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 № 6**

На тему: «Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки динамічних структур.»

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

до:

ВНС Лабораторної Роботи № 10

Алготестер Лабораторної Роботи № 5

Алготестер Лабораторної Роботи № 7-8

Практичних Робіт до блоку № 6

**Виконав:**

Студент групи ШІ-12

Кутельмах Євген Петрович

Львів 2024

**Тема:** Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки динамічних структур

**Мета:** Отримати практичні навички у роботі з динамічними структурами, алгоритмами обробок динамічних структур

**Теоретичні відомості:**

1. *Стек і купа, Черги*

<https://acode.com.ua/urok-111-stek-i-kupa/>

[*https://www.youtube.com/watch?v=Yhw8NbjrSFA&ab\_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD*](https://www.youtube.com/watch?v=Yhw8NbjrSFA&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD)

Завдяки цьому уроці, я зрозумів чим відрізняється стек і купа, коли варто використати певний тип даних. Ще на цьому уроці та через іншу інформацію в інтернеті, зокрема дане відео, я дізнався як можна реалізувати стек власноруч, як чергу, та інші види черг. Витратив на це близько години.

1. *Інші динамічні структури*

[*https://www.youtube.com/watch?v=-25REjF\_atI&t=1189s&ab\_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD*](https://www.youtube.com/watch?v=-25REjF_atI&t=1189s&ab_channel=%D0%91%D0%BB%D0%BE%D0%B3%D0%B0%D0%BD)

На цьому відео я розглянув, як можна створити однозв’язний список, ще виконуючи цю роботу, я розглянув в інтернеті як реалізовуються інші динамічні структури: двозв’язний список, бінарне дерево, динамічний масив, тощо. На це я витратив часу близько 3 годин.

**Виконання роботи:**

*Task 3 - Lab# programming: VNS Lab 10*

Завдання: Написати програму, у якій створюються динамічні структури й виконати їхню обробку у відповідності зі своїм варіантом.

Для кожного варіанту розробити такі функції:

1. Створення списку.

2. Додавання елемента в список (у відповідності зі своїм варіантом).

3. Знищення елемента зі списку (у відповідності зі своїм варіантом).

4. Друк списку.

5. Запис списку у файл.

6. Знищення списку.

