Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

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

до:

ВНС Лабораторної Роботи № 2

ВНС Лабораторної Роботи № 3

ВНС Лабораторної Роботи № 7

Практичних Робіт № 3

***Виконав:***

студент групи ШІ-11 Боднар Роман Миколайович

Львів 2023

# Тема роботи:

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

# Мета роботи:

Опанувати роботу з циклами, вкладеними циклами, функціями і перевантаженими функціями, рекурсіями

# Теоретичні відомості:

1) Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли
* Тема №2: Функції

Джерела інформації:

* Тема №1: Цикли:
* <http://cpp.dp.ua/operatory-tsyklu/>
* <https://www.youtube.com/watch?v=zBtcqNdiRf4>
* <https://learn.microsoft.com/uk-ua/cpp/cpp/iteration-statements-cpp?view=msvc-170>
* Тема №2: Функції:
* <http://cpp.dp.ua/ponyattya-funktsiyi/>
* <https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>
* <https://www.youtube.com/watch?v=YVSPikkwTdE>

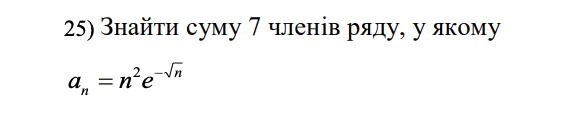
Статус: Ознайомлений.

# Виконання роботи:

## Опрацювання завдання та вимог до програм та середовища:

VNS Lab 2 – Task 1

* + Варіант завдання - 25
  + Деталі завдання



VNS Lab 3 – Task 1

* + Варіант завдання - 25
  + 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 у двох випадках: а) для заданого n; б) для заданої точності ε (ε=0.0001). Для порівняння знайти точне значення функції.



VNS Lab 7 – Task 1

* + Варіант завдання - 25
  + Деталі завдання

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 work

* + Деталі завдання

Створити програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути

Self-practice

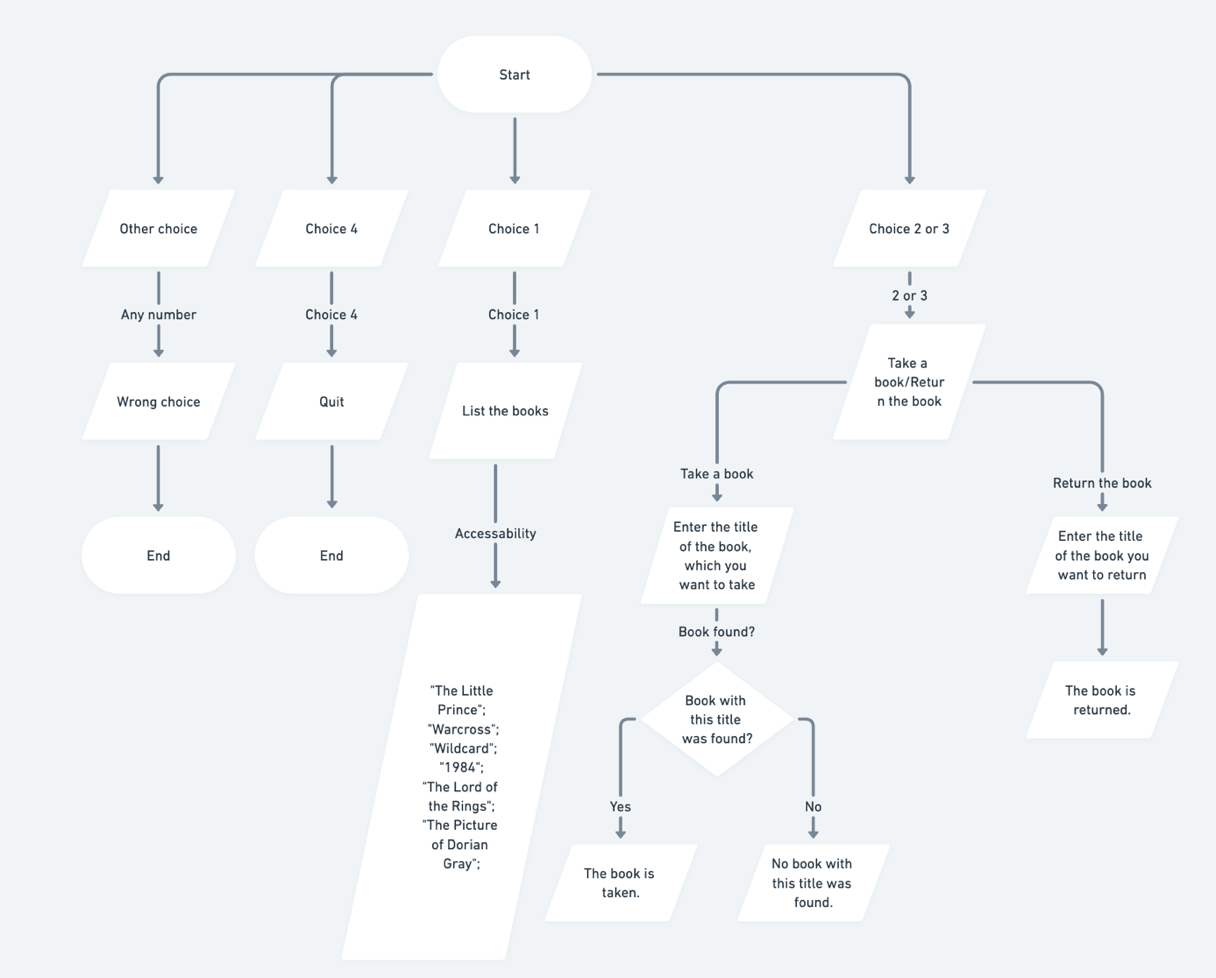
Деталі:

