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(ла):**

Студент(ка) групи ШІ-13

Поліщук Марина Олександрівна

# **Тема роботи:**

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

# **Мета роботи:**

Навчитися працювати з вкладеними циклами, циклами загалом. Зрозуміти як працює перевантаження функцій і для чого воно використовується, зрозуміти, що таке рекурсія.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

-перевантаження функцій

-обчислення результатів із заданою точністю

-рекурсія

-va\_listи

-функції

1. Індивідуальний план опрацювання теорії:

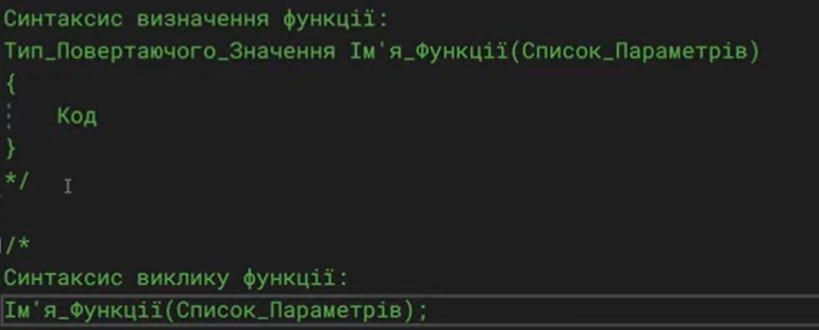
* Тема №1:Перевантаження функцій
  + Джерела Інформації
    - <https://www.youtube.com/watch?v=3KJfisev6SI>
    - методички до робіт
  + Що опрацьовано:
    - Перевантаження функцій потрібне для того, щоб мати змогу зробити кілька функцій, які матимуть однакове ім’я, але матимуть різну кількість параметрів або типів даних і вже при виклику функцій компілятор визначає, яка саме функція з написаних має бути виконана. Наприклад, функція cout перевантажена 17 разів, саме тому ми можемо виводити абсолютно різні значення (різних типів) однією командою.
    - Статус: Ознайомлена
    - Початок опрацювання теми: 22.11.23
    - Звершення опрацювання теми: 26.11.23
* Тема №2: Обчислення результатів із заданою точністю
  + Джерела Інформації
    - <https://youtu.be/Sw5jvi7DcIM?feature=shared>
  + Що опрацьовано:
    - При написанні коду ми використовуємо 10ткові числа, тоді як при компіляції все переводиться в двійкову систему. Справа в тому, що 0.1, наприклад, буде нескінченним періодичним числом, тому у нас виходить, що 0.1+0.2 не буде рівним 0.3, а 0.30000004. Для вирішення цієї проблеми можна використовувати команду std::setprecision() або ж уникати дробових чисел, проводити обчислення з цілими числами.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 29.11.23
  + Звершення опрацювання теми: 29.11.23
* Тема №3: Рекурсія
  + Джерела Інформації:
    - практичні заняття, лекції, методички до робіт
  + <https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%81%D1%96%D1%8F>
  + Що опрацьовано:
    - Рекурсія– це коли через формулу виявляється залежність членів. Тобто, член обчислюється не першим елементом, а попереднім (коли наступний випливає з попереднього)
  + Статус: Ознайомлена
  + Початок опрацювання теми: 25.10.23
  + Звершення опрацювання теми: 27.11.23
* Тема №4 va\_listи
  + Джерела Інформації:
    - https://www.youtube.com/watch?v=3iX9a\_l9W9Y
    - лекційні, практичні заняття, методички до робіт
  + Що опрацьовано:
* Va\_list (**variable** **argument list, змінний список аргументів**)**. Оголошення: тип повертаючої змінної – назва (тип першого елементу, …(вказує на те, що буде ще кілька змінних))**
* va\_start ініціалізує va\_list для доступу до аргументів
* va\_arg отримує значення наступного аргумент
  + Статус: Ознайомлена
  + Початок опрацювання теми: 23.11.23
  + Звершення опрацювання теми: 30.11.23
* Тема №4 функції
  + Джерела Інформації:
    - <https://youtu.be/G8P6SvdqU9s?feature=shared>
    - лекційні, практичні заняття
  + Що опрацьовано: функція-іменований набір коду, який може приймати якусь інформаці. Та повертати результат. Також є можливість використовувати і функції, прописані в інших файлах, головне, щоб вони були в одній папці і в коді, де не прописана функція був вказаний її прототип.
  + 

