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 № 4**

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 № 4

**Виконав:**

Студент групи ШІ-13

Федів Андрій Сергійович

Львів 2024

**Тема:** Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.

**Мета:** зрозуміти основи роботи з одновимірними та двовимірними масивами, а також освоїти концепції вказівників, динамічних масивів і структур даних, що дозволить ефективно реалізовувати алгоритми обробки даних та структури, щоб спростити вирішення складних завдань.

**Теоретичні відомості:**

1. Класи пам'яті у C++

○  Статична пам’ять.

○  Динамічна пам’ять.

○  Поняття стеку.

○  Виділення та вивільнення пам’яті.

1. Вступ до Масивів і Вказівників:

○      Основи масивів: визначення, важливість, приклади використання.

○      Різниця між статичними та динамічними масивами.

○      Основи вказівників: що це таке, як вони працюють.

○      Взаємозв'язок між масивами та вказівниками.

○      Вступ до посилань: основні концепції та відмінності від вказівників.

1. Одновимірні Масиви:

○      Створення та ініціалізація одновимірних масивів.

○      Основні операції: індексація, присвоєння, читання.

○      Цикли та обхід масивів.

○      Використання функцій для роботи з масивами.

○      Приклади алгоритмів сортування та пошуку.

1. Вказівники та Посилання:

○      Використання вказівників для доступу до елементів масиву.

○      Арифметика вказівників.

○      Різниця між вказівниками та посиланнями в контексті функцій.

○      Динамічне виділення пам'яті з використанням вказівників.

○      Використання вказівників для створення складних структур даних.

1. Двовимірні Масиви:

○      Оголошення та ініціалізація двовимірних масивів.

○      Вкладені цикли для обходу двовимірних масивів.

○      Практичні приклади використання двовимірних масивів.

○      Передача двовимірних масивів у функції.

○      Застосування двовимірних масивів для розв'язання задач.

1. Динамічні Масиви:

○      Основи динамічного виділення пам'яті.

○      Створення та управління динамічними масивами.

○      Використання операторів new та delete для управління пам'яттю.

○      Реалізація змінної розмірності масивів.

○      Передача динамічних масивів у функції.

1. Структури Даних:

○      Оголошення та використання структур.

○      Використання масивів та вказівників у структурах.

○      Функції для обробки даних у структурах.

**Індивідуальний план опрацювання теорії:**

Класи пам'яті у C++

Вступ до Масивів і Вказівників

Одновимірні Масиви

Вказівники та Посилання

Двовимірні Масиви

Динамічні Масиви

Структури Даних

1. **Джерела**:

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

**Виконання роботи:**

**VNS Lab 4 Task 1 (15)**

1) Реалізувати з використанням масиву двонаправлене кільце (перегляд

можливий в обидва боки, від останнього елемента можна перейти до

першого).

2) Роздрукувати отриманий масив, починаючи з К-ого елемента і до К-1 (по

кільцю вліво).

3) Знищити з кільця перший й останній елементи.

4) Роздрукувати отриманий масив, починаючи з К-ого елемента (і до К+1 по

кільцю вправо).

**VNS Lab 5 Task 1 (15)**

Задано двовимірний масив. Знайти суму елементів першого стовпця без

одного останнього елемента, суму елементів другого стовпця без двох

останніх, суму елементів третього стовпця без трьох останніх і т.д. Останній

стовпець не обробляється. Серед знайдених сум знайти максимальну

**Algotester Lab 2 (v1)**

У вас є дорога, яка виглядає як N чисел.

Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елементу.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

**Algotester Lab 3 (v2)**

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.

**Class Practice Task:**

Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

**Self Practice Task:**

Вам дано N слів та число K.

