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»

***Виконала:***

студентка групи ШІ-13

Кисців Каріна Володимирівна

# **Тема роботи:**

Ознайомлення з різними типами циклів, функцій, зокрема використання перевантажених та рекурсивних функцій у програмному коді С++ відповідно до вимог завдань.

# **Мета роботи:**

Ознайомитися з базовими видами циклів (for, while, do-while) та вкладеними циклами, вміти аналізувати доцільність їх використання в різних алгоритмах. Навчитися використовувати перевантажені функції та рекурсії для створення програм на мові С++.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли, вкладені цикли.
* Тема №2: Вектори.
* Тема №3: Функції, перевантаження функцій, функції зі змінною кількістю параметрів.
* Тема №4: Рекурсія.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли, вкладені цикли
  + Джерела Інформації
    - <https://acode.com.ua/urok-70-tsykl-while/>
    - <https://acode.com.ua/urok-71-tsykl-do-while/>
    - <https://acode.com.ua/urok-72-tsykl-for/>
    - <https://acode.com.ua/urok-73-operatory-break-i-continue/>
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
  + Що опрацьовано:
    - Опрацьовано теоретичний матеріал про різні видів циклів (for, while, do-while), визначено різницю між ними і алгоритми для коректного застосування
    - Використання циклів для ефективного виконання повторюваних завдань
    - Опрацьовано поняття алгоритмів з вкладеними циклами
  + Статус: Ознайомлена
  + Початок опрацювання теми: 26.11.2023
  + Звершення опрацювання теми: 26.11.2023
* Тема №2: Вектори.
  + Джерела Інформації:
    - <https://acode.com.ua/urok-99-vvedennya-v-std-vector/>
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
  + Що опрацьовано:
    - Оголошення, ініціалізація, розмір та можливість динамічного зміщення розміру вектору
    - Операції над векторами, такі як сортування вектору (sort), знаходження мінімуму та максимуму (min\_element, max\_element).
  + Статус: Ознайомлена
  + Початок опрацювання теми: 26.11.2023
  + Звершення опрацювання теми: 26.11.2023
* Тема №3: Функції, перевантаження функцій, функції зі змінною кількістю параметрів.
  + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
    - <https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>
  + Що опрацьовано:
    - Оголошення та визначення, параметри та повернення значень у функціях
    - Перевантаження функцій(визначення кількох функцій з однаковим ім'ям, але з різними параметрами, визначення різних версій функції для різних типів даних)

▪ Як функції дозволяють розділити програму на логічні блоки та використовувати їх повторно

* + Статус: Ознайомлена
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 27.11.2023
* Тема №4 Рекурсія
  + Джерела Інформації:
    - Конспект лекцій з дисципліни «Алгоритмізація та програмування. Частина 1» Л.І. Мочурад
    - [https://uk.wikipedia.org/wiki/](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%81%D1%96%D1%8F)
    - [https://acode.com.ua/urok-113-rekursiya](https://acode.com.ua/urok-113-rekursiya-i-chysla-fibonachchi/)
  + Що опрацьовано:
    - Визначення функцій, які викликають самі себе.
    - Визначення базового випадку для завершення рекурсивного виклику.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 28.11.2023
  + Звершення опрацювання теми: 28.11.2023

# **Виконання роботи:**

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

Завдання №1 Лабораторна робота №2 (VNS)

* Варіант №18
* Використовуючи оператор циклу, знайти суму ряду з точністю ε=0.0001, загальний член якого

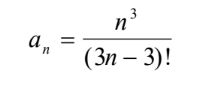


Рисунок 1 Лабораторна робота №2(VNS): формула загального члена ряду

* При визначенні суми членів ряду варто використовувати рекурентну формулу для отримання наступного члена ряду. При складанні програми вважаємо, що точність досягнута, якщо аn<ε.

Завдання №2 Лабораторна робота №3 (VNS)

* Варіант №18
* 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.



Рисунок 2 Лабораторна робота №2(VNS): умова варіанту №18

Завдання №3 Лабораторна робота №7 (VNS). Завдання №1

* Варіант №18
* Написати функцію nok зі змінною кількістю параметрів, що знаходить найменше спільне кратне для декількох чисел.

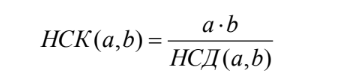


Рисунок 3 Лабораторна робота №7: формула для обчислення НСК

(НСД- найбільший спільний дільник)

Написати викликаючу функцію main, що звертається до функції nok не менше

трьох разів з кількістю параметрів 3, 5, 6.

