Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 2, 3, 7**

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

***з розділу***: «Цикли. Вкладені цикли. Функції та їх перенавантаження. Рекурсія»

***Виконав(ла):***

студент групи ШІ-12

Дашавський Володимир Володимирович

# **Тема роботи:**

Цикли. Вкладені цикли. Функції та їх перенавантаження. Рекурсія.

# **Мета роботи:**

Ознайомитись з використанням функцій, а також перевантажених функцій. Ознайомитись з циклами та вкладеними циклами. Ознайомитись з поняттям рекурсії.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Функції, перевантажені функції
* Тема №2: Поняття рекурсії
* Тема №3: Цикли, вкладені цикли

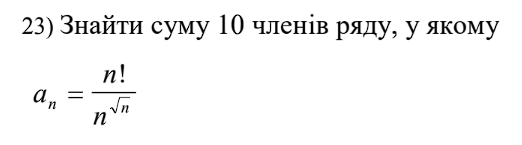
1. Індивідуальний план опрацювання теорії:

* Тема №1: Функції, перевантажені функції
  + Джерела Інформації
    - https://www.w3schools.com/cpp/cpp\_functions.asp
    - https://www.w3schools.com/cpp/cpp\_function\_overloading.asp
  + Що опрацьовано:
    - Опрацював поняття функції, їх викликання, ознайомився з перевантаженими функціями
  + Статус: Ознайомлений
  + Початок опрацювання теми: 23.11.2023
  + Звершення опрацювання теми: 25.11.2023
* Тема №2: Поняття рекурсії
  + Джерела Інформації:
    - https://www.w3schools.com/cpp/cpp\_functions\_recursion.asp
  + Що опрацьовано:
    - Ознайомився з поняттям рекурсії, навчився використовувати її на практиці.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 24.11.2023
  + Звершення опрацювання теми: 25.11.2023
* Тема №3: Цикли, вкладені цикли
  + Джерела Інформації:
    - <http://cpp.dp.ua/operatory-tsyklu/>
    - https://acode.com.ua/urok-72-tsykl-for/
  + Що опрацьовано:
    - Ознайомився з поняттям циклів ( for, while ), вкладених циклів, навчився використовувати їх на практиці
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.2023
  + Звершення опрацювання теми: 24.11.2023

# **Виконання роботи:**

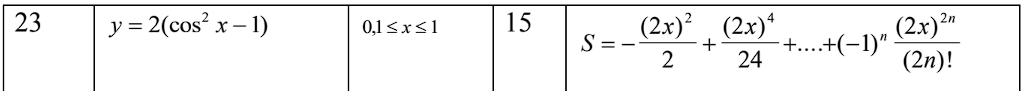
## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS Lab 2 - Task 1

* Варіант завдання - 23
* Деталі завдання
* 
* Важливі деталі для врахування в імплементації програми:

потрібно використати цикл для обрахування факторіала.

Завдання №2 VNS Lab 3 - Task 1

* Варіант завдання - 23
* 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).

Завдання №3 VNS Lab 7 - Task 1

* Варіант завдання - 23
* Деталі завдання: потрібно обрахувати довжину відрізка заданого точками, визначити чи належить точка трикутнику і опуклому многокутнику

Завдання №4 VNS Lab 7 - Task 2

* Варіант завдання - 23
* Деталі завдання: потрібно написати перевантажену функцію а) для масиву, який починається на парне число виконує циклічний зсув вліво
* на кількість елементів, що дорівнює першому елементу масиву.
* б) для масиву, який починається на непарне число виконує циклічний зсув
* вправо на кількість елементів, що дорівнює останньому елементу масиву.

Завдання №5 Class Practice Task

* Варіант завдання - library
* Деталі завдання: потрібно написати бібліотеку в якій можна брати і віддавати книжки.

Завдання №6 Self Practice Task

* Варіант завдання - Коля Вася і Теніс
* Деталі завдання: потрібно визначити хто виграв в ході кількох тенісних матчів;

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

Програма №1 VNS Lab 2 - Task 1

* Планований час на реалізацію 30 хв
* Важливі деталі для врахування в імплементації
* потрібно вміти задавати факторіал за допомогою циклів.