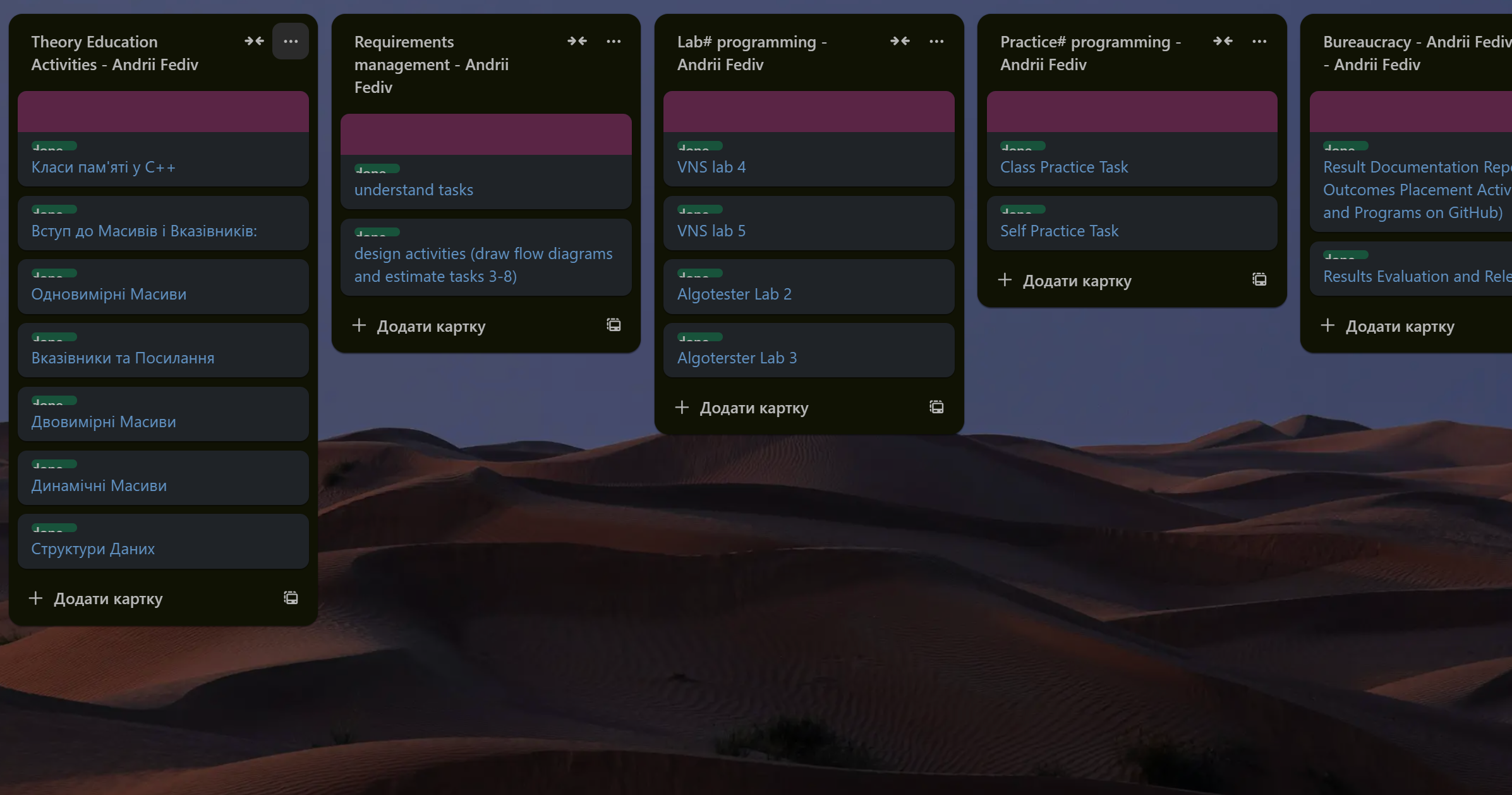
Ваше завдання перечислити букви в словах, які зустрічаються в тексті більше-рівне ніж K разів (саме слово, не буква!).

Великі та маленькі букви вважаються однаковими, виводити необхідно малі, посортовані від останьої до першої у алфавіті. Букву потрібно виводити лише один раз.

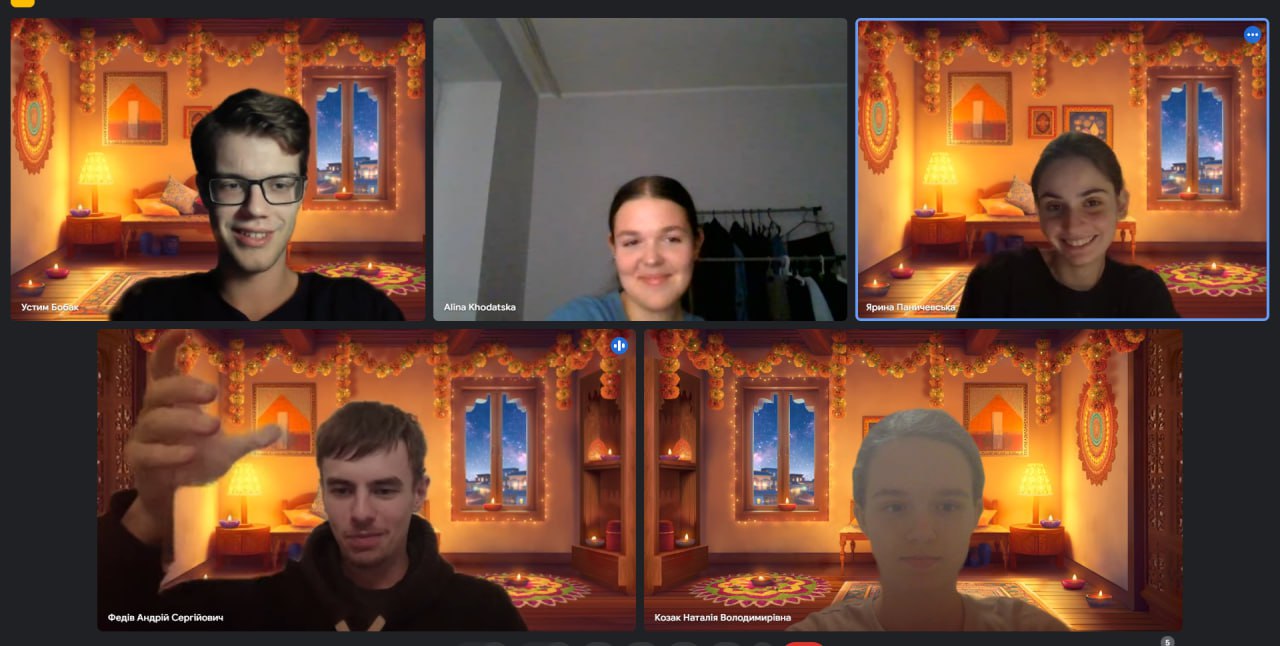
У випадку якщо таких букв немає - вивести "Empty!".