Завдання №4 Лабораторна робота №7 (VNS). Завдання №2

- Варіант №18

- Написати перевантажені функції й основну програму, що їх викликає:

а) для масиву цілих чисел знаходить максимальний елемент;

б) для рядка знаходить довжину найдовшого слова .

Завдання №5 Практична робота №3 (Менеджмент бібліотеки)

- Створення простої програми керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути. Програма має виконувати наступні завдання:

перерахувати всі книги, дозволити взяти книгу (за наявності), дозволити повернення книги.

- Важливі деталі для врахування в імплементації програми: Для зберігання назв книг необхідно використовувати масив або вектор. Для збереження стану доступності кожної книги потрібно використовувати інший масив або вектор. Також у програмному коді програми використовується функція getline і конвертація введеного рядка в ціле число.

Завдання №6 Self-practice «Камінь-ножиці-папір»

- Зеник і Марічка вирішили зіграти n разів у відому гру «камінь-ножиці-папір». Гра відбувається в декілька раундів. У кожному раунді кожен з учасників одночасно обирає один із трьох предметів: камінь, ножиці або папір. Результат раунду ґрунтується на предметах, обраних суперниками: камінь перемагає ножиці, ножиці перемагають папір, а папір перемагає камінь. Звернуть увагу, що в деякому раунді гравці можуть обрати той самий предмет. У такому випадку результатом раунду є нічия.

Марічка знає, у скількох раундах вона обере камінь, ножиці й папір, таку ж інформацію вона має про Зеника.

Марічка хоче дізнатися, скільки максимально раундів вона може виграти. Допоможіть їй дізнатися це.

Вхідні дані:

- Перший рядок містить три цілі числа — кількість раундів, у яких Марічка поставить камінь, ножиці й папір відповідно.

Другий рядок містить три цілі числа — кількість раундів, у яких Зеник поставить камінь, ножиці й папір, відповідно.

Вихідні дані:

В одному рядку виведіть ціле число — максимальну кількість раундів, у яких може перемогти Марічка.

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

Програма №1 Лабораторна робота №2 (VNS)

* Планований час на реалізацію – 1 день

Програма №2 Лабораторна робота №3 (VNS)

* Планований час на реалізацію – 1 день

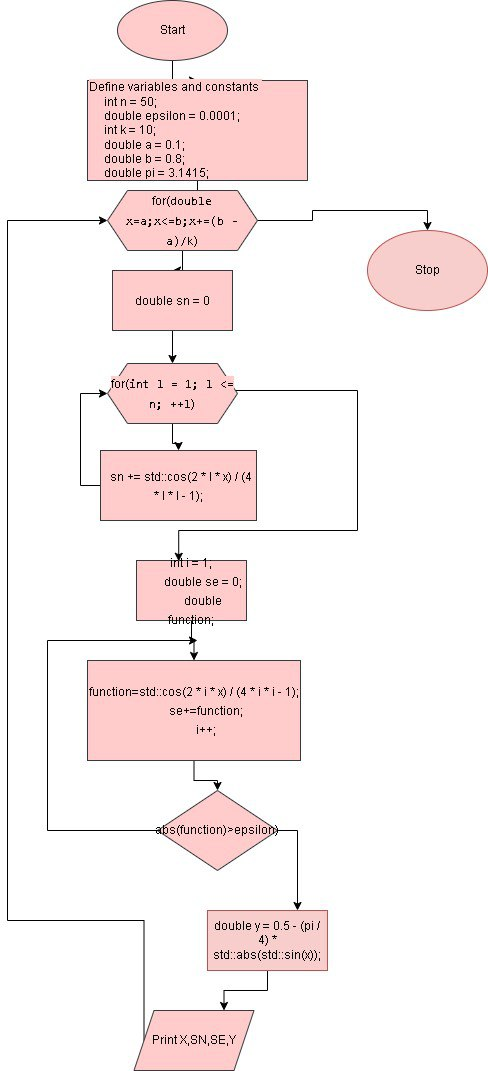


Рисунок 4 Блок-схема до Лабораторної роботи №

Програма №3 Лабораторна робота №7 (VNS). Завдання №1

* Планований час на реалізацію – 1 день

Програма №4 Лабораторна робота №7 (VNS). Завдання №2

- Планований час на реалізацію – 1 день

Програма №5 Практична робота №3 (Менеджмент бібліотеки)

- Планований час на реалізацію – 3 дні

Програма №6 Self-practice «Камінь-ножиці-папір»

- Планований час на реалізацію – 1 день

## 

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

Pull-request: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/599>

Завдання №1 Лабораторна робота №2 (VNS). Використовуючи оператор циклу, знайдено суму елементів ряду.