Рисунок 1 синтаксис визначення та виклику функції



Рисунок 2 синтаксис визначення та виклику функції (коли вони у різних файлах)

* + Статус: Ознайомлена
  + Початок опрацювання теми: 24.11.23
  + Звершення опрацювання теми: 30.11.23

# **Виконання роботи:**

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

Завдання №1 Опрацювання теорії

* Деталі завдання: опрацювання різноманітних матеріалів, аналіз та дослідження відео та статей на задані теми. Вивчення найважливіших моментів.

Завдання № 2 VNS lab 2 task 1

* Варіант завдання: 21
* Деталі завдання: обчислити за формулою значення суми при n= від 1 до13
* Важливі деталі для врахування в імплементації програми: використано цикл, де рекурсивно задається факторіал і обчислення суми. Спочатку присвоюємо сумі значення 0, факторіалу одиницю, після цього при кожній ітерації циклу у нас змінюється число, що підставляється в формулу.
* Планований час на реалізацію: 2 години

Завдання № 3 VNS lab 3 task 1

* Варіант завдання: 21
* Деталі завдання: обчислити значення функції при х від а до б з кроком k, через задану функцію і через ряд Тейлора й Маклорена (40 разів і до заданої точності н-ного члена)
* Важливі деталі для врахування в імплементації програми: розуміння умови задачі, підключено бібліотеку cmath для того, щоб можна було обчислити арктангенс, прописано формулу для ряду.
* Планований час на реалізацію: 9 годин

Завдання №4 VNS lab 7 task 1

* Варіант завдання: 21
* Деталі завдання: написати макровизначення, що знаходить кут за його сторонами. Написати функцію зі змінною к-стю параметрів що знаходить кути н-кутника і main , звертається до функції з кількістю параметрів 3, 9, 11
* Важливі деталі для врахування в імплементації програми: розуміння як працює va\_list, використано формулу для обчислення кута правильного н-кутника.

Завдання №4 VNS lab 7 task 2

* Варіант завдання: 21
* Деталі завдання: написати перевантажені функції й основну програму, що їх викликає

а) для масиву цілих чисел знаходить кількість від’ємних елементів;

б) для рядка знаходить кількість слів, що закінчуються і починаються на ту ж

букву.

* Важливі деталі для врахування в імплементації програми: розуміння як працюють перевантажені функції, їх оголошення, синтаксис, знання того, як розбити рядок на слова

Завдання №5 Class Practice Task

* Деталі завдання: необхідно написати код, який буде давати можливість користувачу дивитись список усіх книг, забирати, повертати їх. Також має бути передбачено вихід (кейс 4) і випадок, коли користувач вводить щось не те.
* Важливі деталі для врахування в імплементації програми: використання циклів, світч кейсу, різні махінації з векторами, адресами елементів.

Завдання №6 Self Practice Task

* Деталі завдання: У єдиному рядку задано два цілих числа n та r — відповідно кількість ярусів та радіус найбільшого яруса в ідеальному торті. У єдиному рядку виведіть дійсне число — радіус найменшого ярусу. Відповідь уважатиметься правильною, якщо її абсолютна чи відносна похибка не перевищуватиме 10^-4.
* Важливі деталі для врахування в імплементації програми: розуміння умови задачі і врахування обмежень.

