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***: «Епік 3: Цикли. Вкладені цикли. Функції. Перевантаження функцій. Рекурсія»

***Виконав(ла):***

студент групи ШІ-13

Золотарчук Максим

# **Тема роботи:**

Робота з циклами та вкладеними циклами. Поняття функції, її типи та способи задання. Розбір та використання перевантажених та рекурсивних функцій та функцій з багатьма змінними.

# **Мета роботи:**

# Набуття базових знань та вмінь, які стосуються роботи з циклами, вкладеними циклами, ітераційними та арифметичними циклами в програмуванні, а також з освоєнням понять функції, її типів та різними способами її опису. Створення моделі обчислення наближеного значення арифметичної функції шляхом її розкладу на складові ряду Маклорена. Дослідження перевантажених та рекурсивних функцій, а також функцій з багатьма змінними для розв’язання різноманітних задач.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли.
* Тема №2: Функції.
* Тема №3: Перевантаження функції.
* Тема №4: Функція зі змінною кількістю параметрів (va\_list).

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли.
  + Джерела Інформації

Відео 1 -https://www.youtube.com/watch?v=ckJtOMcIxyU&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=32

* + - Відео 2 - <https://www.youtube.com/watch?v=pBhaBdXWMmU&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=33>
    - Відео 3 - <https://www.youtube.com/watch?v=QXaSSIjVor8&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=34>
    - Відео 4 - <https://www.youtube.com/watch?v=7ZEAxFb_J8s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=35>
  + Що опрацьовано:
    - Властивості циклу.
    - Визначення та властивості вкладених циклів, арифметичних циклів та ітераційних циклів.
    - Використання різних типів задання циклу (for, while, do-while, range-based for).
    - Властивості та типи операторів переходу та їх використання в різних контекстах.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 28.11.2023
* Тема №2: Функції.
  + Джерела Інформації:
    - Стаття 1 - [C++ Functions](https://www.w3schools.com/cpp/cpp_functions.asp)
    - Відео 1 - https://www.youtube.com/watch?v=G8P6SvdqU9s&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=43
  + Що опрацьовано:
    - Різниця між вбудованими функціями та функціями визначеними користувачем.
    - Спосіб виклику функції.
    - Типи функцій (за параметрами та за типом).
    - Спосіб покращення читабельності коду за рахунок функцій.
    - Область видимості аргументів функції.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 30.11.2023
* Тема №3 Перевантаження функції.
  + Джерела Інформації:
    - Стаття 1 - [C++ Function Overloading](https://www.w3schools.com/cpp/cpp_function_overloading.asp)[Function Overloading in C++](https://www.geeksforgeeks.org/function-overloading-c/)
    - Відео 1 - https://www.youtube.com/watch?v=3KJfisev6SI&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=48
  + Що опрацьовано:
    - Використання одного проголошення функції для виконання різних задач.
    - Визначення перевантаженою функцією, яку саме функцію їй виконувати за типом та кількістю даних.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 30.11.2023
* Тема №4 Функція зі змінною кількістю параметрів.
  + Джерела Інформації:
    - Стаття – 1 [Variadic function templates in C++](https://www.geeksforgeeks.org/variadic-function-templates-c/)
    - Відео – 1 [What are variadic functions (va\_list) in C?](https://www.youtube.com/watch?v=oDC208zvsdg&t=312s&ab_channel=CodeVault)
  + Що опрацьовано:
    - Як визначати та використовувати функції, які можуть приймати різну кількість аргументів
    - Проблема розподілу байтів в обрахуванні заданих циклом аргументів та її вирішення за допомогою варіативної функції va\_list
  + Статус: Ознайомлений
  + Початок опрацювання теми: 22.11.2023
  + Звершення опрацювання теми: 28.11.2023

