Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**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

***Виконала:***

Студентка групи ШІ-13

Жмуд Анастасія Анатоліївна

# **Тема роботи:**

Обчислення функцій з використанням їхнього розкладу в степеневий ряд. Функції зі змінною кількістю параметрів та перевантаження функцій в С++.

# **Мета роботи:**

Знайомство з організацією функцій зі змінною кількістю параметрів та перевантаженням функцій. Практика в організації ітераційних й арифметичних циклів. Отримання навичок у виборі й використанні операторів С.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Керуючі конструкції програми
* Тема №2: Степеневі ряди (ряди Тейлора)
* Тема №3: Функції за змінною кількістю параметрів. Перевантаження функцій

1. Індивідуальний план опрацювання теорії:

* Тема №1: Керуючі конструкції програми
  + Джерела Інформації
    - <http://cpp.dp.ua/operatory-tsyklu/>
    - <https://kleban.page/courses/csharp-basics/module1/lecture4/>
    - <http://cpp.dp.ua/operatory-keruvannya/>
  + Що опрацьовано:
    - Складені оператори
    - Оператори вибору
    - Оператори циклів
    - Оператори переходу
  + Статус: Ознайомлена
  + Початок опрацювання теми: 10.11.23р.
  + Звершення опрацювання теми: 15.11.23р.
* Тема №2: Степеневі ряди (ряди Тейлора)
  + Джерела Інформації:
    - <https://studfile.net/preview/5186968/page:36/e.net/preview/3012007/page:16/>
    - https://core.ac.uk/reader/84826650
  + Що опрацьовано:
    - Інформація про формули та способи розкладу функції в степеневий ряд
    - Задання точності обчислень розкладу функції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.11.23р.
  + Звершення опрацювання теми: 17.11.23р.
* Тема №3: Функції за змінною кількістю параметрів. Перевантаження функцій
  + Джерела Інформації:
    - <https://kazedu.com/referat/132032/9>
    - <https://studfile.net/preview/5206428/page:2/>
    - https://acode.com.ua/urok-108-perevantazhennya-funktsij/

* + Що опрацьовано:
    - Інформація про перевантаження функцій
    - Принцип роботи функції зі змінною кількістю параметрів
  + Статус: Ознайомлена
  + Початок опрацювання теми: 17.11.23р.
  + Звершення опрацювання теми: 25.11.23р.

# **Виконання роботи:**

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

**Завдання №1 Опрацювання теоретичних матеріалів.**

* 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 Перегляд вимог та проектування**

* Деталі завдання:

Введення завдань в Trello, перегляд задач, та створення математичної моделі. Проектування блок-схем в Draw.io.

- Важливі деталі:

Використання різноманітних фігур для певних дій та особливості проектування flow-charts, що включають в себе алгоритми циклу та функції.

**Завдання №3 VNS Lab 2**

* Варіант завдання №23
* Деталі завдання :

Використовуючи оператор циклу, знайти суму елементів, зазначених у конкретному варіанті.

* Важливі деталі для врахування в імплементації програми:

Необхідно використовувати рекурентну формулу для обчислень

**Завдання №4 VNS Lab 3**

* 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 у степеневий ряд.

* Важливі деталі для врахування в імплементації програми:

Обчислення проводяться у двох випадках, для заданого n та для заданої точності.

**Завдання №5 VNS Lab 7 Task 1**

* Варіант завдання №23
* Деталі завдання :

Розв’язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів.

**Завдання №6 VNS Lab 7 Task 2**

* Варіант завдання №23
* Деталі завдання :

Написати перевантажені функції й основну програму, що їх викликає.

**Завдання №7 Class Practice Task**

* Деталі завдання :

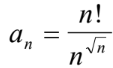
Cтворити просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

* Важливі деталі для врахування в імплементації програми:

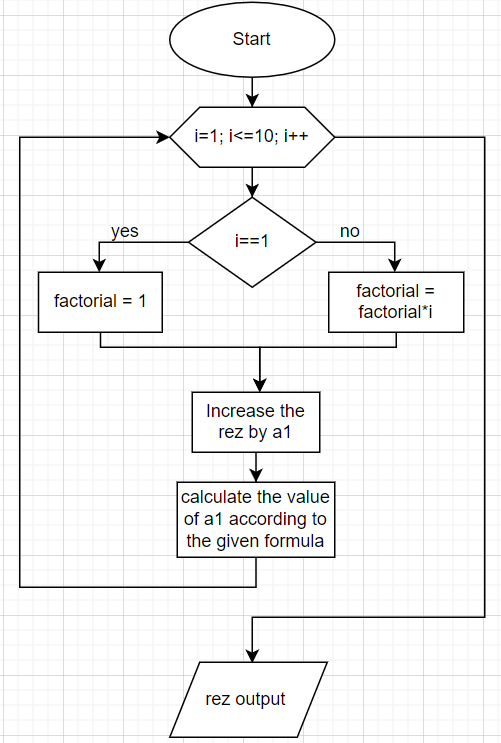
Необхідно використовувати цикли while, for, do while, оператор go to.

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

**Програма №1 VNS Lab 2**

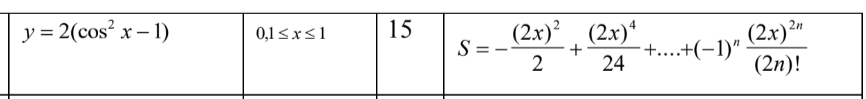
 Знайти суму 10 членів ряду, у якому:

* Блок-схема :

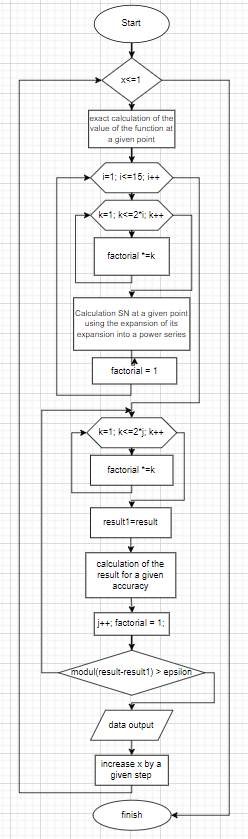


* Планований час на реалізацію: 1 година.
* Важливі деталі для врахування в імплементації:

Необхідно використовувати рекурентну формулу для обчислень

**Програма №2 VNS Lab 3**

* Блок-схема:



* Планований час на реалізацію: 4 години
* Важливі деталі для врахування в імплементації:

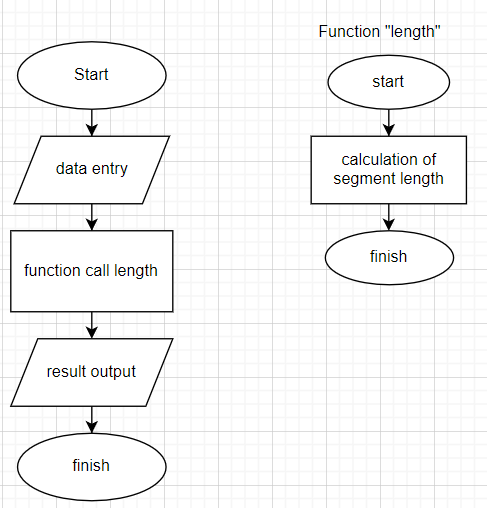
Обчислення проводяться у двох випадках, для заданого n та для заданої точності.

