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****:* «Основи програмування»

до:

ВНС Лабораторної Роботи № 2

ВНС Лабораторної Роботи № 3

ВНС Лабораторної Роботи № 7

Практичних Робіт до блоку № 3

**Виконала:**

Студент групи ШІ-11

Боднар Денис

**Тема роботи:** Ознайомлення з циклами та їх видами: for, while, do-while, та їх значенням у програмуванні. Вивчення способів управління виконанням циклів за допомогою операторів break і continue, а також передчасного завершення циклів. Аналіз вкладених циклів та їх застосування для вирішення складних завдань. Ознайомлення з основами функцій у C++: визначення, параметри, повернення значень, перевантаження функцій та робота з просторами імен. Вивчення розширених можливостей функцій, таких як еліпсис для змінної кількості параметрів, рекурсія, та використання вбудованих функцій для оптимізації коду.

**Мета роботи:** Ознайомитися з основними типами циклів у C++ (for, while, do-while), їхньою роллю та синтаксисом, а також навчитися застосовувати їх у різних завданнях. Навчитися використовувати оператори break та continue, розуміти умови завершення та передчасне завершення циклів. Засвоїти принципи створення функцій, включаючи передавання параметрів, значення за замовчуванням та повернення результатів. Розібратися з поняттям перевантаження функцій та використанням просторів імен для організації коду. Опанувати роботу з функціями зі змінною кількістю параметрів, а також принципами рекурсії та їхнього застосування у складних задачах. Ознайомитися з вбудованими функціями у C++ та їх роллю у спрощенні коду й виконанні стандартних операцій. Навчитися використовувати вкладені цикли та практично застосовувати їх для вирішення комплексних завдань у програмуванні.

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Введення в цикли та їх види в С++. Вкладені цикли.
* Тема № 2: Управління виконанням циклів.
* Тема № 3: Основи Функцій у С++.
* Тема № 4:  Перевантаження Функцій та Простір Імен.
* Тема № 5: Функції зі змінною кількістю параметрів (еліпсис).
* Тема № 6: Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.
* Тема № 7: Вбудовані Функції в С++.

1. Індивідуальний план опрацювання теорії:

Тема №1: Введення в цикли та їх види в С++. Вкладені цикли.

* + Джерела:
  + <https://acode.com.ua/urok-72-tsykl-for>
  + <https://acode.com.ua/urok-70-tsykl-while/>
  + <https://acode.com.ua/urok-69-operator-goto/>
    - Що опрацьовано:
      * Область видимості циклу.
      * Помилка неврахованої одиниці.
      * Пропуск виразів в циклі for та запис в форматі for ( ; i<10; )
      * Тип signed int для лічильників циклу.
    - Статус: Ознайомлений

Тема №2: Управління виконанням циклів.

* + Джерела:
  + <https://acode.com.ua/urok-69-operator-goto/>
    - Що опрацьовано:
      * Нескінченний цикл while та [return](https://acode.com.ua/urok-15-funktsiyi-i-operator-return/), break, exit, [goto](https://acode.com.ua/urok-69-operator-goto/" \t "_blank) для виходу з нього.
      * Оператор goto та лейбл.
    - Статус: Ознайомлений

Тема №3: Основи Функцій у С++.

* + Джерела:
  + <https://acode.com.ua/urok-15-funktsiyi-i-operator-return/>
  + <https://www.youtube.com/watch?v=VKQ242d-Rag>
    - Що опрацьовано:
      * Особливості функцій типу void та їх (виклику) виводу.
      * Переривання роботи функції.
      * Скелет(прототип) функції.
    - Статус: Ознайомлений

Тема №4: Перевантаження Функцій та Простір Імен.

* + Джерела:
  + <https://www.youtube.com/watch?v=kYxNioENAIo>
  + <https://www.youtube.com/watch?v=3KJfisev6SI>
    - Що опрацьовано:
      * Переваги перевантаження функцій.
      * Перевантаження зміною кількості аргументів.
      * Перевантаження зміною типів аргументів.
      * Перевантаження cout.
    - Статус: Ознайомлений

Тема №5: Функції зі змінною кількістю параметрів (еліпсис).