# **Виконання роботи:**

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

**Завдання №1 VNS Lab №2**

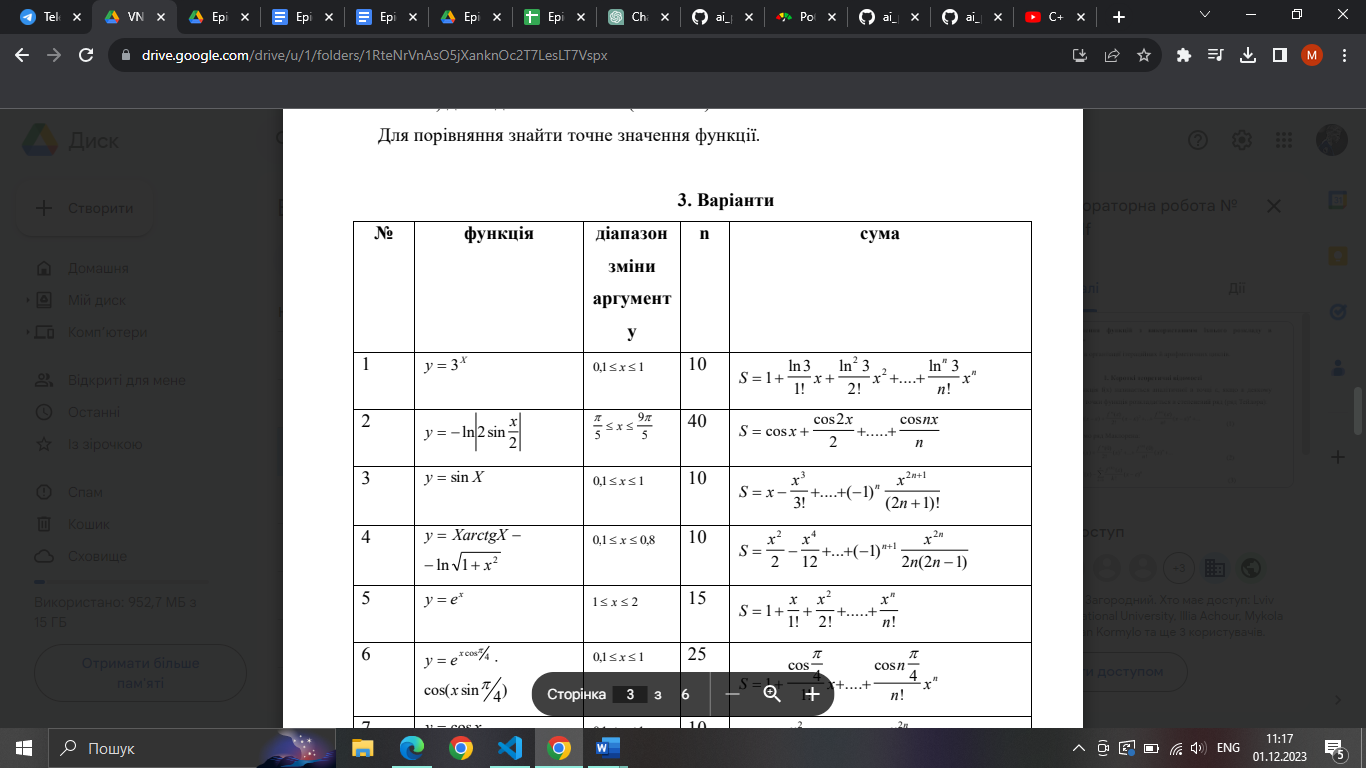
* Варіант завдання – 1

Знайти суму цілих додатніх непарних чисел, менших 200..

* Деталі завдання –
* Використовуючи оператор циклу, знайти суму елементів, зазначених у
* конкретному варіанті. Результат надрукувати, надавши відповідний
* заголовком.

**Завдання №2 VNS Lab №3**

* Варіант завдання – 3



* 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 – 3

Написати функцію sum зі змінною кількістю параметрів, що знаходить суму

чисел типу int за формулою:

S=a1\*a2+a2\*a3+a3\*a4+. . . . .

Деталі завдання

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

кількістю параметрів.

* Важливі деталі для врахування в імплементації програми

Використання перевантаження функції та робота з варіативною функцією va\_list

**Завдання №4 VNS Lab №7 Task 2**

* Варіант завдання – 3

а) для множення цілих чисел;

б) для множення комплексних чисел.

Деталі завдання

Написати перевантажені функції й основну програму, що їх викликає.

**Завдання №5 Class Practice – Library**

* Деталі завдання

Ви створюєте просту програму 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.

### Структури даних

* Використовуйте масив або вектор для зберігання назв книг.
* Використовуйте інший масив або вектор для збереження стану доступності кожної книги.

## **Мета Задачі**

Навчитися користуватися операторами циклів та функцією переходу на мітку:

1. for( ) { … }
2. for each
3. while( ) { … }
4. do { … } while( )
5. go to

### Вимоги:

1. while: продовжувати працювати, доки користувач не вирішить вийти.
2. do while: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
3. for: список усіх книг за допомогою циклу.
4. for each: перевірити наявність кожної книги.
5. goto: якщо користувач вводить неправильний вибір, використовуйте goto, щоб перенаправити його до головного меню.

* Важливі деталі для врахування в імплементації програми

Вказані вимоги.

# **Завдання №6 Self Practice – Робот**

# Варіант завдання

* Деталі завдання

Недавно батьки подарували Петрикові робота, якого можна програмувати. Початково робот стоїть у точці з координатами (0, 0). Петрик уводить роботу набір команд, які той виконує послідовно від першої до останньої. Існує два типи команд:

1. U — перейти на 1 вверх, тобто з точки (x�, y�) у точку (x�, y+1�+1).
2. R — перейти на 1 вправо, тобто з точки (x�, y�) у точку (x+1�+1, y�).

Петрик запрограмував робота послідовністю команд s1�1, s2�2, ..., sn��. Вам задана ця послідовність, а також пара чисел x� та y�.

Визначте, чи може Петрик переставити місцями команди так, щоб робот пройшов через точку (x�, y�).

# Вхідні дані

У першому рядку задано рядок s�, який складається з n� символів, які відповідають за команди, введені Петриком.

У другому рядку задано два цілих числа x� та y�.

# Вихідні дані

У єдиному рядку виведіть YES, якщо Петрик може переставити місцями команди так, щоб робот перейшов через точку (x�, y�).

Якщо досягнути цілі неможливо, виведіть NO.

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

**Програма №2 VNS Lab №3**

* Блок-схема :