1. Requirements management and design activities

Team Trello dashboard for task control

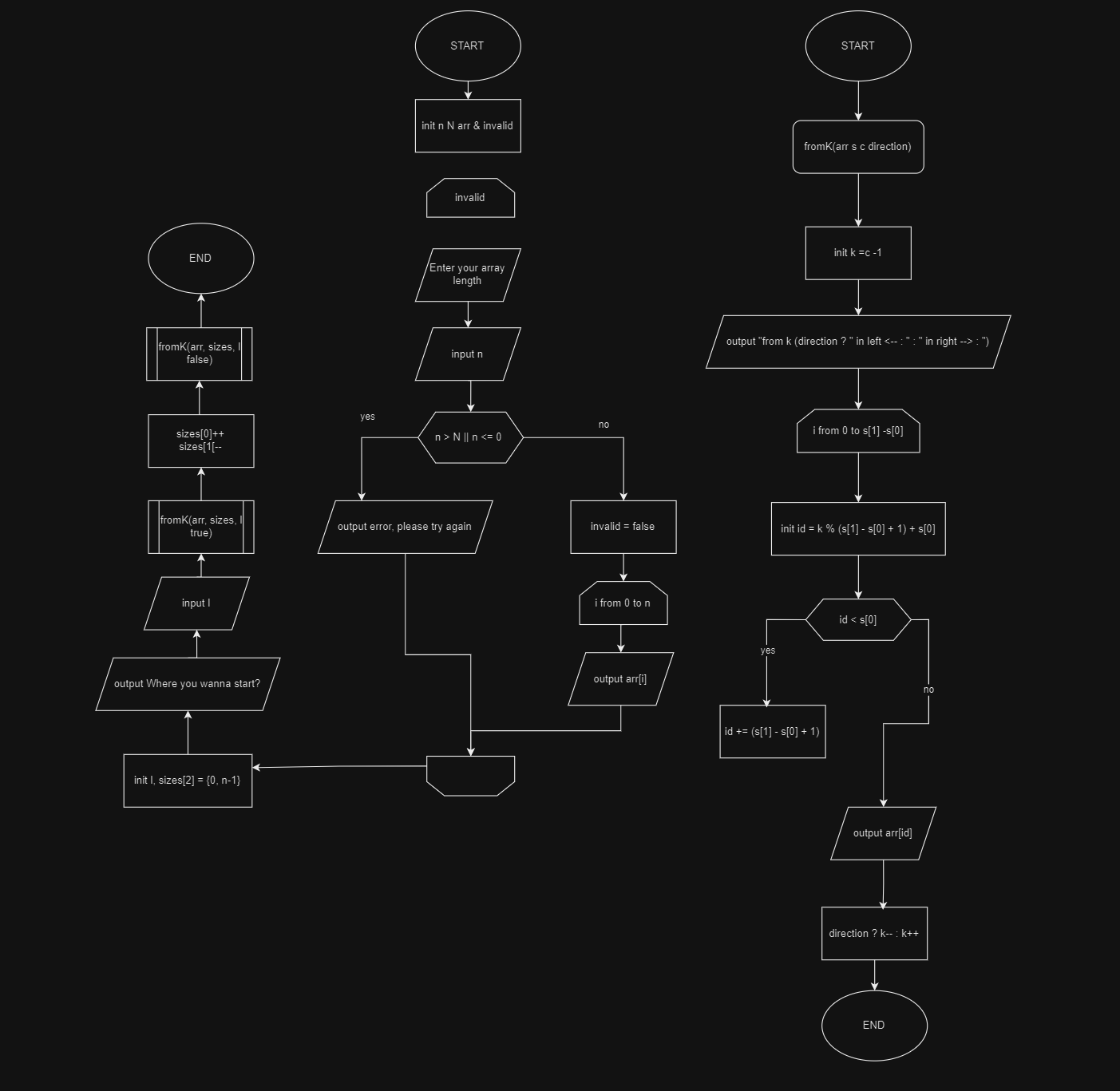