* + Джерела:
  + <https://acode.com.ua/urok-117-elipsys/>
    - Що опрацьовано:
      * доступ до еліпсису через va\_list.
      * Ініціалізуємо va\_list, використовуючи va\_start. Перший параметр - це список, який потрібно ініціалізувати. Другий параметр - це останній параметр, який не є еліпсисом.
      * Використовуємо va\_arg для отримання параметрів з еліпсиса. Перший параметр - це va\_list, який ми використовуємо. Другий параметр - це очікуваний тип параметрів.
      * Виконуємо очищення va\_list, коли вже зробили все необхідне за допомогою va\_end.
      * Параметри еліпсиса не мають оголошень типу даних. При їх використанні компілятор повністю пропускає перевірку типів даних.
    - Статус: Ознайомлений

Тема №6: Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.

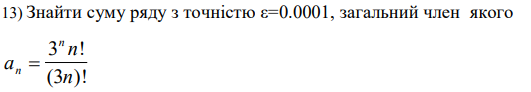
* + Джерела:
  + <https://www.youtube.com/watch?v=V7q9w_s0nns&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=54&t=31s>
    - Що опрацьовано:
      * Рекурсія та її завдання.
      * Недолік рекурсії щодо використання пам’яті через постійне створення змінних.
    - Статус: Ознайомлений

Тема №7: Вбудовані Функції в С++.

* + Джерела:
  + <https://www.youtube.com/watch?v=ZUrdKTBRXWo>
    - Що опрацьовано:
      * Для чого вбудовувати функції.
      * В яких випадках функції не можуть бути вбудованими.
    - Статус: Ознайомлений

**Виконання роботи:**

**Завдання** №1 **VNS Lab 2 - Task 1-13**

****

**Завдання** №2 **VNS Lab 3 - Task 1-13**

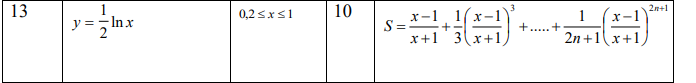
Для х, що змінюється від a до b з кроком (b-a)/k, де (k=10), обчислити

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.

****

1. Алгоритм розв’язання завдання зводиться до трьох циклів, причому два з

них вкладені в третій. Внутрішні цикли підсумують доданки при

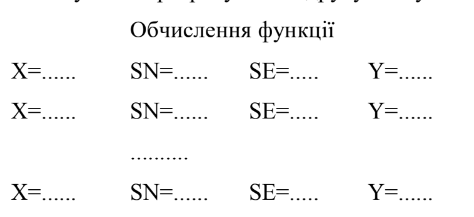
фіксованому параметрі x, один (арифметичний для заданого n), інший

(ітераційний для заданої точності. При організації цих циклів варто

звернути увагу на правильний вибір формули для обчислення елемента ряду

an і правильне присвоєння початкових значень змінним циклу. Зовнішній

цикл організує зміну параметра х.

2. Результати розрахунків надрукувати у такому вигляді: 

Тут X- значення параметра; SN- значення суми для заданого n; SE- значення

суми для заданої точності; Y-точне значення функції.

**Завдання** №3 **VNS Lab 7 - Task 1-13**

Написати функцію зі змінною кількістю параметрів для перекладу чисел з десяткової системи числення в трійкову. Написати викликаючу функцію main, що звертається до цієї функції не менше трьох разів з кількістю параметрів 3, 4, 7.

**Завдання** №4 **VNS Lab 7 - Task 1-13**

Написати перевантажені функції й основну програму, що їх викликає.

а) для перетворення десяткового дробу у звичайний;

б) для перетворення звичайного дробу у десятковий.

**Завдання** №5 **Class Practice Work**

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.

**Завдання** №6 **Self Practice Work**

**Найбільша зростаюча підпослідовність**

Вам задано послідовність із n цілих чисел ai. Ваша задача — знайти довжину найбільшої зростаючої підпослідовності заданої послідовності.

