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: «Програмування: алгоритм, програма, код. Системи числення. Двійкова система числення. Розробка та середовище розробки програми.»

***з дисципліни:*** «Основи програмування»

до:

Практичних Робіт до блоку № 3

**Виконав:**

Студент групи ШІ-13

Федів Андрій Сергійович

Львів 2024

**Тема:** Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції

**Мета:** ознайомлення з основами роботи з циклами та вкладеними циклами, а також способами завершення їх виконання. Розглянути функції, простір імен, перевантаження функцій, функції з еліпсисом, рекурсія та вбудовані функції для поглиблення розуміння структури і поведінки програм.

**Теоретичні відомості:**

1. Введення в Цикли та їх Види в С++:

○      Значення та роль циклів у програмуванні.

○      Огляд видів циклів: for, while, do-while.

○      Синтаксис та основи використання кожного типу циклу.

○      Приклади базових циклів для різних задач.

1. Управління Виконанням Циклів:

○      Застосування операторів break та continue.

○      Умови завершення циклів.

○      Передчасне завершення виконання циклу.

○      Приклади та вправи з управлінням циклами.

1. Вкладені Цикли:

○      Поняття та важливість вкладених циклів.

○      Реалізація вкладених циклів: приклади для різних сценаріїв.

○      Практичні завдання на вкладені цикли.

1. Основи Функцій у С++:

○      Визначення та оголошення функцій.

○      Параметри функцій: передача за значенням і за посиланням.

○  Параметри за замовчуванням.

○      Повернення значень з функцій.

○      Приклади створення та використання функцій.

1. Перевантаження Функцій та Простір Імен:

○      Концепція перевантаження функцій.

○      Правила та приклади перевантаження функцій.

○      Поняття та використання просторів імен.

○  Вкладені простори імен (C++ 17)

○    Роль просторів імен у організації коду.

1. Розширені Можливості Функцій:

○   Функції зі змінною кількістю параметрів (еліпсис): синтаксис та приклади.

○  Область видимості функції – static, extern.

○   Рекурсія: основи, приклади рекурсивних функцій та їх аналіз.

○   Передача масивів та об'єктів як параметрів.

○   Повернення масивів та об'єктів з функцій.

1. Вбудовані Функції в С++:

○  Огляд вбудованих функцій у С++.

○   Приклади використання стандартних функцій у програмуванні.

○   Роль вбудованих функцій у спрощенні коду.

○   Практичні завдання для розуміння вбудованих функцій.

**Індивідуальний план опрацювання теорії:**

1. Введення в Цикли та їх Види в С++
2. Управління Виконанням Циклів
3. Вкладені Цикл
4. Основи Функцій у С++
5. Перевантаження Функцій та Простір Імен
6. Розширені Можливості Функцій

Вбудовані Функції в С++

1. **Джерела**:

* Chat gpt
* Список відтворення на YouTube ( <https://youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&si=sXvmPdnGkwvJLXUi> )
* Власний досвід

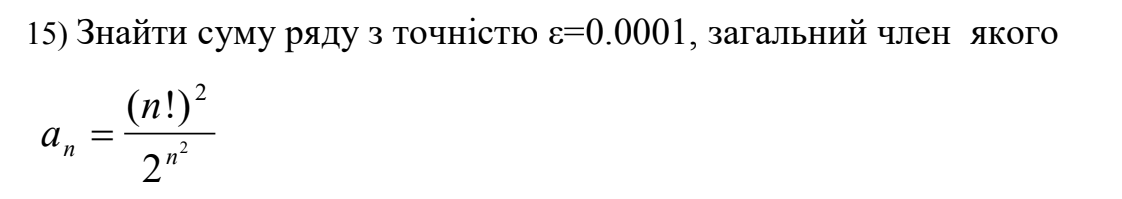
**Виконання роботи:**

**VNS Lab 2 Task 1 (15)**

Використовуючи оператор циклу, знайти суму елементів, зазначених у

конкретному варіанті. Результат надрукувати, надавши відповідний

заголовком.

****

**VNS Lab 3 Task 1 (15)**

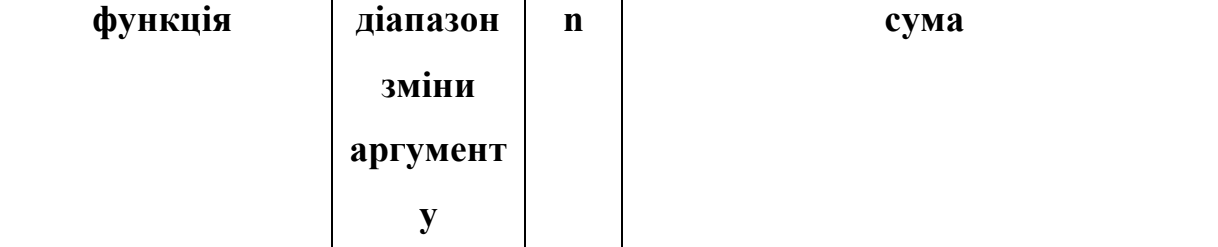
Для х, що змінюється від a до b з кроком (b-a)/k, де (k=10), обчислити

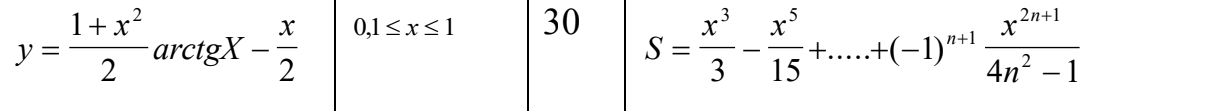
функцію f(x), використовуючи її розклад в степеневий ряд у двох випадках:

а) для заданого n;

б) для заданої точності ε (ε=0.0001).