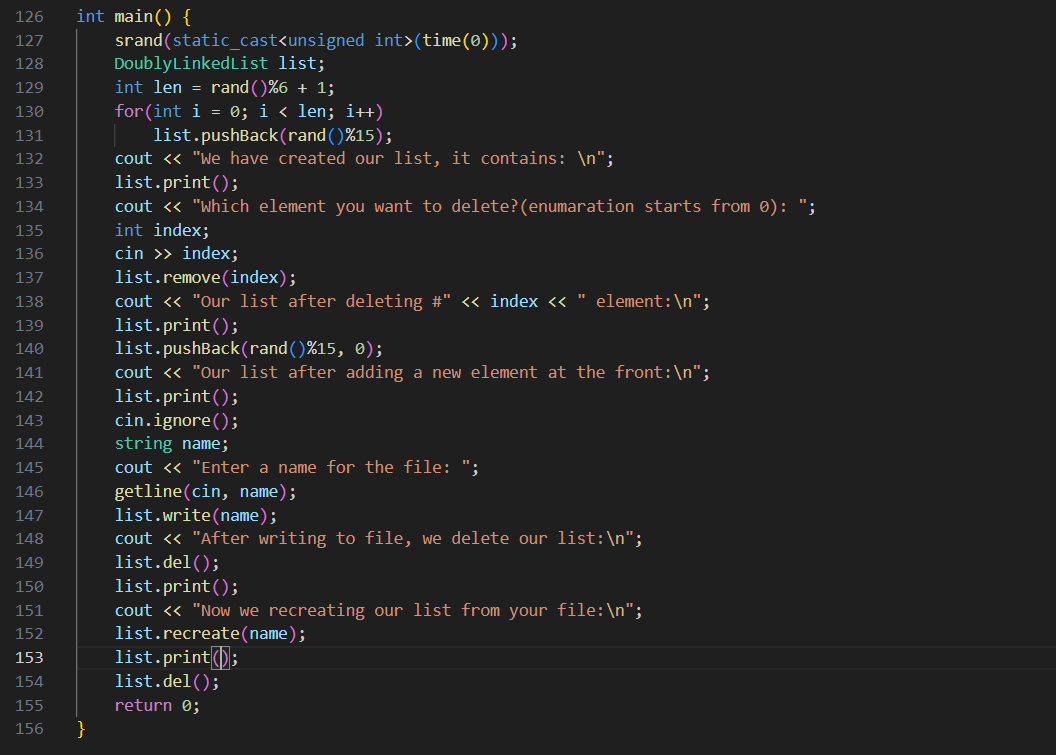
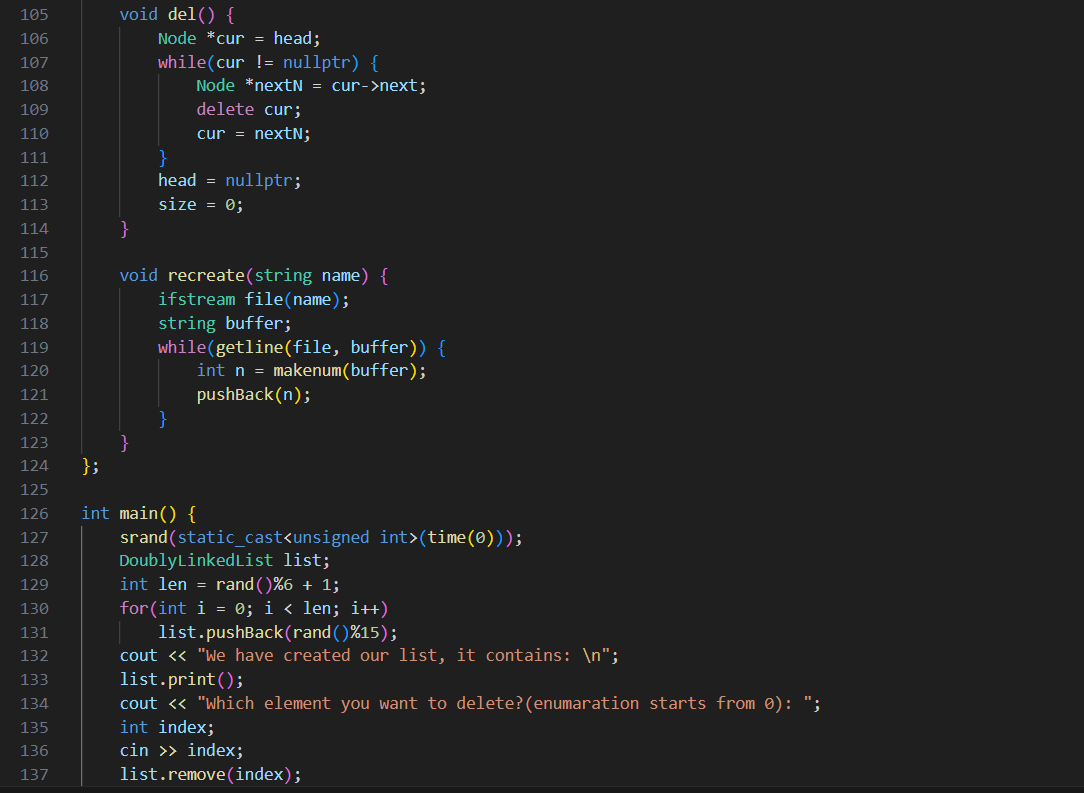