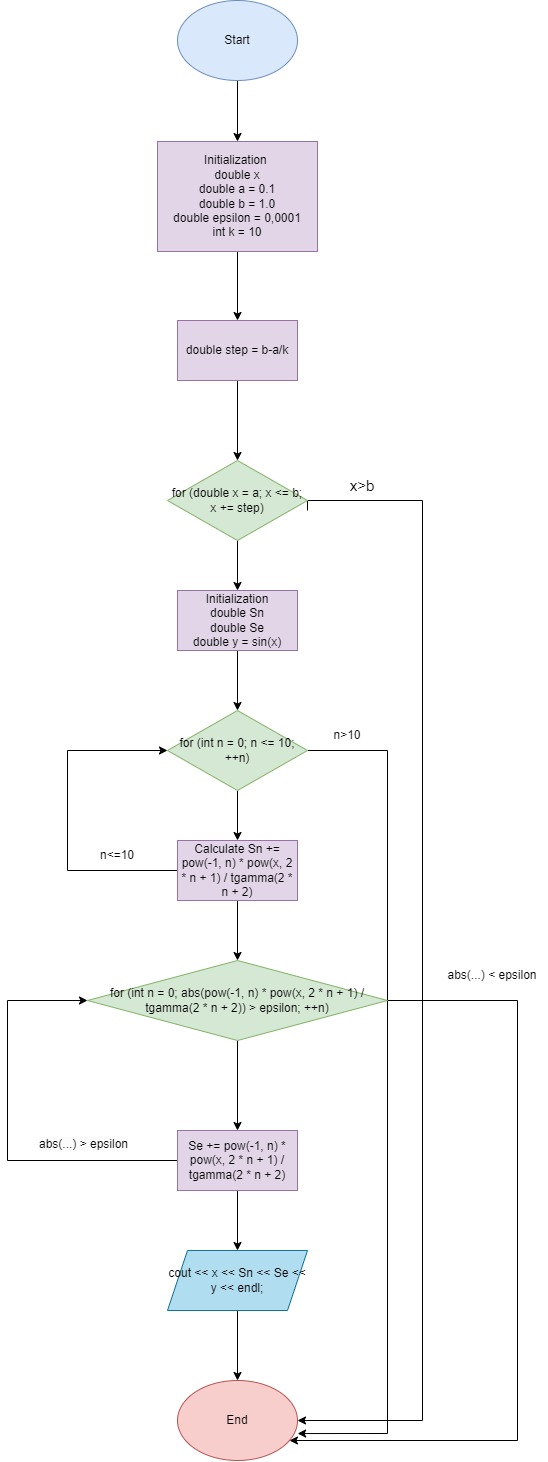


Рисунок Flowchart VNS Lab 3

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

**Завдання №1 - VNS Lab №2**

#include <iostream>

#include <cmath>

#include <cwchar>

#include <Windows.h>

using *namespace* std;

*int* main() {

setlocale(LC\_ALL, "ukr");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

*int* sum = 0;

for (*int* i = 1; i < 200; i += 2) {

  sum += i;

}

wprintf(L"Сума всіх непарних чисел менше 200: %ld " , sum);

return 0;

}

*Код програми №1 VNS Lab №2*

**Завдання №2 - VNS Lab №3**

#include <iostream>

#include <cmath>

using *namespace* std;

*int* main()

{

*double* x;

*double* a = 0.1;

*double* b = 1.0;

*double* epsilon = 0.0001;

*int* k = 10;

    // X- значення параметра; SN- значення суми для заданого n;

    //SE- значення суми для заданої точності; Y-точне значення функції.

*double* step = (b - a) / k;

    for (*double* x = a; x <= b; x += step)

    {

*double* Sn = 0;

*double* Se = 0;

*double* y = sin(x);

        for (*int* n = 0; n <= 10; ++n)

        {

            Sn += pow(-1, n) \* pow(x, 2 \* n + 1) / tgamma(2 \* n + 2);

        }

        for (*int* n = 0; abs(pow(-1, n) \* pow(x, 2 \* n + 1) / tgamma(2 \* n + 2)) > epsilon; ++n)

        {

            Se += pow(-1, n) \* pow(x, 2 \* n + 1) / tgamma(2 \* n + 2);

        }

        cout << "X = " << x << endl;

        cout << "SN = " << Sn << endl;

        cout << "SE = " << Se << endl;

        cout << "Y = " << y << endl;

    }

    return 0;

}

*Код програми №2 VNS Lab №3*

**Завдання №3** - **VNS Lab №7 Task 1**

#include <iostream>

#include <cmath>

#include <cstdarg>

using *namespace* std;

*int* sum(*int* *count*, ...)

{

 va\_list args;

 va\_start(args, *count*);

*int* x = 0;

 for (*int* i = 0; i < *count*; ++i)

 {

  x += va\_arg(args, *int*);

 }

 va\_end(args);

 return x;

}

*int* main()

{

 cout << sum(3, 1, 2, 3) << endl;

 cout << sum(10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10) << endl;

    cout << sum(12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12) << endl;

 return 0;

}

*Код програми №3 VNS Lab №7 Task 1*

**Завдання №4 VNS Lab №7 Task 2**

#include <iostream>

#include <cmath>

#include <complex>

using *namespace* std;

*int* Multi(*int* *a*, *int* *b*)

{

 return *a* \* *b*;

}

complex<*double*> Multi(complex<*double*> *z1*, complex<*double*> *z2*)

{

 return *z1* \* *z2*;

}

*int* main()

{

*int* r = Multi(5, 3);

 cout << "Multiplication " << r << endl;

 //complex<double> c = Multi((2.0, 3.0), (1.0, -2.0));

 complex<*double*> z1(7.0, 6.0); // 7 + 6i

 complex<*double*> z2(9.0, 4.0); // 9 + 4i

 complex<*double*> c = Multi(z1, z2);

 cout << "Complex Multiplication " << c.real() << "+" << c.imag() << "i" << endl;

 return 0;

}

*Код програми №4 VNS Lab №7 Task 2*

**Завдання №5 Class Practice – Library**

#include <iostream>

#include <vector>

#include <string>

using *namespace* std;

*int* main()

{

vector<string> books = {"1984" , "The Great Gatsby", "The Alchemist", "The Art of War"};

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

*int* choice;

string continueChoice;

do {

cout << "1. List all books" << endl;

cout << "2. Take a book" << endl;

cout << "3. Return a book" << endl;;

cout << "Enter your choice: ";

cin >> choice;

switch(choice) {

case 1:

