Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**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:**

Студент групи ШІ-13

Сіренко Юрій Сергійович

# **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:** Оволодіння основними концепціями та навичками, пов'язаними з роботою з циклами, вкладеними циклами, ітераційними та арифметичними циклами в програмуванні, а також з розумінням понять функції, її типів та різними способами її задання. Розроблення моделі визначення наближеного значення арифметичної функції шляхом її розкладу на доданки ряду Маклорена. Вивчення перевантажених та рекурсивних функцій, а також функцій з багатьма змінними для розв'язання різноманітних завдань.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли.
* Тема №2: Функції.
* Тема №3: Перевантаження функції.
* Тема №4: Рекурсія.
* Тема №5: Функція зі змінною кількістю параметрів.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли.
  + Джерела Інформації
    - Стаття 1 - [While loop](https://www.w3schools.com/cpp/cpp_while_loop.asp)
    - Стаття 2 - [For loop](https://www.w3schools.com/cpp/cpp_for_loop.asp)
    - Стаття 3 - [Оператори переходу в циклі](https://www.w3schools.com/cpp/cpp_break.asp)
    - Відео 1 - [For Loops In C++ For Beginners](https://www.youtube.com/watch?v=Y0TYYEhHvIU&ab_channel=Simplilearn)
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Властивості циклу.
    - Визначення та властивості вкладених циклів, арифметичних циклів та ітераційних циклів.
    - Використання різних типів задання циклу (for, while, do-while).
    - Контролювання потоку виконання програми операторами переходу.
    - Властивості та типи операторів переходу та їх використання в різних контекстах.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.2023
  + Звершення опрацювання теми: 29.11.2023
* Тема №2: Функції.
  + Джерела Інформації:
    - Стаття 1 - [C++ Functions](https://www.w3schools.com/cpp/cpp_functions.asp)
    - Стаття 2 - [C++ Function Parameters](https://www.w3schools.com/cpp/cpp_function_param.asp)
    - Відео 1 - [C++ Functions: Syntax, Types and Call Methods](https://www.simplilearn.com/tutorials/cpp-tutorial/cpp-functions)
    - Відео 2 - [Functions In C++ (No parameter, Return value, and single parameter)](https://www.youtube.com/watch?v=UbDcN-D8cpY&ab_channel=GutierrezProgramming)
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + 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: 20.11.2023
  + Звершення опрацювання теми: 29.11.2023
* 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/)
    - Курс - Beginning C++ Programming - From Beginner to Beyond
    - ChatGPT
  + 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: 20.11.2023
  + Звершення опрацювання теми: 29.11.2023
* Тема №4: Рекурсія.
  + Джерела Інформації:
    - Книжка.
    - Відео 1 - [Recursion and Recursive Function in C++](https://www.youtube.com/watch?v=nIqgWLpoX68&ab_channel=Simplilearn)
    - Відео 2 - [Functions In C++ (No parameter, Return value, and single parameter)](https://www.youtube.com/watch?v=UbDcN-D8cpY&ab_channel=GutierrezProgramming)
    - Стаття 1 - [C++ Recursion](https://www.w3schools.com/cpp/cpp_functions_recursion.asp).
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Визначення рекурсії, яка дозволяє викликати функції саму себе до настання деяких умов.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.2023
  + Звершення опрацювання теми: 29.11.2023
* Тема №5 Функція зі змінною кількістю параметрів.
  + Джерела Інформації:
    - Відео – 1 [What are variadic functions (va\_list) in C?](https://www.youtube.com/watch?v=oDC208zvsdg&t=312s&ab_channel=CodeVault)
    - Стаття – 1 [Variadic function templates in C++](https://www.geeksforgeeks.org/variadic-function-templates-c/)
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + 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: 20.11.2023
  + Звершення опрацювання теми: 29.11.2023

# **Виконання роботи:**

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

**Завдання №1 Опрацювання теорії**

* Деталі завдання: опрацювання вище перечисленої теорії. Розбір статей, перегляд відео та імплементація теорії на практиці. Вивчення найосновніших тем.

**Завдання №2 VNS Lab №2**

* Варіант завдання – 1

1) Знайти суму цілих додатніх чисел, кратних 3 і менших 200.

* Деталі завдання –

Використовуючи оператор циклу, знайти суму елементів, зазначених у

конкретному варіанті. Результат надрукувати, надавши відповідний

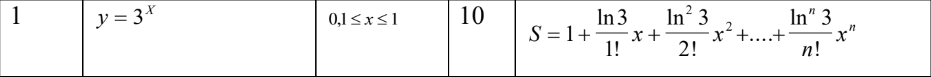
заголовком.