Для порівняння знайти точне значення функції.





**VNS Lab 7 Task 1 & 2 (15)**

**1:** Розв’язати зазначене у варіанті завдання, використовуючи функції зі змінною кількістю параметрів.

Написати функцію зі змінною кількістю параметрів для перетворення чисел

з вісімкової системи числення в десяткову. Написати викликаючу main, що

звертається до цієї функції не менше трьох разів з кількістю параметрів 3, 6, 7.

**2:** Написати перевантажені функції й основну програму, що їх викликає.

а) цілі числа підносить до степеня n;

б) з десяткових чисел бере корінь степеня n.

**Class Practice Task:**

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

**Self Practice Task:**

Вам дано 2 масиви розміром N та M. Значення у цих масивах унікальні.

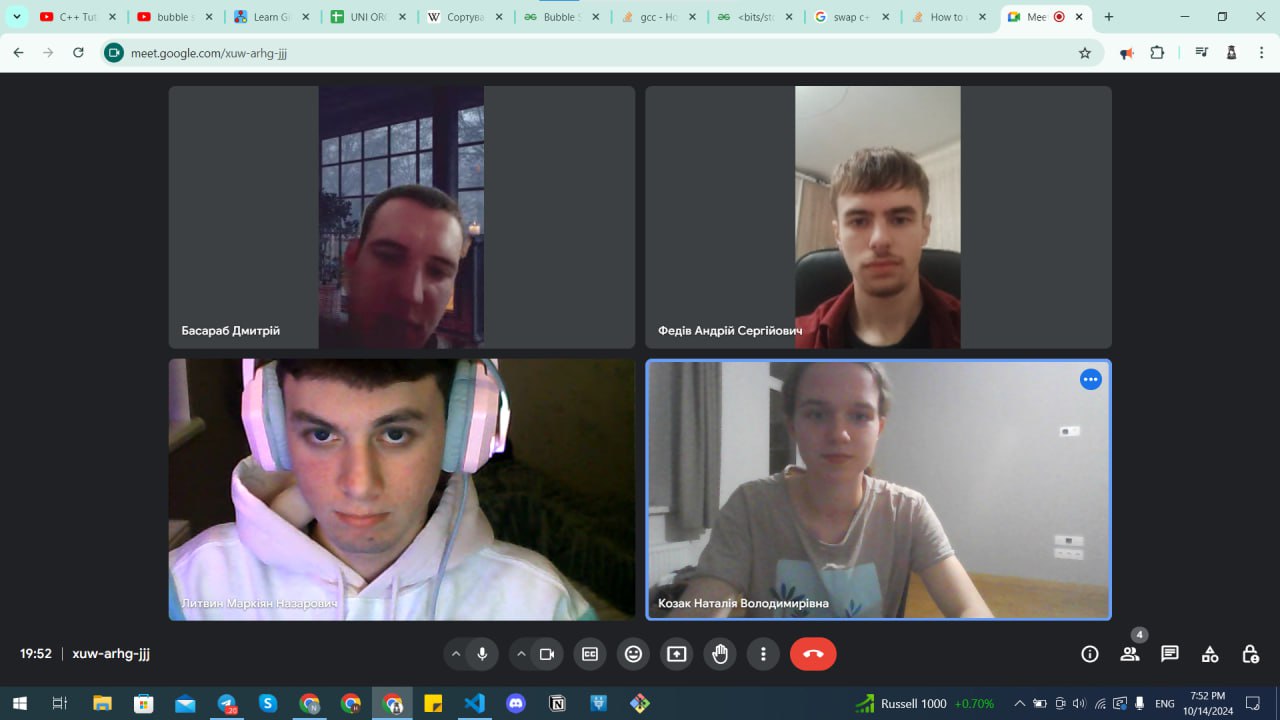
Ваше завдання вивести у першому рядку кількість елементів, які наявні в обох масивах одночасно, у другому кількість унікальних елементів в обох масивах разом.