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

Програма №5 Class Practice

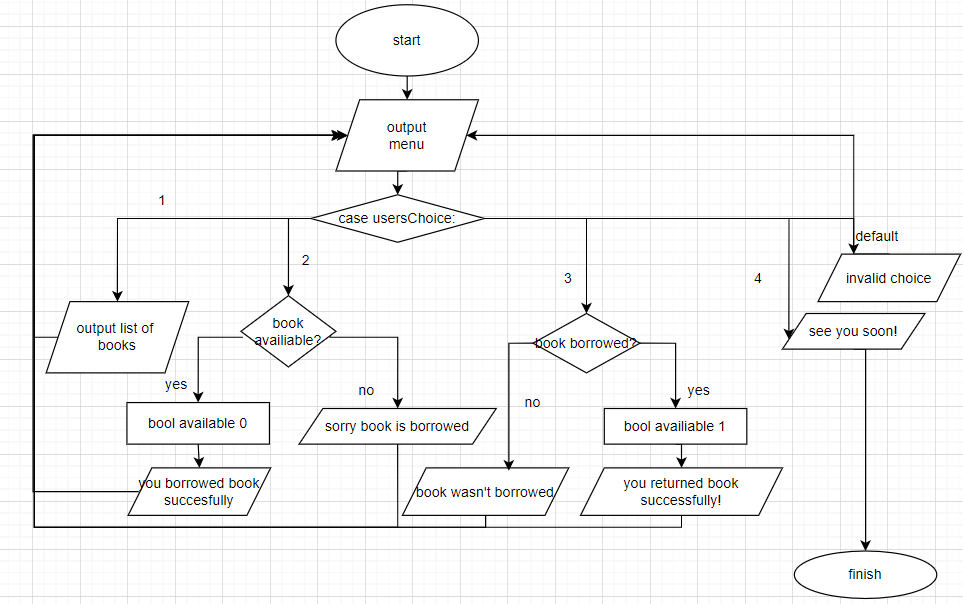
* Блок-схема
* 

Рисунок 3 Class Practice

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

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

Завдання № 1 vns lab2.task1 Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <cmath>

int main ()

{

  unsigned long long factorial=1;

  double sum=0;

  float a=0;

  for (int n=1; n <=13; n++ )

  {

    factorial \*=n;

    a=(log(factorial))/pow(n,2);

    sum +=a;

  }

  std::cout << "result = " << sum << std::endl;

  return 0;

}

## ai\_13/maryna\_polishchuk/epic3\_practice\_and\_labs\_maryna\_polishchuk2/vns\_lab\_2\_task\_1\_maryna\_polishchuk.cpp

## Завдання № 2 vns lab 3 task 1 Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

    double x, a=0.1;

double Y, SN = 0;

    int b=1, k=10;

    for (x=a; x <= b; x+=(b-a)/k ){

    Y = atan(x);

    double result = x;

    for(int i = 1; i <= 40; i++)

    {

        double

        m = pow(-1, i),

        top = pow(x, 2\*i + 1),

        bottom = 2\*i + 1;

        result += m \* (top / bottom);

    }

    double result1 = x;

    for(int i = 1; true; i++)

    {

        double

        m = pow(-1, i),

        top = pow(x, 2\*i + 1),

        bottom = 2\*i + 1;

        result1 += m \* (top / bottom);

        if (abs(m \* (top / bottom)) < 0.0001) {

            break;

        }

    }

    cout << "X: " << x << "      SN: " << result << "    SE: " << result1 << "  Y: " << Y << endl;

    }

}

## ai\_13/maryna\_polishchuk/epic3\_practice\_and\_labs\_maryna\_polishchuk2/vns\_ lab\_ 3\_ task\_1\_maryna\_polishchuk.cpp

## Завдання № 3 vns lab 7 task 1 Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <cmath>

#include <cstdarg>

using namespace std;

#define pi 3.1415926535

#define FIND\_ANGLE\_1(a,b,c) acos(-(a\*a-b\*b-c\*c) / (2.0\*b\*c)) \* (180/pi)

#define FIND\_ANGLE\_2(a,b,c) acos(-0.5 \* (b\*b-a\*a-c\*c) / (a\*c)) \* (180/pi)

#define FIND\_ANGLE\_3(a,b,c) acos(-(c\*c-a\*a-b\*b) / (2.0\*a\*b)) \* (180/pi)

double angles(double firstSide, ...)