* Важливі деталі для врахування в імплементації програми

Використання циклу While для досягнення суми чисел, кратних 3 та меншим за 200.

**Завдання №3 VNS Lab №3**

* Варіант завдання – 1



* 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).

Для порівняння знайти точне значення функції.

* Важливі деталі для врахування в імплементації програми

Імплементація рядів Маклорена в коді C++, порівняння двох отриманих значень зі справжнім. Використання власної функції обрахування факторіала.

**Завдання №4 VNS Lab №7 Task 1**

* Варіант завдання – 1

1) Написати функцію sum зі змінною кількістю параметрів, що знаходить суму

чисел типу int. Написати викликаючу функцію main, що звертається до

функції sum не менш трьох разів з кількістю параметрів 3, 7, 11.

* 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 – 1

а) для додавання цілих чисел;

б) для додавання комплексних чисел.

Деталі завдання

Написати перевантажені функції й основну програму, що їх викликає.

* Важливі деталі для врахування в імплементації програми

Використання перевантаженої функції та розрізнення типів даних у програмі.

**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 – Офісна Вулиця. Частина 1**

Варіант завдання

[Algotester](https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40792)

* Деталі завдання

Зустрілися якось працівники великих компаній і почали... Обговорювати план вулиці.

Виявляється, всі приміщення, які орендуватимуть ці компанії, збудують вздовж однієї вулиці.

i-та компанія орендуватиме офіс довжиною ii метрів. Офіси будуватимуть один за одним, починаючи з точки 0. Всі працівники приїжджатимуть на стоянку, яку побудують в точці 0, та будуть йти до офісів своїх компаній.

Тобто, якщо офіси будуть збудовані в порядку p1,p2,...,pn, то перший офіс почнеться в точці 0 і закінчиться в точці lp11, другий почнеться в lp11 і закінчиться в lp1+lp2 і т.д. Двері кожного офісу завжди є в кінці будинку, який є ближчим до стоянки.

Ваше завдання — допомогти розмістити офіси компаній на цій вулиці в такому порядку, щоб сумарна відстань від точки 0 до усіх офісів була мінімальною.

# Вхідні дані

У першому рядку задане ціле число n — кількість компаній.

У наступному рядку задано n цілих чисел li через пробіл — довжини офісів усіх компаній.

# Вихідні дані

У єдиному рядку виведіть n чисел від 1 до n — порядок компаній, в якому варто будувати офіси.

Якщо існує декілька оптимальних порядків — виведіть будь-який із них.

# Обмеження:

1≤n≤10^5,

1≤li≤10^4.

* Важливі деталі для врахування в імплементації програми

Вказані обмеження.

Розуміння рекурсії та її використання у функції мердж сортування. Створення

двовимірного масиву для кращого прослідковування офісів на вулиці.

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

**Програма №1 VNS Lab №2**

* Блок-схема

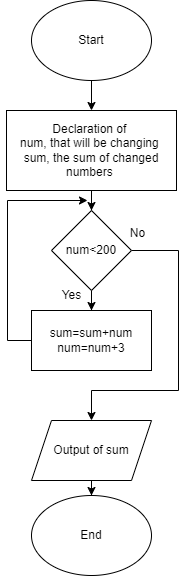


Рисунок Flowchart to VNS Lab №2

* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 20 хвилин
* Важливі деталі для врахування в імплементації

Використання циклу While для досягнення суми чисел, кратних 3 та меншим за 200.

**Програма №2 VNS Lab №3**

* Блок-схема - відсутня
* Планований час на реалізацію – 2 години
* Час затрачений на виконання завдання – 45 хвилин
* Важливі деталі для врахування в імплементації

Імплементація рядів Маклорена в коді C++, порівняння двох отриманих значень зі справжнім. Використання власної функції обрахування факторіала.

**Програма №3 VNS Lab №7 Task 1**

* Блок-схема - відсутня
* Планований час на реалізацію – 3 години
* Час затрачений на виконання завдання – 1 година 20 хвилин
* Важливі деталі для врахування в імплементації

Використання перевантаження функції та робота з варіативною функцією va\_list

**Програма №4 VNS Lab №7 Task 2**

* Блок-схема - відсутня
* Планований час на реалізацію – 3 години
* Час затрачений на виконання завдання – 1 година 40 хвилин
* Важливі деталі для врахування в імплементації

Використання перевантаженої функції та розрізнення типів даних у програмі.