1. Requirements management and design activities

Team Trello dashboard for task control

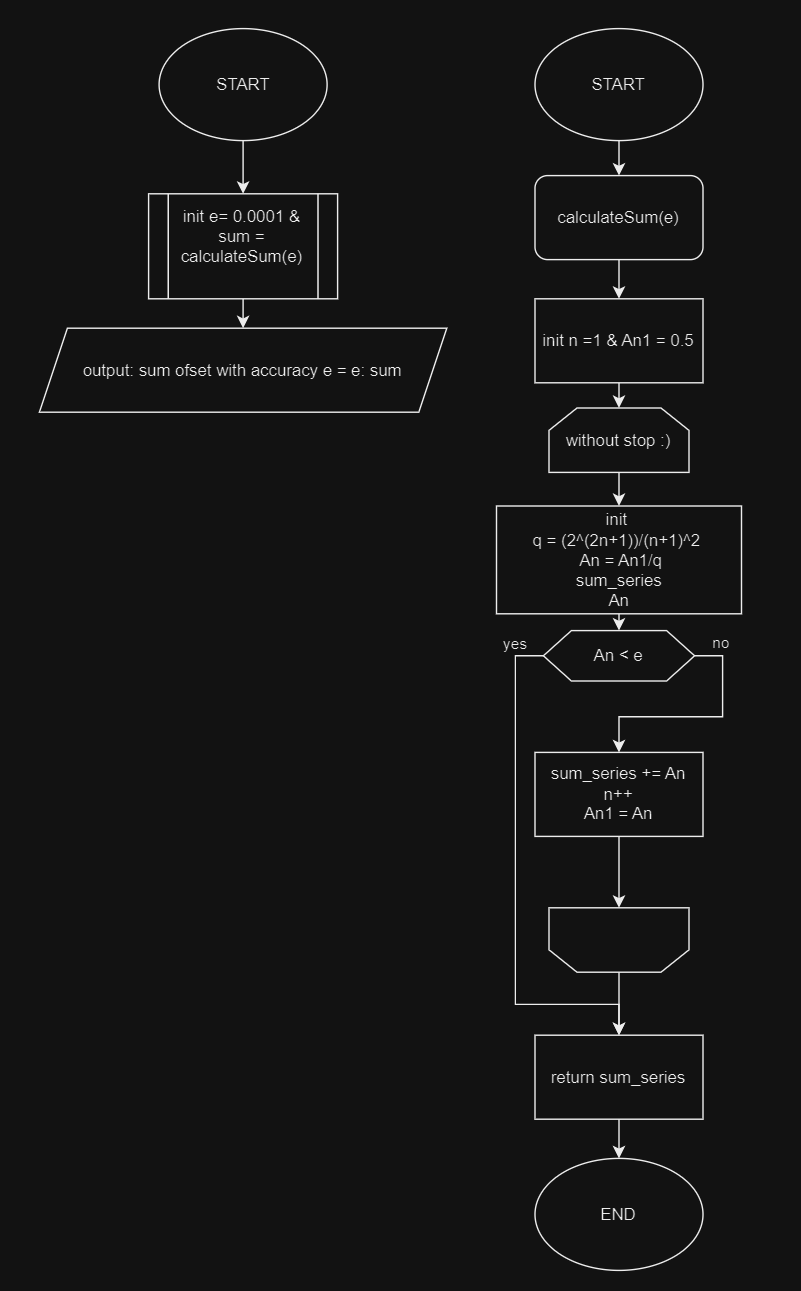


Team meeting in zoom



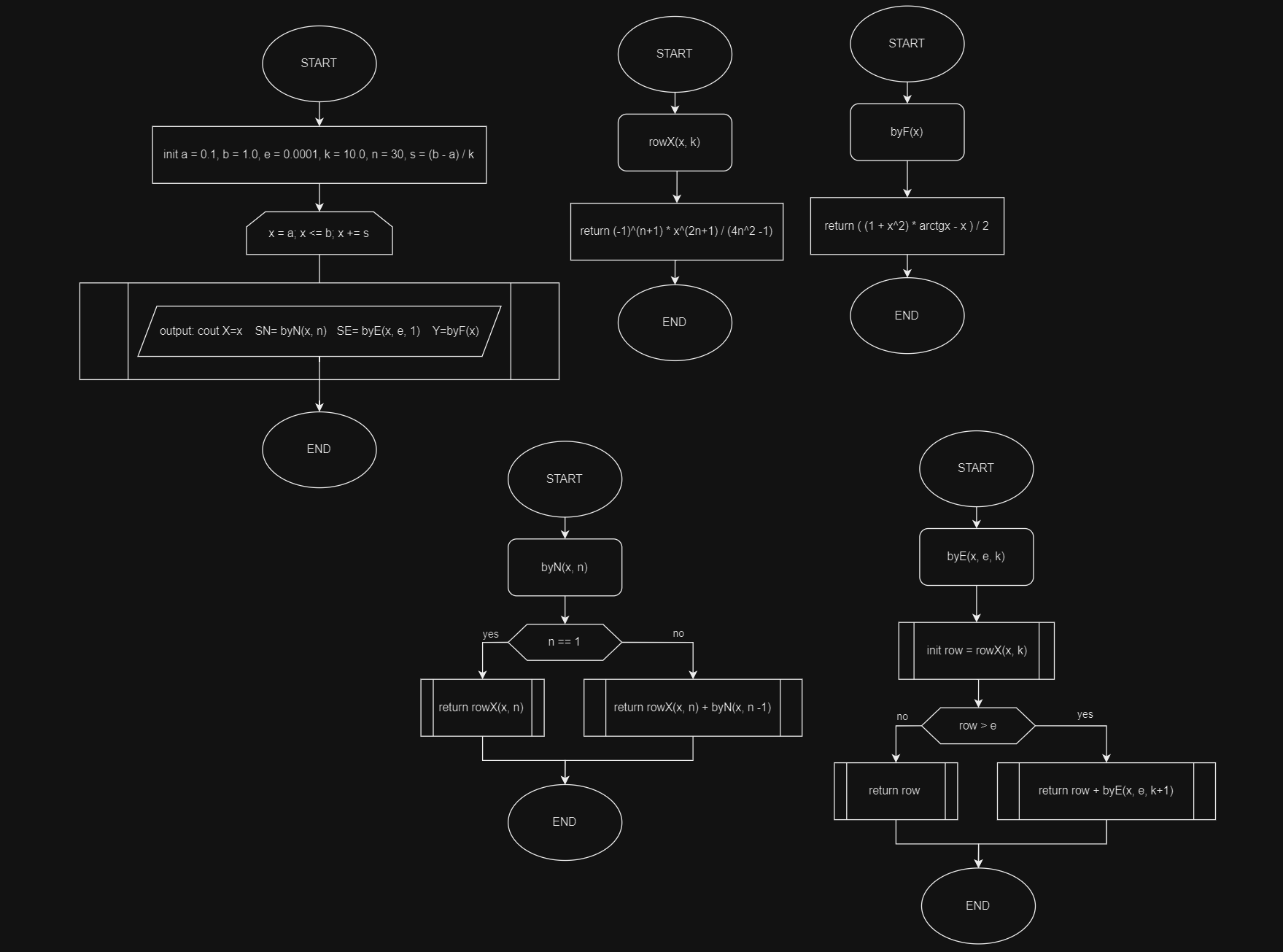
**UML-diagram block-scheme for each task**