[Посилання на файл програми у пул-запиті GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/599/files" \l "diff-703796bc3f7296f27fd369e402f0868bc946a40afd73fb70478bb738cfbde3c9)

#include <iostream>

#include <cmath>

using namespace std;

int factorial(int fn){

    if (fn<=1){

        return 1;

    }

    else{

        return fn\*factorial(fn-1);

    }

   }

int main(){

    int n=1;

    float an=pow(n,3)/factorial(3\*n -3);

    float totalsum =0;

    do{

        totalsum += an;

        an\*=pow((n+1),3)/(3\*(pow(n,4))\*(3\*n-2)\*(3\*n-1));

        n++;

    }while(an>=0.0001);

  cout<<totalsum;

return 0;

    }

Програмний код №1 - Лабораторна робота №2 (VNS)

Завдання №2 Лабораторна робота №3 (VNS)

Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <cmath>

const double pi = 3.1415;

int main(){

    const int n = 50;

    const double epsilon = 0.0001;

    const int k = 10;

    const double a = 0.1;

    const double b = 0.8;

  for (double x=a; x<=b; x+= (b - a) / k) {

        // Обчислення суми за фіксованою кількістю членів n

        double sn = 0;

        for (int l = 1; l <= n; ++l) {

            sn += std::cos(2 \* l \* x) / (4 \* l \* l - 1);

        }

      // Обчислення суми з точністю epsilon

        int i = 1;

        double se = 0;

        double function;

        do {

            function=std::cos(2 \* i \* x) / (4 \* i \* i - 1);

        se+=function;

        i++;

        }while (abs(function)>epsilon);

          // Точне значення функції

        double y = 0.5 - (pi / 4) \* std::abs(std::sin(x));

        std::cout << "X=" << x << " SN=" << sn << " SE=" << se << " Y=" << y << "\n";

    }

    return 0;

}

Програмний код №2 - Лабораторна робота №3 (VNS)

Завдання №3 Лабораторна робота №7 (VNS). Завдання №1

[Посилання на файл програми у пул-запиті GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/599/files#diff-9d7d4452dfad44d7372f154486b1d8028ab5d84dc951a7ef5f27764ee2503385)

#include <iostream>

#include <cstdarg>

using namespace std;

int nod(int a,int b){

  while(a && b) if (a < b) b %= a; else a %= b;

    return a + b;

}

int nok(int count,...){

    va\_list list;

    va\_start(list,count);

    int nsk=1;

    for(int i=0;i<count;i++){

        int parameter=va\_arg(list,int);

         nsk =(nsk\*parameter)/nod(nsk,parameter);

    }

    va\_end(list);

    return nsk;

}

int main(){

cout<<"NOK(6,2,3) = "<<nok(3,6,2,3)<<endl;

cout<<"NOK(2,4,8,32,64) = "<<nok(5,2,4,8,32,64)<<endl;

cout<<"NOK(7, 19, 11, 87, 6, 3) = "<<nok(6,7,19,11,87,6,3)<<endl;

    return 0;

}

Програмний код №3 - Лабораторна робота №7 (VNS). Завдання №1

Завдання №4 Лабораторна робота №7 (VNS). Завдання №2

[Посилання на файл програми у пул-запиті GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/599/files#diff-6202ea619d509aa24849f25f4a7e37f7f21ed4f087a5a2e1da3cb8cb5c0f2d86)

#include <iostream>

#include <cstring>

using namespace std;

// a) Знаходження максимального елемента

int max1(int size, int array[]) {

    int max = array[0];

    for (int i = 1; i < size; ++i) {

        if (array[i] > max) {

            max = array[i];

        }

    }

    return max;

}

// b) Знаходження довжини найдовшого слова у рядку

int max1(string str) {

   int length = str.length();

   int temp = 0;

   int step = 0;

   for (int i = 0; i < length; i++) {

      if (str[i] != ' ')

         step++;

      else {

         temp = max(temp, step);

         step = 0;

      }

   }

   return max(temp, step);}

int main() {

    // a) Тестування функції для масиву цілих чисел

    int intArray[] = {3, 7, 1, 5, 9, 2, 8, 4, 6};

    int arraySize = sizeof(intArray) / sizeof(int);

    int max = max1(arraySize, intArray);

    std::cout << "maximum element of array: " << max << std::endl;