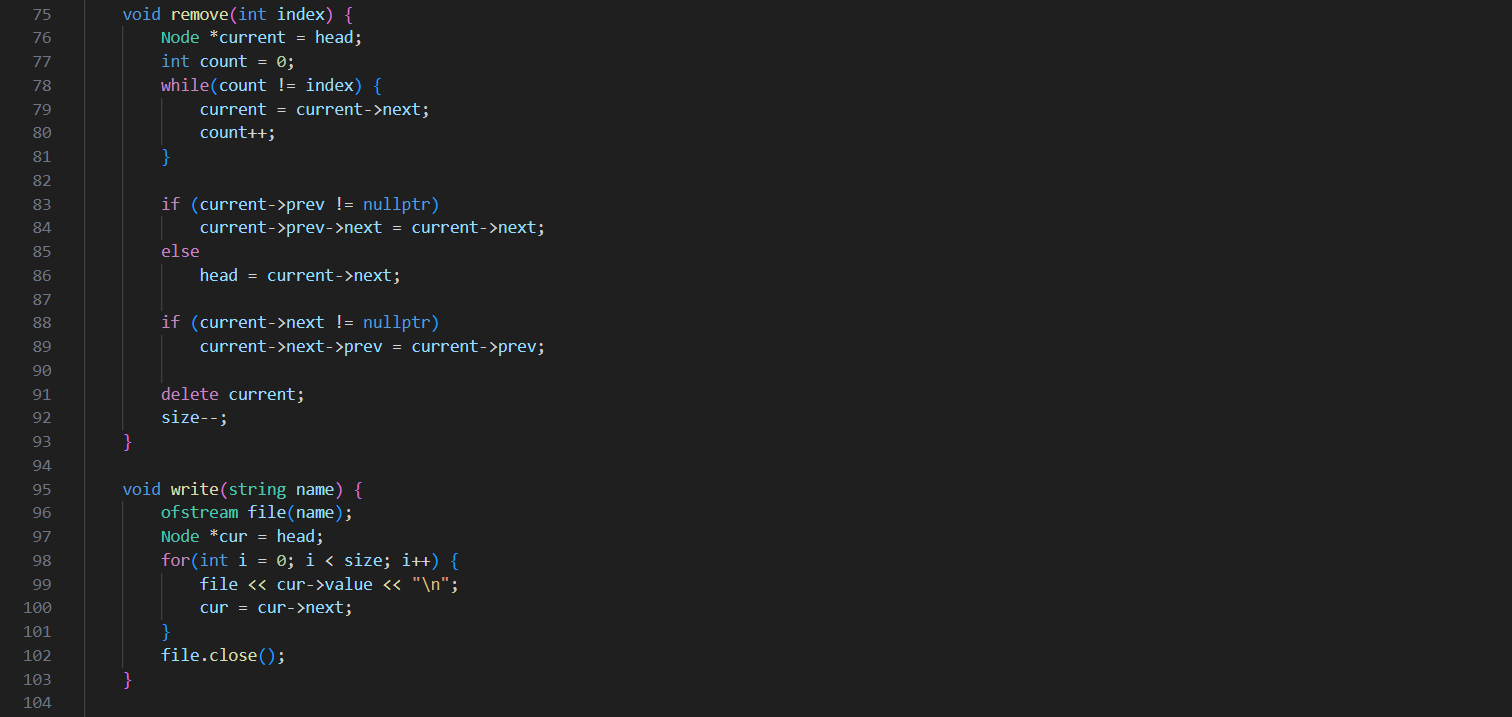
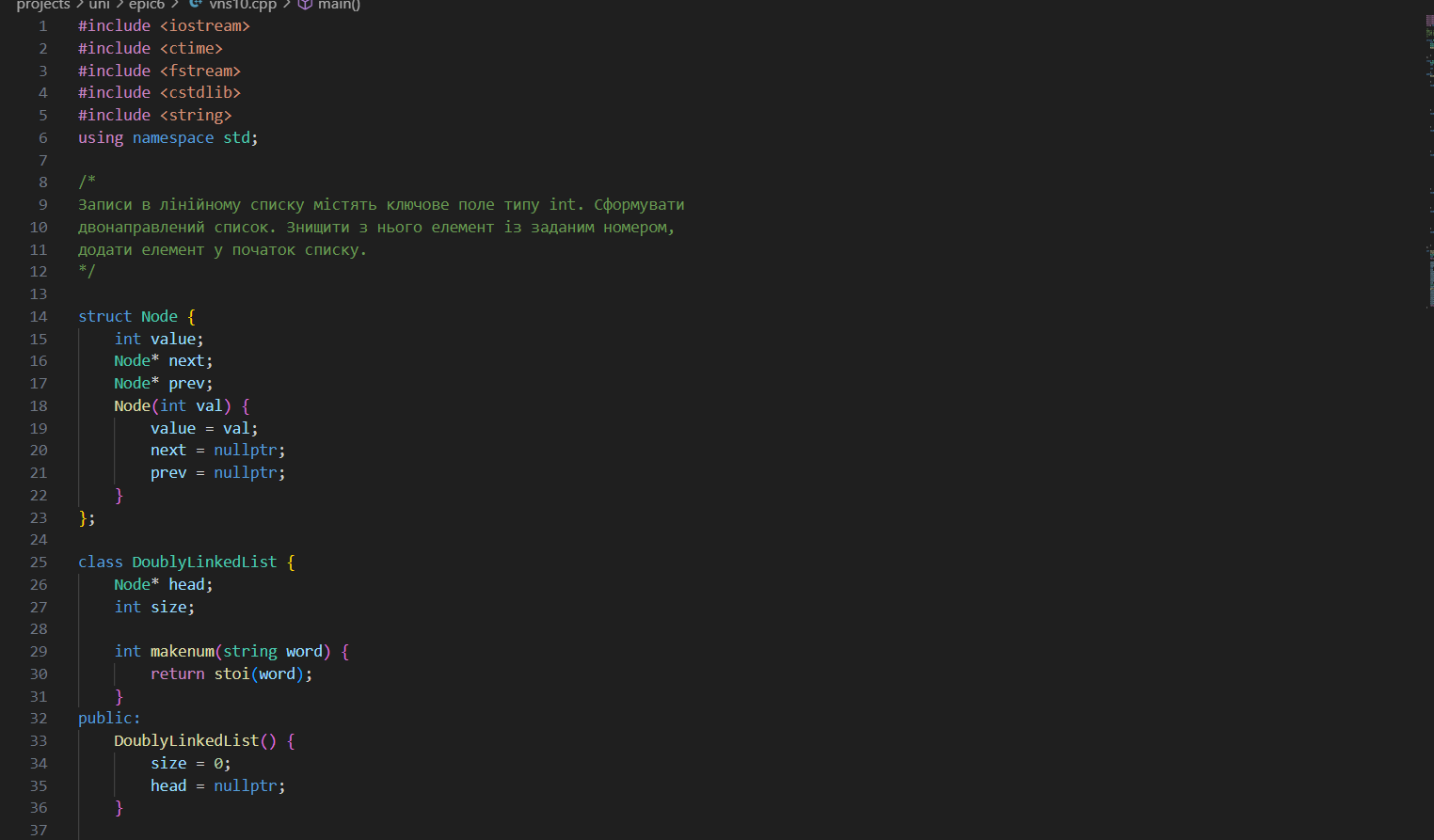
7. Відновлення списку з файлу.