**Програма №5 Class Practice - Library**

* Блок-схема – відсутня
* Планований час на реалізації – 3 години
* Час затрачений на виконання завдання – 6 годин
* Важливі деталі для врахування в імплементації

Вказані вимоги.

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

* Блок-схема

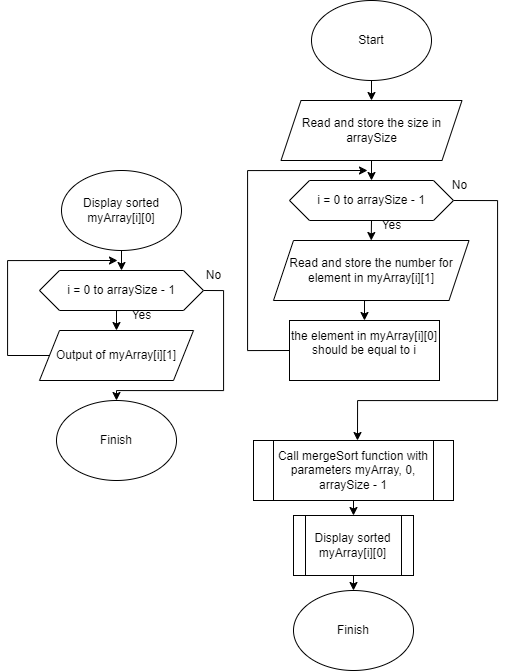


Рисунок Flowchart to Self Practice Main Function

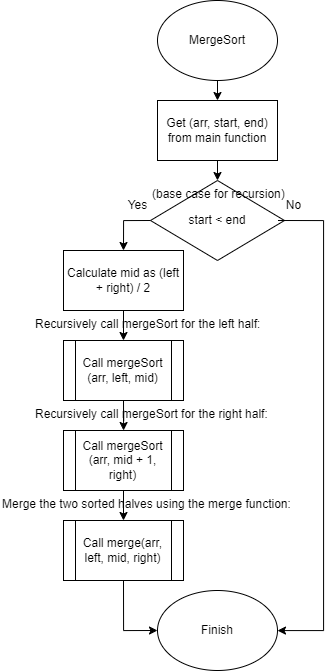


Рисунок Flowchart to Self Practice Merge Sort Function

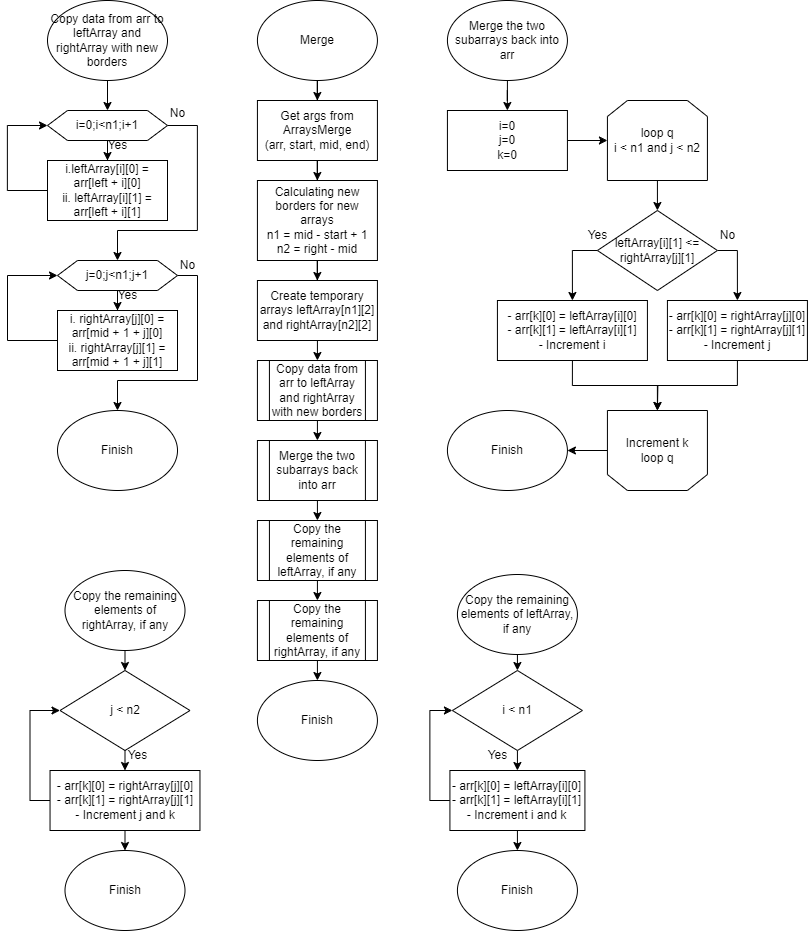


Рисунок Flowchart to Self Practice Merge Function and it's components

* Планований час на реалізацію – 6 годин
* Час затрачений на виконання завдання – 2 доби
* Важливі деталі для врахування в імплементації

Вказані обмеження

Розуміння рекурсії та її використання у функції мердж сортування. Створення двовимірного масиву для кращого прослідковування офісів на вулиці.