    // b) Тестування функції для рядка

    string str = "hello Ivan have a nice day:)";

    cout <<"maximum length of a word is: "<<max1(str);

    return 0;

}

Програмний код №4 - Лабораторна робота №7 (VNS). Завдання №2

Завдання №5 Практична робота №3 (Менеджмент бібліотеки)

[Посилання на файл програми у пул-запиті GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/599/files#diff-ccae97689cfd1ef36d7aaf6d9139adb6ae34a0bdaa1dba357414a368bd53e704)

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main() {

    vector<string> books = {"Hunger Games", "The Lord of the Rings", "Da Vinci Code", "Murder on the Orient Express"};

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

    int choice;

    string bookName;

cout << "Welcome to the library!\n";

menu:

    cout << "1. List all books\n";

    cout << "2. Borrow a book\n";

    cout << "3. Return a book\n";

    cout << "4. Exit\n";

    cout << "Print your choice: ";

    string choiceStr;

    getline(cin, choiceStr);

    try {

        choice = stoi(choiceStr);

    } catch (const invalid\_argument& e) {

        cout << "Invalid choice. Please enter a number.\n";

        goto menu;

    }

    switch (choice) {

        case 1:

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

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

            }

            break;

        case 2:

            cout << "Enter book name to borrow: ";

            getline(cin, bookName);

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

                if (books[i] == bookName) {

                    if (available[i]) {

                        available[i] = false;

                        cout << "Enter book name to borrow:\n";

                    } else {

                        cout << "Book borrowed successfully!\n";

                    }

                    break;

                }

            }

            break;

        case 3:

            cout << "Enter book name to return:";

            getline(cin, bookName);

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

                if (books[i] == bookName) {

                    if (!available[i]) {

                        available[i] = true;

                        cout << "Book returned successfully!\n";

                    } else {

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

                    }

                    break;

                }

            }

            break;

        case 4:

            cout << "Goodbye!\n";

            return 0;

        default:

            cout << "Wrong choice\n";

            goto menu;

    }

    char continueChoice;

    do {

        cout << " Do you want to perform another operation? (Y/N):";

        cin >> continueChoice;

    } while (continueChoice != 'Y' && continueChoice != 'y' && continueChoice != 'N' && continueChoice != 'n');

    if (continueChoice == 'Y' || continueChoice == 'y') {

        goto menu;

    } else {

        cout << "Have a nice day!!\n";

        return 0;

    }

}

Програмний код №5 - Практична робота №3 (Менеджмент бібліотеки)

Завдання №6 Self-practice «Камінь-ножиці-папір»

[Посилання на файл програми у пул-запиті GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/599/files#diff-1cc95b3faba0413ccce404219f0809008bd5690f596ea48292e5a1f54c6833d9)

#include <iostream>

using namespace std;

int main()

{

    int rockM = 0, pprM = 0, scrsM = 0;

    cin >> rockM >> scrsM >> pprM;

    int rockZ = 0, pprZ = 0, scrsZ = 0;

    cin >> rockZ >> scrsZ >> pprZ;

    int win = 0;

    while (rockM > 0 && scrsZ > 0) {

        win++;

        rockM--; scrsZ--;

    }

    while (pprM > 0 && rockZ > 0) {

        win++;

        pprM--; rockZ--;

    }

    while (scrsM > 0 && pprZ > 0) {

        win++;

        scrsM--; pprZ--;

    }

    cout <<  win;

    return 0;

}

Програмний код №6 - Self-practice «Камінь-ножиці-папір»

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

Завдання №1. Лабораторна робота №2 (VNS). Використовуючи оператор циклу, знайдено суму елементів ряду.