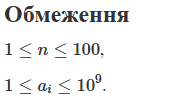
**Вхідні дані**

У першому рядку задано ціле число n.

У наступному рядку задано n цілих чисел — послідовність ai.

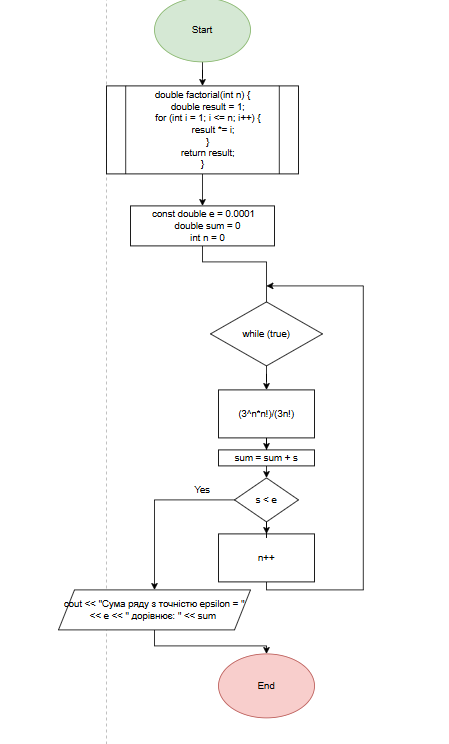
**Вихідні дані**

Виведіть довжину найбільшої зростаючої підпослідовності.

