Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 2,3**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Цикли. Вкладені цикли. Функції. Перевантаження функцій. Рекурсія»

***Виконав(ла):***

студент групи ШІ-12

Ширяєв Іван

# **Тема роботи:**

Опрацювання циклів, функцій багатьох змінних, перевантаження та рекурсії функцій в мові C/C++.

# **Мета роботи:**

Навчитись використовувати різні види циклів, функції багатьох змінних, перевантаження та рекурсію і комбінації цих засобів для розв’язання практичних завдань на мові C/C++.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли.
* Тема №2: Функції.
* Тема №3: Перевантаження функцій.
* Тема №4: Рекурсія функцій.
* Тема №5: Функції багатьох змінних.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли.
  + Джерела Інформації
    - <https://www.geeksforgeeks.org/cpp-loops/>
    - <https://www.w3schools.com/cpp/cpp_for_loop.asp>
    - <https://www.programiz.com/cpp-programming/for-loop>
  + Що опрацьовано:
    - Опрацьовано різні типи циклів такі як : for, while, do while, for range, їх особливості, властивості і кейси використання.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.11.2023
  + Звершення опрацювання теми: 12.11.2023
* Тема №2: Функції.
  + Джерела Інформації:
    - <https://www.w3schools.com/cpp/cpp_functions.asp>
    - <https://www.geeksforgeeks.org/functions-in-cpp/>
    - <https://www.youtube.com/watch?v=V9zuox47zr0>
  + Що опрацьовано:
    - Опрацьовано самі по собі функції, як та для чого ними потрібно користуватися
    - Опрацьовано типи функцій (void, int, string тощо.) їх типи повернення
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.11.2023
  + Звершення опрацювання теми: 12.11.2023
* Тема №3: Перевантаження функцій.
  + Джерела Інформації:
    - <https://www.geeksforgeeks.org/function-overloading-c/>
    - <https://www.programiz.com/cpp-programming/function-overloading>
  + Що опрацьовано:
    - Опрацьовано перегрузку функцій та кейси її використання.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 13.11.2023
  + Звершення опрацювання теми: 15.11.2023
* Тема №4 Рекурсія функцій.
  + Джерела Інформації:
    - <https://www.w3schools.com/cpp/cpp_functions_recursion.asp>
    - <https://www.geeksforgeeks.org/cpp-recursion/>
    - <https://www.youtube.com/watch?v=MwfvXDfaZeI>
  + Що опрацьовано:
    - Опрацьована рекурсія, кейси її використання.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 05.11.2023
  + Звершення опрацювання теми: 06.11.2023
* Тема №5: Функції багатьох змінних
  + Джерела Інформації
    - <https://en.cppreference.com/w/c/variadic>
    - <https://www.geeksforgeeks.org/variadic-function-templates-c/>
    - <https://www.youtube.com/watch?v=3iX9a_l9W9Y>
  + Що опрацьовано:
    - Функції багатьох змінних, va\_list, для чого потрібні, ситуації використання.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.2023
  + Звершення опрацювання теми: 25.11.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 2 - Task 1-1

* Варіант 1
* Деталі завдання

Знайти суму цілих додатних чисел, кратних 3 і менших 200.

* Важливі деталі для врахування в імплементації програми

Використовуємо int для знаходження кількості чисел кратних 3 з 200 бо він відкидає дробову частину.

Завдання №2 VNS Lab 3 - Task 1-1

* Варіант 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.

Моя функція : 3^x

* Важливі деталі для врахування в імплементації програми

Всі значення записуємо в double для максимальної точності обчислень, використовуємо бібліотеку cmath для функції pow(), та рекурсію.

Завдання №3 VNS Lab 7 - Task 1-1

* Варіант 1
* Деталі завдання

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.

* Важливі деталі для врахування в імплементації програми

Для виконання завдання використано va\_list.

Завдання №4 VNS Lab 7 - Task 1-2

* Варіант завдання 1
* Деталі завдання

Написати перевантажені функції й основну програму, що їх викликає.

а) для додавання цілих чисел;

б) для додавання комплексних чисел.

* Важливі деталі для врахування в імплементації програми

Використовуємо бібліотеку complex

Завдання №5 Class practice work

* Варіанту немає
* Деталі завдання

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

* Важливі деталі для врахування в імплементації програми

Необхідно використати різні цикли та go to.