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

**Завдання №1 - VNS Lab №2**

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/562/files#diff-5b9a453c0986daf8412e59745df4365fa68e6c0465dd66d47a8e7e199cf7e8a3>

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

    int sum = 0;

    int num = 0;

    while(num<200)

    {

        sum+=num;

        num+=3;

    }

    cout <<"The sum of positive integers that\nare "

            << "multiples of 3 and less than 200 is: \n"<< sum;

    return 0;

}

*Код програми №1 VNS Lab №2*

**Завдання №2 - VNS Lab №3**

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/562/files#diff-2453299f56003e736f394931382931a8435f3ed5eec47aa57fddba67bf35b286>

#include <iostream>

#include <cmath>

using namespace std;

int factorial(int f)

{

    int result=1;

    for(int i = 1;i<=f;i++)

    {

        result \*=i;

    }

    return result;

}

int main()

{

    double s = 1;

    for(double x =0.1;x<=1;x+=(1-0.1)/10)

    {

        cout<<"X="<<x<<" ";

        double s = 1;

        for(int n = 1;n<=10;n++)

        {

            s+=(log(pow(3,n))\*x)/factorial(n);

        }

        cout<<"SN="<<s<<" ";

        double t = 1;

        for(int n = 1;(log(pow(3,n))\*x)/factorial(n)>0.0001;n++)

        {

            t+=(log(pow(3,n))\*x)/factorial(n);

        }

        cout<<"SE="<<t<<" ";

        cout<<"Y="<<pow(3,x)<<"\n";

    }

    return 0;

}

*Код програми №2 VNS Lab №3*

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

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/562/files#diff-b68d77cdbf746b2d101e5ee08984696e050b21c3eb862baf65592cc38e63fc9e>

#include <cstdarg>

#include <iostream>

using namespace std;

int sum(int n, ...)

{

    int s = 0;

    va\_list array;

    va\_start(array, n);

    for(int i = n;i!=0;i--)

    {

        int x = va\_arg(array, int);

        s += x;

    }

    va\_end(array);

    return s;

}

int main()

{

    cout<<sum(3,1,2,3)<<"\n";

    cout<<sum(7,1,2,3,4,5,6,7)<<"\n";

    cout<<sum(11,101,202,303,404,504,606,707,808,909,1010,1111)<<"\n";

    return 0;

}

*Код програми №3 VNS Lab №7 Task 1*

**Завдання №4 VNS Lab №7 Task 2** Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/562/files#diff-0716c5d1dc1f8ac8cdb5a2f16c8ab1294e7c255f0059e83ad3d18802fa478801>

#include <iostream>

#include <cmath>

using namespace std;

struct complex

{

    int real;

    int imaginary;

};

int sum(int a,int b)

{

    return(a+b);

}

complex sum(complex a,complex b)

{

    complex result;

    result.real=a.real+b.real;

    result.imaginary=a.imaginary+b.imaginary;

    return result;

}

int main()

{

    string answer;

    cout<<"Enter, what type of number you want to add:\n1) real\n2) complex\n";

    cin>>answer;

    if(answer=="real")

    {

        int x,y;

        cout<<"Enter numbers you want to add:\n";

        cin>>x>>y;

        cout<<"Sum = "<<sum(x,y)<<"\n";

    }

    else if(answer=="complex")

    {

        int x,xi,y,yi;

        cout <<"Enter values in real and imaginary world for 2 numbers\n";

        cin>>x>>xi>>y>>yi;

        complex c1={x, xi};

        complex c2={y, yi};

        complex complexsum = sum(c1,c2);

        cout<<"Complex sum = "<<complexsum.real<< " + "<<complexsum.imaginary<<"i\n";

    }

    return 0;

}

*Код програми №4 VNS Lab №7 Task 2*

**Завдання №5 Class Practice – Library**

Деталі по програмі:

Посилання на файл програми у пул-запиті GitHub

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/562/files#diff-fc3f53f3c7003ea55223d2066fc478ecc177fb58800392a4ebc5f201fbc5b80a>

#include <iostream>

#include <vector>

using namespace std;

int main()