Програма №2 VNS Lab3 - Task1

* Планований час на реалізацію 45 хв
* Важливі деталі для врахування в імплементації

треба вміти використовувати цикл for

Програма №3 VNS Lab7 - Task1

Планований час на реалізацію 1 год 20 хв

* Важливі деталі для врахування в імплементації:

потрібно правильно використати формули для визначення чи належить точка многокутнику.

Програма №4 VNS Lab7 - Task2

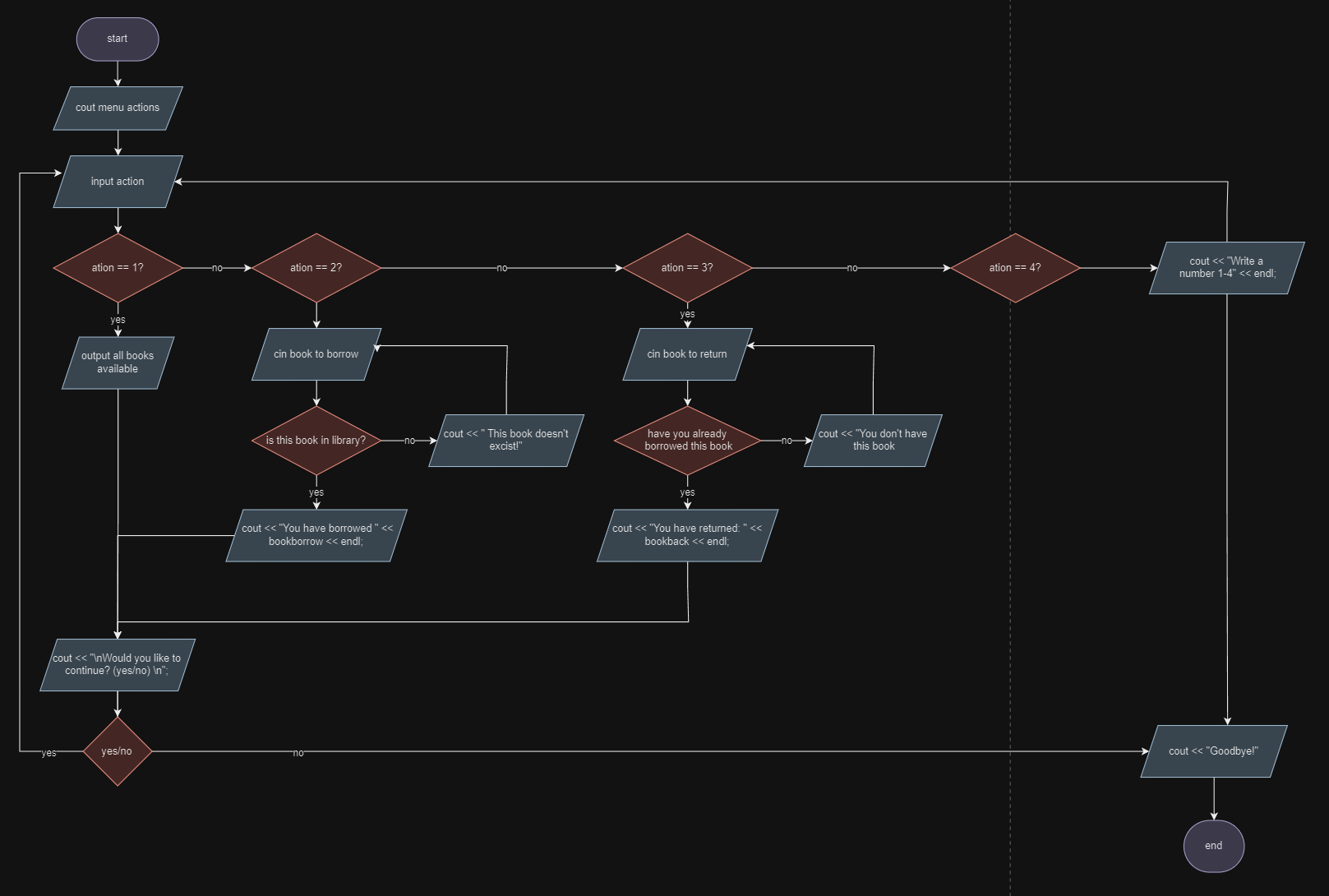
Планований час на реалізацію 40 хв

* Важливі деталі для врахування в імплементації:

потрібно написати перевантажену функцію, розуміти властивості масивів.