Завдання №6 Self Practice work

* Варіанту немає
* Деталі завдання

Потрібно знайти чи зустрічається слово в string введеному користувачем певну кількість разів.

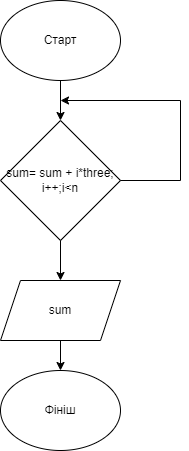
* Важливі деталі для врахування в імплементації програми

Використовуємо функцію find(), та size().

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

Програма №1 VNS Lab 2 - Task 1-1

* Блок-схема

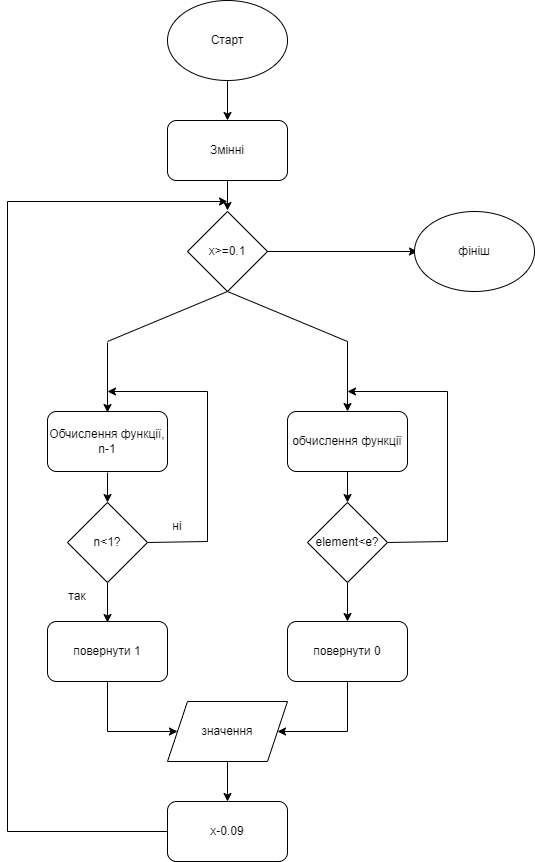


*Блок-схема VNS Lab 2 - Task 1-1*

* Планований час на реалізацію: 1 година.