{

    va\_list args;

    va\_start(args, firstSide);

    int count = 1;

    while (va\_arg(args, int) != 0) {

        count++;

    }

    double angle = (180 \* (count-2)) / count;

    return angle;

}

int main()

{

    int a = 3, b = 4, c = 5;

    if(a+b>c && a+c>b && b+c>a)

    {

        cout << "angle1 = " << FIND\_ANGLE\_1(a,b,c) << endl;

        cout << "angle2 = " << FIND\_ANGLE\_2(a,b,c) << endl;

        cout << "angle3 = " << FIND\_ANGLE\_3(a,b,c) << endl;

    }

    double angleOfFigure = angles(5, 5, 5, 0);

    cout << "angle of 3angle: " << angleOfFigure << endl;

    double angleOfFigure2 = angles(5, 5, 5, 5, 5, 5, 5, 5, 5, 0);

    cout << "angle of 9angle: " << angleOfFigure2 << endl;

    double angleOfFigure3 = angles(5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 0);

    cout << "angleof 11angle: " << angleOfFigure3 << endl;

}

**ai\_13/maryna\_polishchuk/epic3\_practice\_and\_labs\_maryna\_polishchuk2/vns\_lab\_7\_task\_1\_maryna\_polishchuk.cpp**

Завдання № 4 **vns lab7 task 2** Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <string>

using namespace std;

int FindNegativeNum(int nums[], int length)

{

    int output = 0;

    for(int i = 0; i < length; i++)

    {

        if(nums[i] < 0)

            output++;

    }

    return output;

}

int FindNegativeNum(double nums[], int length)

{

    int output = 0;

    for(int i = 0; i < length; i++)

    {

        if(nums[i] < 0)

            output++;

    }

    return output;

}

void FindLatters(string str)

{

    str += ' ';

    int result = 0;

    bool isWordStarted = false;

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

    {

        int temp; //індекс початку слова

        // char currentChar = str[i];

        if(str[i] != ' ' && !isWordStarted)

        {

            temp = i;

            isWordStarted = true;

        }

        if(str[i] == ' ' && isWordStarted)

        {

            if(str[temp] == str[i-1])

                result++;

            isWordStarted = false;        }

    }

    cout << "last and first letter are equel: " << result << endl;

}

int main()

{

    int arr[] = {1, 5, -4, 6, -9, -5, -2},

        length = sizeof(arr) / sizeof(arr[0]);

    int result = FindNegativeNum(arr, length);

    cout << "amount of negative numbers: " << result << endl;

    string str = "   sdsds fkwe f q  sddfsm";

    FindLatters(str);

    return 0;

}

}

**ai\_13/maryna\_polishchuk/epic3\_practice\_and\_labs\_maryna\_polishchuk2/vns\_lab\_7\_task\_2\_maryna\_polishchuk.cpp**

## Завдання № 5 class practice

## Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

#include <cstdio>

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main() {

    vector<string> books = {"HarryPotter", "Moriarty the patriot", "Sherlock Holmes", "1984", "Murder on the Orient Express"};

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

    int usersChoice;

    string name;

while (true){

    menu:

    cout << "\n";

    cout << "\tLibrary Managment\n";

    cout << "\tMenu\n";

    cout << "1. List all books\n";

    cout << "2. Borrow a book\n";

    cout << "3. Return a book\n";

    cout << "4. Exit\n";

    cout << "\nWhat you want to do? ";

    cin >> usersChoice;

    switch(usersChoice) {

        case 1:

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

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

            }

             goto menu;

        case 2:

            cin.ignore();

            cout << "Enter a book name you would like to borrow: ";

            getline(cin, name);

            for (const auto &book : books) {

                if (book == name) {

                    int id = &book - &books[0];

                    if (available[id]) {

                        available[id] = false;

                        cout << "You borrowed the book successfully!\n";

                        goto menu;