Team meeting in zoom



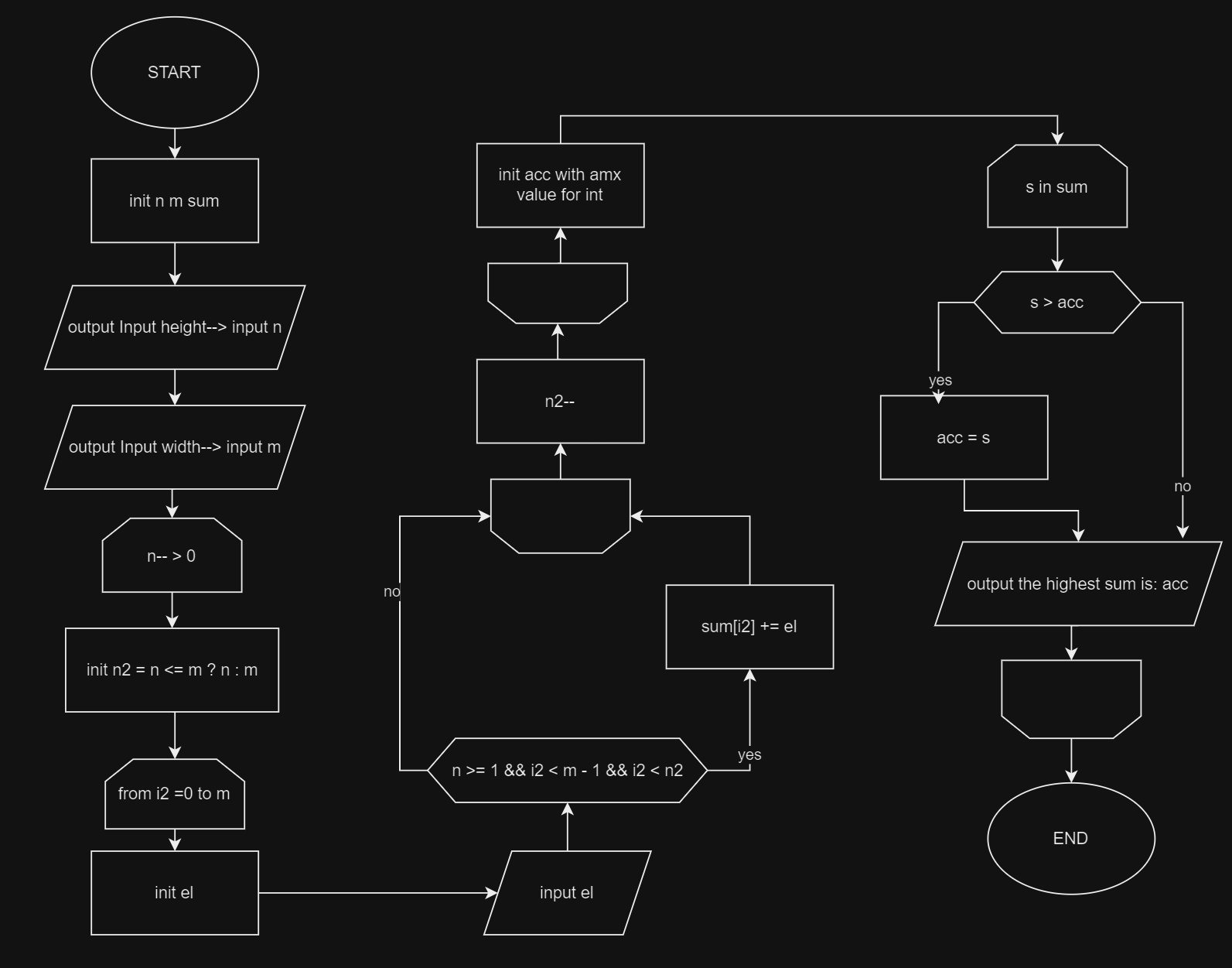
**UML-diagram block-scheme for each task**