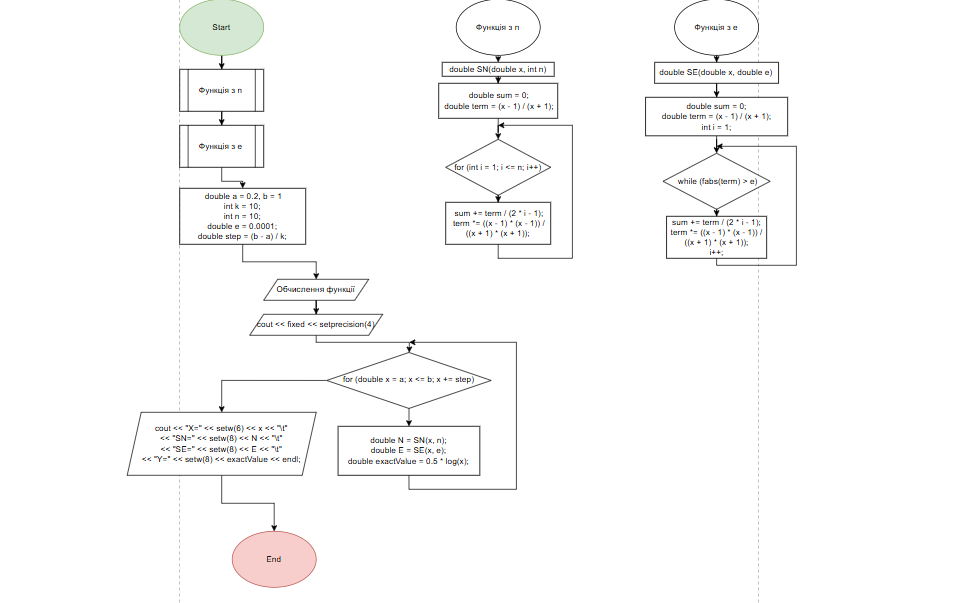


**Дизайн та планована оцінка часу виконання завдань:**

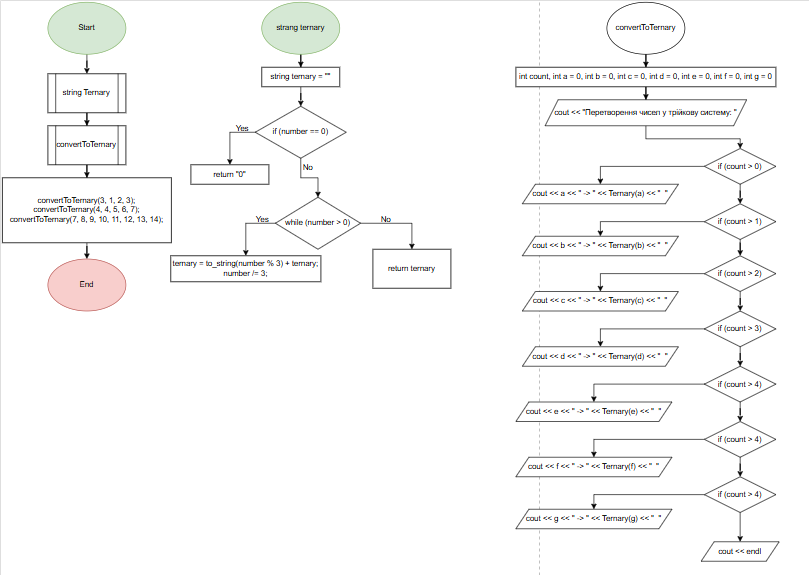
**Завдання** №1 **VNS Lab 2 - Task 1-13**



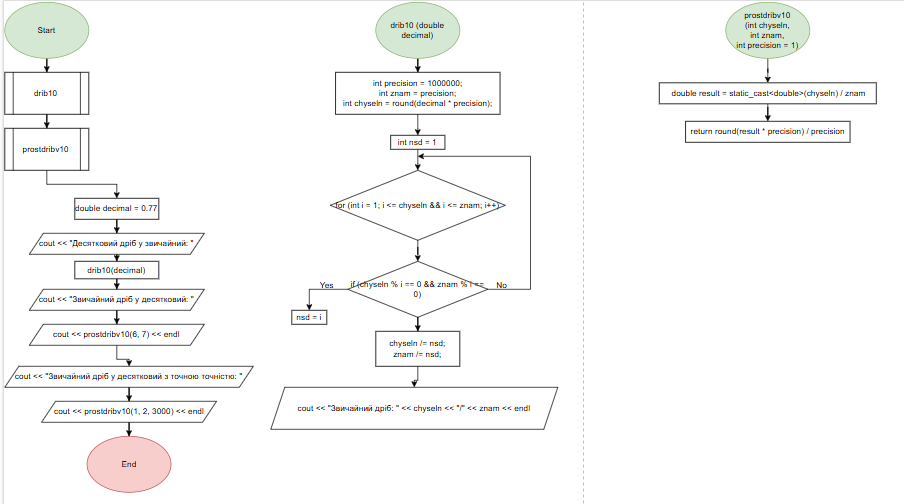
**Завдання** №2 **VNS Lab 3 - Task 1-13**