                    } else {

                        cout << "Sorry, book is already borrowed ^\_^\n";

                    }

                }

            }

            break;

        case 3:

            cin.ignore();

            cout << "Enter the book name you would like to return: ";

            getline(cin, name);

            for (const auto &book : books) {

                if (book == name) {

                    int id = &book - &books[0];

                    if (!available[id]) {

                        available[id] = true;

                        cout << "You returned the book successfully!\n";

                        goto menu;

                    } else {

                        cout << "Book wasn't borrowed ^\_^\n";

                        goto menu;

                    }

                }

            }

            break;

        case 4:

            cout << "See you soon!";

            return 0;

        default:

            cout << "Invalid choice.\n";

            goto menu;

    }

}

}

ai\_13/maryna\_polishchuk/epic3\_practice\_and\_labs\_maryna\_polishchuk2/class\_ practice\_epic\_3\_polishchuk\_maryna.cpp.cpp

Завдання №6 self practice Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

//торт для петрика

# include <iostream>

#include <cmath>

using namespace std;

int main (){

    int n, r;

    double res=0;

    cin >>n;

    cin >>r;

res=sqrt(pow(r, 2)/n);

cout << res;

    return 0;

}

## ai\_13/maryna\_polishchuk/epic3\_practice\_and\_labs\_maryna\_polishchuk2/self\_practice\_work\_algotester\_task\_0191\_maryna\_polishchuk.cpp