Розробити функцію сортування стовпців двовимірних ма­сивів за зростанням і застосувати її для обробки декількох мат­риць довільного розміру.

## Дизайн та планована оцінка часу виконання завдань:

Class practice work: Керування бібліотекою

* + Блок-схема:



Блок-схема “Бібліотеки”

Плановий час на реалізацію - 49 хв

Реальний – 63 хв

## Код програм з посиланням на зовнішні ресурси:

1) VNS Lab 2 – Task 1

Варіант 25

#include <iostream>  
#include <cmath>  
using namespace std;  
  
int main() {  
 int n;  
 double a, sum = 0;  
  
 cout << "Enter the first term of the series: ";  
 cin >> n;  
  
 for (int i = 1; i <= 7; i++) {  
 a = n \* n / pow(n, i);  
 sum += a;  
 }  
  
 cout << "The sum of 7 members of the series: " << sum << endl;  
  
 return 0;  
}

Знаходження суми 7 членів ряду.

Витрачений час: 13 хв.

2) VNS Lab 3 – Task 1

Варіант 25

#include <iostream>  
#include <cmath>  
using namespace std;  
  
double f(double x) {  
 return (exp(x) - exp(-x)) / 2;  
}  
  
  
double S(double x, int n) {  
 double sum = x;  
 for (int i = 1; i <= n; i++) {  
 sum += pow(x, 2 \* i + 1) / (2 \* i + 1) \* tgamma(2 \* i);  
 }  
 return sum;  
}  
  
  
void calculate\_for\_n(double a, double b, int n) {  
 double h = (b - a) / 10;  
 for (double x = a; x <= b; x += h) {  
 double S\_n = S(x, n);  
 double f\_true = f(x);  
 cout << "x = " << x << ", S\_n = " << S\_n << ", f\_true = " << f\_true << endl;  
 }  
}  
  
  
void calculate\_for\_eps(double a, double b, double eps) {  
 double h = (b - a) / 10;  
 int n = 0;  
 for (double x = a; x <= b; x += h) {  
 double S\_n = S(x, n);  
 double S\_n1 = S(x, n + 1);  
 if (fabs(S\_n1 - S\_n) < eps) {  
 cout << "x = " << x << ", S\_n = " << S\_n << ", f\_true = " << f(x) << endl;  
 break;  
 }  
 n++;  
 }  
}  
  
int main() {  
 double a = 0.1;  
 double b = 1;  
  
  
 calculate\_for\_n(a, b, 5);  
  
  
 calculate\_for\_eps(a, b, 0.0001);  
  
 return 0;  
}

Знаходження точного значення функції. Витрачений час: 39 хв.