**Програма № 3 VNS Lab 7 Task 1**

Частина 1:

Написати функцію (або макровизначення), що знаходить площу трикутника

за його сторонами.

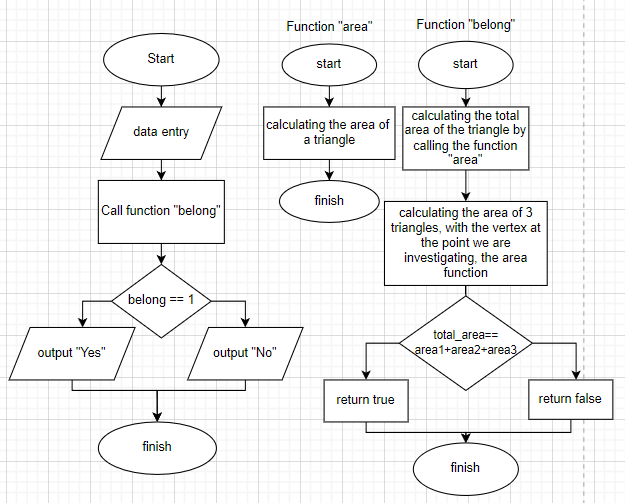
* Блок-схема:
* Планований час на реалізацію: 1 година

Частина 2:

Написати функцію square зі змінною кількістю

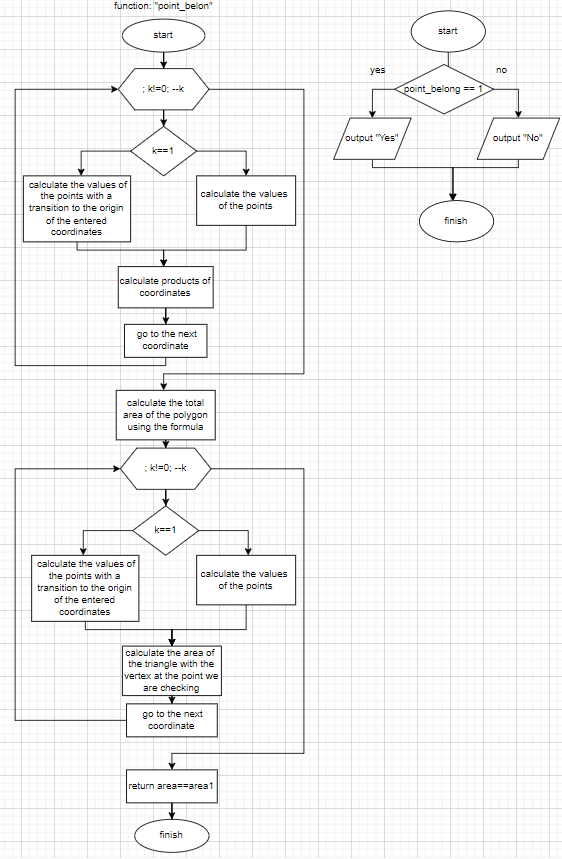
параметрів, що знаходить площу n-кутника за заданими сторонами.

- Блок-схема:



- Планований час на реалізацію: 4 години

Частина 3:

- Блок-схема:

- Планований час на реалізацію: 9 годин

**Програма №4 VNS Lab 7 Task 2**

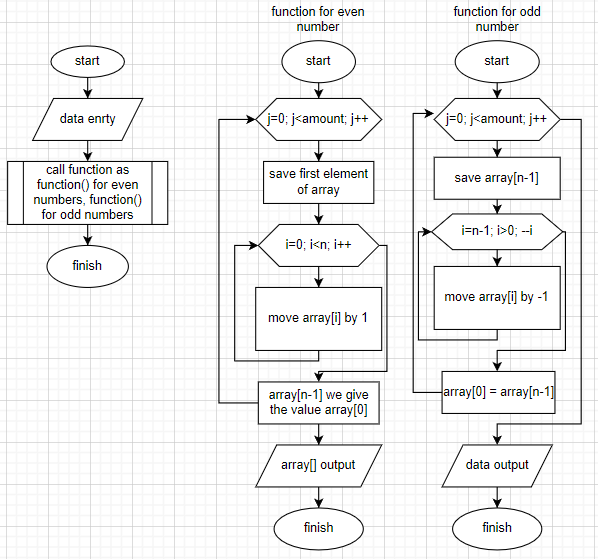
Написати перевантажену функцію

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.

- Блок-схема:

- Планований час на реалізацію: 2 годин

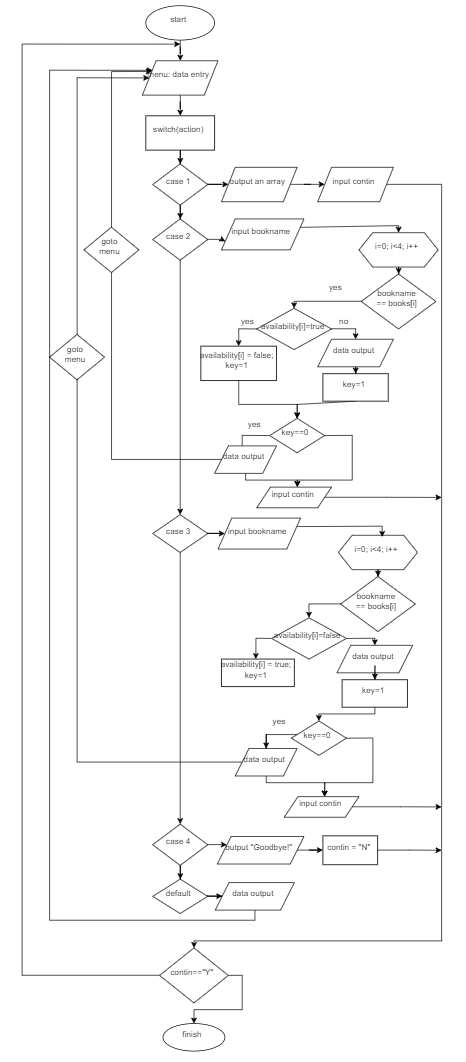
**Програма №5 Class Practice Task**

Створення простої програми керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

Програма повинна вміти

* Перерахувати всі книги.
* Дозволити взяти книгу (за наявності).
* Дозволити повернення книги.