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

    {

    cout << i+1 << ". " << books[i];

        if(available[i]){

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

        }

        else {

        cout << "not available";

        }

        cout << endl;

                }

        break;

case 2:

cout << "Enter the number of the book you want to take: ";

cin >> choice;

    if(available[choice-1]) {

    available[choice-1] = false;

    cout << "You took the book " << books[choice-1] << endl;

}

    else {

     cout << "This book is unavailable " << endl;

    }

    break;

case 3:

cout << "Enter the number of the book you want to return: ";

cin >> choice;

available[choice-1] = true;

cout << "You returned the book " << books[choice-1] << endl;

break;

default:

cout << "Wrong choice. Try again." << endl;

continue;

        }

cout << "Do you want to perform another operation? (Yes/No): ";

cin >> continueChoice;

    } while(continueChoice == "Yes");

    return 0;

}

*Код програми №5 Class Practice – Library*

**Завдання №6 Self Practice – Офісна Вулиця. Частина 1**

Деталі по програмі:

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/562/files#diff-a97f971fa41f16ddaed412c27b236f0f2c5bf52cc3dd46431278daa2a59a525d>

#include <iostream>

#include <string>

#include <cmath>

using namespace std;

int main() {

string s;

cin >> s;

int x,y;

cin >> x >> y;

int R = 0;

int U = 0;

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

if(s[i] == 'R'){

++R;

}

else if(s[i] == 'U'){

++U;

}

}

if(x>=0 && y>=0 && x<=R && y<=U){

cout << "YES" << endl;

}

else {

cout << "NO" << endl;

}

return 0;

}}

*Код програми №6 Self Practice – Робот.*

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

**Завдання №1 VNS Lab №2**

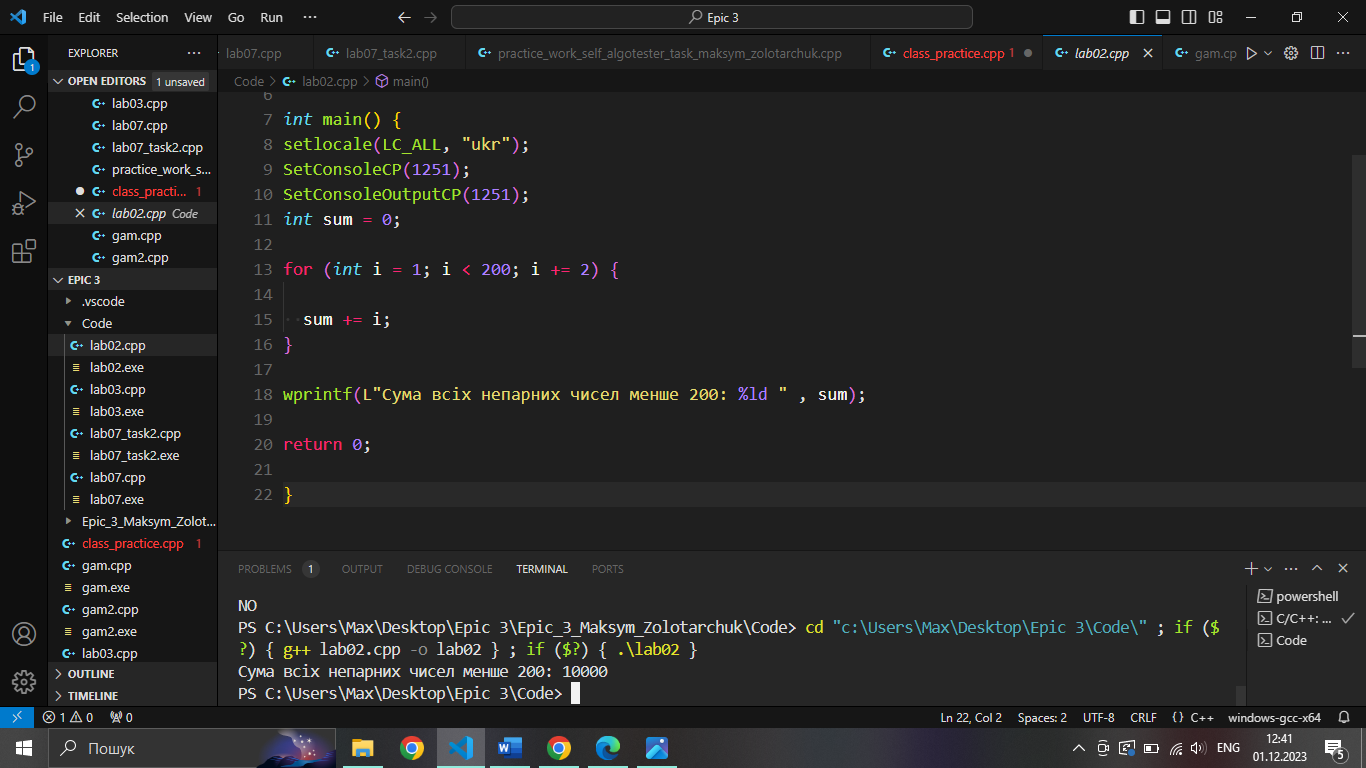


Рисунок 2 Результат завдання №1 VNS Lab №2

**Завдання №2** **VNS Lab №3** Програма виконує лише одну функцію – обрахування функції y=sin(x);

Деталі по виконанню і тестуванню програми:

X- значення параметра; SN- значення суми для заданого n; SE- значення суми для заданої точності; Y-точне значення функції.