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

Завдання № 1 VNS lab 2 task 1

Деталі по виконанню і тестуванню програми



Рисунок 4: VNS lab 2 task 1

Час затрачений на виконання завдання: 5 годин

Завдання № 2 VNS lab 3 task 1

Деталі по виконанню і тестуванню програми

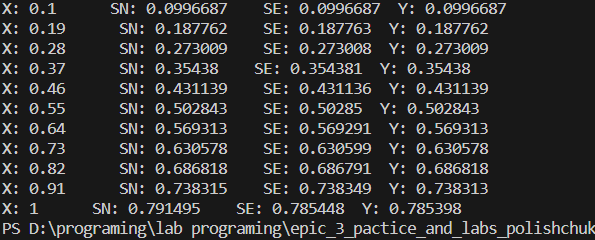


Рисунок 5: VNS lab 3 task 1

Час затрачений на виконання завдання: 3 дні

Завдання № 3 VNS lab 7 task 1

Деталі по виконанню і тестуванню програми

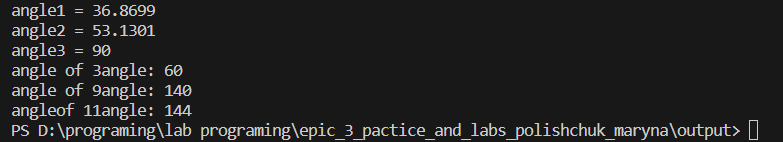


Рисунок 6 VNS lab 7 task 1

Час затрачений на виконання завдання: 3 дні

Завдання № 3 VNS lab 7 task 2

Деталі по виконанню і тестуванню програми

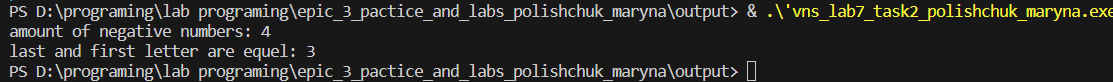


Рисунок 7 VNS lab 7 task 2

Час затрачений на виконання завдання: 3 дні

Завдання № 5 Class Practice

Деталі по виконанню і тестуванню програми

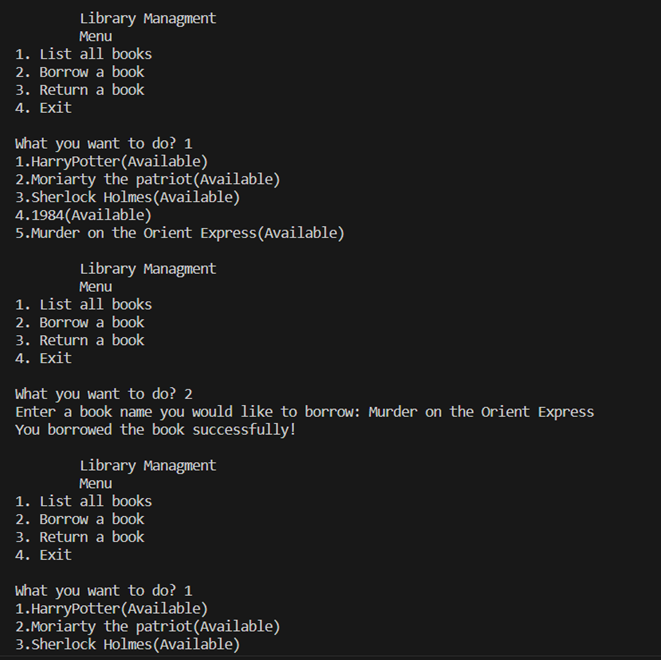


Рисунок 8 Class Practice test

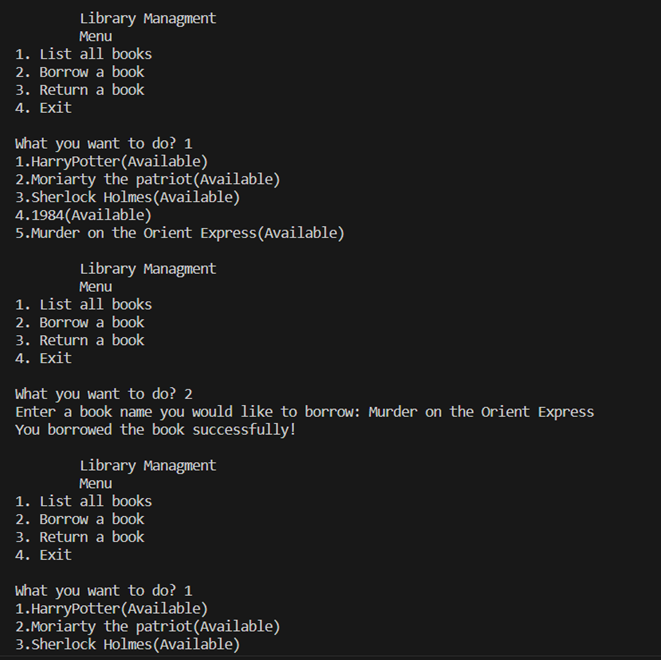


Рисунок 9 Class Practice test

Завдання № 6 Self Practice

Деталі по виконанню і тестуванню програми

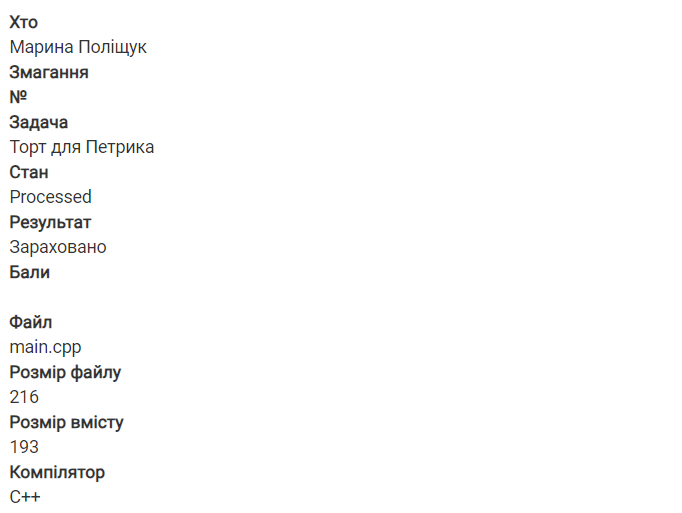
Деталі по виконанню і тестуванню програми 