Програма №2 VNS Lab 3 - Task 1-1

* Блок-схема

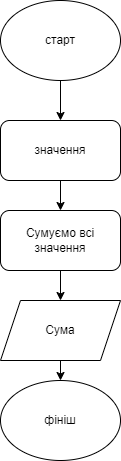


*Блок-схема VNS Lab 3 - Task 1-1*

* Планований час на реалізацію: 2 години

Програма №3 VNS Lab 7 - Task 1-1

* Блок-схема

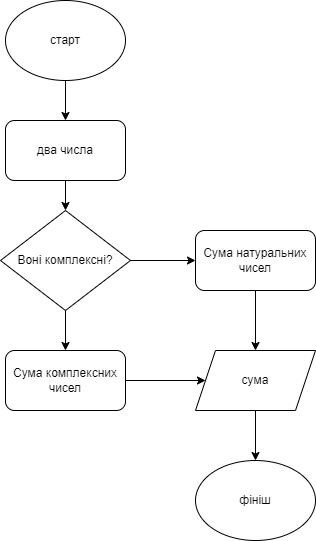


*Блок-схема VNS Lab 7 - Task 1-1*

* Планований час на реалізацію: 1 година

Програма №4 VNS Lab 7 - Task 1-2

* Блок-схема

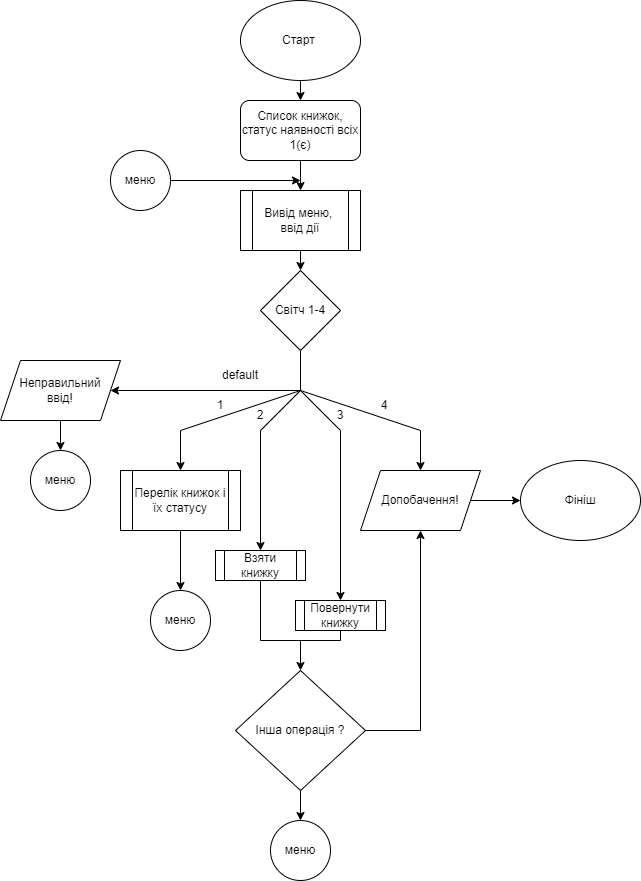


*Блок-схема VNS Lab 7 - Task 1-2*

* Планований час на реалізацію: 1 година

Програма №5 Class practice work

* Блок-схема

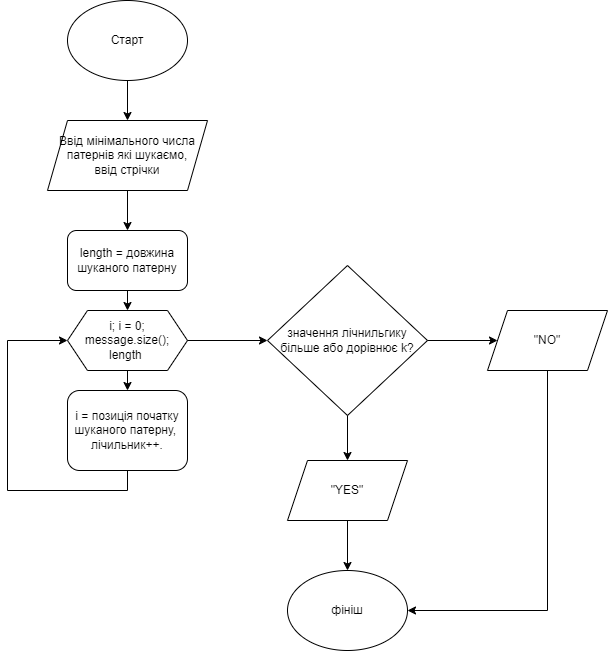


*Блок-схема Class practice work*

* Планований час на реалізацію: 3 години

Програма №6 Self Practice work

* Блок-схема



*Блок-схема Self Practice work*

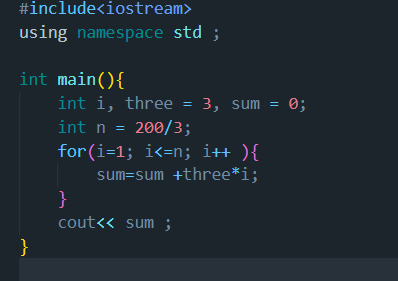
* Планований час на реалізацію: 1 година

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

Додаткова конфігурація не потребується.

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

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

*код VNS Lab 2 - Task 1-1*

Знаходимо кількість чисел кратних 3 з 200, це 200/3, через те що використано тип даних int дробова частина відкидається і отримуємо ціле число, потім за допомогою циклу for знаходимо суму всіх цих чисел.

Завдання №2 Деталі по програмі. Посилання на файл програми у пул-запиті GitHub

*код VNS Lab 3 - Task 1-1*

Спочатку створюємо функцію що буде рекурсивно розраховувати факторіал так як це буде необхідно для подальших обчислень, після цього є дві функції і обидві також рекурсивні, перша - обчислює суму ряду рівно 10 разів, друга - обчислює суму ряду до певної точності, створюємо цикл що буде паралельно виводити значення цих функцій для x що змінює значення від 1 до 0.1 з кроком 0.09. Так само виводимо значення з цим аргументом для функції 3^х для порівняння.