**-** Блок-схема:

****

- Планований час на реалізацію: 4 години

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

**Завдання № 1**

**VNS Lab 2**

#include<iostream>

#include<math.h>

using namespace std;

int main(){

    double rez, a1=1, factorial=0;

    for(int i=1; i<=10; i++){

        if(i==1){

            factorial=1;

        }

        else{

            factorial=factorial\*i;

        }

        rez+=a1;

        a1=(factorial/pow(i+1,(pow(i+1, 0.5)-1)));

    }

    cout<<rez;

}

Спочатку потрібно підключити усі необхідні для роботи програми бібліотеки, у цьому випадку це «iostream» та «math.h». Після, оголошуємо змінні типу double. Далі в циклі оголошуємо умову і рахуємо факторіал, в тому ж циклі рахуємо суму за формулою та додаємо її до попередньої суми. Після виходу з циклу виводимо результат.

**Завдання № 2**

**VNS Lab 3**

#include <iostream>

#include <cmath>

using namespace std;

int main(){

    double x\_step = 0.09, epsilon = 0.0001, x = 0.1, factorial=1;

    long double result=0, y=0, j=1, SN=0, result1;

    while(x<=1){

     y=2\*(pow(cos(x), 2)-1);

     SN=0;

     j=1;

     result=0;

     for (int i = 1; i <= 15; i++) {

      for(int k=1; k<=2\*i; k++){

        factorial \*= k;

      }

      SN += (pow((-1), i)\*(pow(2\*x, 2\*i)/factorial));

      factorial = 1;

     }

     do{

          for(int k=1; k<=2\*j; k++){

          factorial \*= k;

          }

          result1 = result;

          result+=(pow((-1), j)\*(pow(2\*x, 2\*j)/factorial));

          j++;

          factorial = 1;

     }

       while(abs(result-result1)>epsilon);

        cout<<"x = "<<x<<"   "<<"SN = "<<SN<<"   "<<"SE = "<<result<<"   "<<"Y = "<<y<<endl;

        x+=x\_step;

    }

}

Першим етапом роботи для виконання даної задачі є підключення бібліотек «cmath» та «iostream». Далі ініціалізуємо потрібні змінні в типах double та long double. Після, в циклі рахуємо точне значення функції для даного x. Наступним етапом є цикл в якому ми обраховуємо факторіал та суму для заданого n. Далі, в циклі do while, обчислюємо факторіал та суму для заданої точності. Після, здійснюється вивід результатів та збільшуємо х на заданий крок.

Завдання №3

**VNS Lab 7 Task 1**

Частина 1:

#include<iostream>

#include <cmath>

using namespace std;

double length(double x1,double y1,double x2,double y2){

    return sqrt(pow(x2-x1, 2)+ pow(y2-y1, 2));

}

int main(){

    double x1, x2, y1, y2;

    cout<<"enter the coordinates of the first point:"<<endl;

    cin>>x1>>y1;

    cout<<"enter the coordinates of the second point:"<<endl;

    cin>>x2>>y2;

    cout<<length(x1, y1, x2, y2);

}

Спочатку потрібно підключити усі необхідні для роботи програми бібліотеки, у цьому випадку це «iostream» та «сmath». Далі оголошуємо функцію length з параметрами х1, у1, х2, у, що обчислює довжину, за формулою та повертає значення. В функції main вводимо усі необхідні координати та виводимо обчислення, проведені для цих координат в функції length.

Частина 2:

#include <iostream>

#include<cmath>

using namespace std;

     double area(int x1, int y1, int x2, int y2, int x3, int y3) {

        return abs(0.5\*((x1\*y2+x2\*y3+x3\*y1)-(y1\*x2+y2\*x3+y3\*x1)));

    }

    bool belong(int x, int y, int x1, int y1, int x2, int y2, int x3, int y3) {

     double total\_area = area(x1, y1, x2, y2, x3, y3);

     double area1 = area(x, y, x2, y2, x3, y3);

     double area2 = area(x1, y1, x, y, x3, y3);

     double area3 = area(x1, y1, x2, y2, x, y);

    if(total\_area==area1+area2+area3){

        return true;

    } else{

        return false;

    }

}

int main() {

    int x1, y1;

    cin>>x1>>y1;

    int x2, y2;

    cin>>x2>>y2;

    int x3, y3;

    cin>>x3>>y3;

    int x, y;

    cin>>x>>y;

    if (belong(x, y, x1, y1, x2, y2, x3, y3)==1) {

      cout << "YES" << endl;

    } else {

      cout << "NO" << endl;

    }

    return 0;

}

Першим етапом роботи для виконання даної задачі є підключення бібліотек «cmath» та «iostream». Оголошуємо функцію area(), що обчислює площу трикутника за координатами його вершин та повертає отримане значення. Наступним кроком є оголошення функції belong(), в якій ми викликаємо функцію area() для обчислення загальної площі трикутника та площі менших трикутників з вершиною в точці, яку ми перевіряємо. Після, перевіряємо чи сума площ менших трикутників рівна площі більшого трикутника, то belong() повертає значення true, в іншому випадку – false.

В функції main вводимо координати точок та викликаємо функцію belong(), і в умові if перевіряємо чи належить точка трикутнику, чи ні, та виводимо результат відповідно до виконання чи не виконання умови.

Частина 3:

#include <iostream>

#include <cmath>

using namespace std;

bool point\_belong(int k, int x, int y,  int x1, int y1, int x2, int y2, int x3, int y3,...){

    int\* p=&y;

    int\* p1=&y;

    int s=0, s1=0;

    double area=0.0, area1=0.0;

    int x0, y0, x01,y01, k1=k;

    for(; k!=0; --k){

        if(k==1){

            x0=\*(++p);

            y0=\*(2+p1);

            y01=\*(1+p);

            x01=\*(++p1);

        }

        else{

            x0=\*(++p);

            y0=\*(3+p);

            y01=\*(1+p);

            x01=\*(2+p);

        }

        s+=x0\*y0;

        s1+=y01\*x01;

        p++;

    }

    p=&y;

    area=0.5\*abs(s-s1);

    x0 =0, y0=0, y01=0, x01=0;

    for(; k1!=0; --k1){

        if(k1==1){

            x0=\*(p1);

            y0=\*(++p1);

            x01=\*(++p);

            y01=\*(1+p);

        }

        else{

        x0=\*(p1);

        y0=\*(++p1);

        x01=\*(1+p1);

        y01=\*(2+p1);}

        p1++;

        area1+=0.5\*abs((x\*y0+x0\*y01+x01\*y)-(y\*x0+y0\*x01+y01\*x));