VNS Lab 4 Task 1



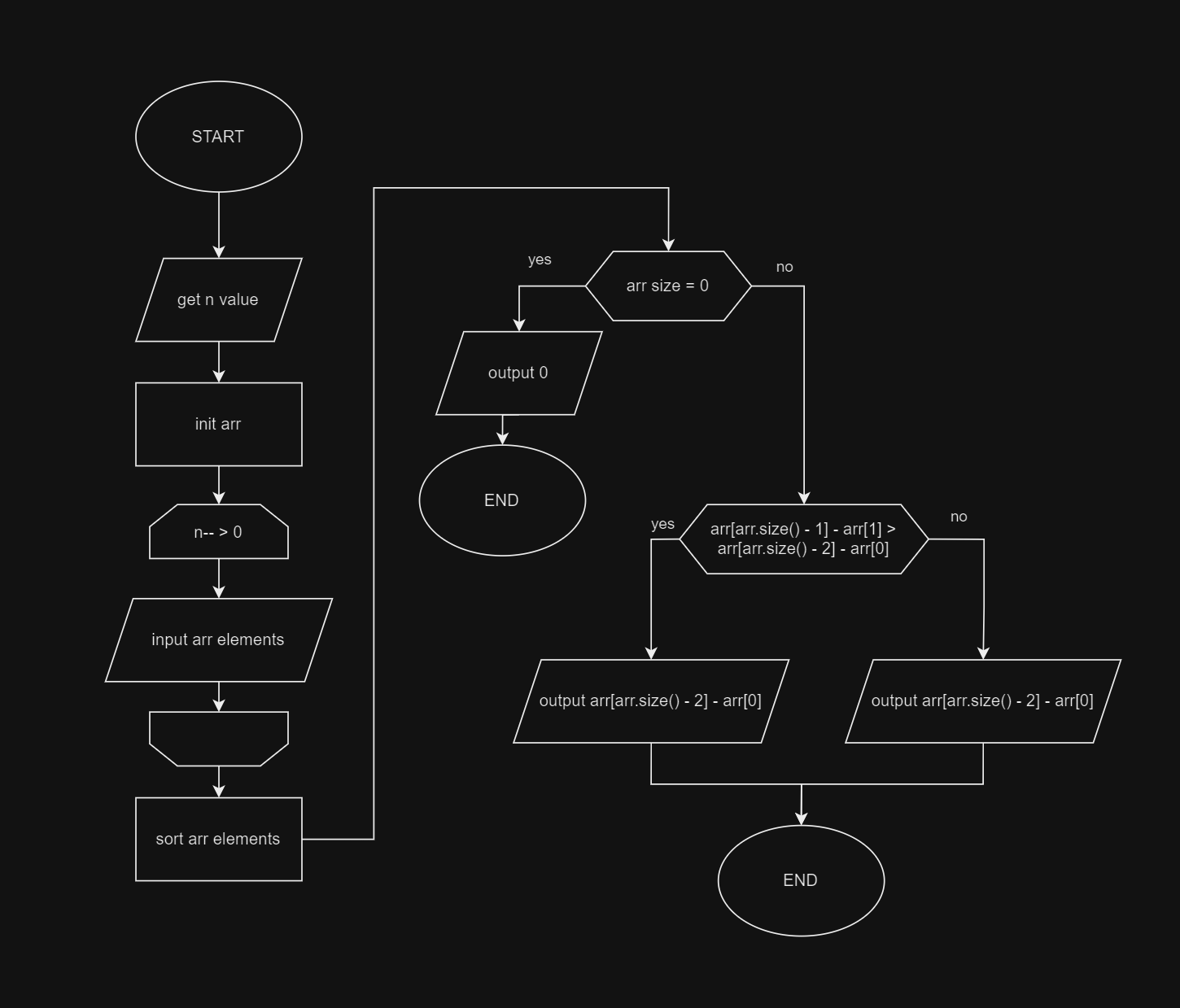
*~40хв*

*VNS Lab 5 Task 1*



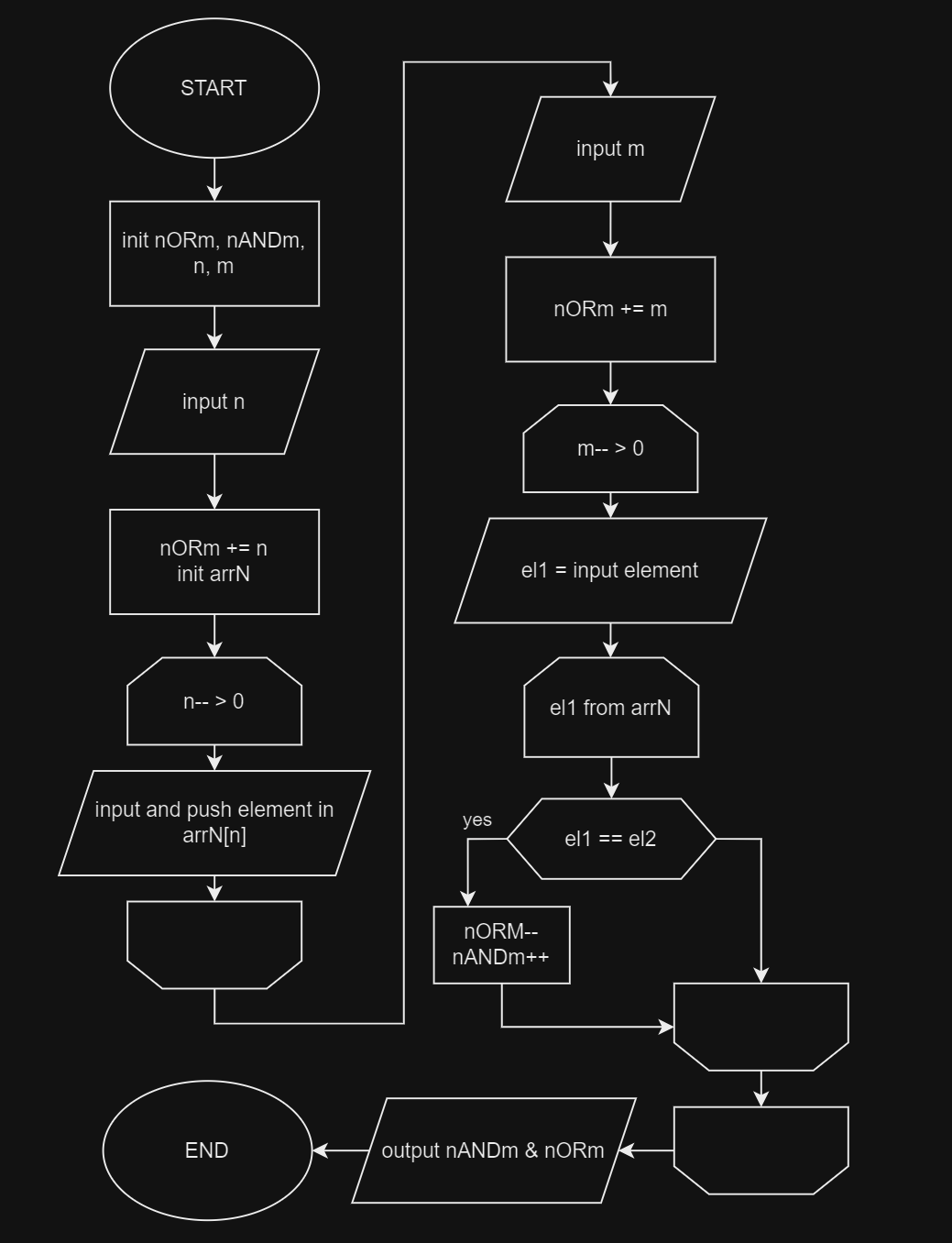
~20хв

Algotester Lab 2 v1



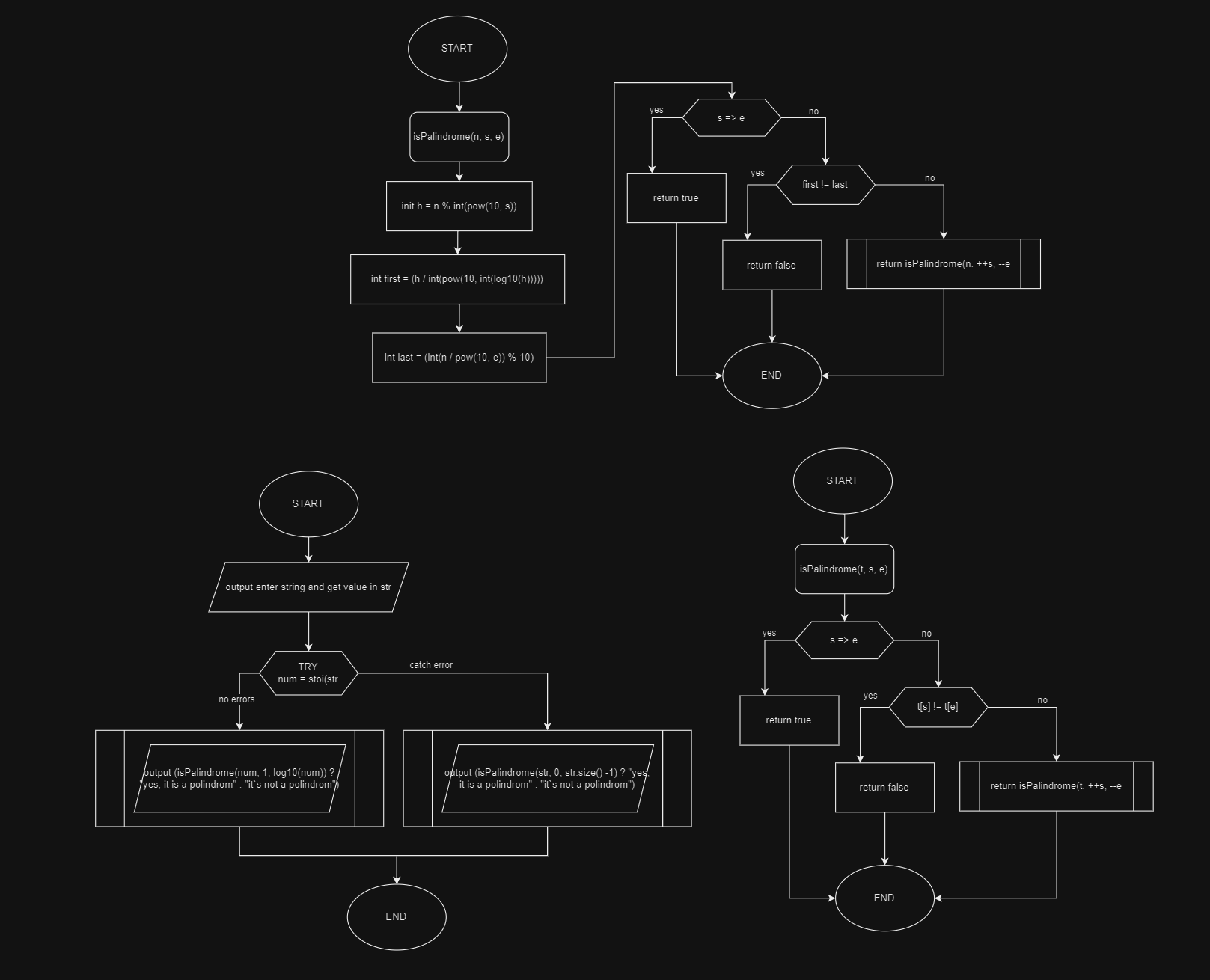
~35хв

ALgotester lab 3 v2



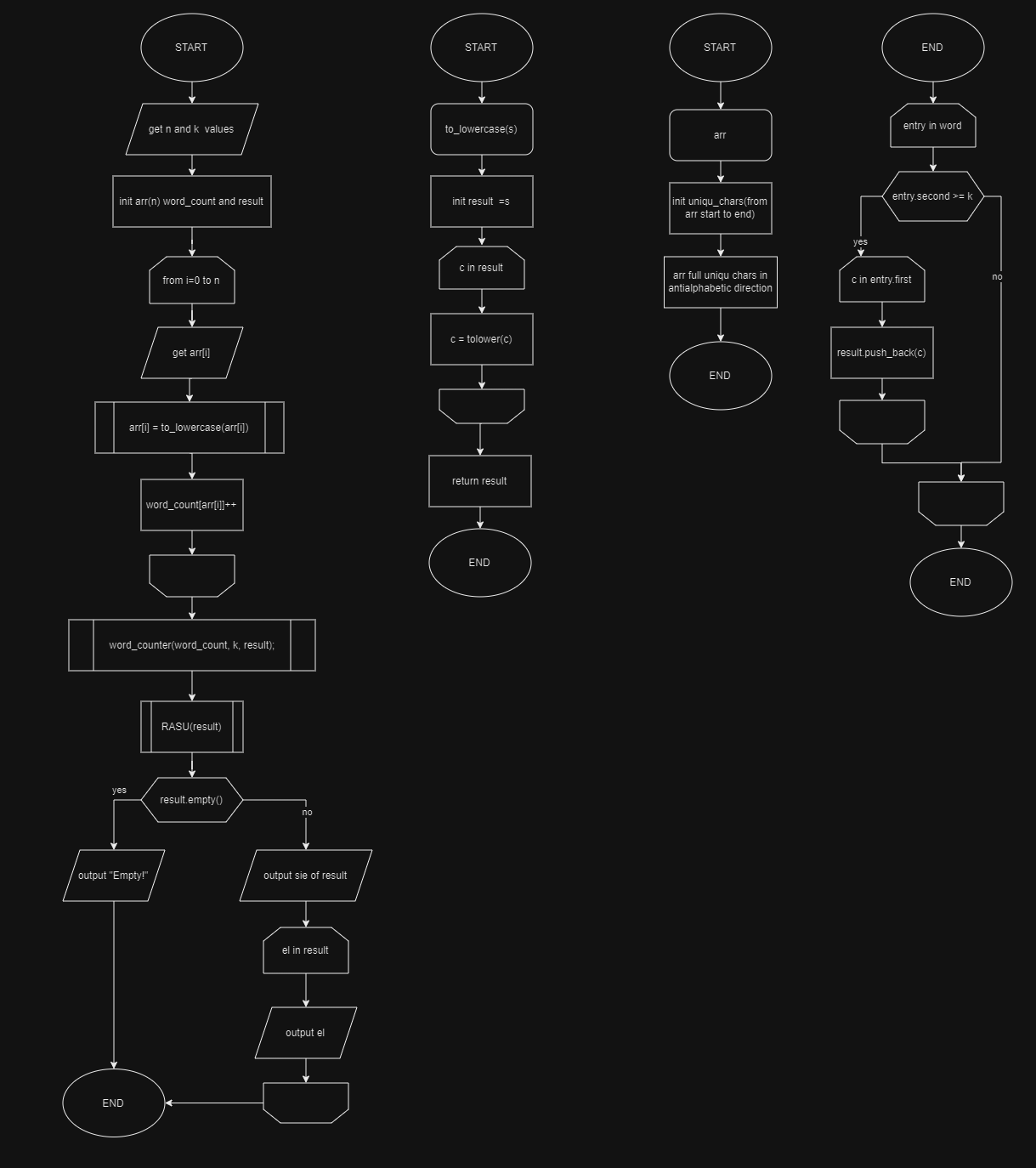
~45хв

Class Practice Task



~20хв

Self Practice Task



