

x = 0.46 SN = 1.0164 SE = 0.899605 y = 1.2684

x = 0.505 SN = 1.0217 SE = 0.880283 y = 1.3552

x = 0.55 SN = 1.0282 SE = 0.859633 y = 1.4448

x = 0.595 SN = 1.0357 SE = 0.837771 y = 1.5372

x = 0.64 SN = 1.0446 SE = 0.814813 y = 1.6328

x = 0.685 SN = 1.0548 SE = 0.790877 y = 1.7317

x = 0.73 SN = 1.0666 SE = 0.766094 y = 1.8341

x = 0.775 SN = 1.0799 SE = 0.740586 y = 1.9402

x = 0.82 SN = 1.0950 SE = 0.714478 y = 2.0503

x = 0.865 SN = 1.1120 SE = 0.687896 y = 2.1645

x = 0.91 SN = 1.1309 SE = 0.660968 y = 2.2831

x = 0.955 SN = 1.1519 SE = 0.633807 y = 2.4063

PS J:\Epic3>

**10** Lab# programing: VNS lab 3 code result

**Lab# programing: VNS lab 7 code**

#include <iostream>

#include <cmath>

using namespace std;

struct Point {

    double x, y;

};

// Функція для знаходження довжини відрізка між двома точками

double findDistance(const Point& p1, const Point& p2) {

    return sqrt(pow(p2.x - p1.x, 2) + pow(p2.y - p1.y, 2));

}

// Функція для обчислення площі трикутника за координатами його вершин

double square(const Point& A, const Point& B, const Point& C) {

    double a = findDistance(B, C);

    double b = findDistance(A, C);

    double c = findDistance(A, B);

    // Застосовуємо формулу Герона для площі трикутника

    double s = 0.5 \* (a + b + c);

    return sqrt(s \* (s - a) \* (s - b) \* (s - c));

}

// Функція площі трикутника з найбільгою діагоналлю багатокутника

double square1(const initializer\_list<Point>& vertices) {

    double maxDiagonalLength = 0.0;

    // Знаходимо довжину найбільшої діагоналі

    for (auto it1 = vertices.begin(); it1 != vertices.end(); ++it1) {

        for (auto it2 = next(it1); it2 != vertices.end(); ++it2) {

            double diagonalLength = findDistance(\*it1, \*it2);

            if (diagonalLength > maxDiagonalLength) {

                maxDiagonalLength = diagonalLength;

            }

        }

    }

    // Площа трикутника (формула герона)

    double s = 0.5 \* maxDiagonalLength;

    return sqrt(3.0) / 4.0 \* pow(maxDiagonalLength, 2);

}

int main() {

    // Площа трикутника ABC

    float ax;

    float ay;

    float bx;

    float by;

    float cx;

    float cy;

    Point A = {ax, ay};

    Point B = {bx, by};

    Point C = {cx, cy};

    double areaABC = square(A, B, C);

    cout << "Square of triangle ABC: " << areaABC << endl;

    cout <<"Enter ax:";

        cin>> ax;

    cout <<"Enter ay:";

        cin>>ay;

    cout <<"Enter bx:";

        cin>>bx;

    cout <<"Enter by:";

        cin>>by;

    cout <<"Enter cx:";

        cin>>cx;

    cout <<"Enter cy:";

        cin>>cy;

    // Площа трикутника з діагоналлю найбільшої довжини

    double areaWithMaxDiagonal = square1({A, B, C});

    cout << "Square with  longer diagonal: " << areaWithMaxDiagonal << endl;

    return 0;

}

**11** Lab# programing: VNS lab 7 code

**Lab# programing: VNS lab 7 code result.**

Enter ax:1

Enter ay:2

Enter bx:4

Enter by:2

Enter cx:0.5

Enter cy:6

Square of triangle ABC: 0

Square with longer diagonal: 3.30532e+66

PS J:\Epic3>

**12** Lab# programing: VNS lab 7 code result.

**Meeting with Team**



13 Meeting call team 6

# **Висновки:**

**Детально ознайомився з циклами for/while/do while навсився створювати функцію та дослідив overload function.**