{

    string option;

    string book;

    vector<string>list\_initial;

    vector<string>list={"book1","book2","book3","book4"};

    for(int i = 0;i<list.size();i++)

    {

        list\_initial.push\_back(list[i]);

    }

    list = list\_initial;

    cout<<"Welcome to the Yura's library.";

    while(option != "exit")

    {

        menu:

        cout<<"\nChoose option to do: \n -(list) of books\n -(take) book\n -(return) book \n -(exit)\n";

        cin>>option;

        if(option == "list")

        {

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

            {

                cout <<" -"<< list[i]<<"- \n";

            }

            cout << "\nenter - to move to main menu\nenter (exit) to exit library\n";

            cin>>option;

            if(option == "exit")

            {

                break;

            }

            goto menu;

        }

        if(option == "take")

        {

            cout<<"What book do you want to take?\n";

            cin>>book;

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

            {

                if(book == list[i])

                {

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

                    cout<<"Succes. Now book -"<<book<<"- is yours!\nenter - to move to main menu\nenter (exit) to exit library\n";

                    cin>>option;

                    if(option == "exit")

                    {

                        break;

                    }

                    goto menu;

                }

            }

            for(int i = 0;i<list\_initial.size();i++)

            {

                if(book == list\_initial[i])

                {

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

                    cout<<"Sorry, there is no -"<<book<<"- at this moment.\nenter - to move to main menu\nenter (exit) to exit library\n";

                    cin>>option;

                    if(option == "exit")

                    {

                        break;

                    }

                    goto menu;

                }

            }

            cout<<"There is no -"<<book<<"- in our library at all.\n\n\nenter - to move to main menu\nenter (exit) to exit library\n";

            cin>>option;

            if(option == "exit")

            {

                return 0;

            }

            goto menu;

        }

        if(option == "return")

        {

            cout<<"What book do you want to return?\n";

            cin>>book;

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

            {

                if(book == list[i])

                {

                    cout<<"It looks like this book is in library right now\n";

                    goto menu;

                }

            }

            for(int i = 0;i<list\_initial.size();i++)

            {

                if(book == list\_initial[i])

                {

                    list.push\_back(book);

                    cout<<"Succes. Now book -"<<book<<"- is in library again!\nenter - to move to main menu\nenter (exit) to exit library\n";

                    cin>>option;

                    if(option == "exit")

                    {

                        break;

                    }

                    goto menu;

                }

            }

            cout<<"Do you want to share with us your new book?\n type (y) for yes, (n) for no\n";

            cin>>option;

            if(option == "y")

            {

                list.push\_back(book);

                list\_initial.push\_back(book);

                cout<<"Succes. Now book -"<<book<<"- is in library again!\nenter - to move to main menu\nenter (exit) to exit library\n\n";

                cin>>option;

                if(option == "exit")

                {

                    break;

                }

                goto menu;

            }

            else if(option == "n")

            {

                goto menu;

            }

        }

    }

}

*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>

using namespace std;

void sortArray(int array[][2] ,int start,int mid, int end)

{

    int left\_array\_size=mid-start+1;

    int right\_array\_size=end-mid;

    int left\_array[left\_array\_size][2];

    int right\_array[right\_array\_size][2];

    for(int i = 0;i<left\_array\_size;i++)

    {

        left\_array[i][0]=array[i+start][0];

        left\_array[i][1]=array[i+start][1];

    }

    for(int i = 0;i<right\_array\_size;i++)

    {

        right\_array[i][0]=array[i+mid+1][0];

        right\_array[i][1]=array[i+mid+1][1];

    }

    int left\_itter=0;

    int right\_itter=0;

    int main\_itter=start;

    while(left\_itter < left\_array\_size && right\_itter < right\_array\_size)

    {

        if(left\_array[left\_itter][0]<=right\_array[right\_itter][0])

        {

            array[main\_itter][0]=left\_array[left\_itter][0];

            array[main\_itter][1]=left\_array[left\_itter][1];

            left\_itter++;

        }

        else

        {

            array[main\_itter][0]=right\_array[right\_itter][0];

            array[main\_itter][1]=right\_array[right\_itter][1];

            right\_itter++;

        }

        main\_itter++;

    }

    while(left\_itter < left\_array\_size)

    {

        array[main\_itter][0]=left\_array[left\_itter][0];

        array[main\_itter][1]=left\_array[left\_itter][1];

        left\_itter++;

        main\_itter++;

    }

    while(right\_itter < right\_array\_size)