~30хв

Код програми з посиланням на зовнішні ресурси

VNS Lab 2 Task 1: **./src/vns\_lab\_4\_task\_andrii\_fediv.cpp**

VNS Lab 3 Task 1: **./src/vns\_lab\_5\_task\_andrii\_fediv.cpp**

Algotester Lab 2v1: **./src/algotester\_lab\_2\_task\_andrii\_fediv.cpp**

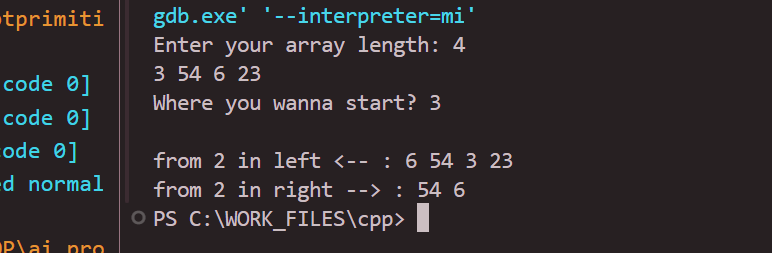
Algotester Lab 3v2: **./src/algotester\_lab\_3\_task\_andrii\_fediv.cpp**

Class Practice Task: **./src/practice\_work\_team\_tasks\_andrii\_fediv.cpp**

Self Practice Task: **./src/practice\_work\_self\_algotester\_tasks\_andrii\_fediv.cpp**

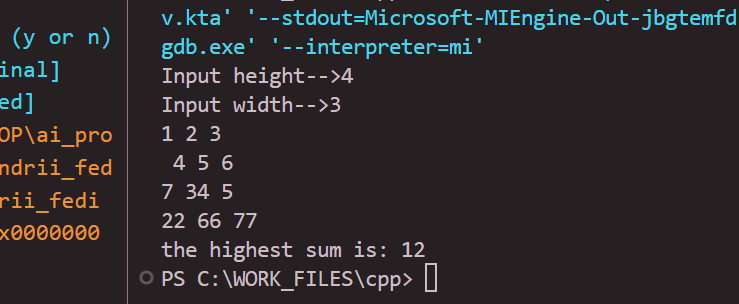
**Результати виконаних завдань, тестування та фактично затрачений час**