Програма №5 VNS Self Practice Task

блок - схема:



Планований час на реалізацію 1 год 45 хв

* Важливі деталі для врахування в імплементації:

потрібно написати бібліотеку, вміти працювати з динамічними масивами, використовувати умовні оператори.

Програма №6 Self Practice Task

Планований час на реалізацію 1 год.

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

Завдання №1 VNS Lab 2 Task 1

Для цього завдання достатньо базового розширення c++ І бібліотеки <cmath>.

Завдання №2 VNS Lab 3 Task 1

Для цього завдання достатньо базового розширення c++ І бібліотеки <cmath>.

Завдання №3 VNS Lab 7 Task 1

Для цього завдання достатньо базового розширення c++ І бібліотек <vector> та <cmath>.

Завдання №4 VNS Lab 7 Task 2

Для цього завдання достатньо базового розширення c++ І бібліотеки <vector>.

Завдання №5 Class Practice Task

Для цього завдання достатньо базового розширення c++ І бібліотек <vector> та <string>.

Завдання №6 Self Practice Task

Для цього завдання достатньо базового розширення c++ І бібліотеки <cmath>.

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

Завдання №1 VNS Lab 2 Task 1

#include <iostream>

#include <cmath>

using namespace std;

int fact(int n)

{

if (n == 1)

{

return 1;

}

return n \* fact(n - 1);

}

int main()

{

double num = 10;

double sum = 0;

for (int n = 1; n <= num; n++)

{

sum = (fact(n)) / (pow(n, sqrt(n)));

}

cout << sum;

return 0;

}

блок з кодом до програми VNS Lab 2 Task 1

Завдання №2 VNS Lab 3 Task 1

#include <iostream>

#include <cmath>

using namespace std;

double b = 1.0;

double a = 0.1;

int k = 10;

double step = (b - a) / k;

const double epsilon = 0.0001;

int factorial(int n)

{

if (n == 1)

{

return 1;

}

return n \* factorial(n - 1);

}

int main()

{

for (double c = a; c <= b; c = c + step)

{

double sn = 0.0;

double se = 0.0;

double y = 2 \* (pow(cos(c), 2) - 1);

// Обчислення sn:

for (int n = 1; n <= 15; n++)

{

double an = pow(-1, n) \* ((pow(2 \* c, 2 \* n)) / (2 \* n) / factorial(2 \* n));

sn = sn + an;

}

// Обчислення se:

int n = 1;

double en = pow((-1), n) \* ((pow(2 \* c, 2 \* n)) / (2 \* n) / factorial(2 \* n));

while (fabs(en) >= epsilon)

{

se = se + en;

n++;

en = pow((-1), n) \* ((pow(2 \* c, 2 \* n)) / (2 \* n) / factorial(2 \* n));

}

cout << " X = " << c << "; sn = " << sn << "; se = " << se << "; y = " << y << endl;

}

return 0;

}

блок з кодом до програми VNS Lab 3 Task 1

Завдання №3 VNS Lab 7 Task 1

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

// Функція обрахування довжини відрізка за координатами

int X1, X2, Y1, Y2;

cout << "Write coordinates of first dot" << endl;

cin >> X1 >> Y1;

cout << "Write coordinates of second dot" << endl;

cin >> X2 >> Y2;

int vectorx = X2 - X1;

int vectory = Y2 - Y1;

double side;

side = pow((pow(vectorx, 2) + pow(vectory, 2)), 0.5);

cout << side << endl;

обрахування довжини відрізка

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double Mx, My;

cout << "Type coordinates of M dot" << endl;

cin >> Mx >> My;

double x1, y1, x2, y2, x3, y3;

cout << "Type coordinates of dot A of the triangle" << endl;

cin >> x1 >> y1;

cout << "Type coordinates of dot B of the triangle" << endl;

cin >> x2 >> y2;

cout << "Type coordinates of dot C of the triangle" << endl;

cin >> x3 >> y3;

double AB, BC, CA;

AB = pow(((pow((x2 - x1), 2) + pow((y2 - y1), 2))), 0.5);