    }

    return area==area1;

}

int main(){

    if(point\_belong(5, 0,0,-2,-2,1,-3,3,0,1,3,-1,2)){

        cout<<" Point (0;0) belong to this polygon";

    }

    else{

        cout<<" Point (0;0) don`t belong to this polygon";

    }

}

Спочатку потрібно підключити усі необхідні для роботи програми бібліотеки, у цьому випадку це «iostream» та «сmath». Потім, оголошуємо функцію point\_belong() зі змінною кількістю параметрів Вона отримує на вхід кількість вершин k та координати (x, y) точки, а також координати вершин багатокутника (x1, y1), (x2, y2), (x3, y3), ... У функції використовуються вказівники для доступу до координат вершин та обчислення площі багатокутника за допомогою методу "геометричної суми". Використовується формула для обчислення площі багатокутника, яка базується на сумі площ трикутників, утворених вершинами полігона та тестової точки (x, y). Далі перевіряється чи дорівнює сума площ менших трикутників площі всього многокутника і повертає відповідне значення.

В функції main() викликаємо функцію point\_belong() і відповідно до значення яке вона поверне виводимо результат

**Завдання №4**

**VNS Lab 7 Task 2**

#include <iostream>

using namespace std;

int function(int n ,int arr[], int c) {

    int amount = arr[0];

        for (int j = 0; j < amount ; ++j) {

            int temp = arr[0];

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

                arr[i] = arr[i + 1];

            }

            arr[n - 1] = temp;

        }

        for(int i=0; i<n; ++i){

            cout<<arr[i];

        }

}

int function(int n, int arr[], double c) {

        int amount = arr[n-1];

        for (int j = 0; j < amount ; ++j) {

            int temp = arr[n-1];

            for (int i = n-1; i >0 ; --i) {

                arr[i] = arr[i-1];

            }

            arr[0] = temp;

        }

        for(int i=0; i<n; ++i){

            cout<<arr[i];

        }

}

int main() {

    int n;

    cin>>n;

    int array1[n] = {4, 1, 2, 3, 2};

    int c=array1[0]%2;

    function(n, array1, c);

    return 0;

}

Перш за все, під’єднаємо бібліотеку «iostream». Оголошуємо функцію «function» для зсуву вліво масиву, що починається на парний елемент, одним із параметрів функції є параметр с (типу int), який перевіряє чи парним є перший елемент. В ній ми оголошуємо перший цикл, зберігаємо перший елемент масиву в змінній temp і оголошуємо другий цикл в якому і відбувається зсув кожного елемента масиву, після виходу з другого циклу присвоюємо останньому значенню масиву значення першого елемента. Останнім кроком є виведення масиву.

Далі, ще раз оголошуємо функцію «function» для масиву, який починається на непарне число виконує циклічний зсув вправо, одним із параметрів функції є параметр с (типу double), який перевіряє чи непарним є перший елемент. В даній функції спочатку зберігаємо значення останнього елемента, потім оголошуємо цикл, в змінній temp зберігаємо останнє значення масиву. Оголошуємо ще один цикл в якому робимо зсув кожного елемента та після виходу з нього присвоюємо першому елементу масиву значення temp. Останнім кроком є виведення масиву.

У функції main() вводимо масив та перевіряємо чи є перше його значення парним чи непарним, і відповідно до цього значення викликаємо потрібну функцію.

**Завдання №5**

**Class Practice Task**

#include<iostream>

#include<string>

using namespace std;

int main(){

    string books[4]= {"Kobzar", "Aeneid", "City", "Moment"};

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

    int key=0;

    string contin;

    do{

    menu:

    key=0;

    int action;

    cout<<"Libruary management\n";

    cout<<"1. List of books\n";

    cout<<"2. Take a book\n";

    cout<<"3. Return book\n";

    cout<<"4. Finish\n\n";

    cout<<"Enter a number: ";

    cin>>action;

    switch (action){

    case 1:{

        for (int i=0; i<4; i++){

            cout<<books[i]<<"   ";

        }

        cout<<"\nDo you want to do something else? (Y/N):  ";

        cin>>contin;

        break;

    }

    case 2:{

        string bookname;

        cout << "\nEnter book name to borrow: ";

        cin>>bookname;

        for(int i=0; i<4; i++){

            if(bookname == books[i]){

                if(availability[i]==true){

                availability[i]=false;

                cout<<"You have successfully taken the book!";

                key = 1;

                }

                else cout<<"Book is already borrowed";

                key = 1;

            }

        }

    }

        if(key==0){

            cout<<"Your input is wrong. Please, try again.\n";

            goto menu;

        }

        cout<<"\nDo you want to do something else? (Y/N):  ";

        cin>>contin;

        break;

    case 3:{

        string bookname;

        cout << "\nEnter book name to return: ";

        cin>>bookname;

        for(int i=0; i<4; i++){

            if(bookname == books[i]){

                if(availability[i]==false){

                availability[i]=true;

                cout<<"You have successfully return the book!";

                key = 1;

                }

                else cout<<"Book is already returned";

                key = 1;

            }

        }

        if(key==0){

            cout<<"Your input is wrong. Please, try again.\n";

            goto menu;

        }

        cout<<"\nDo you want to do something else? (Y/N):  ";

        cin>>contin;

        break;

    }

    case 4:{

        cout<<"Goodbye!";

        contin="N";

        break;

    }

    default:{

        cout<<"Your input is wrong. Please, try again.\n";

        goto menu;

    }

    }

    }

    while(contin=="Y");

}

Після оголошення бібліотек «iostream» і «string», об'являються необхідні змінні, такі як масив books, який містить імена книг, масив availability, що показує доступність кожної книги, змінні для управління вибором користувача тощо. Використовується цикл do-while, щоб створити можливість вибору операцій у користувача. Ставиться мітка menu і користувачу пропонується вибрати дію з меню.

Користувач обирає номер операції, введений через cin, і на основі обраної опції виконується певна дія за допомогою оператора switch:

case 1: Виводиться список усіх книг. Після виведення списку книг користувачу пропонується знову обрати дію.

case 2: Користувач може взяти книгу. Вводиться назва книги, яку користувач хоче взяти. Програма перевіряє, чи книга є у списку. Якщо так, і вона доступна (availability[i]==true), вона видається користувачу (змінюється availability[i] на false). Якщо книга вже взята, виводиться повідомлення про це. Після завершення операції користувачу пропонується знову обрати дію.