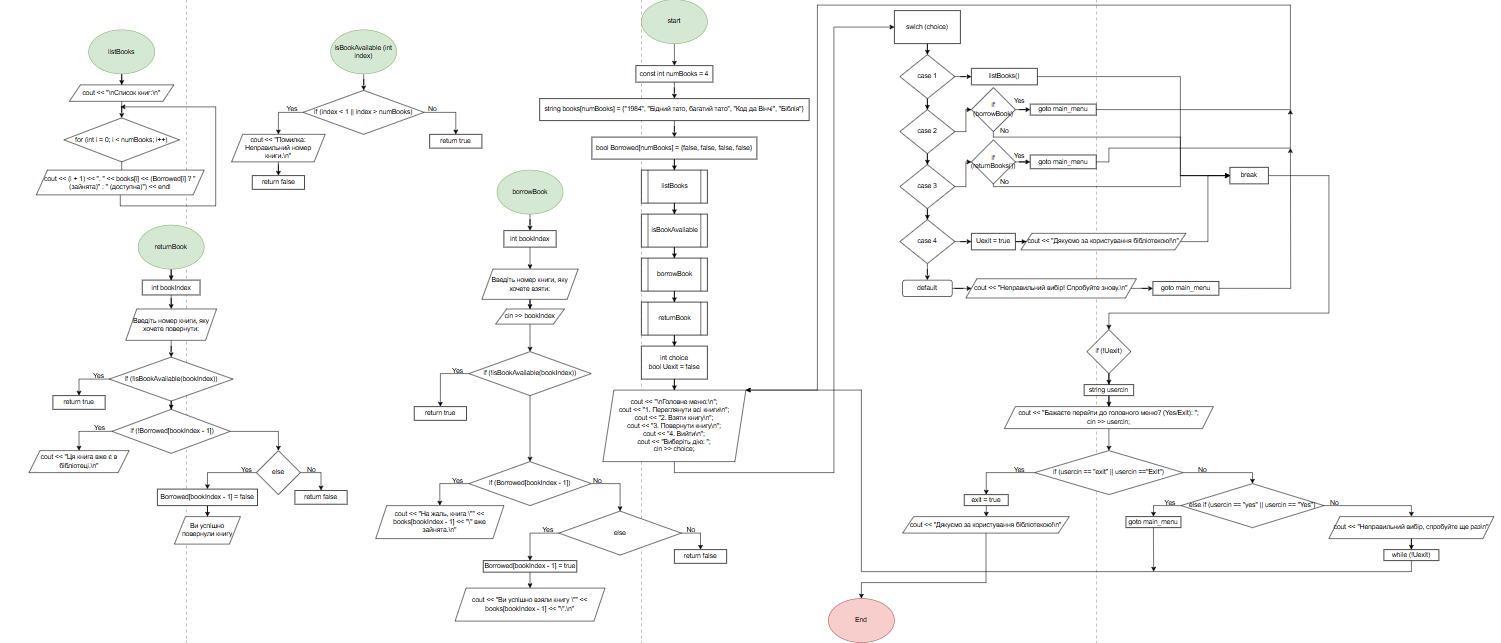
**Завдання** №3 **VNS Lab 7 - Task 1-13**



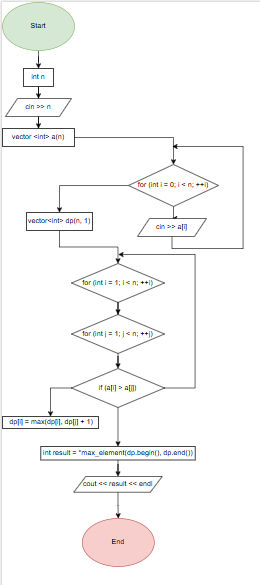
**Завдання** №4 **VNS Lab 7 - Task 2-13**