BC = pow(((pow((x3 - x2), 2) + pow((y3 - y2), 2))), 0.5);

CA = pow(((pow((x1 - x3), 2) + pow((y1 - y3), 2))), 0.5);

double AM, BM, CM;

AM = pow(((pow((Mx - x1), 2) + pow((My - y1), 2))), 0.5);

BM = pow(((pow((Mx - x2), 2) + pow((My - y2), 2))), 0.5);

CM = pow(((pow((Mx - x3), 2) + pow((My - y3), 2))), 0.5);

double Sabc;

double p;

p = ((AB + BC + CA) / 2);

Sabc = pow(p \* (p - AB) \* (p - BC) \* (p - CA), 0.5);

Sabc = (int)Sabc;

double Samb;

double p1;

p1 = ((AB + BM + AM) / 2);

Samb = pow(p1 \* (p1 - AB) \* (p1 - BM) \* (p1 - AM), 0.5);

double Sbmc;

double p2;

p2 = ((BM + CM + BC) / 2);

Sbmc = pow(p2 \* (p2 - BM) \* (p2 - CM) \* (p2 - BC), 0.5);

double Scma;

double p3;

p3 = ((CM + AM + CA) / 2);

Scma = pow(p3 \* (p3 - CM) \* (p3 - AM) \* (p3 - CA), 0.5);

double s;

s = Samb + Sbmc + Scma;

s = (int)s;

cout << Sabc << endl;

cout << s << endl;

if (Sabc == s)

{

cout << "true \n";

}

else

{

cout << "false";

}

}

}

визначення чи належить задана точка трикутнику

#include <iostream>

#include <vector>

using namespace std;

bool point\_in\_polygon(const vector<pair<double, double>> &polygonpoints, const pair<double, double> &point)

{

for (const auto &pointp : polygonpoints)

{

if (pointp == point)

{

return true;

}

}

int n = polygonpoints.size();

int count = 0;

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

{

int j = (i + 1) % n;

if (point.first > polygonpoints[i].first && point.first < polygonpoints[j].first)

{

double x = (point.first - polygonpoints[i].first) / (polygonpoints[j].first - polygonpoints[i].first) \* (polygonpoints[j].second - polygonpoints[i].second) + polygonpoints[i].second;

if (x > point.second)

{

count++;

}

}

}

return count % 2 == 1;

}

int main()

{

int n;

cout << "number of sides: ";

cin >> n;

vector<pair<double, double>> polygonpoints(n);

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

{

cout << "coordinates of dot " << i + 1 << ": ";

cin >> polygonpoints[i].first >> polygonpoints[i].second;

}

pair<double, double> point;

cout << "coordinates of dot we're checking: ";

cin >> point.first >> point.second;

bool is\_in = point\_in\_polygon(polygonpoints, point);

if (is\_in)

{

cout << "true." << endl;

}

else

{

cout << "false." << endl;

}

return 0;

}

визначення чи належить задана точка опуклому многокутнику.

Завдання №4 VNS Lab 7 Task 2

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

void cyclicShiftLeft(vector<int> &arr)

{

if (arr.empty() || arr[0] % 2 != 0)

{

return;

}

int shiftCount = arr[0] % arr.size();

rotate(arr.begin(), arr.begin() + shiftCount, arr.end());

}

void cyclicShiftRight(vector<int> &arr)

{

if (arr.empty() || arr.back() % 2 == 0)

{

return;

}

int shiftCount = arr.back() % arr.size();

rotate(arr.rbegin(), arr.rbegin() + shiftCount, arr.rend());

}

int main()

{

vector<int> arr1 = {4, 1, 2, 3, 5};

cyclicShiftLeft(arr1);

cout << "The result of cycle shift left: ";

for (int num1 : arr1)

{

cout << num1 << " ";

}

cout << endl;

vector<int> arr2 = {3, 6, 8, 2, 7};

cyclicShiftRight(arr2);

cout << "The result of cycle shift right: ";

for (int num2 : arr2)

{

cout << num2 << " ";

}

cout << endl;

return 0;

}

циклічний зсув

Завдання №5 Class Practice Task

#include <iostream>

#include <vector>

#include <string>

using namespace std;

void exit() { exit(0); }

int main()