Рисунок 5 Результат виконання Лабораторної роботи №2

Час затрачений на виконання завдання – 1 день

Завдання №2. Лабораторна робота №3 (VNS)

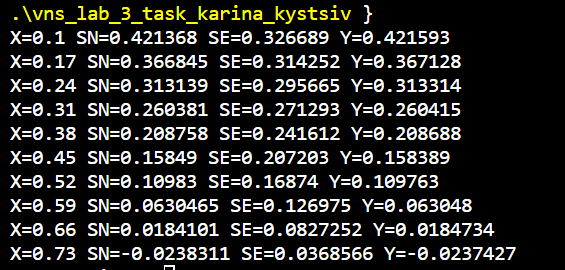


Рисунок 6 Результат виконання Лабораторної роботи №3

Час затрачений на виконання завдання – 1 день

Завдання №3. Лабораторна робота №7 (VNS). Завдання №1

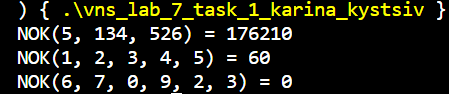
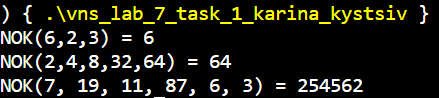
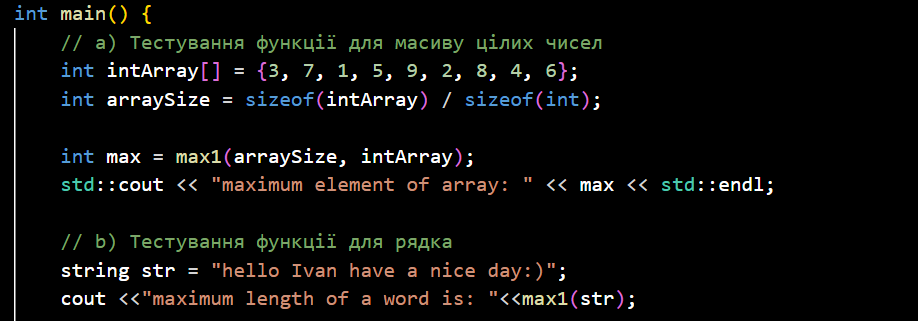


Рисунок 7 Результати виконання лабораторної роботи №7. Завдання 1

Час затрачений на виконання завдання – 1 день

Завдання №4. Лабораторна робота №7 (VNS). Завдання №2



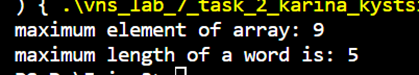
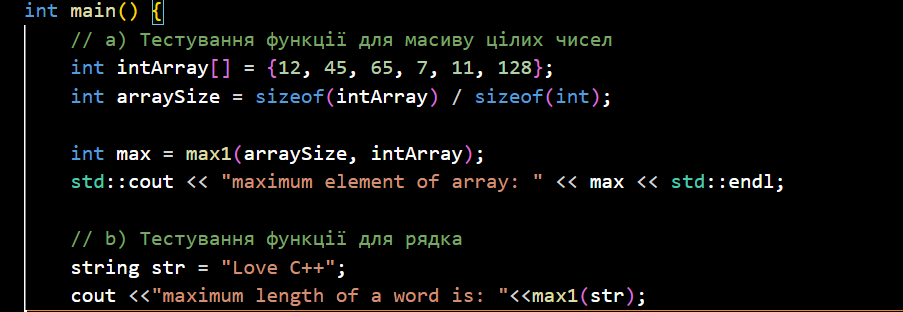


Рисунок 8 Результати виконання лабораторної роботи №7. Завдання 2, варіант 1



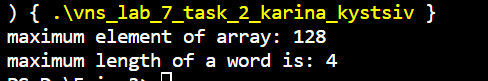


Рисунок 9 Результати виконання лабораторної роботи №7. Завдання 2, варіант 2

Час затрачений на виконання завдання – 2 дні

Завдання №5. Практична робота №3 (Менеджмент бібліотеки)