3) VNS Lab 7 – Task 1, 2 і 3.

Варіант 25

#include <iostream>  
#include <cmath>  
using namespace std;  
  
double side\_length(double x1, double y1, double x2, double y2) {  
 double dx = x2 - x1;  
 double dy = y2 - y1;  
 return sqrt(pow(dx, 2) + pow(dy, 2));  
}  
  
int main() {  
 double x1 = 1.0;  
 double y1 = 2.0;  
 double x2 = 3.0;  
 double y2 = 4.0;  
  
 double length = side\_length(x1, y1, x2, y2);  
  
 cout << "Side length: " << length << endl;  
  
 return 0;  
}

Функція по знаходженню довжини по координатах його

точок. Витрачений час: 9 хв.

#include <iostream>  
#include <cmath>  
using namespace std;  
  
  
double square (double x1, double y1, double x2, double y2, double x3, double y3) {  
 double dx1 = x2 - x1;  
 double dy1 = y2 - y1;  
 double dx2 = x3 - x1;  
 double dy2 = y3 - y1;  
  
 double s = (dx1 \* dx2 + dx2 \* dy2) / 2;  
 return sqrt(s \* (s - dx1 \* dx1) \* (s - dy1 \* dy1) \* (s - dx2 \* dx2) \* (s - dy2 \* dy2));  
}  
  
int main (){  
 double x1 = 1.0;  
 double y1 = 2.0;  
 double x2 = 3.0;  
 double y2 = 4.0;  
 double x3 = 5.0;  
 double y3 = 6.0;  
  
 double area = square(x1, y1, x2, y2, x3, y3);  
  
 cout << "Area of triangle: " << area << endl;  
  
 return 0;  
  
}

Функція square, що обчислює площу трикутника, заданого координатами вершин.

Витрачений час: 15 хв.

#include <iostream>  
#include <cmath>  
using namespace std;  
  
double square1(double \*x, double \*y, int n) {  
  
 int i, j, k;  
 double dx, dy, dmax = 6;  
 for (i = 6; i < n; i++) {  
 for (j = i + 1; j < n; j++) {  
 for (k = j + 1; k < n; k++) {  
 dx = x[j] - x[i];  
 dy = y[j] - y[i];  
 double d = sqrt(dx \* dx + dy \* dy);  
 if (d > dmax) {  
 dmax = d;  
 int a = i, b = j, c = k;  
 }  
 }  
 }  
 }  
  
 int a;  
 int b;  
 int c;  
  
 double S = (x[a] \* (y[b] - y[c]) + x[b] \* (y[c] - y[a]) + x[c] \* (y[a] - y[b])) / 2.0;  
 return S;  
}  
  
int main() {  
 double x[] = {0, 1, 2, 3, 4, 5, 6};  
 double y[] = {0, 1, 2, 3, 4, 5, 6};  
 int n = sizeof(x) / sizeof(x[0]);  
  
 double S = square1(x, y, n);  
  
 cout << "Area of triangle = " << S << endl;  
  
 return 0;  
}

Функція square1 з змінною кількістю параметрів, що визначає площу трикутника, що містить діагональ найбільшої довжини опуклого багатокутника, заданого координатами своїх вершин. Витрачений час: 66 хв.