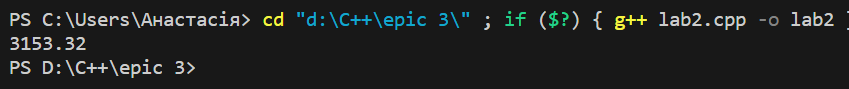
case 3: Користувач може повернути книгу. Вводиться назва книги, яку користувач хоче повернути. Програма перевіряє, чи книга є у списку. Якщо так, і вона вже взята (availability[i]==false), книга повертається (змінюється availability[i] на true). Якщо книга вже повернута, виводиться повідомлення про це. Після завершення операції користувачу пропонується знову обрати дію.

case 4: Програма завершує свою роботу, виводиться повідомлення "Goodbye!" і змінна contin встановлюється в "N", що виводить користувача з циклу do-while.

default: Якщо введено недійсний номер операції, виводиться повідомлення про помилку, користувачу пропонується спробувати знову, і програма переходить до мітки menu знову.

Програма продовжує виконуватися, доки змінна contin дорівнює “Y”.

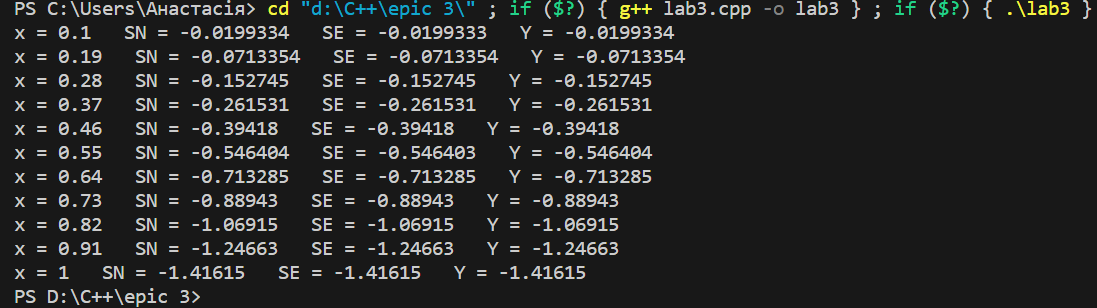
## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1

Приклад виконання завдання VNS Lab 2

Час затрачений на виконання завдання: 1 година

Завдання №2

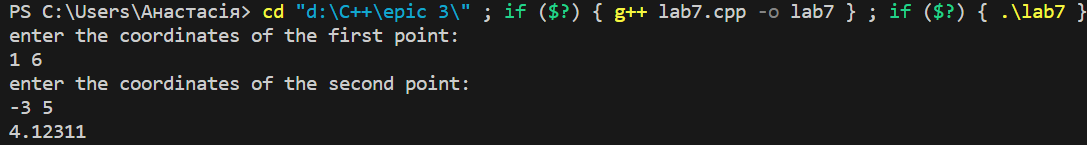


Приклад виконання завдання VNS Lab 3

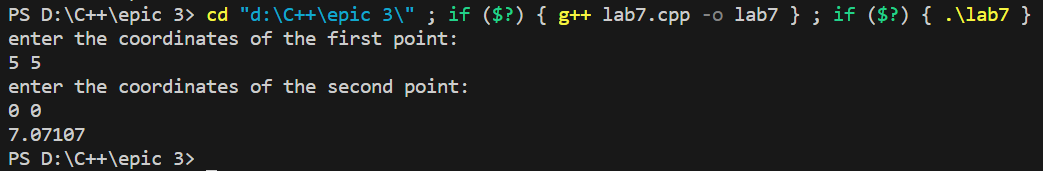
Час затрачений на виконання завдання: 3 години

Завдання №3

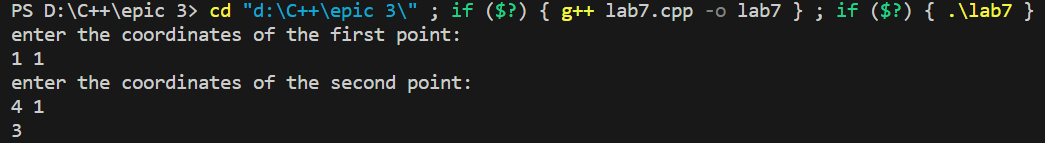
Частина 1:



Тест 1 VNS Lab 7 Task 1 (part 1)

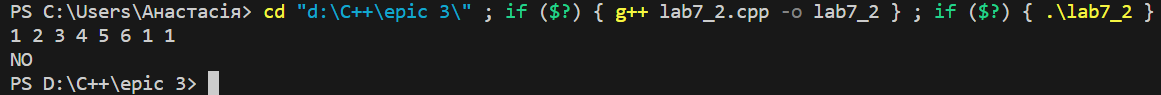


Тест 2 VNS Lab 7 Task 1 (part 1)

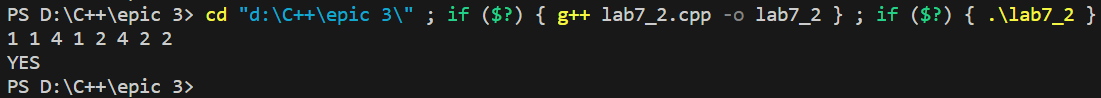


Тест 3 VNS Lab 7 Task 1 (part 1)

Частина 2:

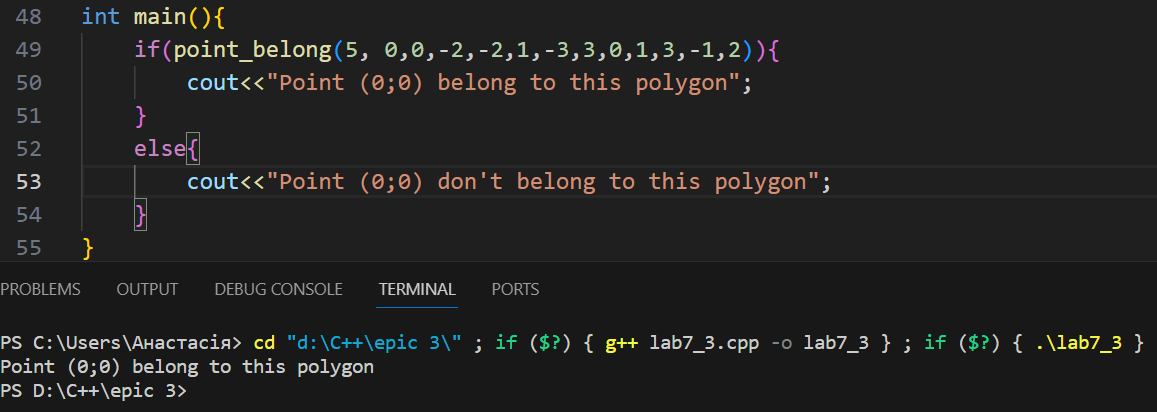


Тест 1 VNS Lab 7 Task 1 (part 2)