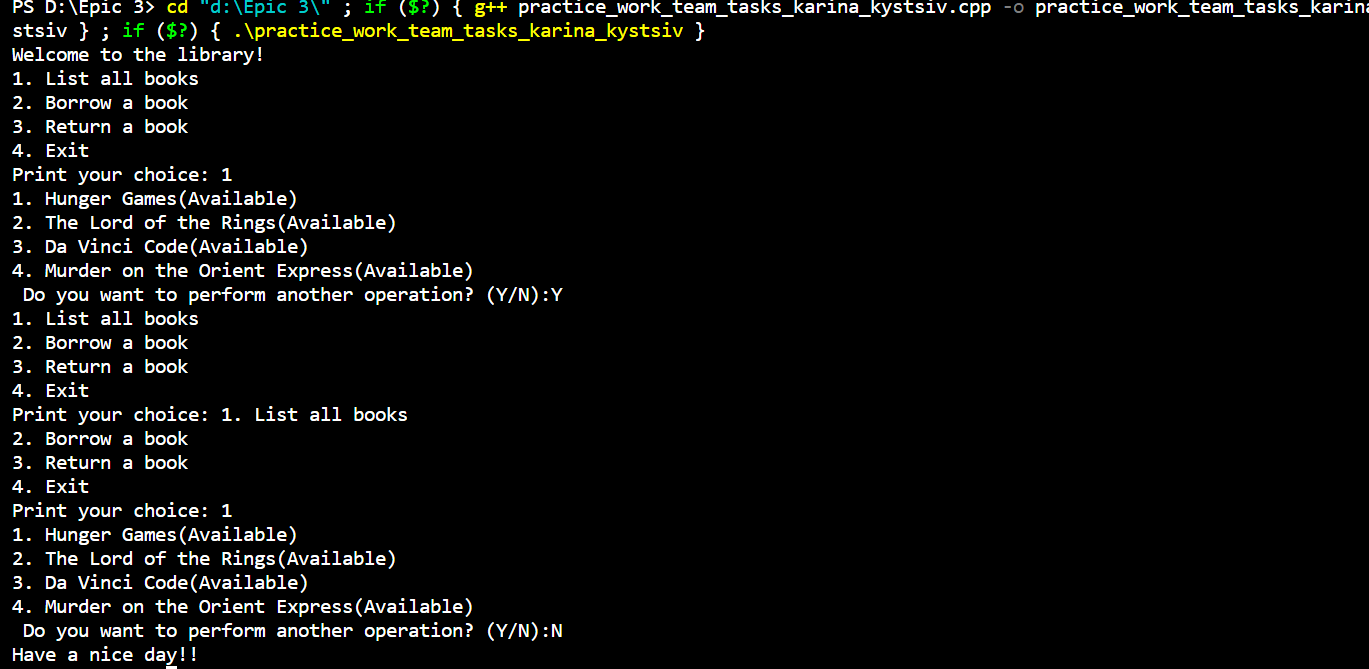


Рисунок 10 Результати виконання практичної роботи №3

Час затрачений на виконання завдання – 2 дні

Завдання №6 Self-practice

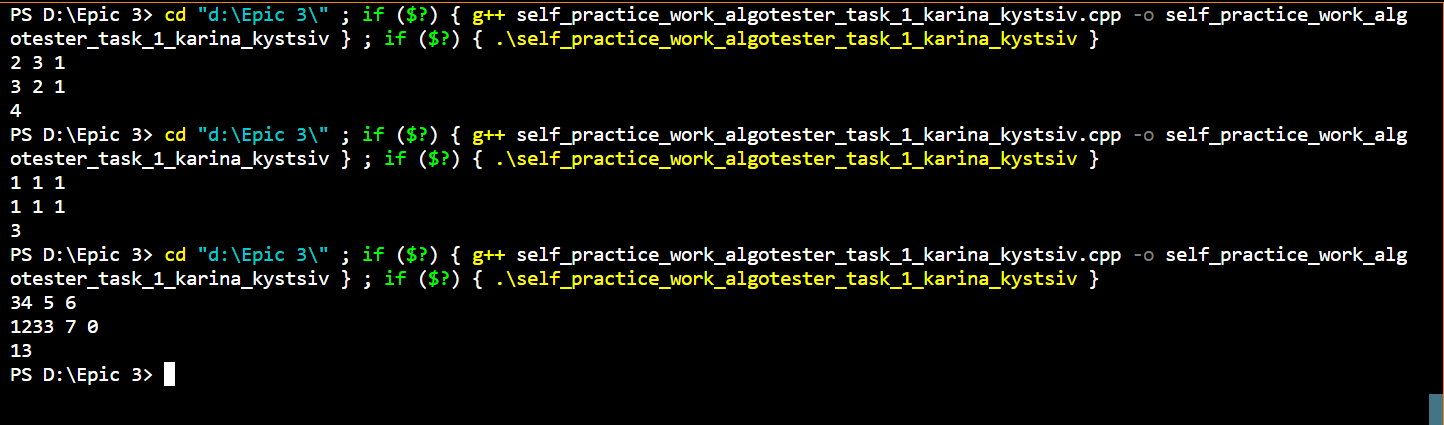


Рисунок 11 Результати виконання програми «Камінь-ножиці-папір»