Варіант 6: Записи в лінійному списку містять ключове поле типу int. Сформувати

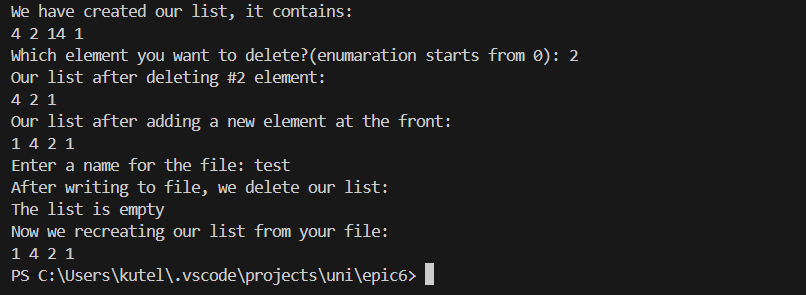
двонаправлений список. Знищити з нього елемент із заданим номером,

додати елемент у початок списку.

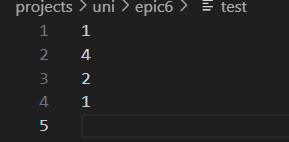
Програма:



Результат:



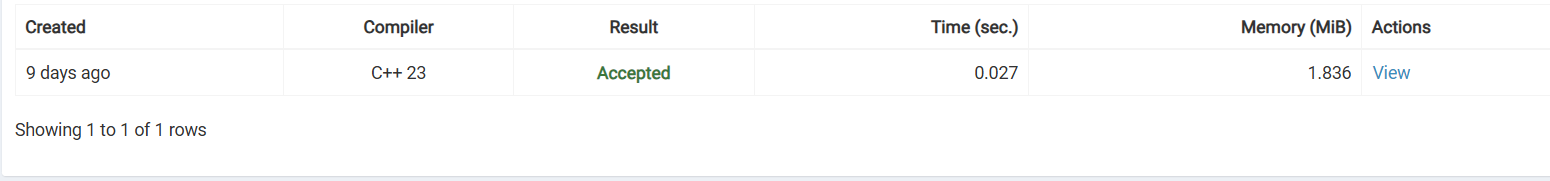
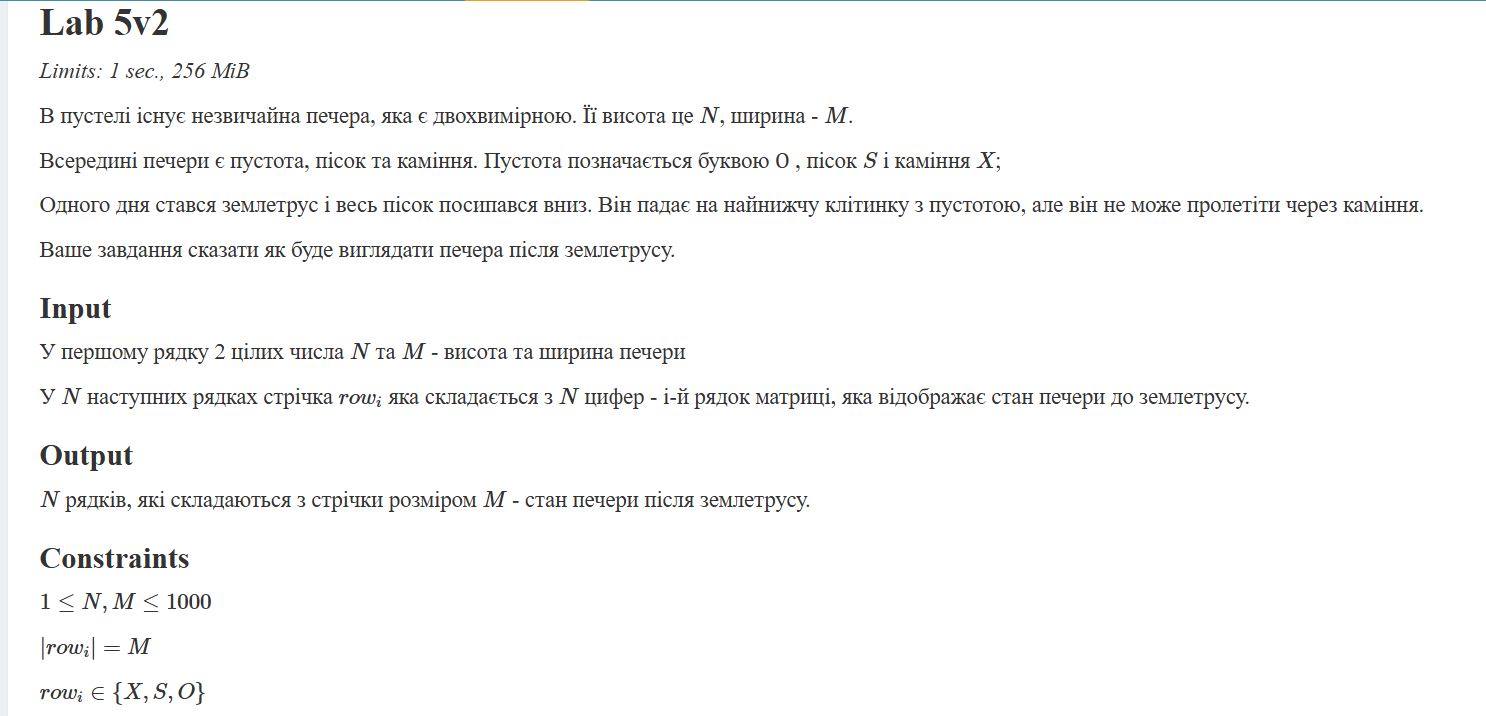
test:

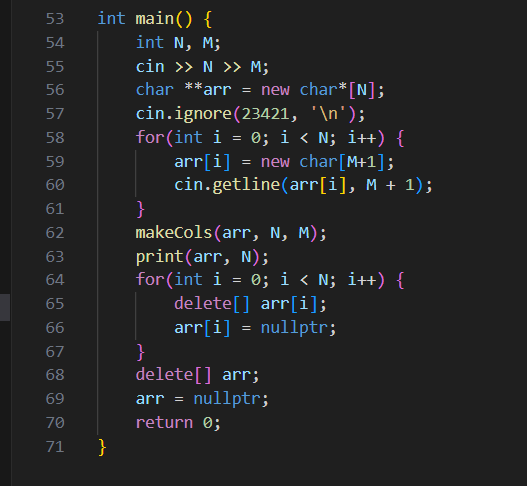
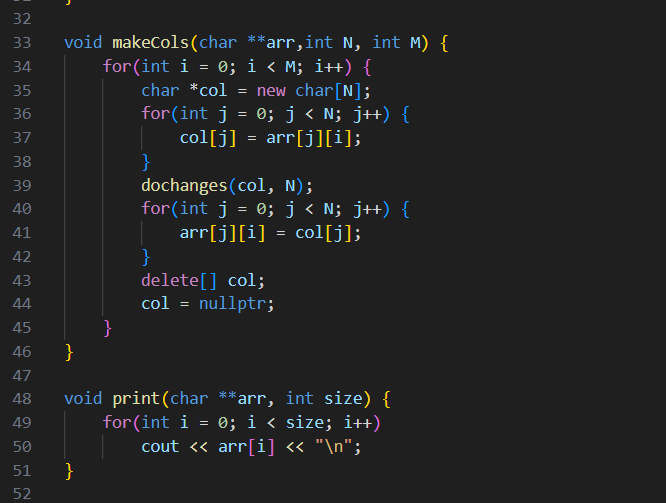
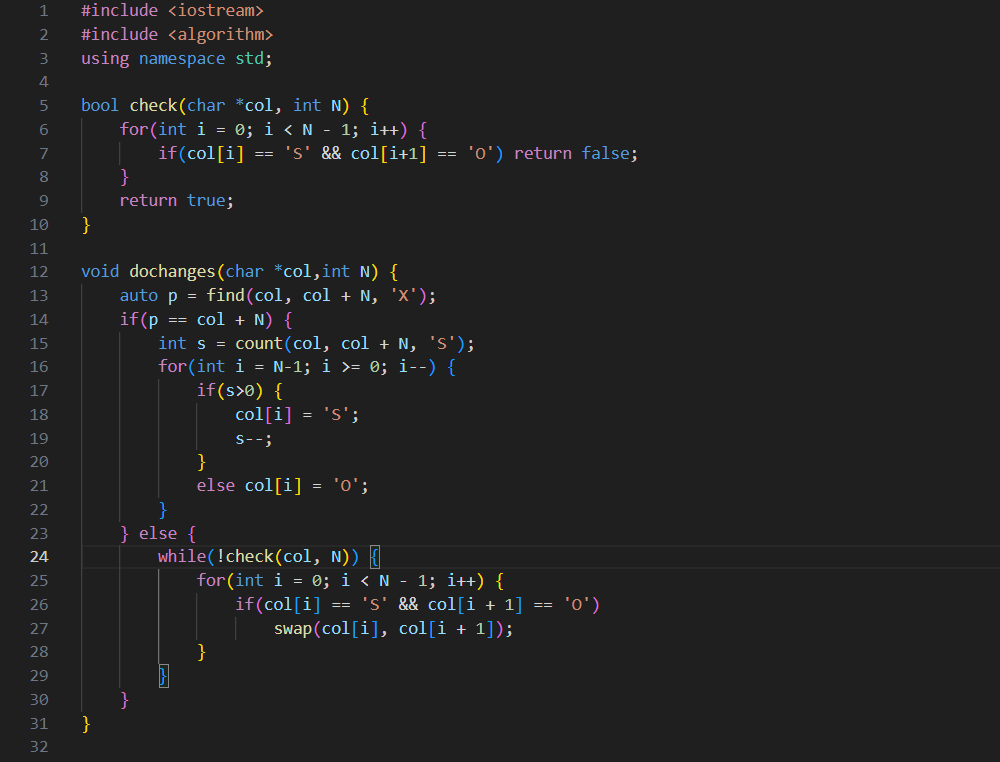