VNS Lab 2 Task 1



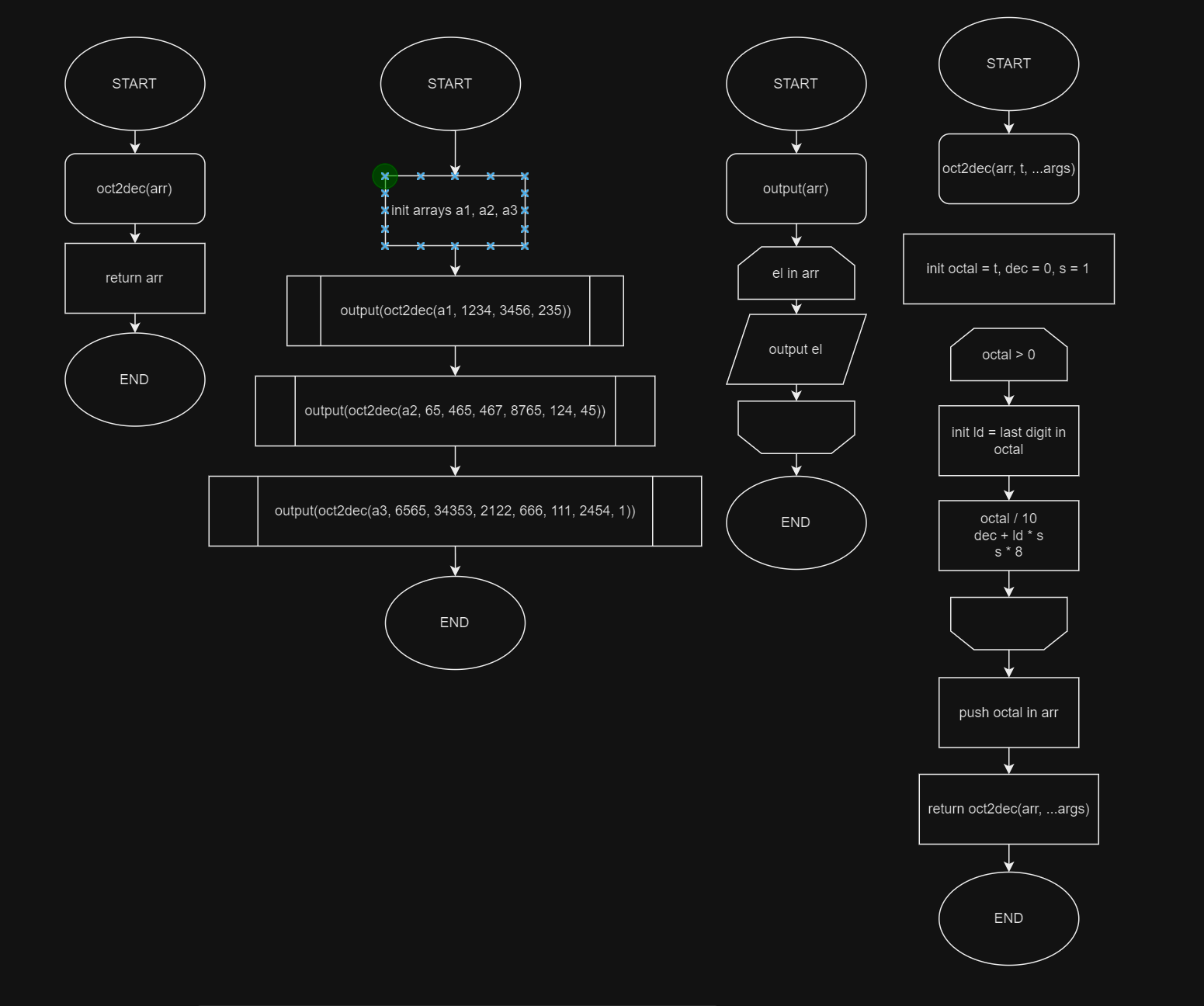
*~30хв*

*VNS Lab 3 Task 1*



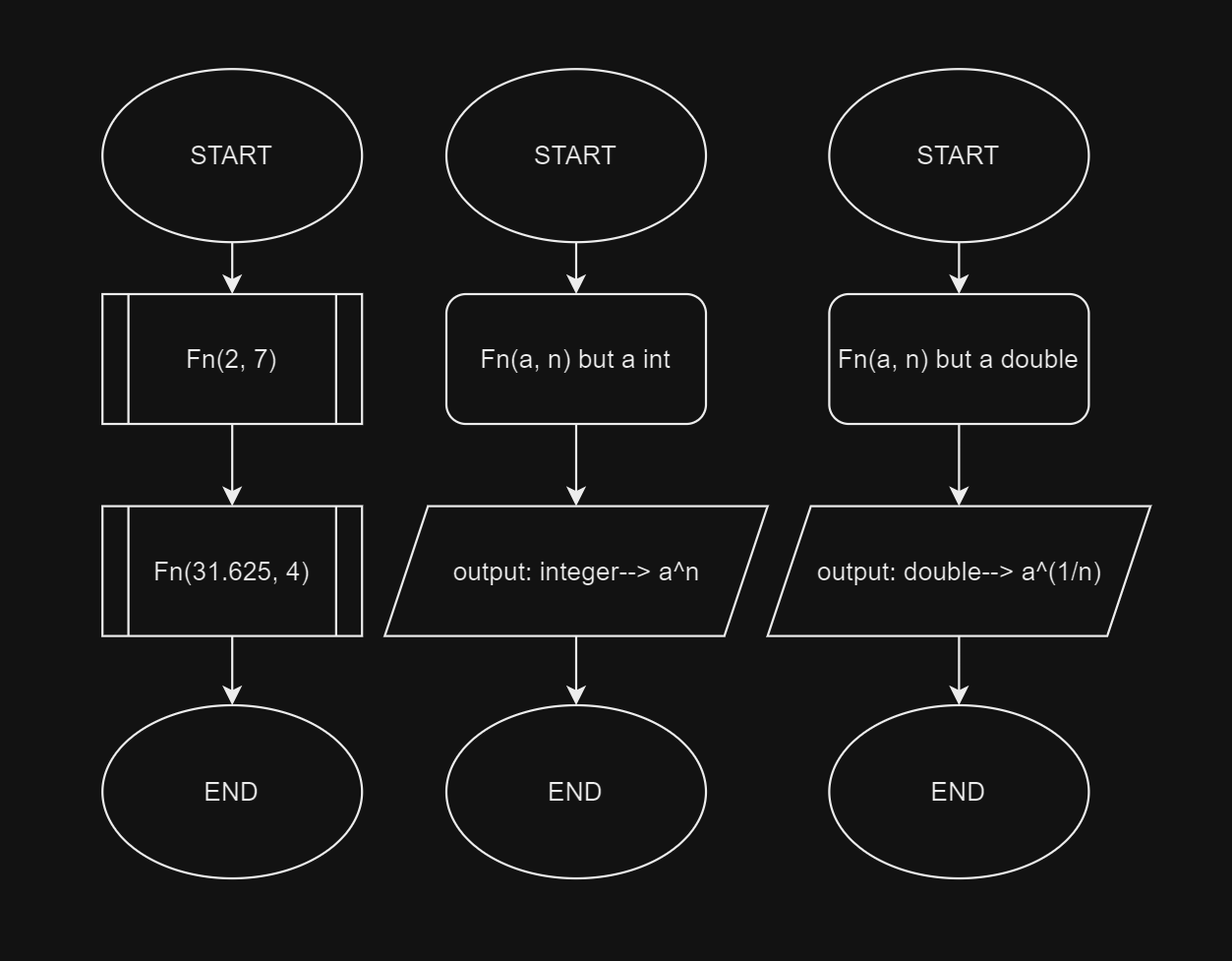
~1год

VNS Lab 7 Task 1



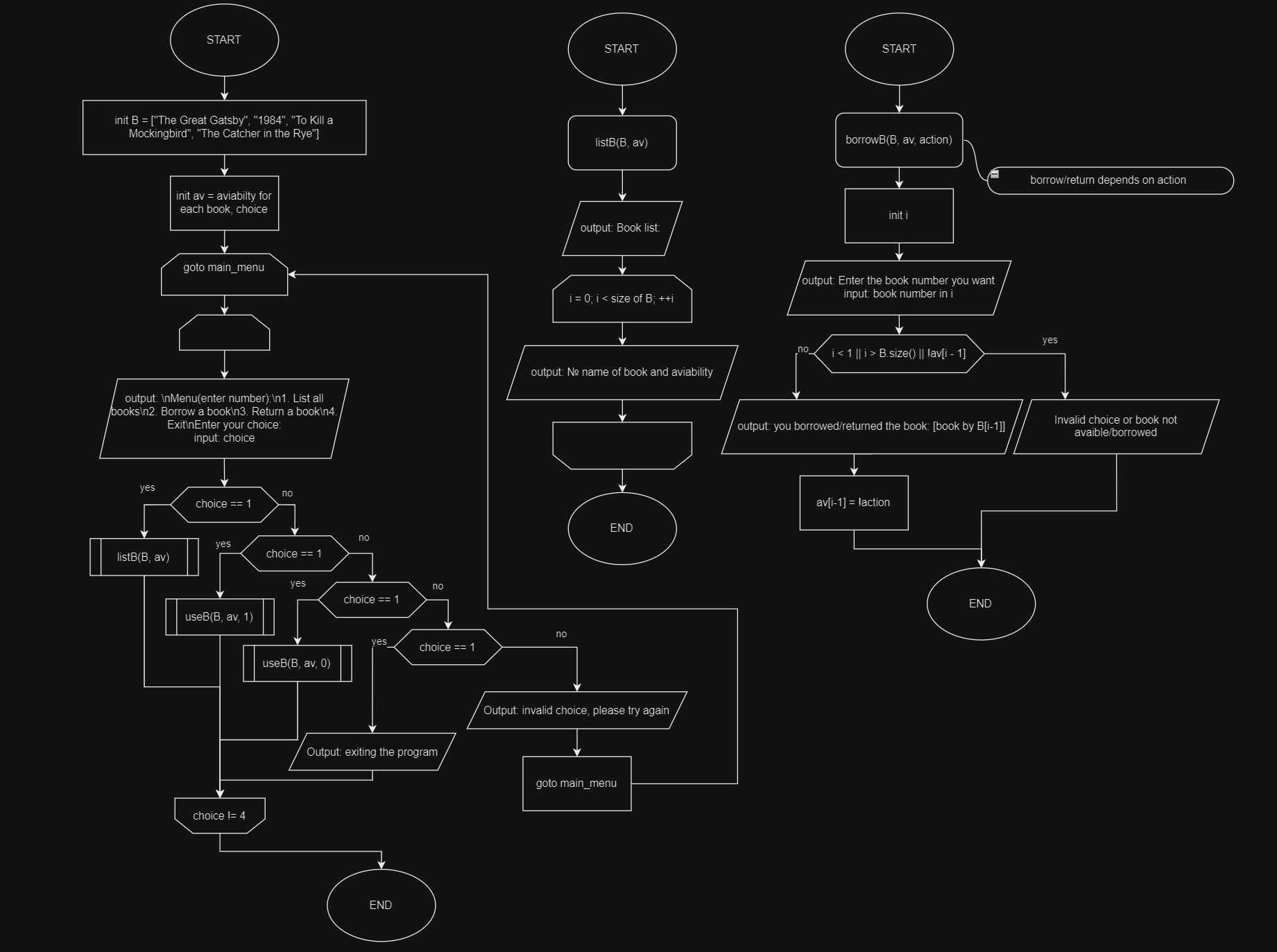
~25хв

VNS Lab 7 Task 2



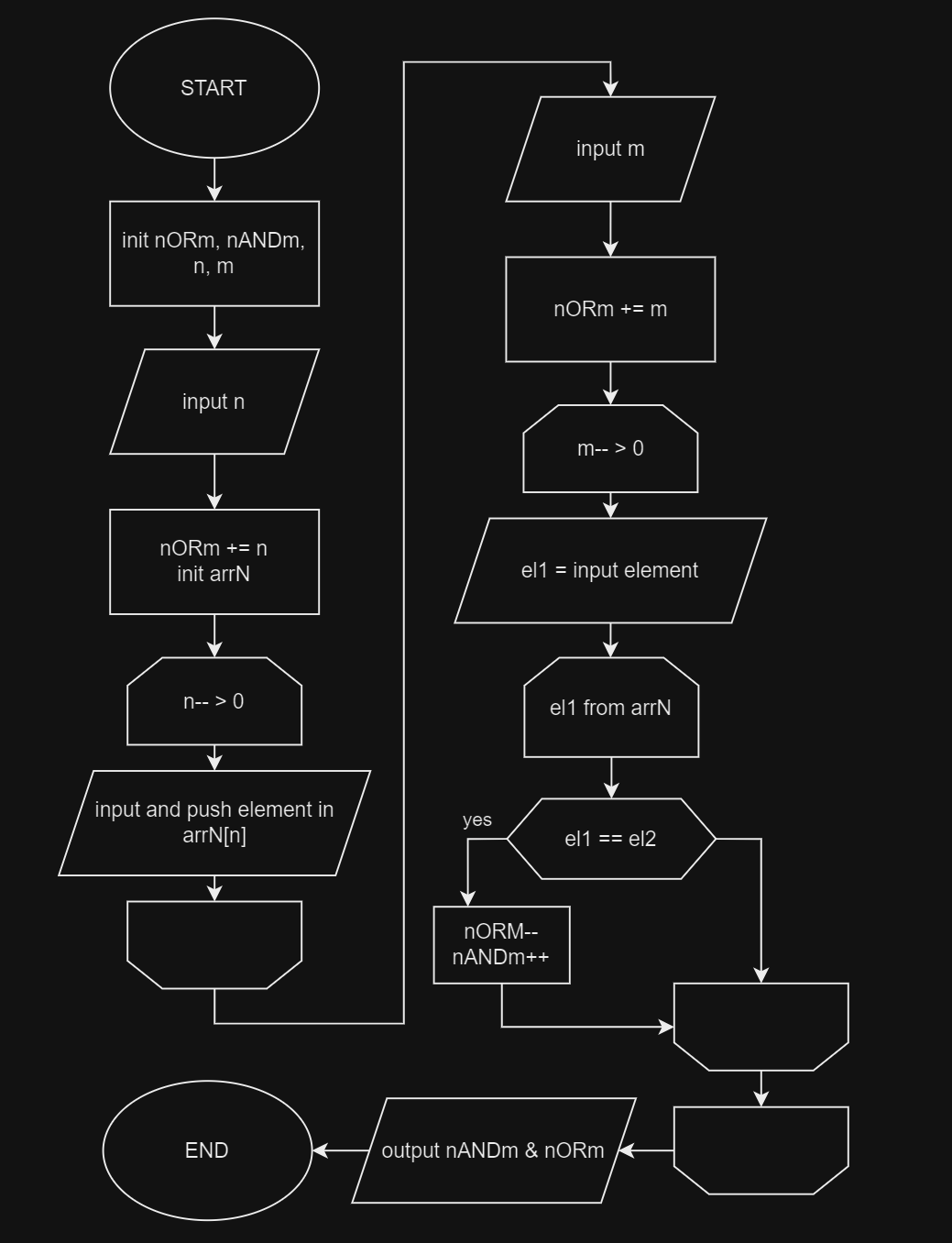