    {

        array[main\_itter][0]=right\_array[right\_itter][0];

        array[main\_itter][1]=right\_array[right\_itter][1];

        right\_itter++;

        main\_itter++;

    }

}

void sort(int array[][2],int start, int end)

{

    if(start<end)

    {

        int mid = start +(end-start)/2;

        sort(array,start,mid);

        sort(array,mid+1,end);

        sortArray(array,start,mid,end);

    }

}

int main()

{

    int n;

    cin >> n;

    int street[n][2];

    for (int i = 0; i < n; i++)

    {

        cin >> street[i][0];

    }

    for (int i = 0; i < n; i++)

    {

        street[i][1] = i;

    }

    sort(street, 0, n-1 );

    for (int i = 0; i < n; ++i)

    {

        cout << street[i][1]+1 << " ";

    }

    return 0;

}

*Код програми №6 Self Practice – Офісна Вулиця. Частина 1*

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

**Завдання №1 VNS Lab №2** Програма виконує лише одну функцію - обрахування суми чисел, кратних 3 та меншим за 200.



Рисунок Результат завдання №1 VNS Lab №2

**Завдання №2** **VNS Lab №3** Програма виконує лише одну функцію – обрахування функції y=3^x

Деталі по виконанню і тестуванню програми:

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

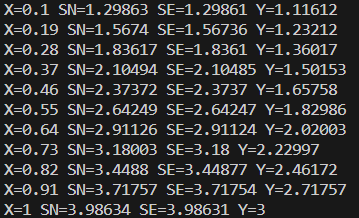


Рисунок Результат завдання №2 VNS Lab №3

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

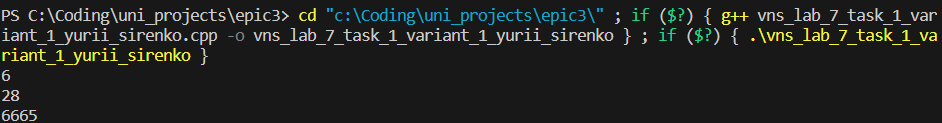


Рисунок Результат завдання №3 VNS Lab №7 Task 1

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

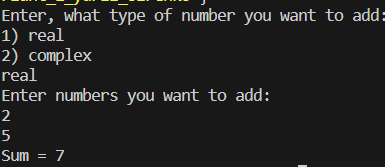


Рисунок Результат завдання №3 VNS Lab №7 Task 2 у випадку додавання справжніх чисел

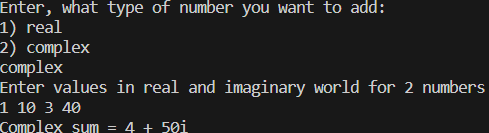
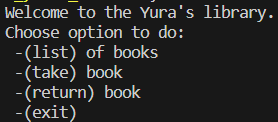


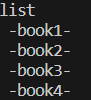
Рисунок Результат завдання №3 VNS Lab №7 Task 2 у випадку додавання уявних чисел

Важливо! Обрання типу чисел на початку використовується лише для правильного виведення результату, на обрання функції воно ніяк не впливає!

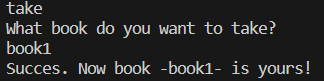
**Завдання №5 Class Practice - Library** Деталі по виконанню і тестуванню програми



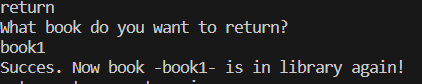
Завдання №5 Class Practice - Library Головне меню



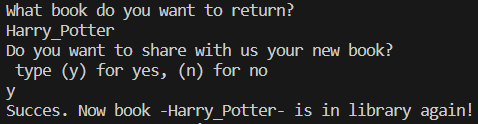
Завдання №5 Class Practice - Library Список наявних книжок



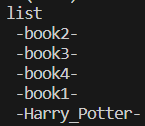
Завдання №5 Class Practice - Library Опція Взяття книги



Завдання №5 Class Practice - Library Опція повернення книги



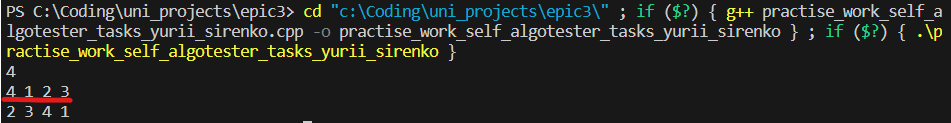
Завдання №5 Class Practice - Library Опція повернення неіснуючої книги (додавання нової)