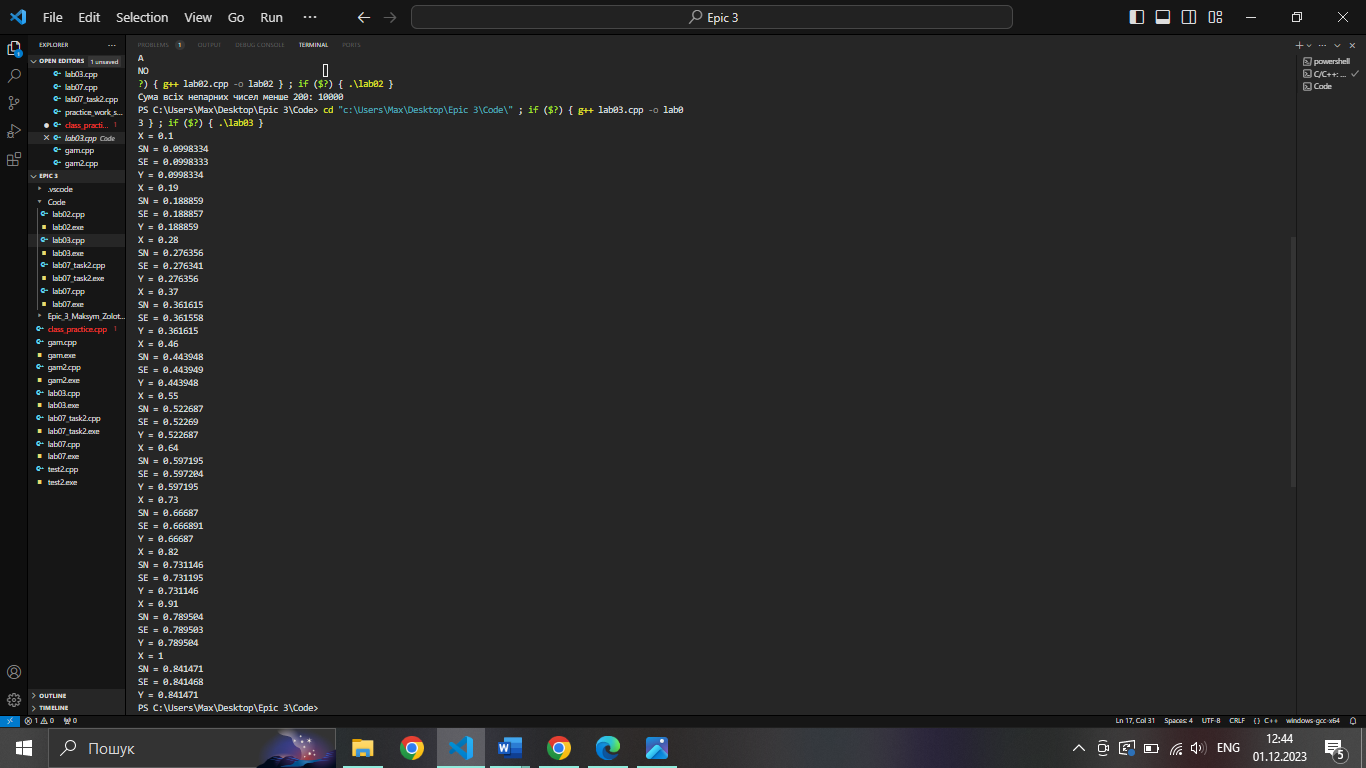


Рисунок 3 Результат завдання №2 VNS Lab №3

**Завдання №3 VNS Lab №7 Task 1** Деталі по виконанню і тестуванню програми

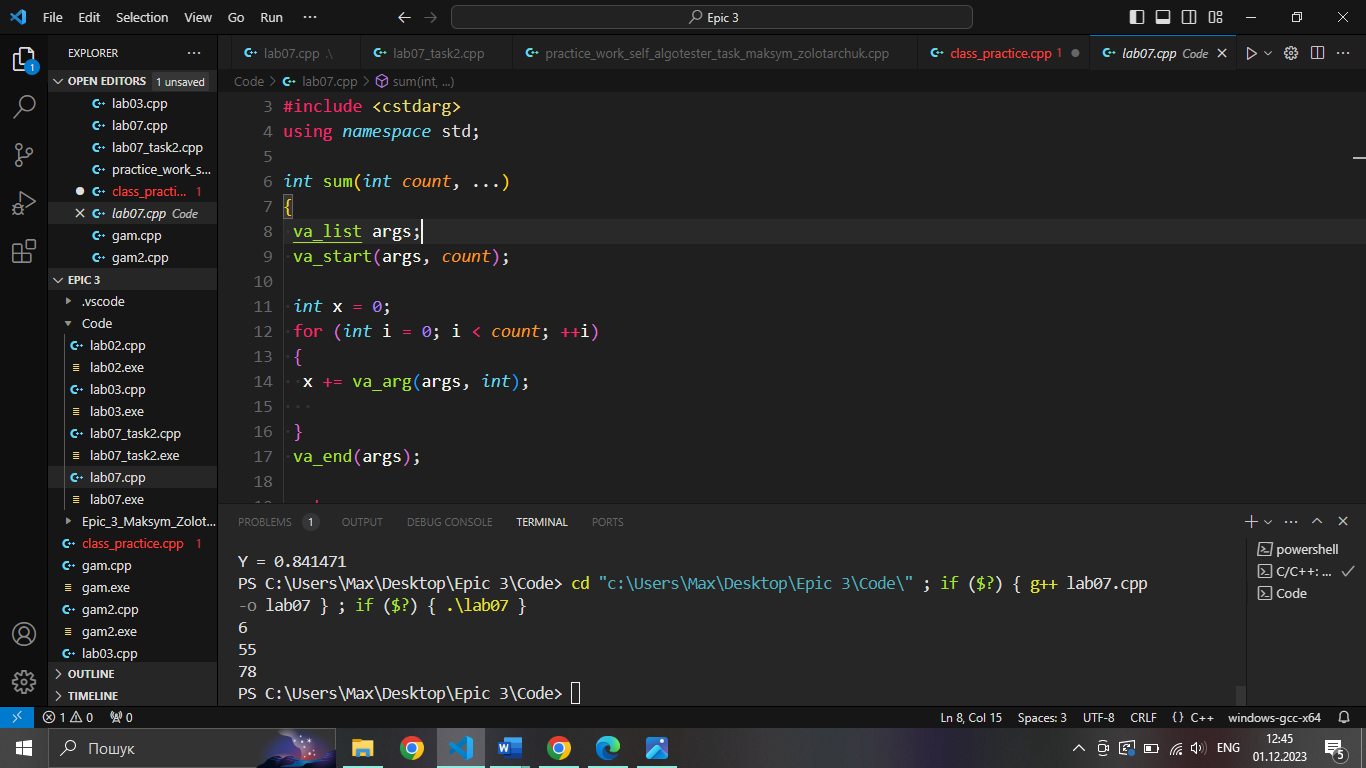