~15хв

Class Practice Task



~30хв

Self Practice Task



~20хв

Код програми з посиланням на зовнішні ресурси

VNS Lab 2 Task 1: **./src/vns\_lab\_2\_task\_1\_andrii\_fediv.cpp**

VNS Lab 3 Task 1: **./src/vns\_lab\_3\_task\_1\_andrii\_fediv.cpp**

VNS Lab 7 Task 1: **./src/vns\_lab\_7\_task\_1\_variant\_15\_andrii\_fediv.cpp**

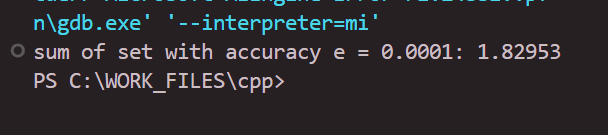
VNS Lab 7 Task 2: **./src/vns\_lab\_7\_task\_2\_variant\_15\_andrii\_fediv.cpp**

Class Practice Task: **./src/practice\_work\_task\_1\_andrii\_fediv.cpp**

Self Practice Task: **./src/self\_practice\_work\_algotester\_task\_1\_andrii\_fediv.cpp**

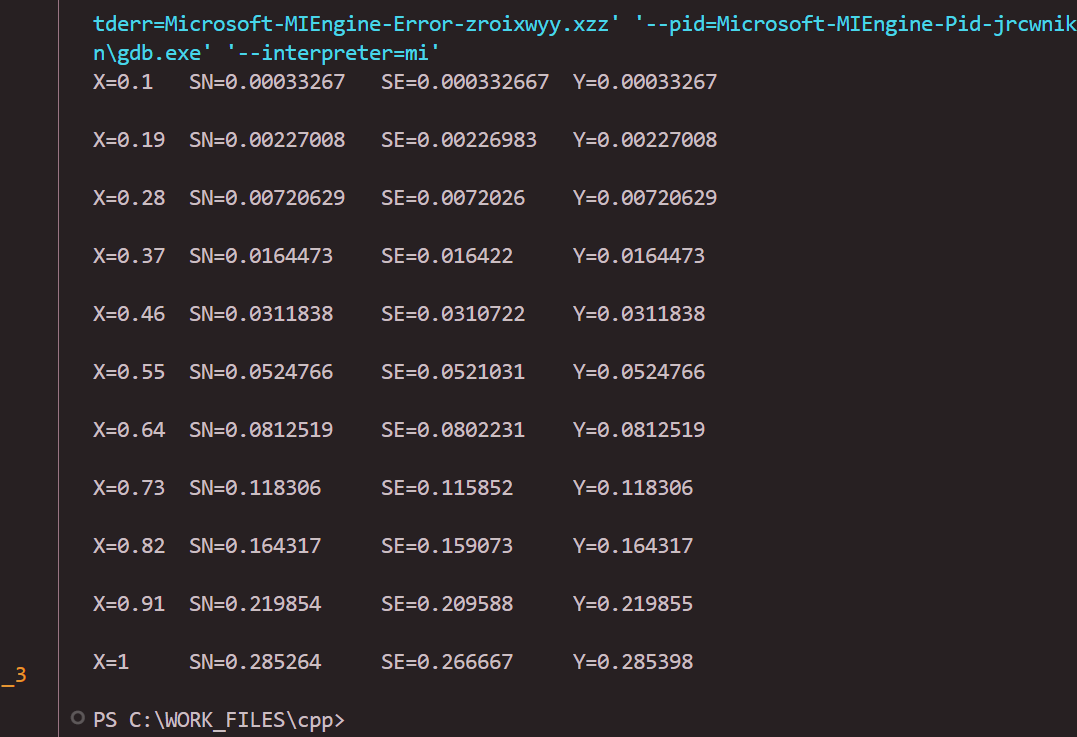
**Результати виконаних завдань, тестування та фактично затрачений час**