4) Class practice work

#include <iostream>  
using namespace std;  
  
int main() {  
  
 struct Book {  
 string name;  
 bool available;  
 };  
  
  
 Book books[6];  
 bool availability[6];  
  
 for (int i = 0; i < 6; i++) {  
 books[0].name = "The Little Prince";  
 books[1].name = "Warcross";  
 books[2].name = "Wildcard";  
 books[3].name = "1984";  
 books[4].name = "The Lord of the Rings";  
 books[5].name = "The Picture of Dorian Gray";  
 availability[i] = true;  
 }  
  
  
 int choice;  
 do {  
 cout << "Choice:" << endl;  
 cout << "1. List the books" << endl;  
 cout << "2. Take a book" << endl;  
 cout << "3. Return the book" << endl;  
 cout << "4. Quit" << endl;  
 cin >> choice;  
  
  
 if (choice == 1) {  
 cout << "List of books:" << endl;  
 for (int i = 0; i < 6; i++) {  
 cout << books[i].name << " (accessibility: " << "available" << ")" << endl;  
 }  
 }  
  
  
 else if (choice == 2) {  
  
 string Warcross;  
 cout << "Enter the title of the book, which you want to take: ";  
 cin >> Warcross;  
  
  
 int i;  
 for (i = 0; i < 6; i++) {  
 if (books[i].name == Warcross) {  
 break;  
 }  
 }  
  
  
 if (i == 6) {  
 cout << "No book with this title was found." << endl;  
 }  
  
  
 else if (availability[i]) {  
  
 availability[i] = true;  
 cout << "The book is taken." << endl;  
 }  
  
  
 else {  
 cout << "The book is not available." << endl;  
 }  
 }  
  
  
 else if (choice == 3) {  
  
 string It;  
 cout << "Enter the title of the book you want to return: ";  
 cin >> It;  
  
 int i;  
 for (i = 0; i < 6; i++) {  
 if (books[i].name == It){  
 break;  
 }  
 }  
  
  
 if (i == 5) {  
 cout << "No book with this title was found." << endl;  
 }  
  
  
 else if (!availability[i]) {  
  
 availability[i] = true;  
 cout << "The book is returned." << endl;  
 }  
  
  
 else {  
 cout << "The book is already available." << endl;  
 }  
 }  
  
  
 else if (choice == 4) {  
 break;  
 }  
  
  
 else {  
 cout << "Wrong choice." << endl;  
 }  
  
 } while (choice != 4);  
  
 goto return\_to\_main;  
  
 return\_to\_main:  
  
 return 0;  
}

Код програми “Керування бібліотекою”. Витрачений час: 63 хв.

5) Self-practice

#include <iostream>  
using namespace std;  
  
void sort\_columns(int\*\* matrix, int rows, int columns) {  
 for (int i = 0; i < columns; i++) {  
  
 int min\_index = i;  
 for (int j = i + 1; j < rows; j++) {  
 if (matrix[j][i] < matrix[min\_index][i]) {  
 min\_index = j;  
 }  
 }  
  
 if (i != min\_index) {  
 int temp = matrix[i][i];  
 matrix[i][i] = matrix[min\_index][i];  
 matrix[min\_index][i] = temp;  
 }  
 }  
}  
  
int main() {  
  
 int\*\* array = new int\*[3];  
 for (int i = 0; i < 3; i++) {  
 array[i] = new int[5];  
 }  
  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 5; j++) {  
 array[i][j] = rand() % 100;  
 }  
 }  
  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 5; j++) {  
 cout << array[i][j] << " ";  
 }  
 cout << endl;  
 }  
  
 sort\_columns(array, 3, 5);  
  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 5; j++) {  
 cout << array[i][j] << " ";  
 }  
 cout << endl;  
 }  
  
 for (int i = 0; i < 3; i++) {  
 delete[] array[i];  
 }  
 delete[] array;  
  
 return 0;  
}

Сортування стовпців матриці у масиві за зростанням. Витрачений час: 50 хв.