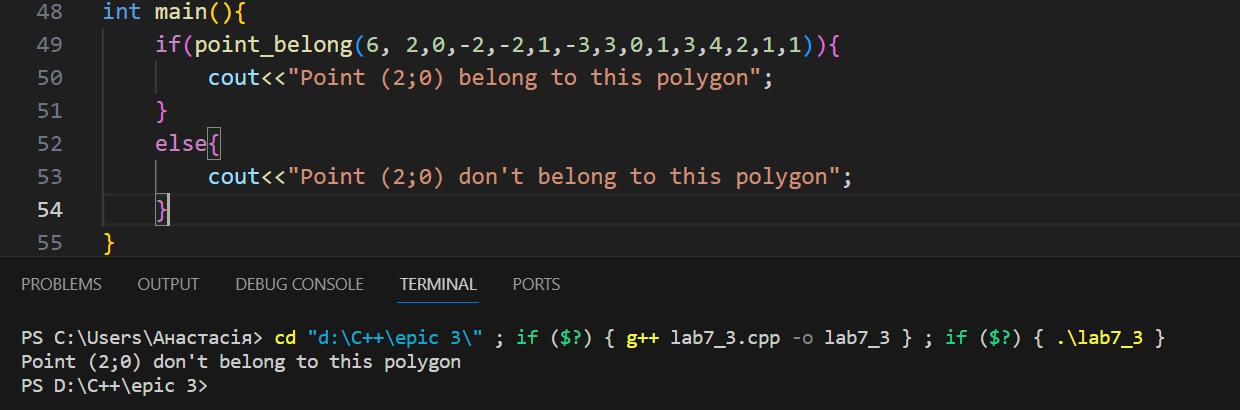


Тест 2 VNS Lab 7 Task 1 (part 2)

Частина 3:

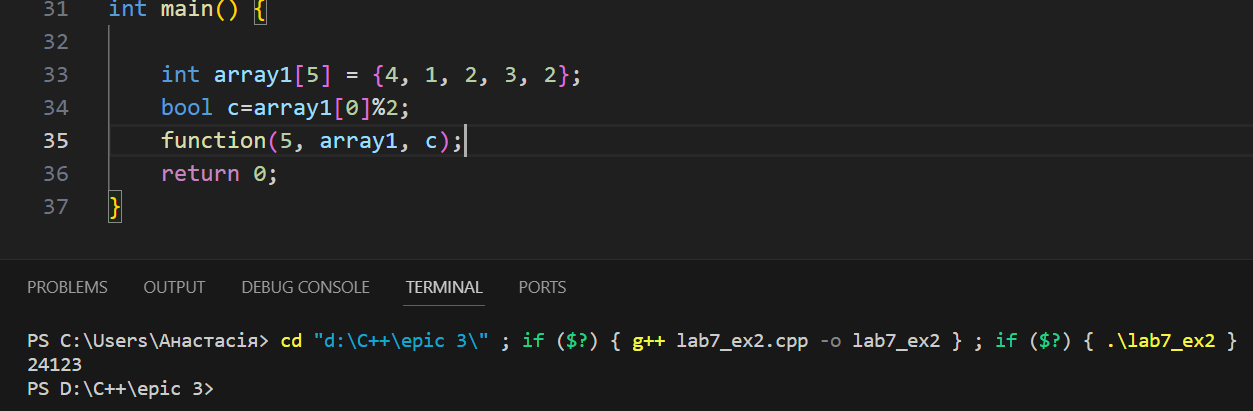


Тест 1 VNS Lab 7 Task 1 (part 3)

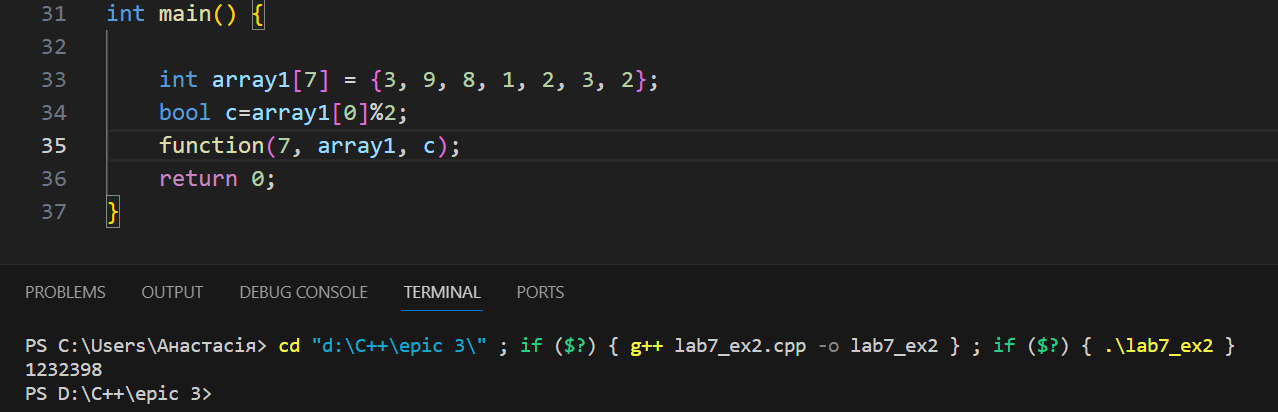


Тест 2 VNS Lab 7 Task 1 (part 3)