VNS Lab 2 Task 1

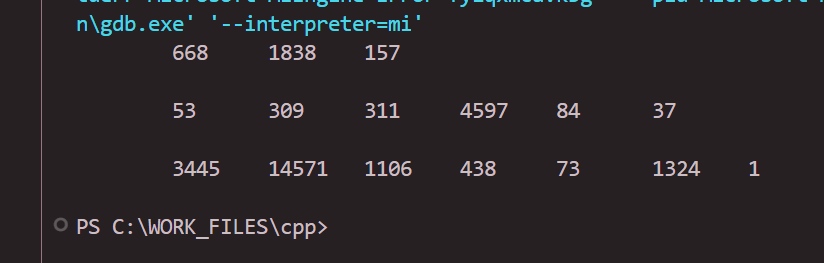


Затратність ~45хв

VNS Lab 3 Task 1

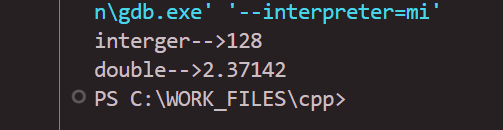


Затратність ~1год (+ психіка)

VNS Lab 7 Task 1 

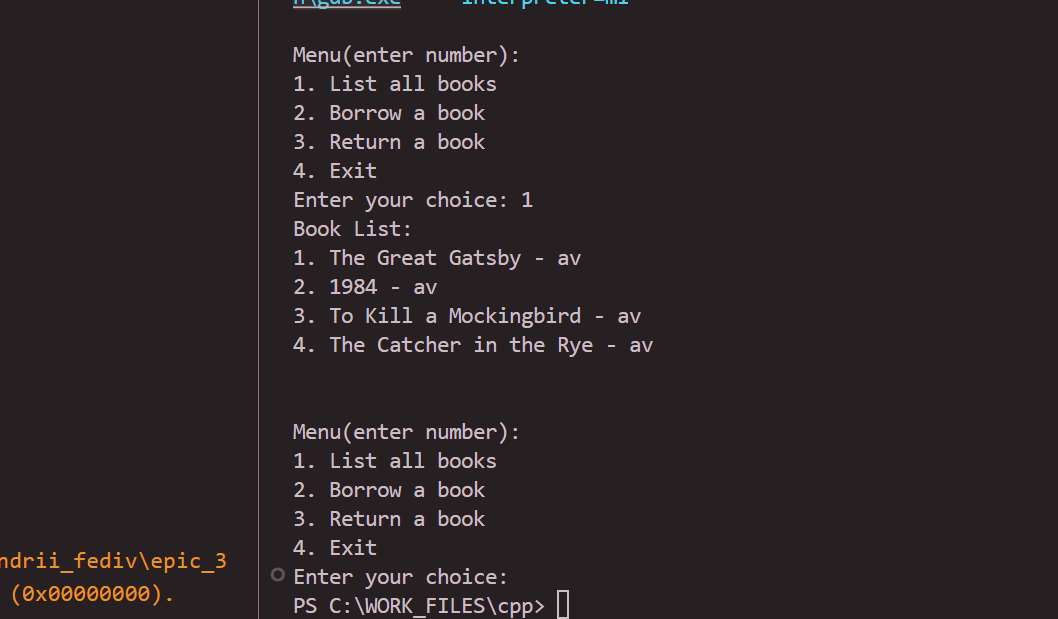
Затратність ~20хв

VNS Lab 7 Task 2



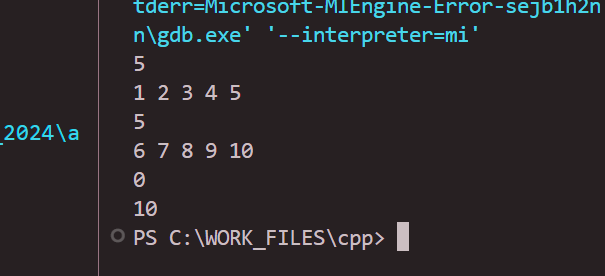
Затратність ~5хв

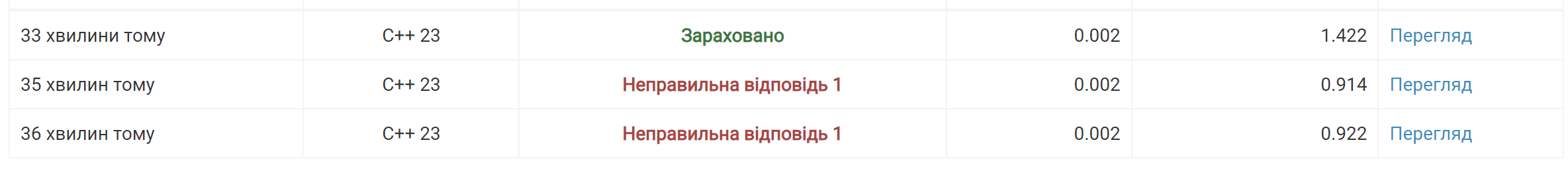
Class Practice Task



Затратність ~40хв

Self Practice Task





Затратність ~15хв

**Висновки:**

Виконуючи цей епік я поглибив свої знання з циклів, що допоможе мені оптимізувати виконання програм. Розуміння функцій, перевантаження та рекурсії дозволить мені створювати гнучкі й потужні рішення для різних завдань. Знання вбудованих функцій і роботи з простором імен допомагало уникати конфліктів та забезпечити більш чітку організацію коду.