На це завдання я витратив близько 2 годин часу.

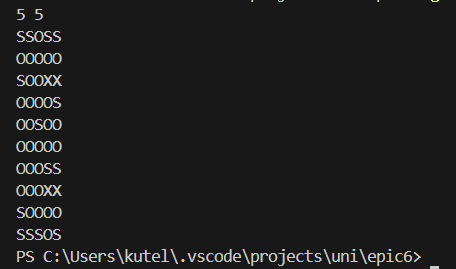
*Task 4 - Lab# programming: Algotester Lab 5*

Умова:

Програма:



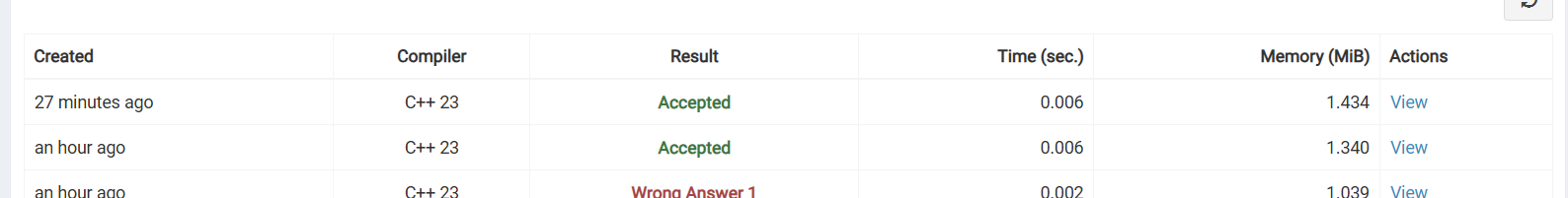