Рисунок 4 Результат завдання №3 VNS Lab №7 Task 1

**Завдання №4 VNS Lab №7 Task 2** Деталі по виконанню і тестуванню програми

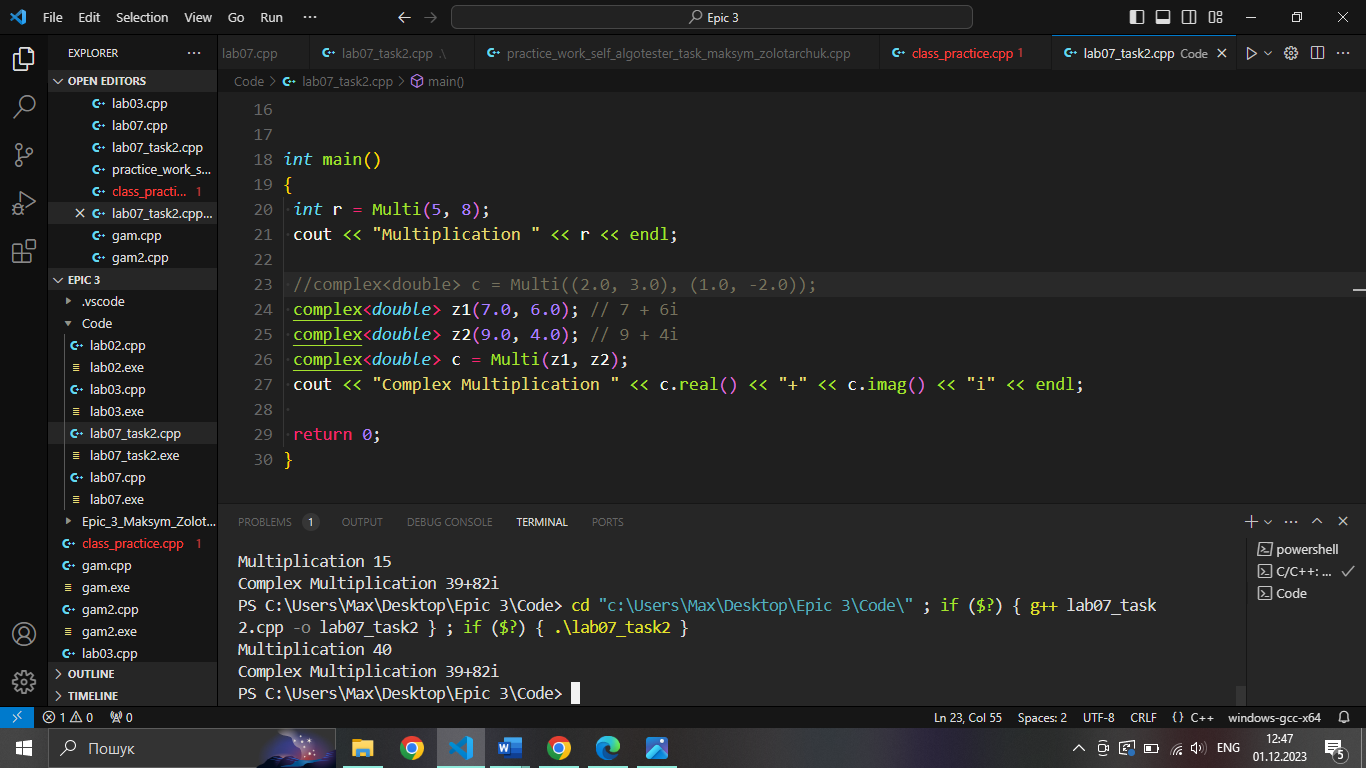
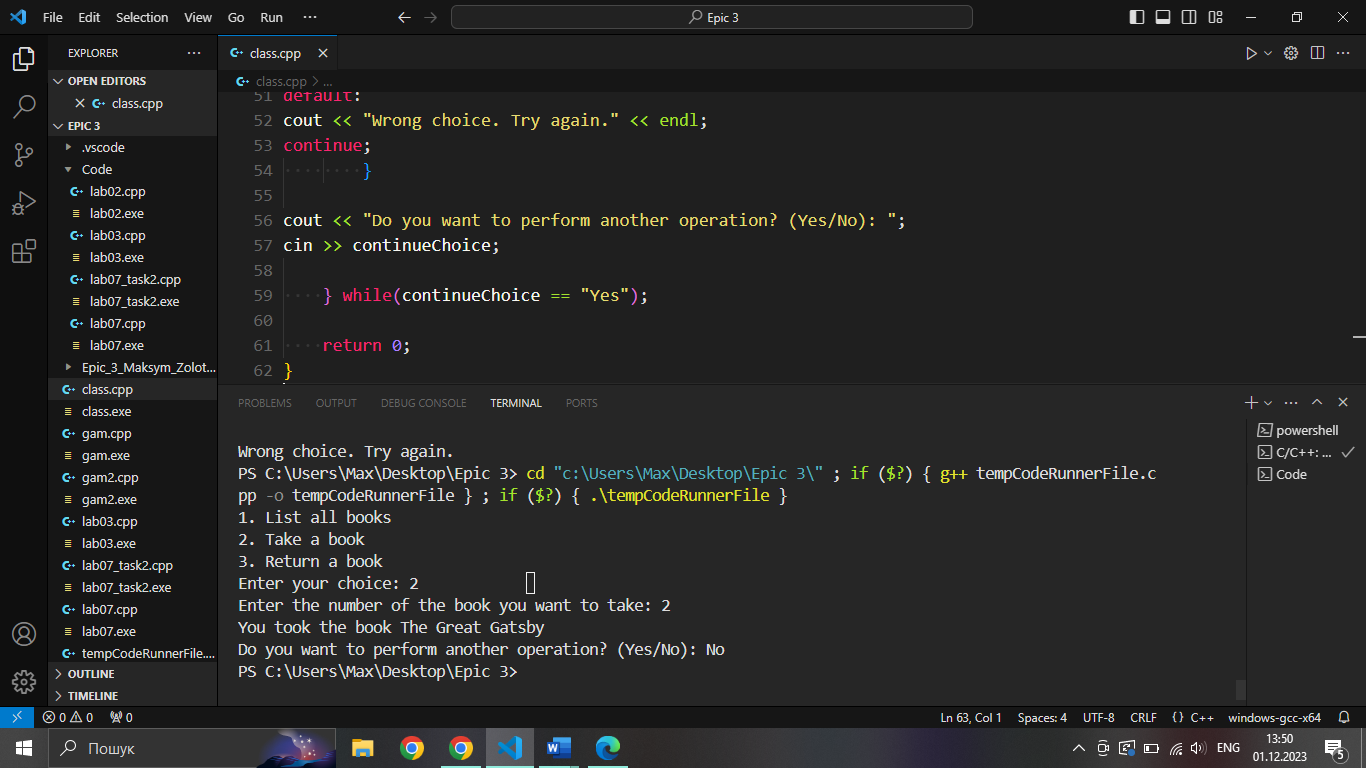