Завдання №4

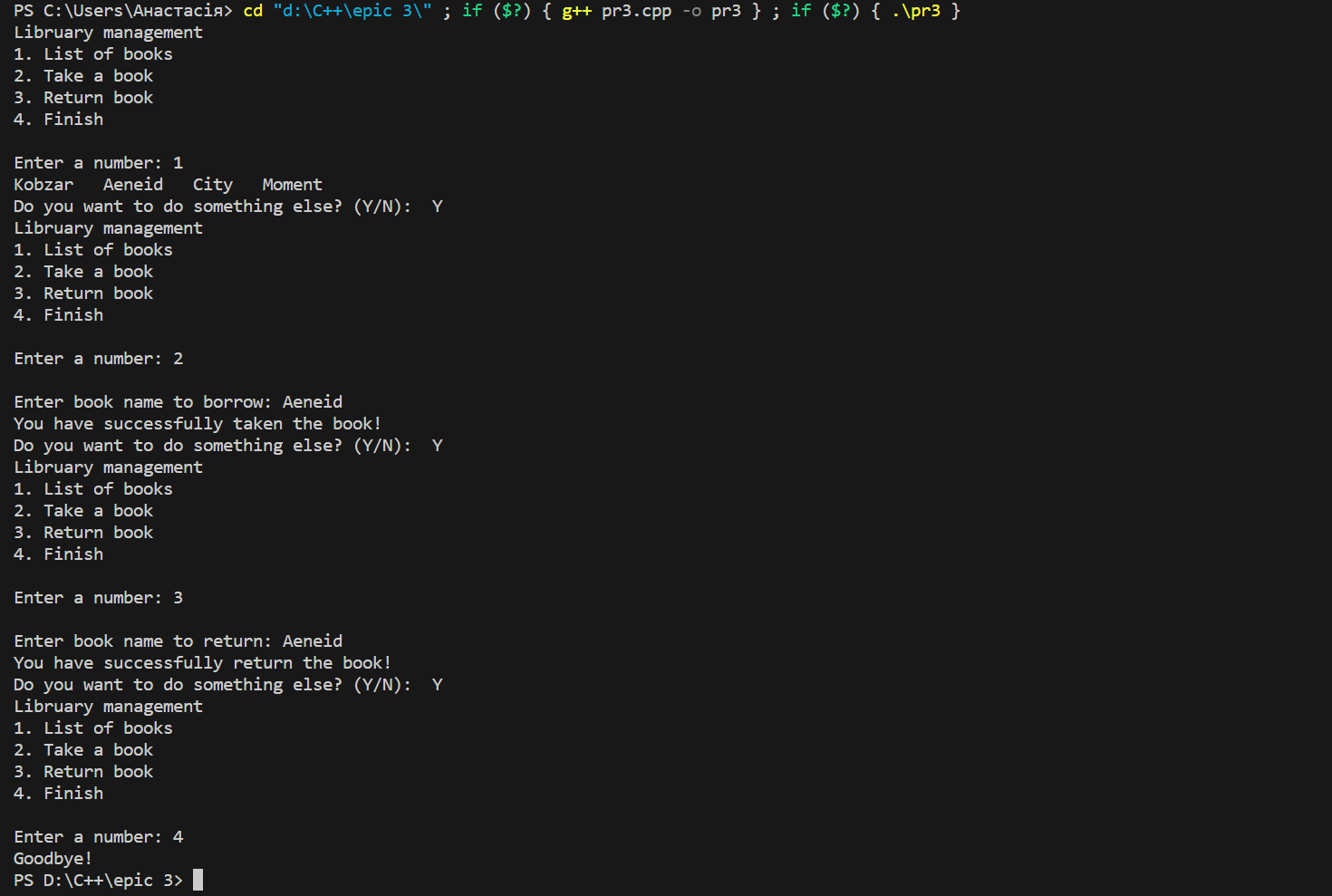


Тест 1 VNS Lab 7 Task 2

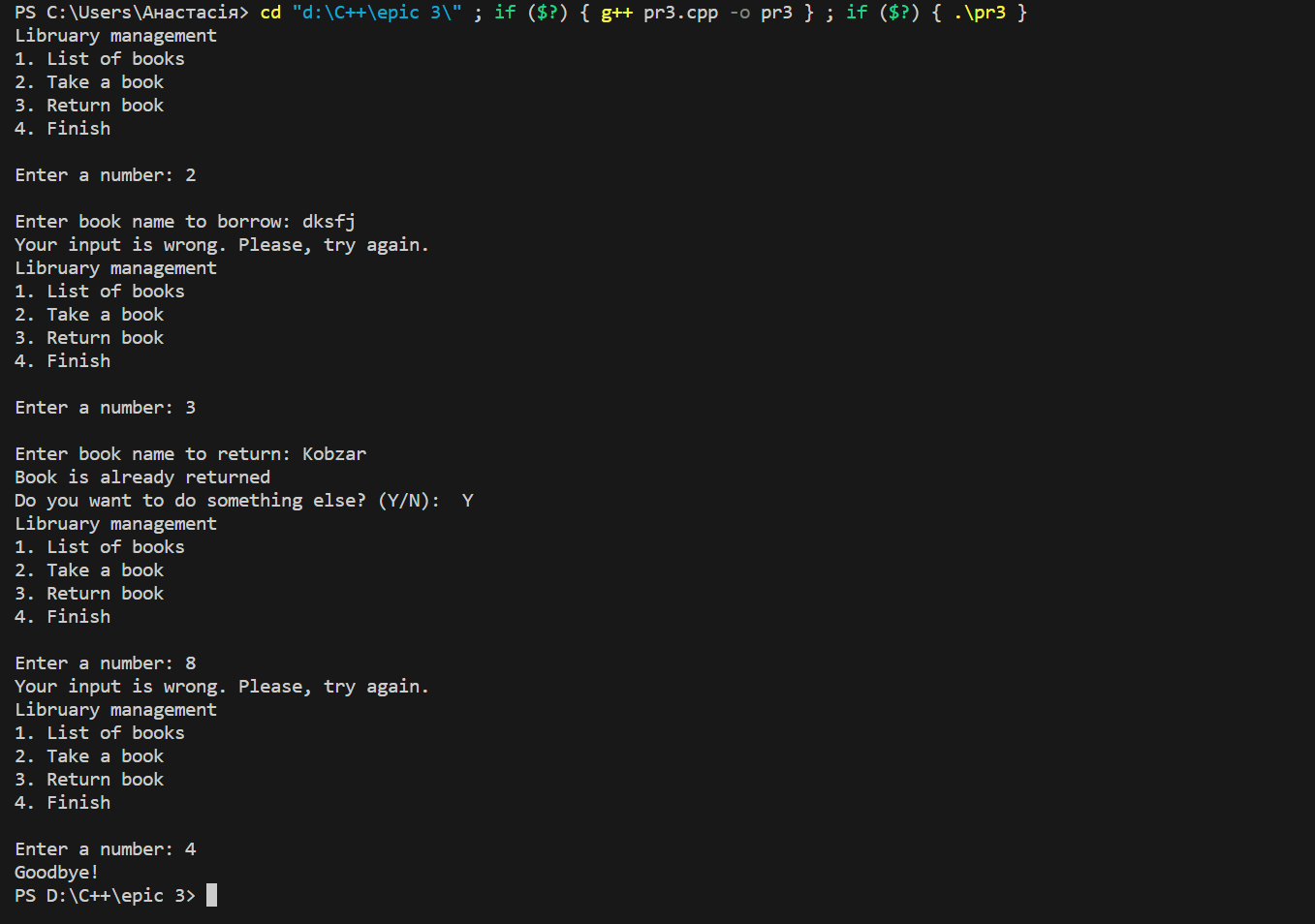


Тест 2 VNS Lab 7 Task 2

Завдання №5



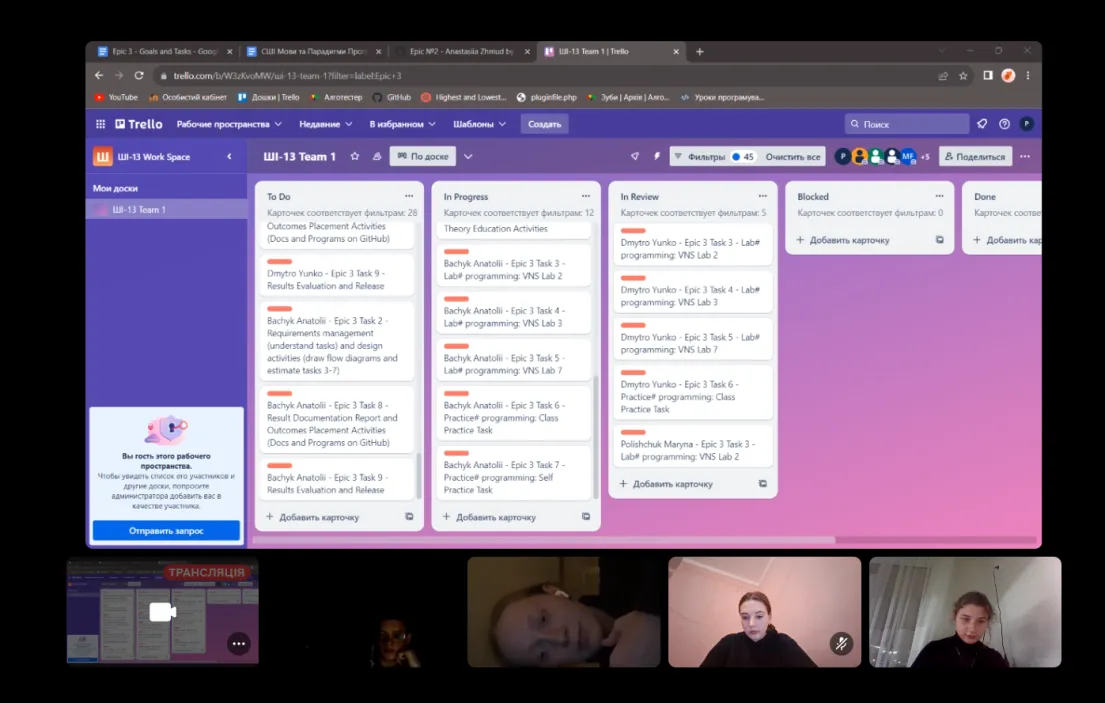
Тест 1 Class Practice Task



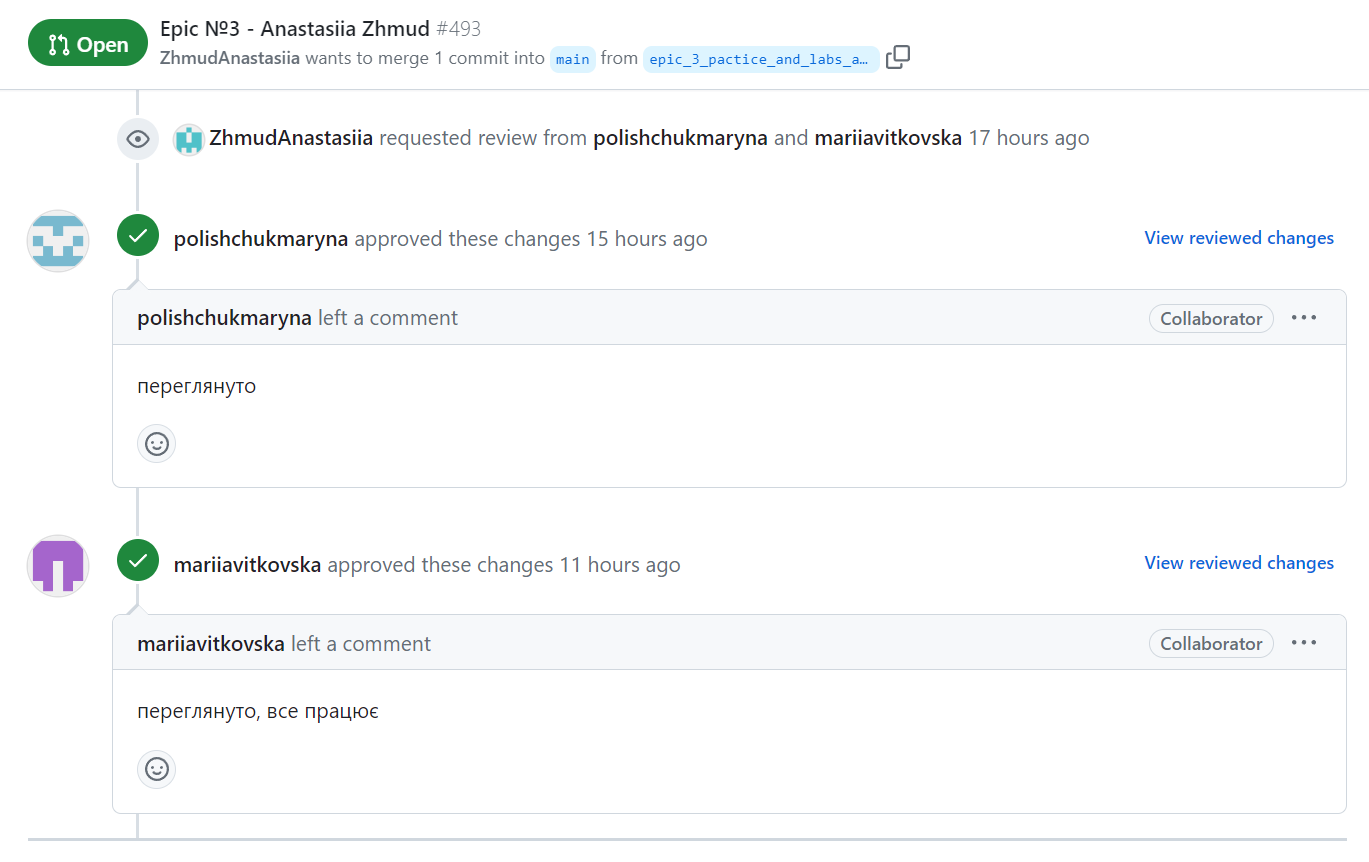
Тест 2 Class Practice Task

## **6. Кооперація з командою:**

* Скрін з зустрічі з командою по обговорення задач Епіку та Скрін прогресу по Трелло:



- Скрін з 2-му коментарями від учасників команди на пул реквесті з Ревю Роботи:



# **Висновки:**

Під час виконання лабораторної роботи я отримала знання щодо функцій зі змінною кількістю параметрів та перевантаження функцій, розширила розуміння особливостей роботи з функціями у мові програмування C++. Отримала навички роботи з циклами та вкладеними циклами. Робота над цією лабораторною роботою сприяла практичному використанню отриманих знань через розв'язання конкретних завдань з програмування, що допомогло краще зрозуміти вказані вище теми.