{

vector<string> books = {"Harry Potter", "The Shining", "Lord of the rings", "Sherlock Holmes", "1984", "The picture of Dorian Gray", "Kobzar", "The thing"};

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

int number = books.size();

int action;

linkmenu:

cout << "L I B R A R Y \n";

cout << "Choose the action (write only number): \n";

cout << "1: Show all books \n";

cout << "2: Take a book \n";

cout << "3: Give a book back \n";

cout << "4: quit \n";

cin >> action;

if (action == 1)

{

for (string word : books)

{

cout << word << endl;

}

goto yesno;

}

else if (action == 2)

{

cout << "Choose the book you'd like to borrow: ";

cin.ignore();

string bookborrow;

getline(cin, bookborrow);

bool available = false;

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

{

if (books[i] == bookborrow && availability[i])

{

availability[i] = false;

books.erase(books.begin() + i);

availability.erase(availability.begin() + i);

available = true;

break;

}

}

}

else if (action == 3)

{

cout << "What book would you like to give back?\n";

cin.ignore();

string bookback;

getline(cin, bookback);

books.push\_back(bookback);

availability.push\_back(true);

number++;

cout << "You have returned: " << bookback << endl;

}

else if (action == 4)

{

cout << "\nGoodbye! \n";

exit();

}

yesno:

cout << "\nWould you like to continue? (yes/no) \n";

string answer;

cin >> answer;

if (answer == "yes")

{

goto linkmenu;

}

else if (answer == "no")

{

cout << "\nGoodbye! \n";

exit();

}

else if (answer != "yes" || "no")

{

cout << "You have to write yes or no" << endl;

goto yesno;

}

return 0;

}

бібліотека

Завдання №6 Self Practice Task

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

long long n;

int K = 0;

int V = 0;

int A = 0;

int B = 0;

cin >> n;

char arr[n];

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

{

cin >> arr[i];

}

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

{

if (arr[i] == 'K')

{

++K;

}

else if (arr[i] == 'V')

{

++V;

}

if (K >= 11 && (K - V) >= 2)

{

A++;

K = 0;

V = 0;

}

if (V >= 11 && (V - K) >= 2)

{

B++;

V = 0;

K = 0;

}

if (i == (n - 1))

{

cout << A << ":" << B << endl;

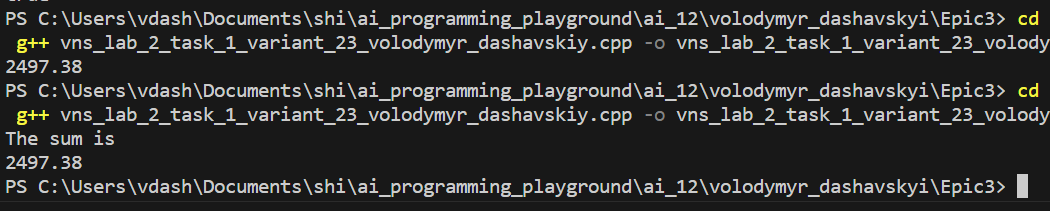
cout << K << ":" << V << endl;

}

}

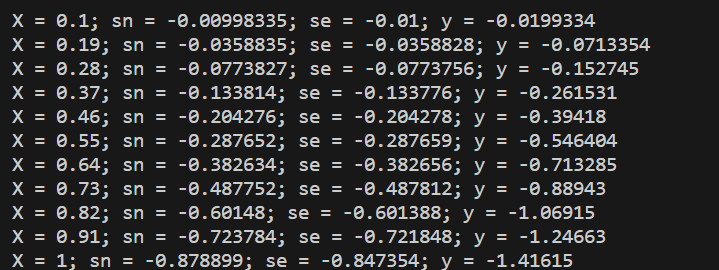
return 0;

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

Завдання №1 VNS Lab 2 Task 1

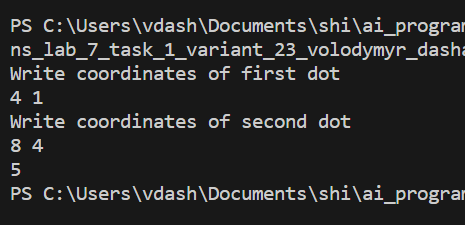
Час затрачений на виконання завдання 25 хв