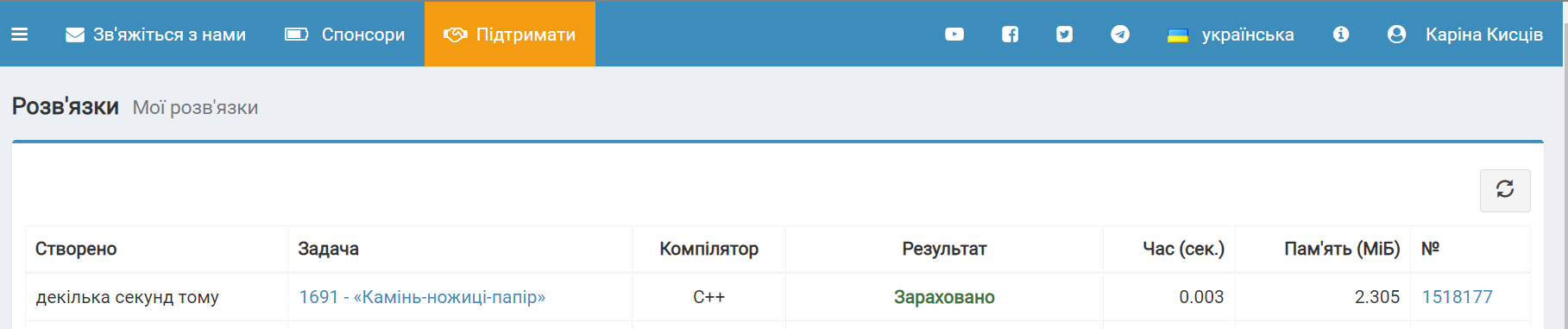


Рисунок 12 Скріншот з Алготестера

Час затрачений на виконання завдання – 1 день

## **5. Кооперація з командою:**

# 

Рисунок 13 Зустріч з командою №1

Зустріч №2



Рисунок 14 Зустріч з командою №2

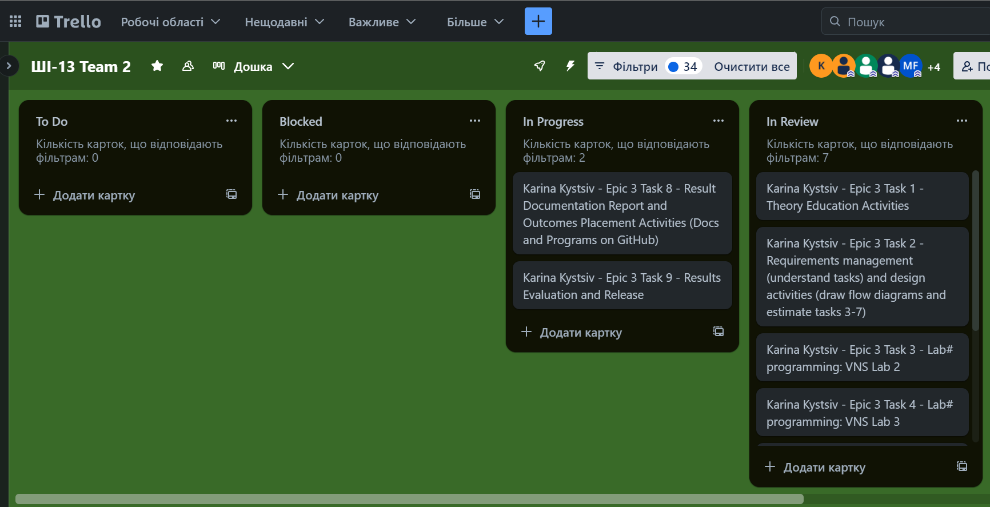


Рисунок 15 Скріншот прогресу в Трелло

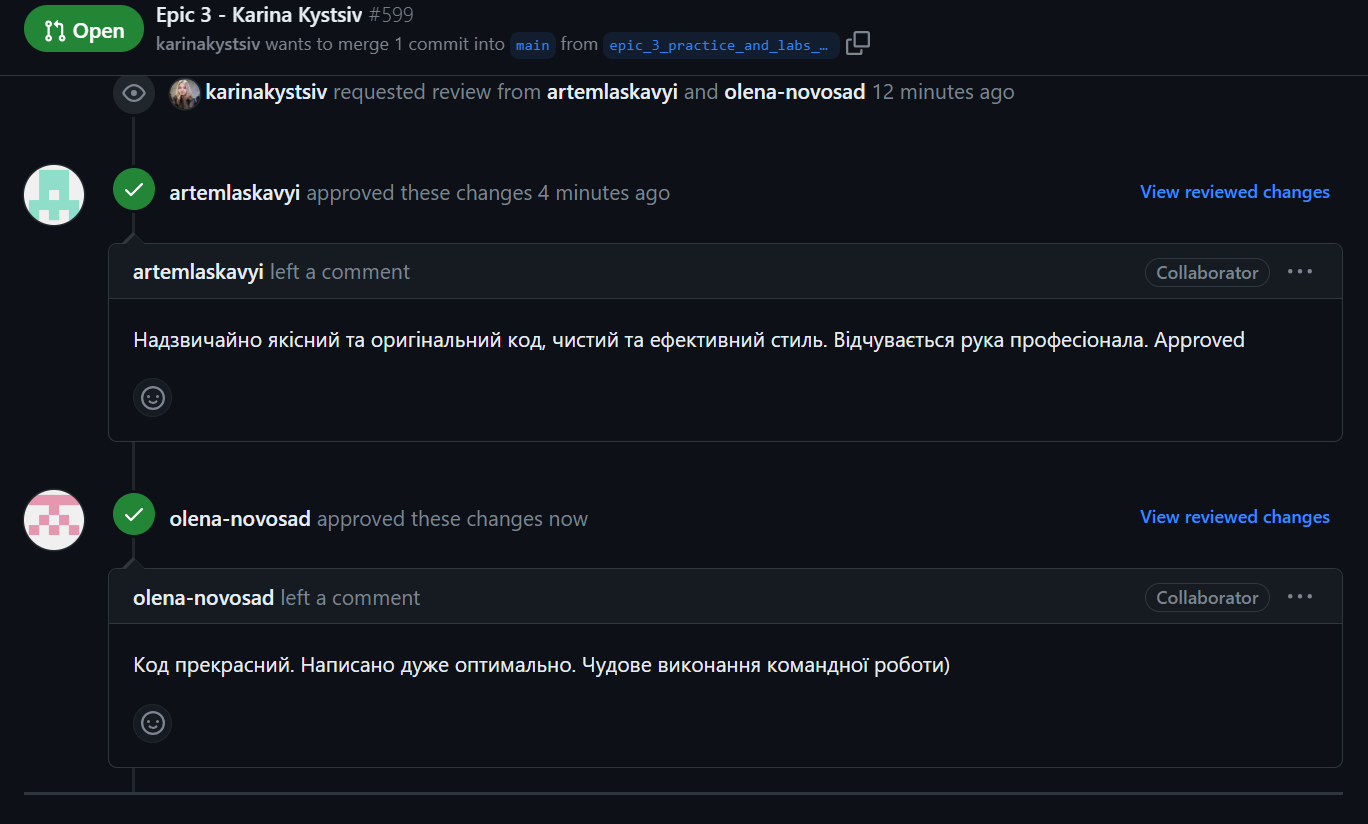


Рисунок 16 Код рев’ю від учасників команди

# **Висновки:**

При виконанні цієї роботи, я зосереджувалась на вивченні та практичному використанні ключових концепцій програмування: циклів, вкладених циклів, функцій, перевантаження функцій та рекурсії. Ці концепції є важливими для будь-якого програміста та можуть бути використані для вирішення різних завдань в програмуванні. Дана робота була також доволі складною з математичної точки зору, адже завдання з лабораторних робіт №2 та №3 базувалися на знаходженні суми ряду. Багато часу витратила і на лабораторну роботу №7, а саме 1 завдання, оскільки там потрібно було ознайомитись з variadic function і для роботи з нею потрібно було навчитись користуватись va\_list.

Отож, цикли в програмуванні є потужним інструментом для повторення блоку коду зазначену кількість разів або до виконання певної умови. For, while, та do-while є основними видами циклів, які використовуються для різних сценаріїв.

Вкладені цикли використовуються, коли потрібно повторити певний блок коду всередині іншого циклу. Вони корисні для обробки декількох рівнів даних чи виконання багаторівневих завдань.

Функції дозволяють розділяти програму на логічні блоки, полегшуючи підтримку та розуміння коду. Використання функцій сприяє модульності програми, покращує читабельність та можливість використання коду знову.

Перевантаження функцій дозволяє використовувати одне ім'я функції для різних сценаріїв використання, залежно від параметрів, викликається відповідна функція.

Рекурсія визначається тим, що функція може викликати сама себе. Використання рекурсії може спростити деякі завдання та полегшити розуміння коду.

Використання векторів у C++ дозволяє зручно та ефективно працювати з динамічними масивами даних та використовувати багато функцій стандартної бібліотеки для роботи з ними.