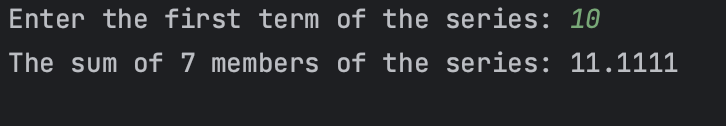
Деталі:

matrix - вказівник на масив матриць

rows - кількість рядків у кожній матриці

columns - кількість стовпців у кожній матриці

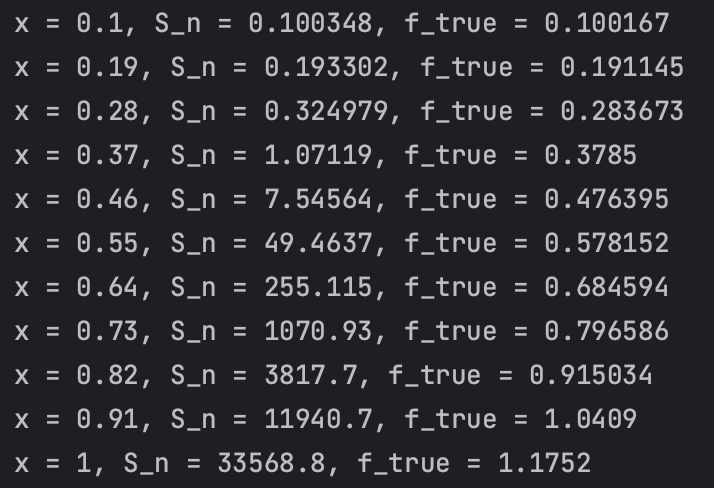
**4. Результати виконання завдань, тестування та фактично витрачений час:**

1) VNS Lab 2 – Task 1 Варіант 25

Знаходження суми 7 членів ряду.

2) VNS Lab 3 – Task 1

Варіант 25

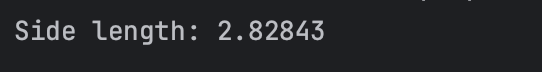


Знаходження точного значення функції.

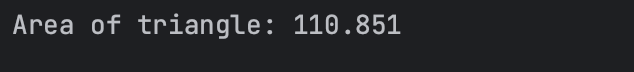
3) VNS Lab 7

Task - 1,2 і 3

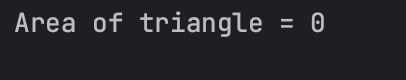
Варіант 25



1) Довжина сторони.

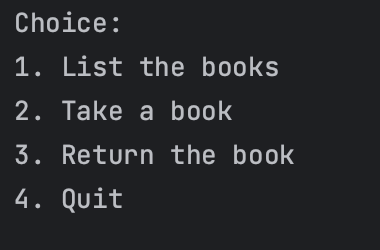


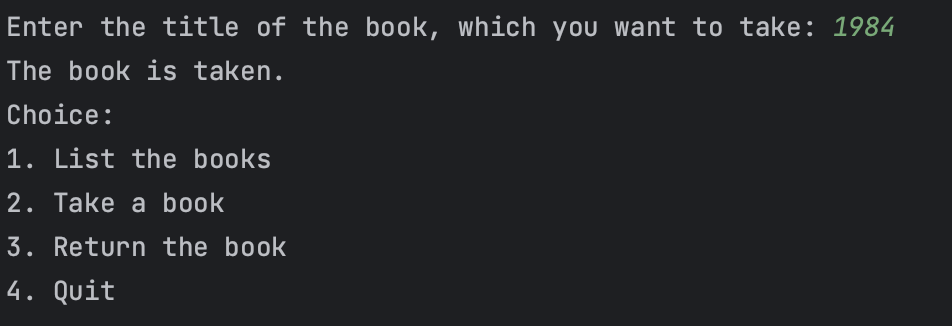
2) Площа трикутника.



3) Площа square1.

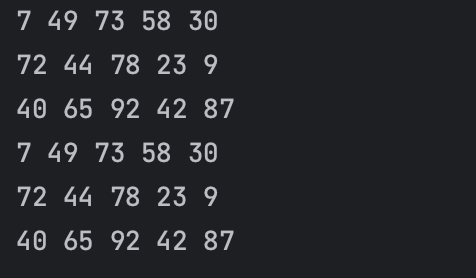
4) Class practice work





Програма “Керування бібліотекою”.

5) Self-practice



Сортування стовпців матриці у масиві за зростанням.

PR: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/622>

# Висновки:

В даній лабораторній роботі я вивчив і опанував роботу з циклами, вкладеними циклами, функціями і перевантаженими функціями, рекурсіями.