Завдання №2 VNS Lab 3 Task 1

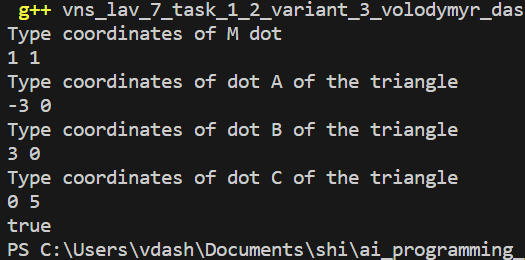


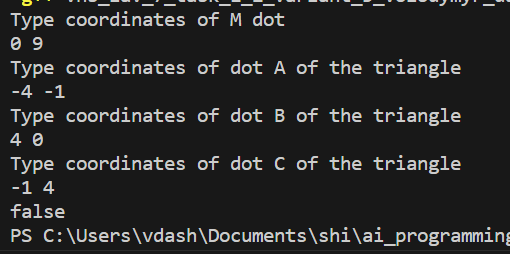
Час затрачений на завдання 35 хв

Завдання №3 VNS Lab 7 Task 1: обрахування довжини відрізка за координатами його точок

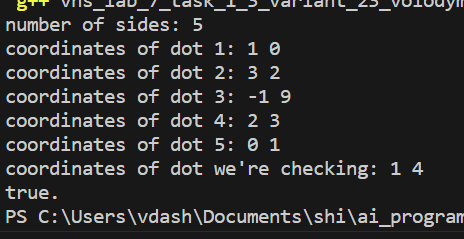


визначення чи знаходиться точка всередині трикутника

(точка всередині трикутника)

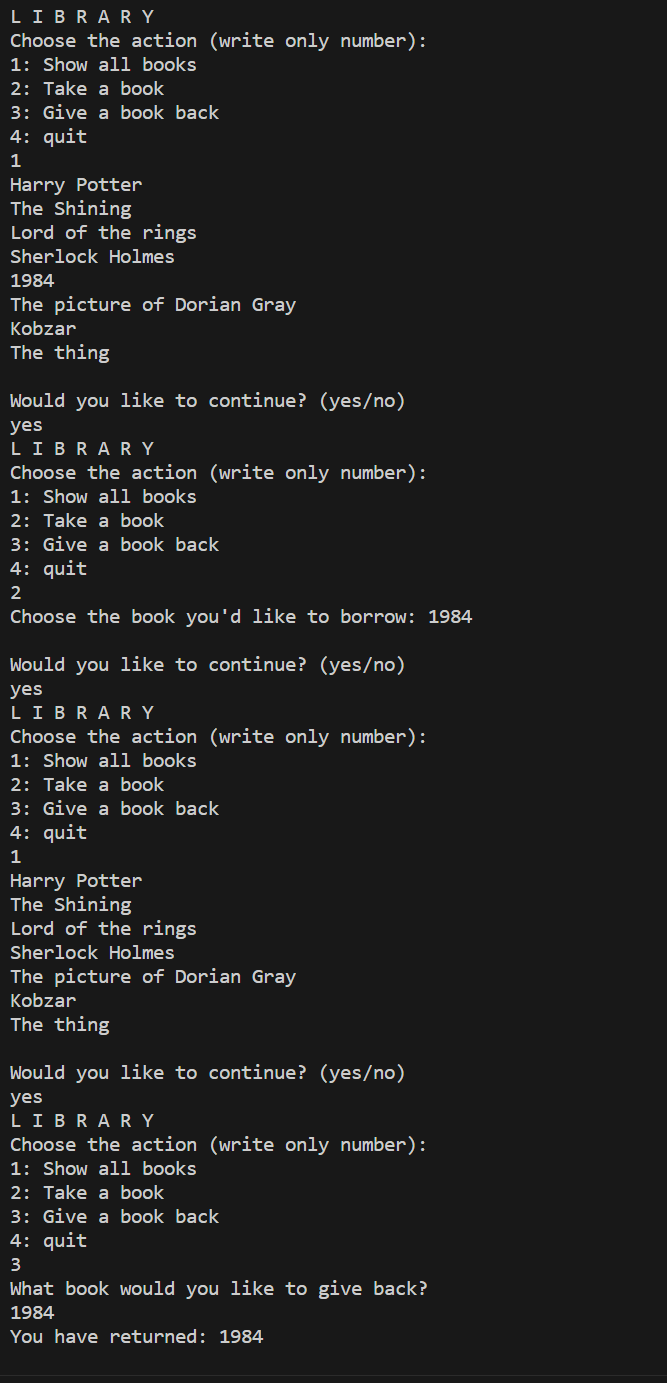
(точка зовні трикутника)

перевірка чи знаходиться точка всередині опуклого многокутника.