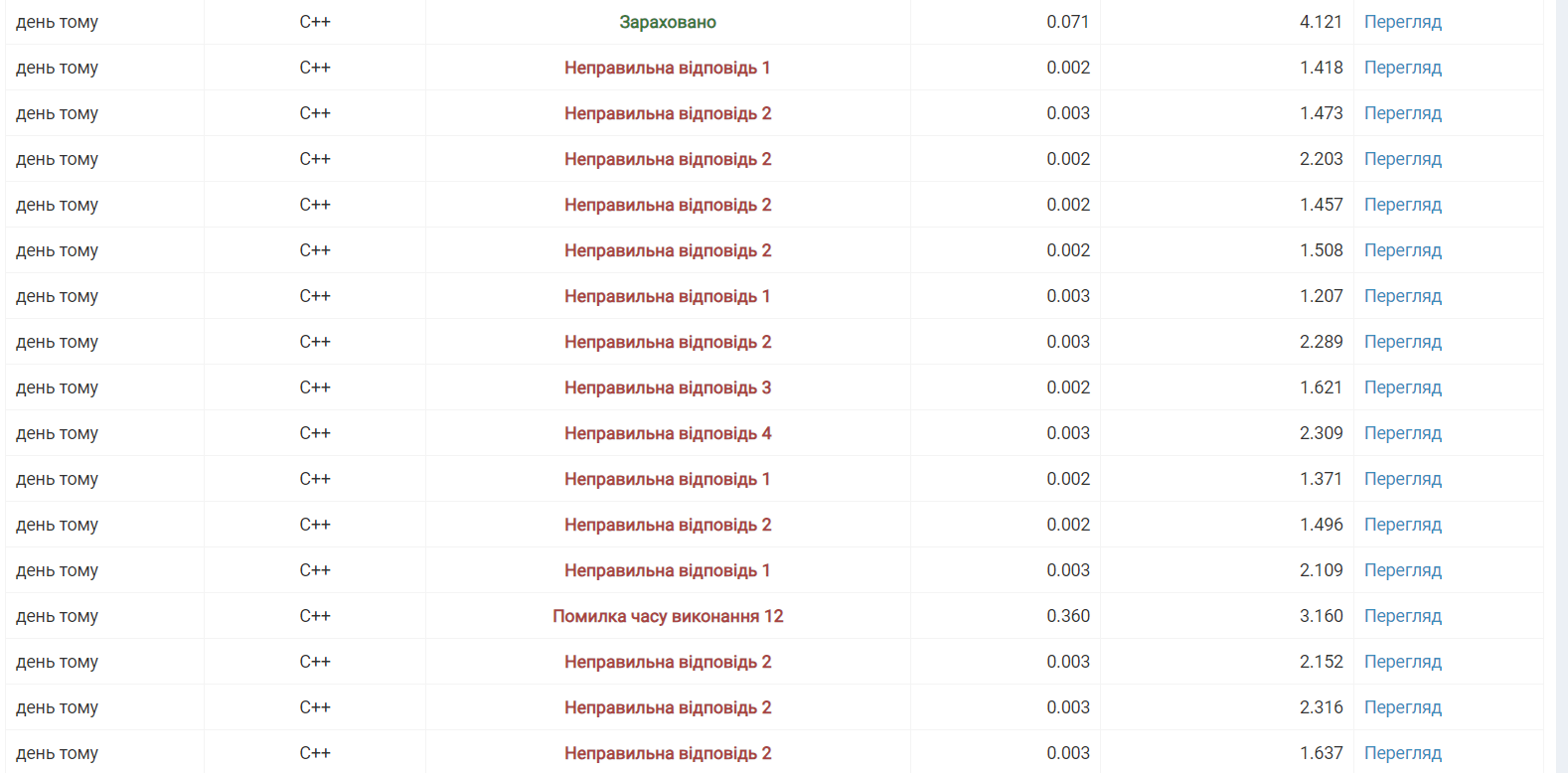


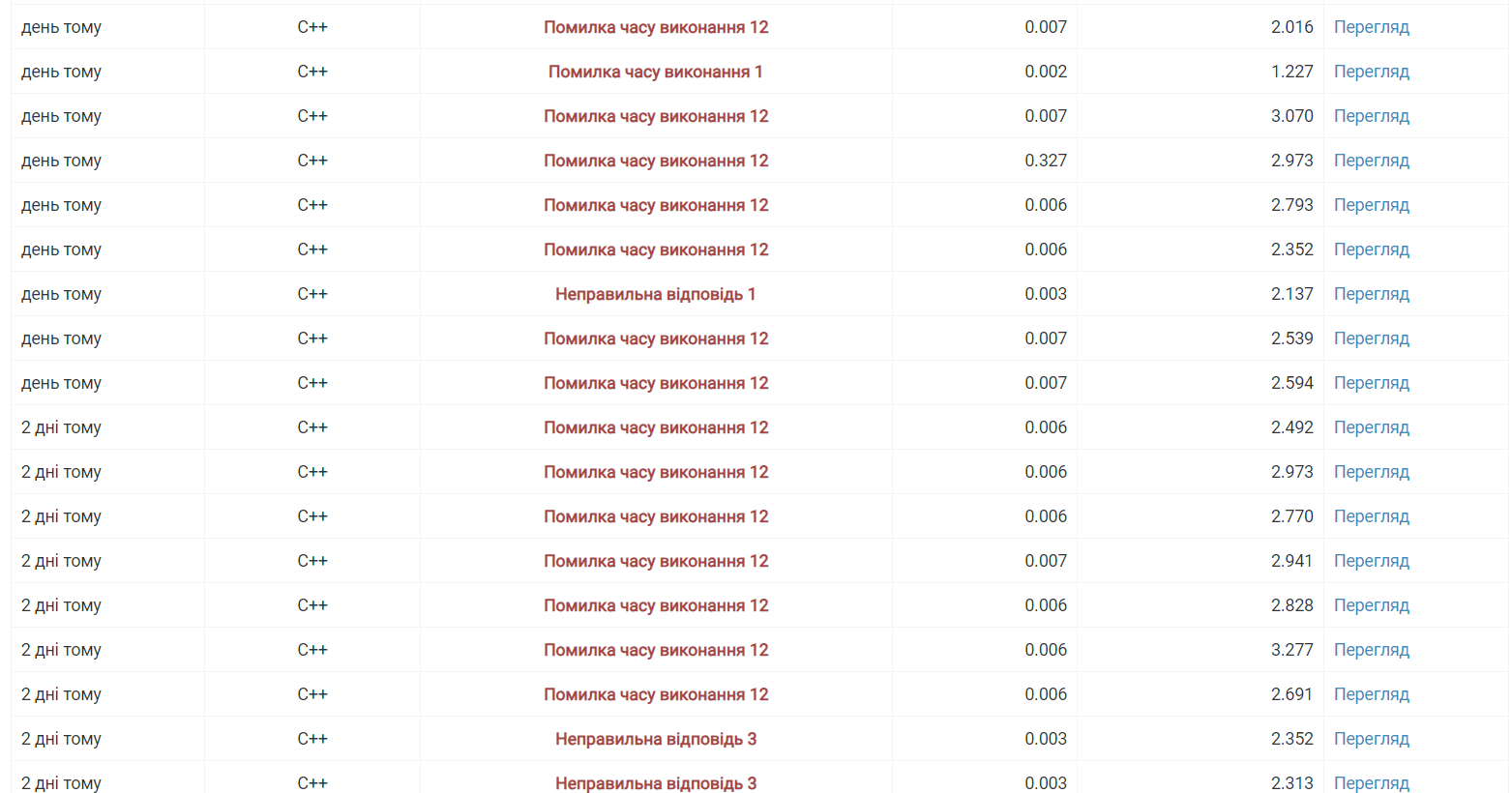
Завдання №5 Class Practice - Library Результат операцій

By -exit- program is returning to 0 and ending.

**Завдання №6 Self Practice – Офісна Вулиця. Частина 1** Деталі по виконанню і тестуванню програми







# **Висновки:**

Під час вивчення теми "Цикли. Вкладені цикли. Функції. Перевантаження функцій. Рекурсія ", я здобув глибокі знання та отримав практичні навички. Це допомогло мені освоїти основи програмування на мові C++ та набути впевненості у написанні програм.

Робота з з циклами, вкладеними циклами, ітераційними та арифметичними циклами, контролювання потоку виконання програми операторами переходу дала мені розуміння того, як ефективно структурувати програми для розв'язання різноманітних завдань.

Застосування отриманих знань та навичок у майбутньому дозволить мені ефективно вирішувати завдання програмування, покращувати якість коду і підвищувати продуктивність роботи. Цей курс підготував мене до подальшого вивчення програмування, і я з нетерпінням чекаю нових викликів.

Цей звіт демонструє, що систематичне вивчення теоретичного матеріалу, поетапна обробка кожної теми та практичне застосування отриманих знань є важливим етапом успішного освоєння програмування. Завдяки цьому досвіду я маю необхідну базу для подальшого вивчення мов програмування C/C++ і розумію, як ефективно використовувати ці навички на практиці.