Завдання №3 Деталі по програмі. Посилання на файл програми у пул-запиті GitHub

*код VNS Lab 7 - Task 1-1*

Тут створюємо функцію зі змінною кількістю параметрів за допомогою va\_list, numberofelements це аргумент що задає кількість параметрів функції, створюємо сам va\_list, ініціалізуємо його за допомогою va\_start, перебираємо введені значення та сумуємо їх за допомогою va\_arg, завершуємо обробку va\_end, виводимо суму.

Завдання №4 Деталі по програмі. Посилання на файл програми у пул-запиті GitHub

*код VNS Lab 7 - Task 1-2*

Створюємо дві функції з однаковою назвою, тобто перевантажуємо їх, для того щоб з однією функцією можна було знаходити суму як натуральних так і комплексних чисел, тип другої функції має бути complex<int> через те що вона повертає саме комплексне значення.

Завдання №5 Деталі по програмі. Посилання на файл програми у пул-запиті GitHub

*код для Class practice work частина 1*

*код для Class practice work частина 2* 

Для того щоб не засмічувати switch case великою кількістю коду я вирішив винести дії в бібліотеці в окремі функції, функцію mainmenu використовуємо в якості аргументу для світч кейсу бо вона не тільки виводить текст а ще й приймає і повертає значення яке ми будемо використовувати для навігації по різним випадкам, перша опція - перерахувати книжки і її статус, тут використано for range цикл для виводу назв книжок та тернарний оператор що з’ясовує чи є книжка в наявності чи ні, після цього знову виводимо меню за допомогою go to що повертає нас до світч кейсу, функції забрати і повернути книжку майже однакові, вони беруть вектор що зберігає статус книжок, просить значення у користувача і змінює статус наявності відповідно до введеного індексу книжки, також ці функції перевіряють чи правильно введено індекс книжки а також якщо наприклад користувач хоче повернути книгу яка є в наявності виводить повідомлення про те що цього зробити не можна, у цих функцій тип вектору бо вони повертають відповідний змінений вектор.Останнім варіантом є вихід, він завершує програму. Також за допомогою циклу do while після кожної операції виводиться дві опції: продовжити роботу або закінчити, якщо y то робота продовжується, знов повертаємось до меню, n - кінець, інакше - чекаємо правильного вводу.

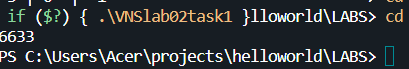
Завдання №6 Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

*код Self practice algotester* 

Спочатку зчитуємо мінімальну кількість патернів що шукаємо в стрічці, потім - саму стрічку, після цього за допомогою функції find() шукаємо патерн, функція повертає положення першого елементу тому цикл while саме в рамках розміру самої стрічки, якщо значення більше довжини - завершуємо цикл. Всередині самого циклу рахуємо кількість знайдених патернів та зміщуємо параметр на довжину самого патерна щоб функція продовжувала шукати, після цього якщо кількість патернів більше дорівнює k виводимо “YES” інакше - “NO”.

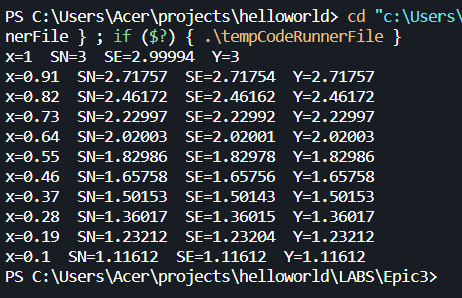
## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 Деталі по виконанню і тестуванню програми

*Результат VNS Lab 2 - Task 1-1*

Час затрачений на виконання завдання: 40 хвилин

Завдання №2 Деталі по виконанню і тестуванню програми



*Результат VNS lab 3 - Task 1-1*

Для суми до певної точності є невелика похибка через відсутність в формулі залишкового члена.

Час затрачений на виконання завдання: 3 години

Завдання №3 Деталі по виконанню і тестуванню програми



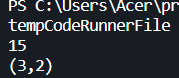
*Результат VNS lab 7 - Task 1-1*

перше це сума 3 чисел 1 2 1, друге - сума 6 чисел 1, 1, 1, 1, 1, 1, 0, а третє - сума 11 чисел

1,1,1,4,5,7,2,3,3,3,11.