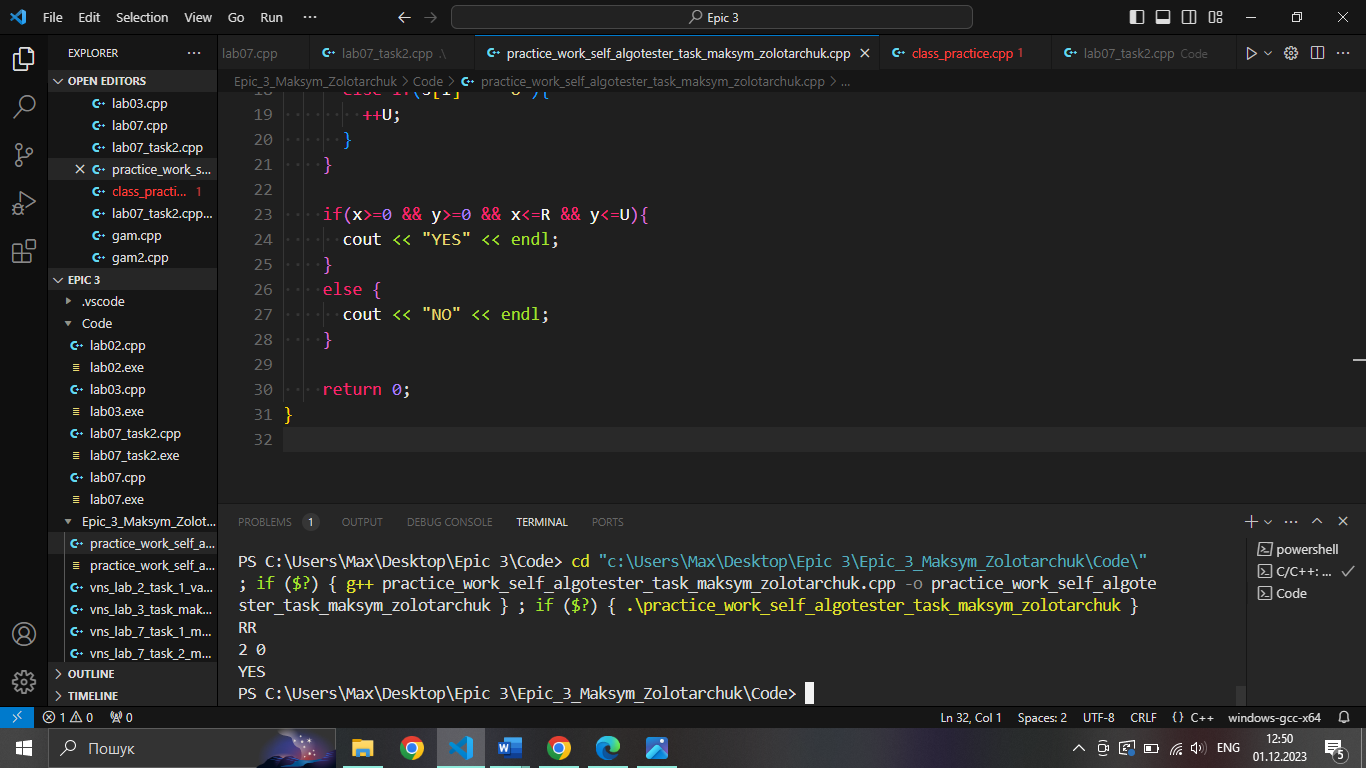
Рисунок 5 Результат завдання №3 VNS Lab №7 Task 2