****

**Завдання** №5 **Class Practice Work**

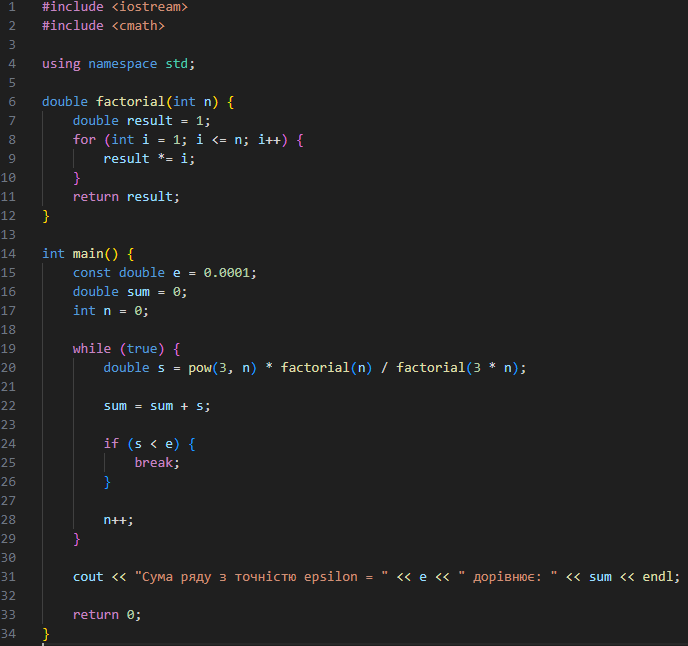


**Завдання** №6 **Self Practice Work**

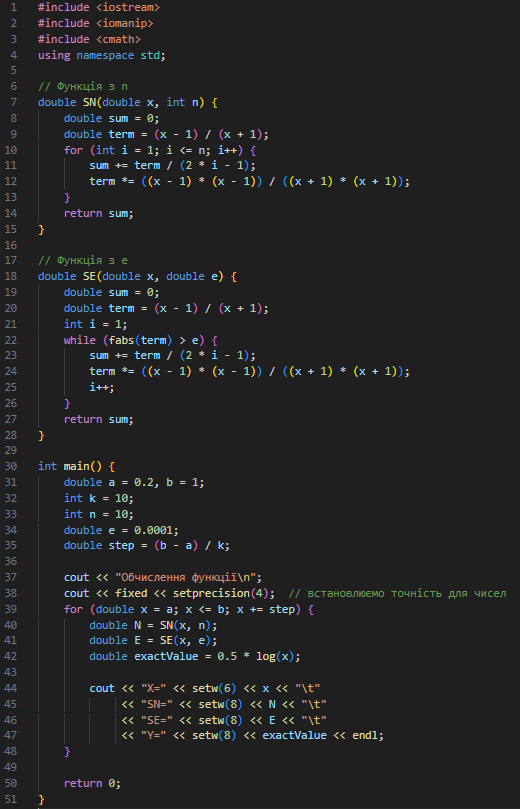


**Код програм з посиланням на зовнішні ресурси**

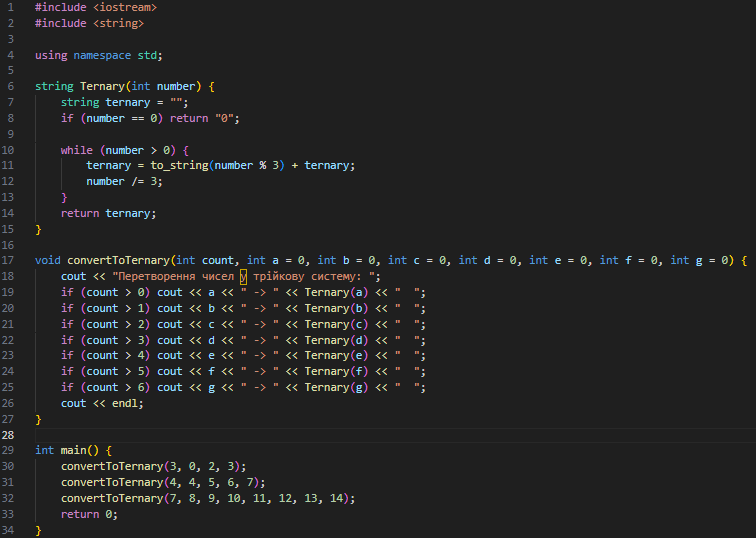
**Завдання №1**

****

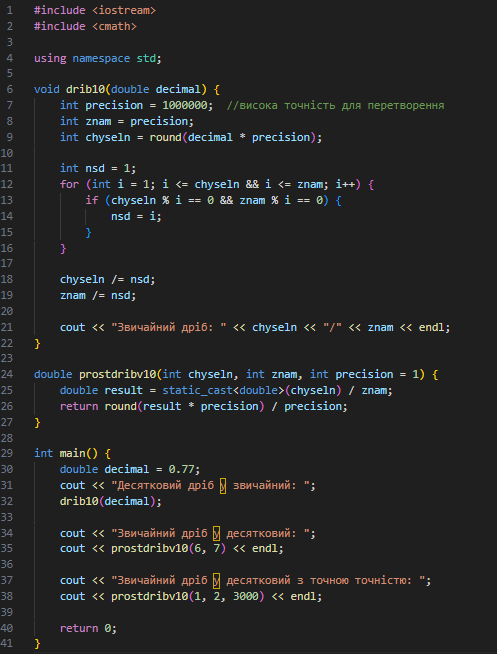
**Завдання №2**

****

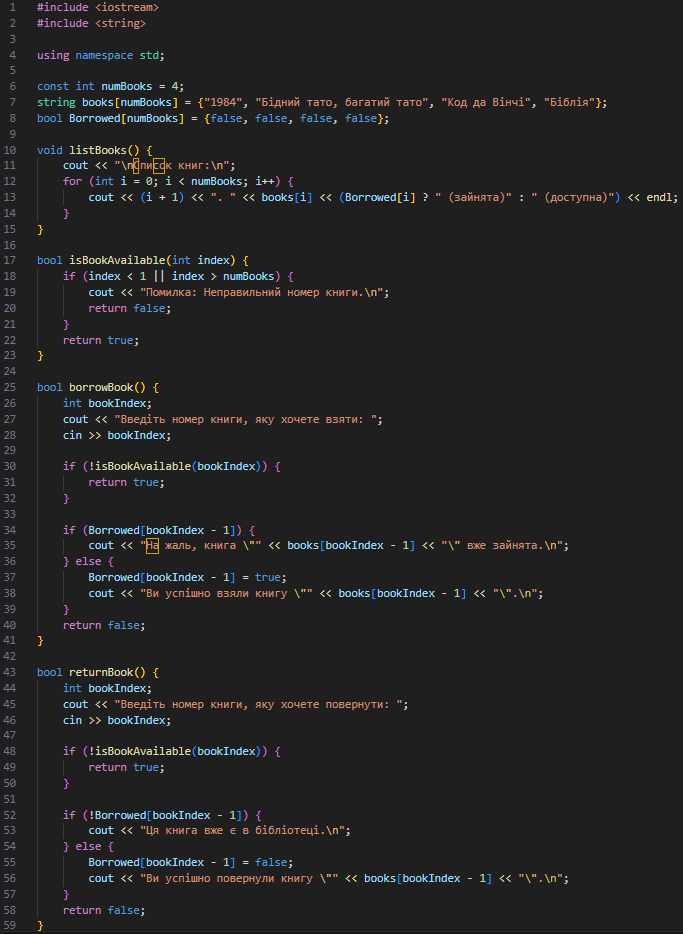
**Завдання №3**

****

**Завдання №4**

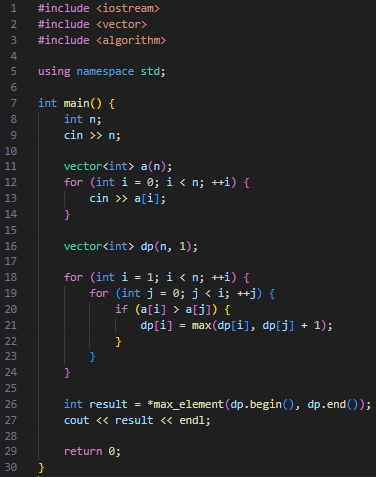
****

**Завдання №5**

****

****

**Завдання №6**

****

**Результати виконання завдань, тестування та фактично затрачений час:**

**Завдання** №1 **VNS Lab 2 - Task 1-13**

****

Фактично затрачений час: 40хв