VNS Lab 4 Task 1



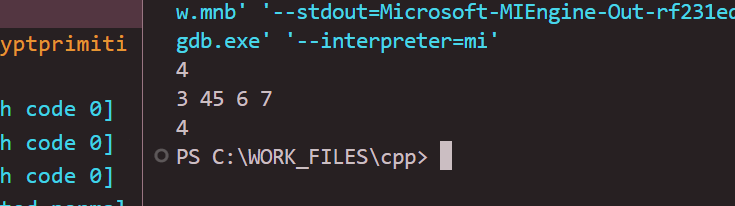
Затратність ~1год

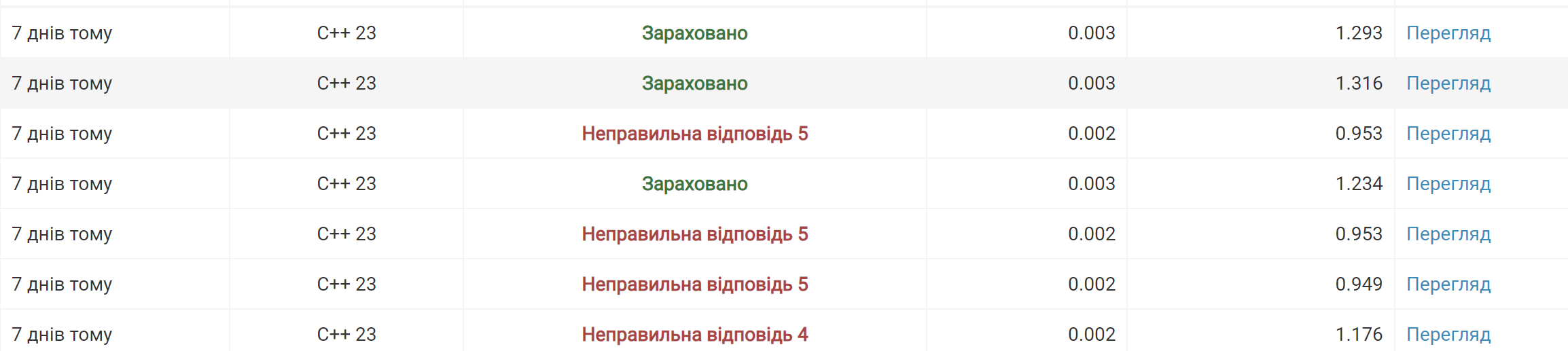
VNS Lab 5 Task 1



Затратність ~40хв

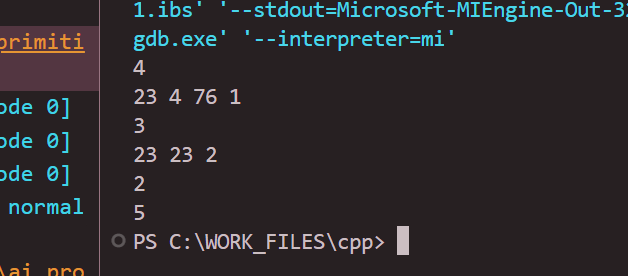
Algotester Lab 2v1

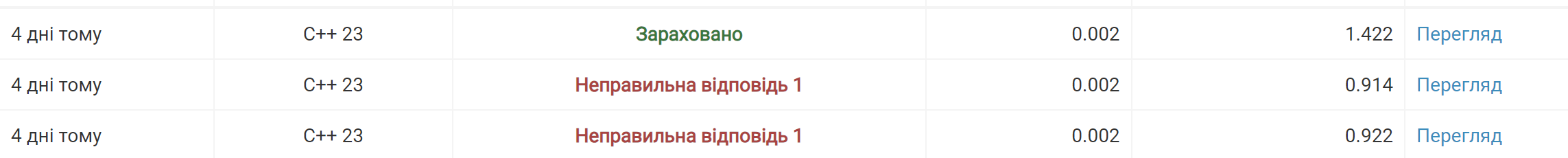




Затратність ~10хв

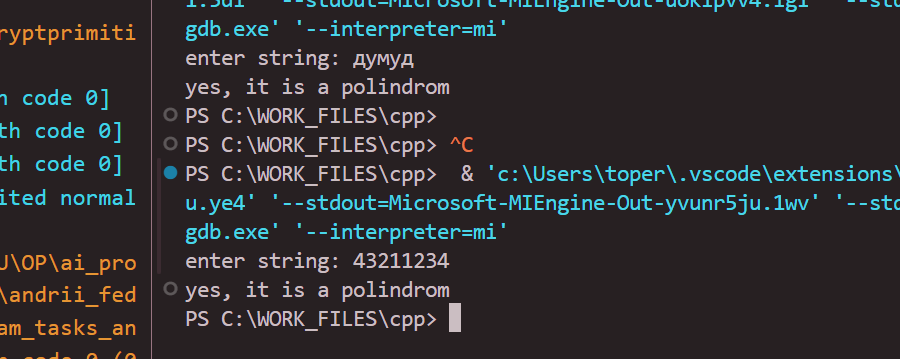
Algotester Lab 3v2





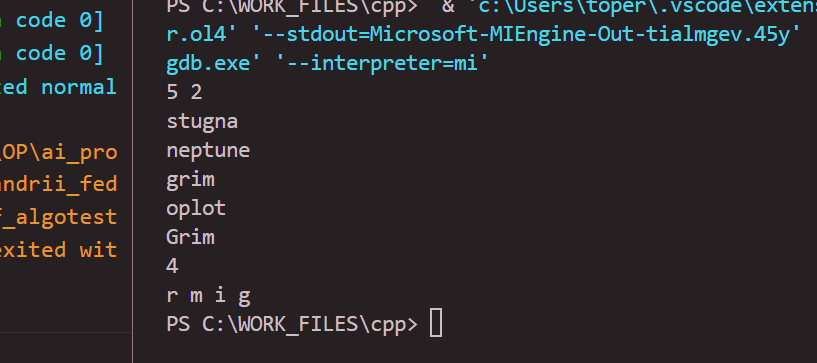
Затратність ~15хв

Class Practice Task



Затратність ~50хв (а на легшу версію 3хв ☹)

Self Practice Task



****

Затратність ~1.5год

**Висновки:**

Виконуючи цей епік я поглибив свої знання у зберіганні та обробці даних, що суттєво спрощує програмування. Застосуванн вказівників і динамічних масивів дозволяє зекономити пам'ять і гнучко управляти даними, а розуміння структур даних та алгоритмів обробки допомагає створювати оптимізовані рішення для різноманітних задач.