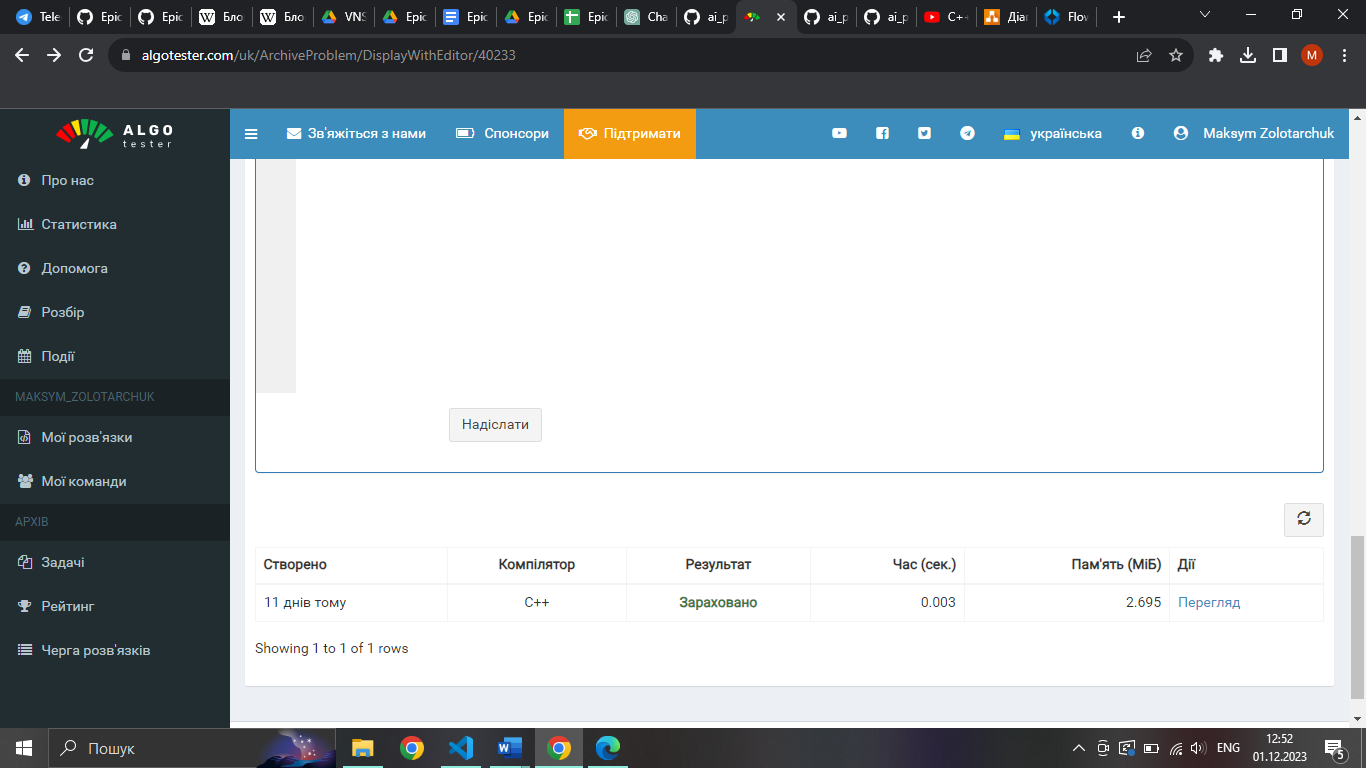
**Завдання №5 Class Practice - Library**



Завдання №5 Class Practice

**Завдання №6 Self Practice – Робот.**





# **<https://github.com/artificial-intelligence-department/ai_programming_playground/tree/epic_3_pactice_and_labs_maksym_zolotarchuk/ai_13/maksym_zolotarchuk/Epic_3_Maksym_Zolotarchuk/Code>**

# **Висновки:**

Я вивчав тему “Цикли. Вкладені цикли. Функції. Перевантаження функцій. Рекурсія” і отримав багато корисних знань та навичок. Це допомогло мені засвоїти основи програмування на C++ та навчитися писати програми. Я навчився працювати з циклами, вкладеними циклами, ітераційними та арифметичними циклами, використовувати оператори переходу для контролю за потоком виконання програми. Це дало мені можливість створювати програми для розв’язання різних задач. У майбутньому я зможу застосовувати отримані знання та навички для ефективного програмування, покращення якості коду і збільшення продуктивності роботи. Цей курс підготував мене до наступного етапу вивчення програмування, і я готовий до нових викликів. Цей звіт показує, що ретельне вивчення теорії, послідовне опрацювання кожної теми та практичне використання знань є важливими умовами успішного навчання програмування. Завдяки цьому досвіду я маю необхідні основи для дальшого вивчення мов програмування C/C++ і розумію, як правильно використовувати ці навички на практиці.