Час затрачений на завдання 1 год 55 хв

Завдання №5 Class Practice Task



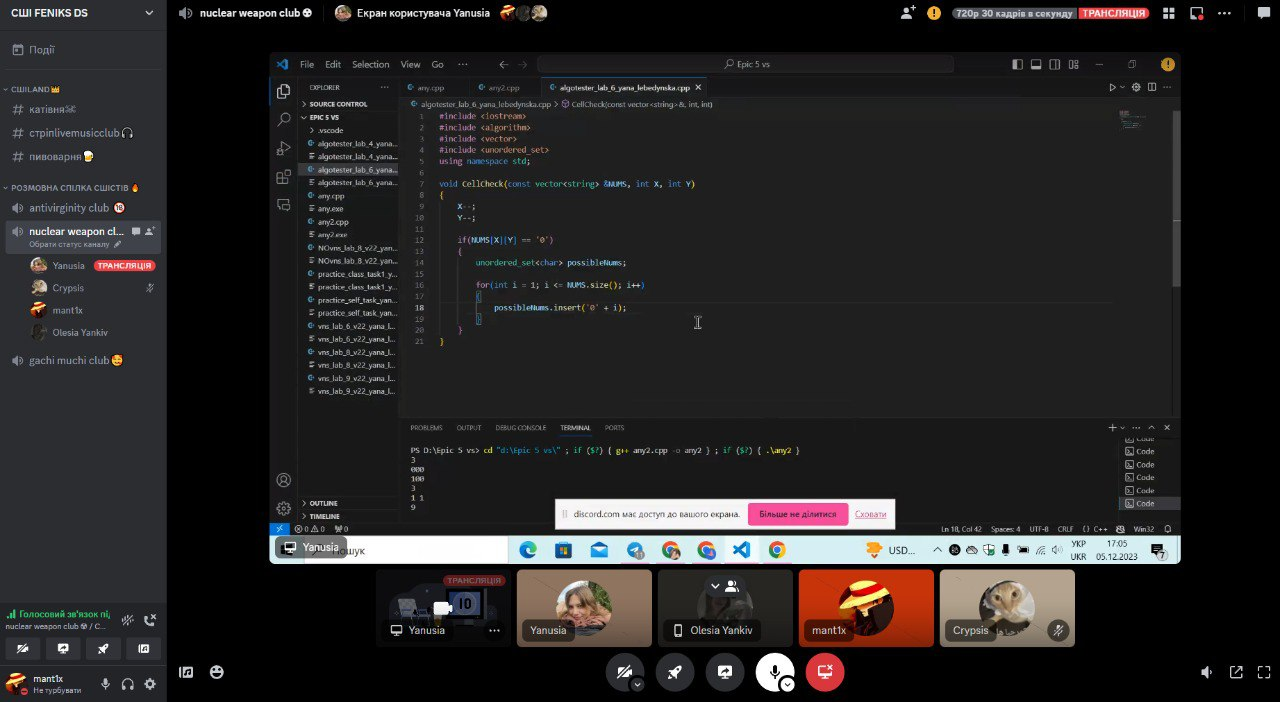
Час затрачений на завдання 3 год 10 хв

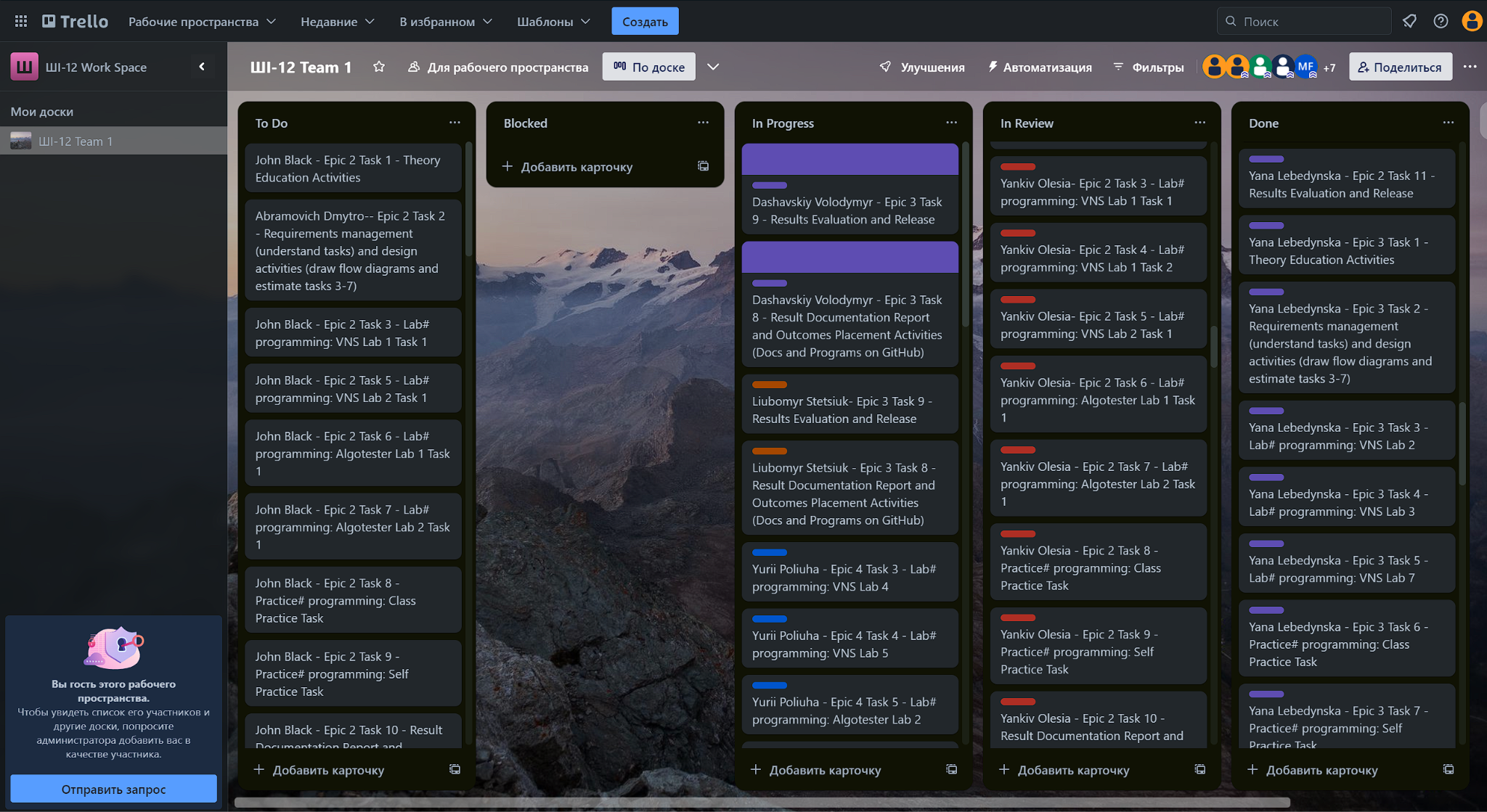
Завдання №6 Self Practice Task

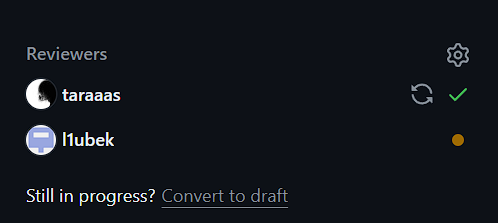
<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/13>

Час затрачений на завдання 50 хв

## **6. Кооперація з командою:**





* 

# **Висновки:**

В цій темі ми навчились оголошувати функції , робити перенавантажені функції, а також робити рекурсивні функції, для зручного обчислення факторіалу, також навчились працювати з різними циклами , такими як for each, for , do(while), while , goto, робити вкладені цикли.