Час затрачений на виконання завдання:40 хвилин

Завдання №4 Деталі по виконанню і тестуванню програми

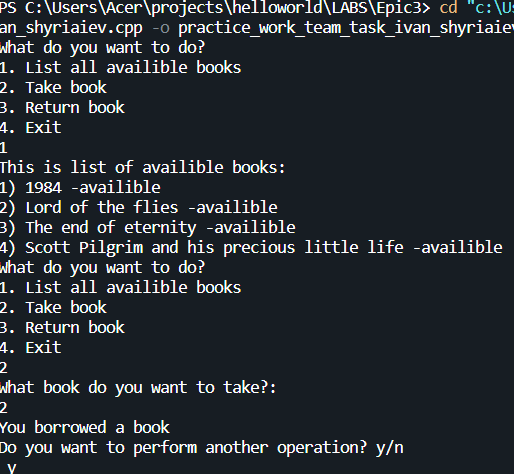


*Результат VNS lab 7 - Task 2-1*

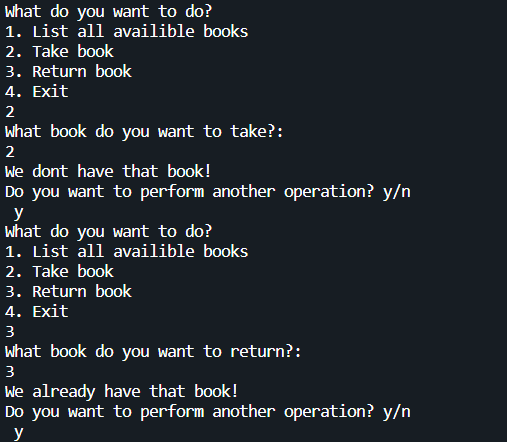
Час затрачений на виконання завдання: 40 хвилин

Перше число - сума 10 і 5, друге - сума (1,-2) і (2,4) тобото 1-2і та 2+4і.

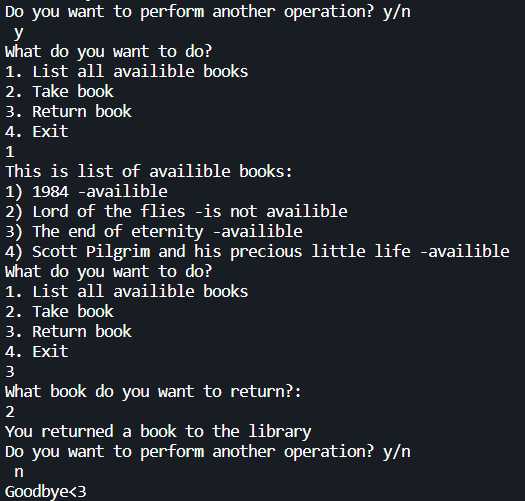
Завдання №5 Деталі по виконанню і тестуванню програми



*результат Class practice work 1/3*



*результат Class practice work 2/3*

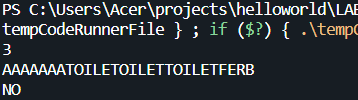


*результат Class practice work 3/3*

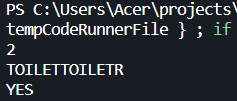
Хід програми детально розписано вище, тут нічого неординарного.

Час затрачений на виконання завдання: 2 дні

Завдання №6 Деталі по виконанню і тестуванню програми

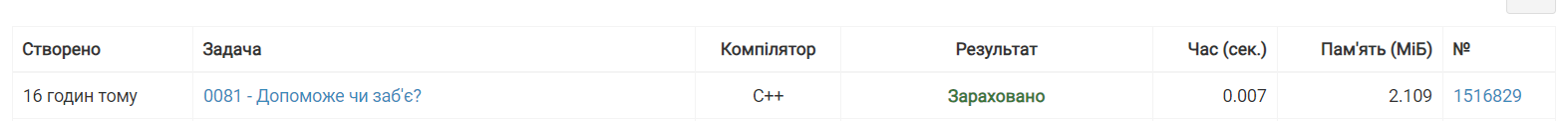


*Результат Self practice 1/2*

**

*Результат Self practice 2/2*

Шукаємо патерн “TOILET”, в першому “ні” через те що патерн зустрічається тільки 2 рази з мінімум 3,а в другому “так” бо патерн зустрічається мінімально допустиму кількість разів.



*Результат на алготестері*

# **Висновки:**

Після виконання завдань для цього епіку я навчився користуватися різними видами циклів, використовувати функції різної кількості змінних, перевантажувати їх, дізнався що таке рекурсія в мові C++ та як її можна використовувати для розв’язання поставлених задач.