**Завдання** №2 **VNS Lab 3 - Task 1-13**

****

Фактично затрачений час: 1год

**Завдання** №3 **VNS Lab 7 - Task 1-13**

****

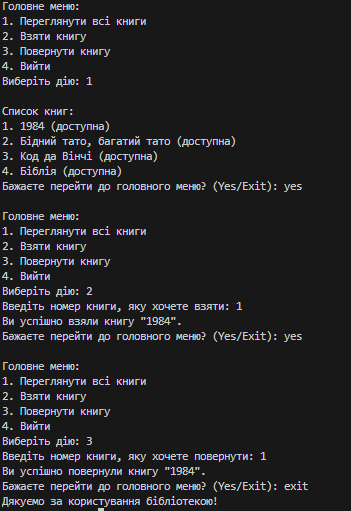
Фактично затрачений час: 1,5год

**Завдання** №4 **VNS Lab 7 - Task 2-13**

****

Фактично затрачений час: 45хв

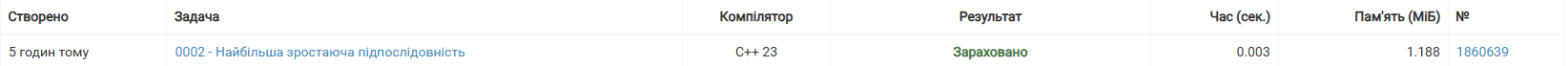
**Завдання** №5 **Class Practice Work**

****

Фактично затрачений час: 3год

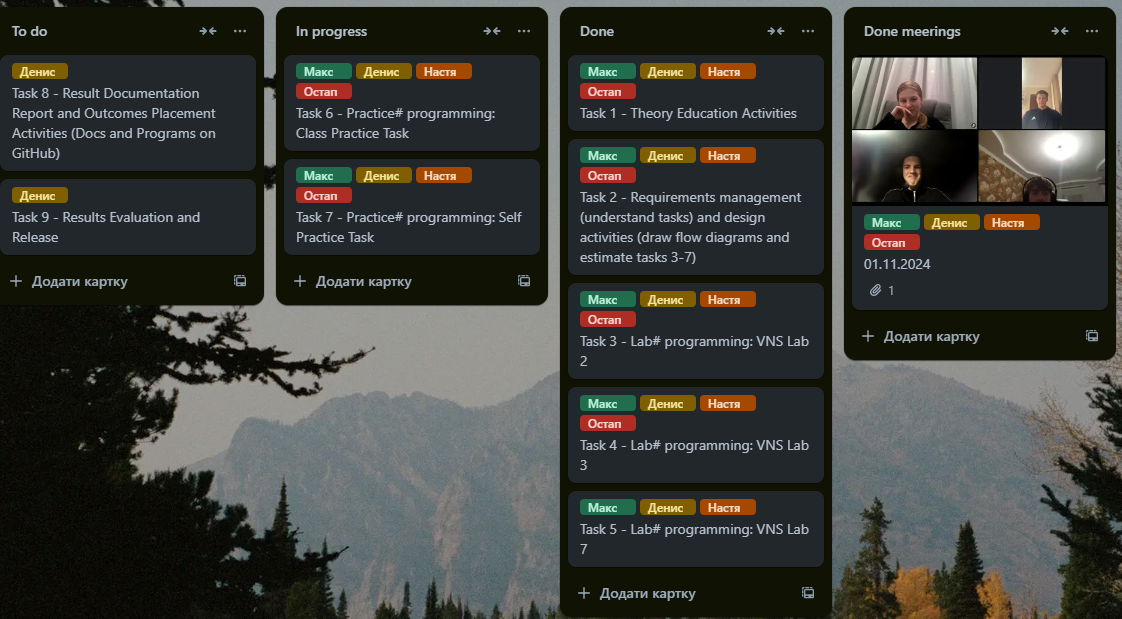
**Завдання** №6 **Self Practice Work**

****

****

Фактично затрачений час: 30хв

**Кооперація з командою:**



Ми зустрічалися один раз з командою, на цій зустрічі кожен розпитав, що не розуміє ми один одному помогли а також розставили пріоритети в дошці трелло

**Висновок:**Виконуючи третій епік, я ознайомився з різними видами циклів у C++ і зрозумів, як вони можуть спростити реалізацію повторюваних завдань у програмуванні. Навчився керувати виконанням циклів за допомогою операторів break і continue, що дозволяє змінювати логіку їх роботи. Дослідження функцій, їх параметрів і перевантаження допомогло мені краще організовувати код і робити його більш зрозумілим. Також я опанував застосування рекурсії та функцій зі змінною кількістю параметрів, що розширило моє уявлення про можливості функцій у складних завданнях. Завдяки вивченню вбудованих функцій, я побачив, як C++ може оптимізувати виконання стандартних операцій і полегшити написання коду. Ці знання я закріпив на практиці, написавши кілька програм.