Рисунок 10 Self Practice algo

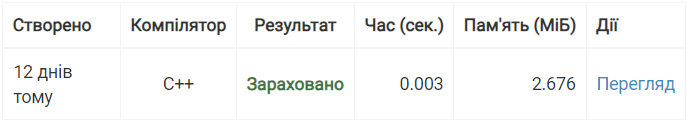


Рисунок 11 Self Practice algotester

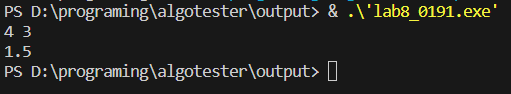


Рисунок 12: Self Practice тест 1

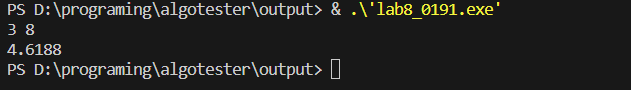


Рисунок 13: Self Practice тест 2

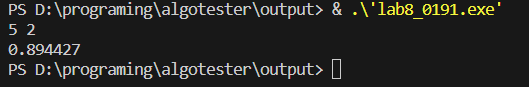


Рисунок 14: Self Practice тест 3

Час затрачений на виконання завдання: 1.5 години

**Кооперація з командою:**

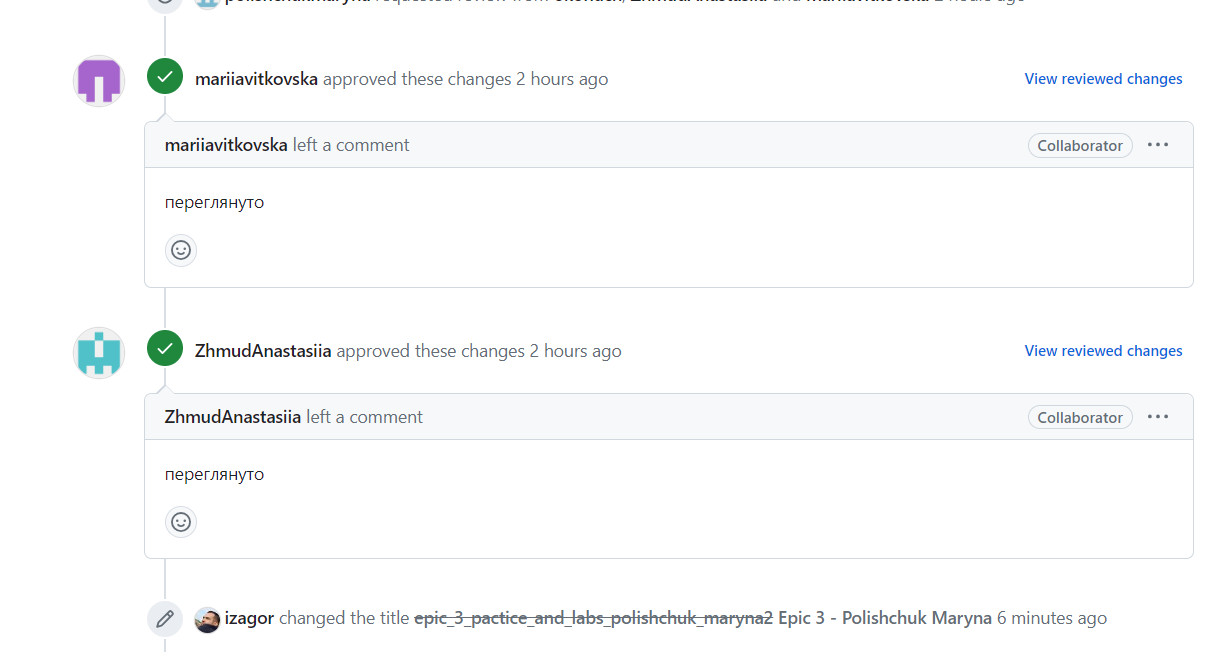


Рисунок 15 підтвердження на гіт від команди

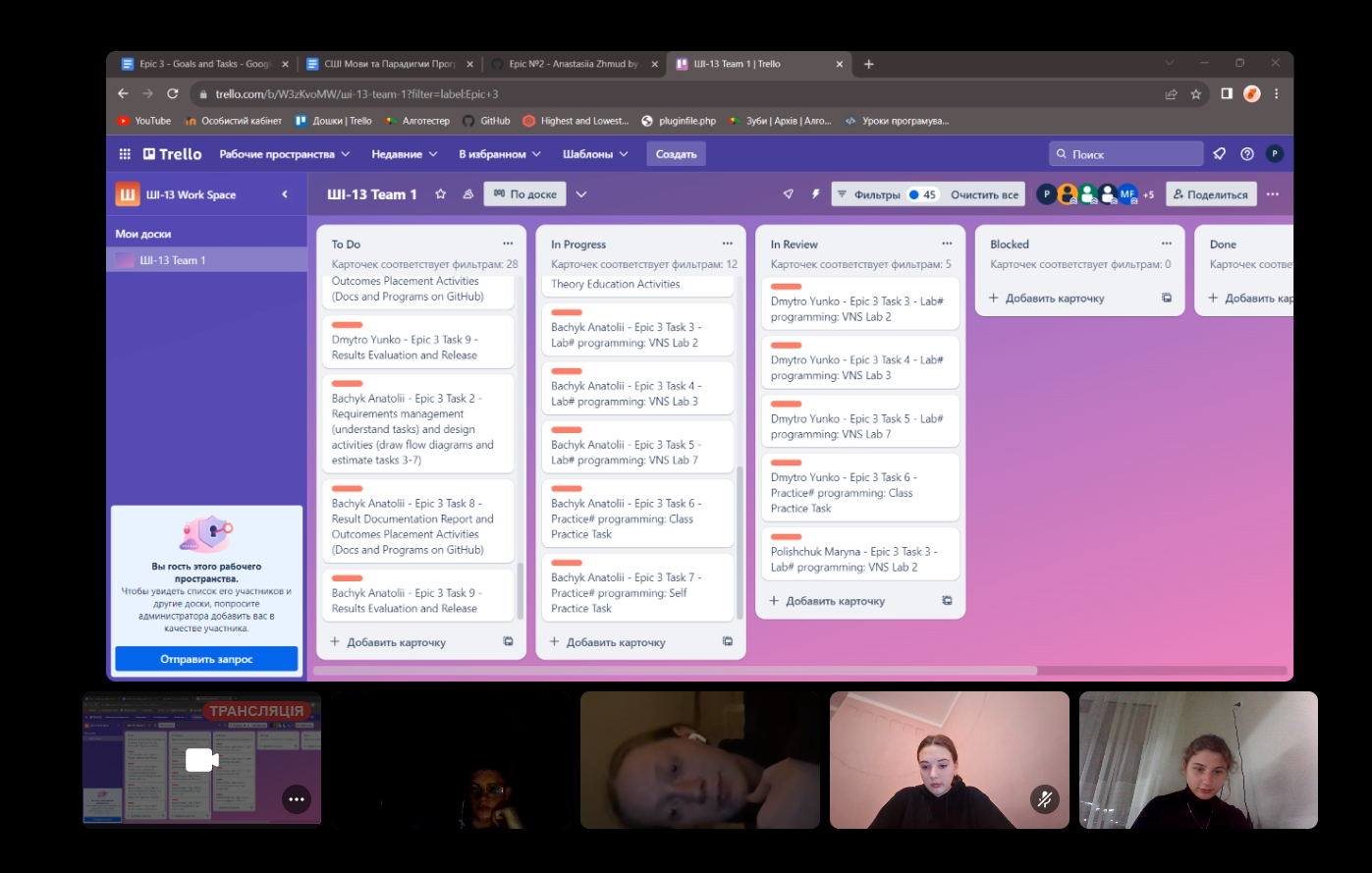


Рисунок 16 мітинг з командою

# **Висновки:**

Навчилися працювати з вкладеними циклами, циклами загалом. Зрозуміли як працює перевантаження функцій і для чого воно використовується, зрозуміли, що таке рекурсія.