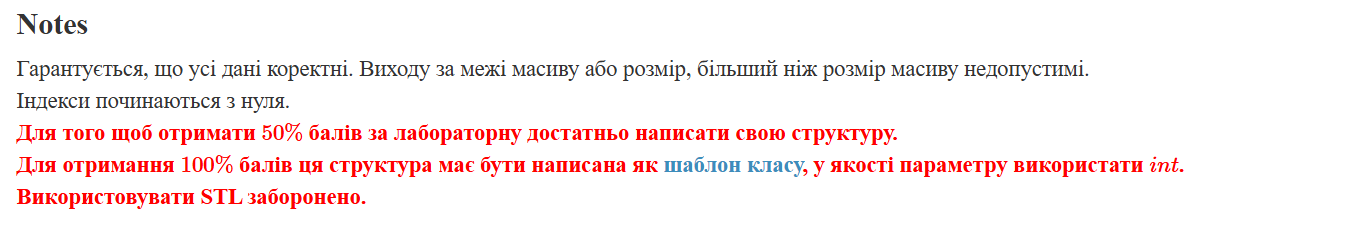
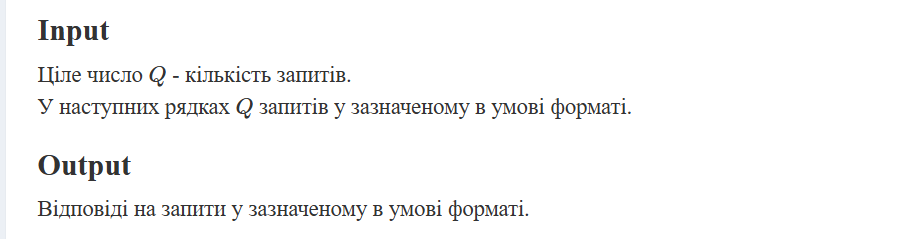
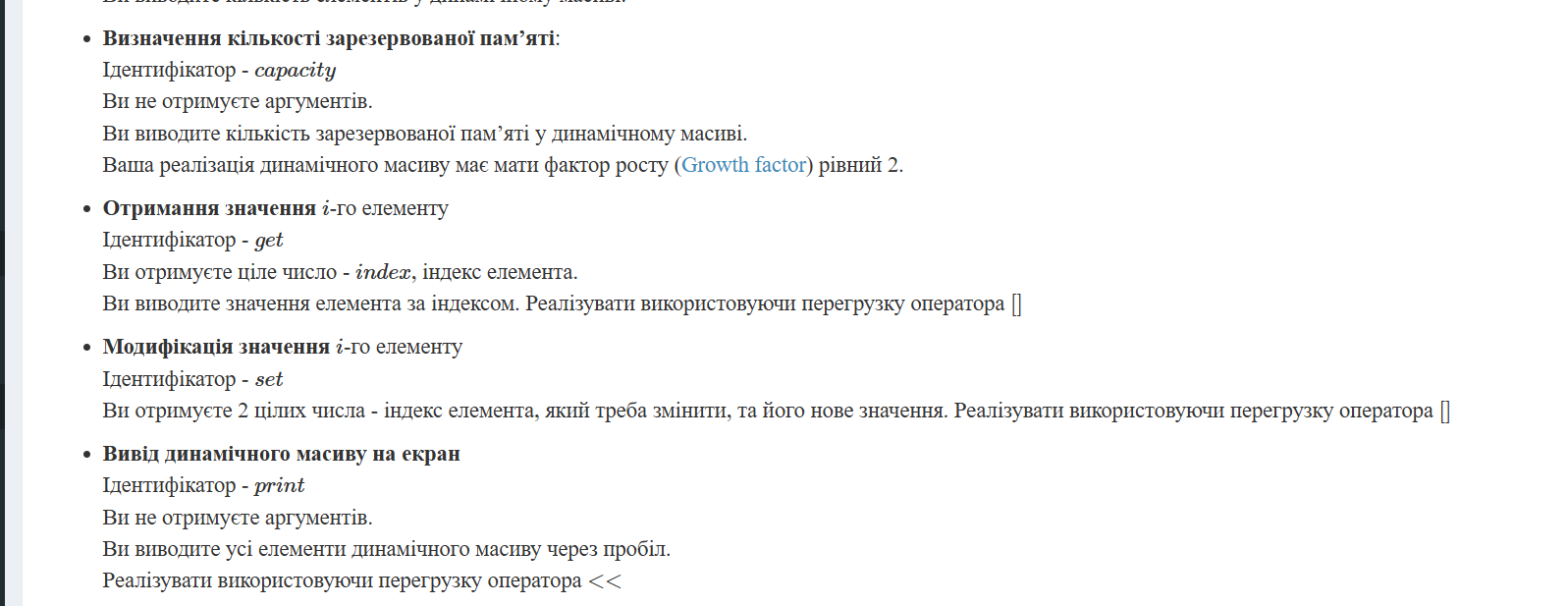
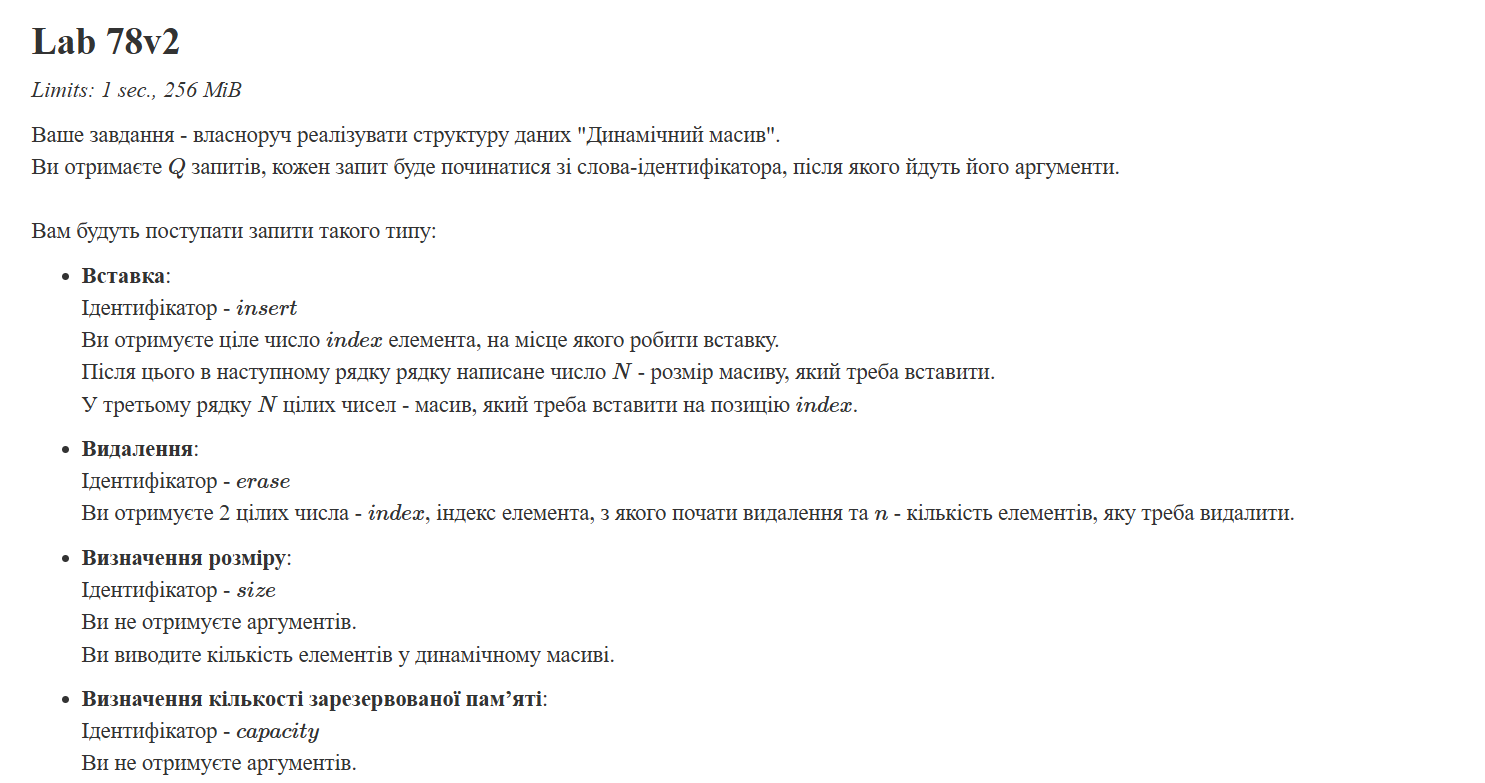
Результат:



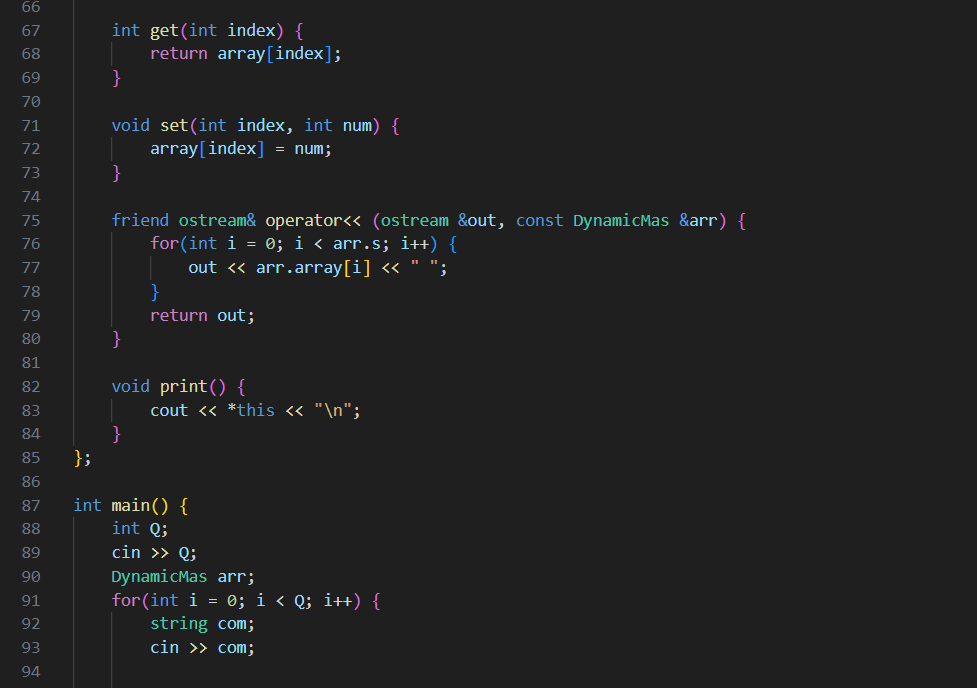
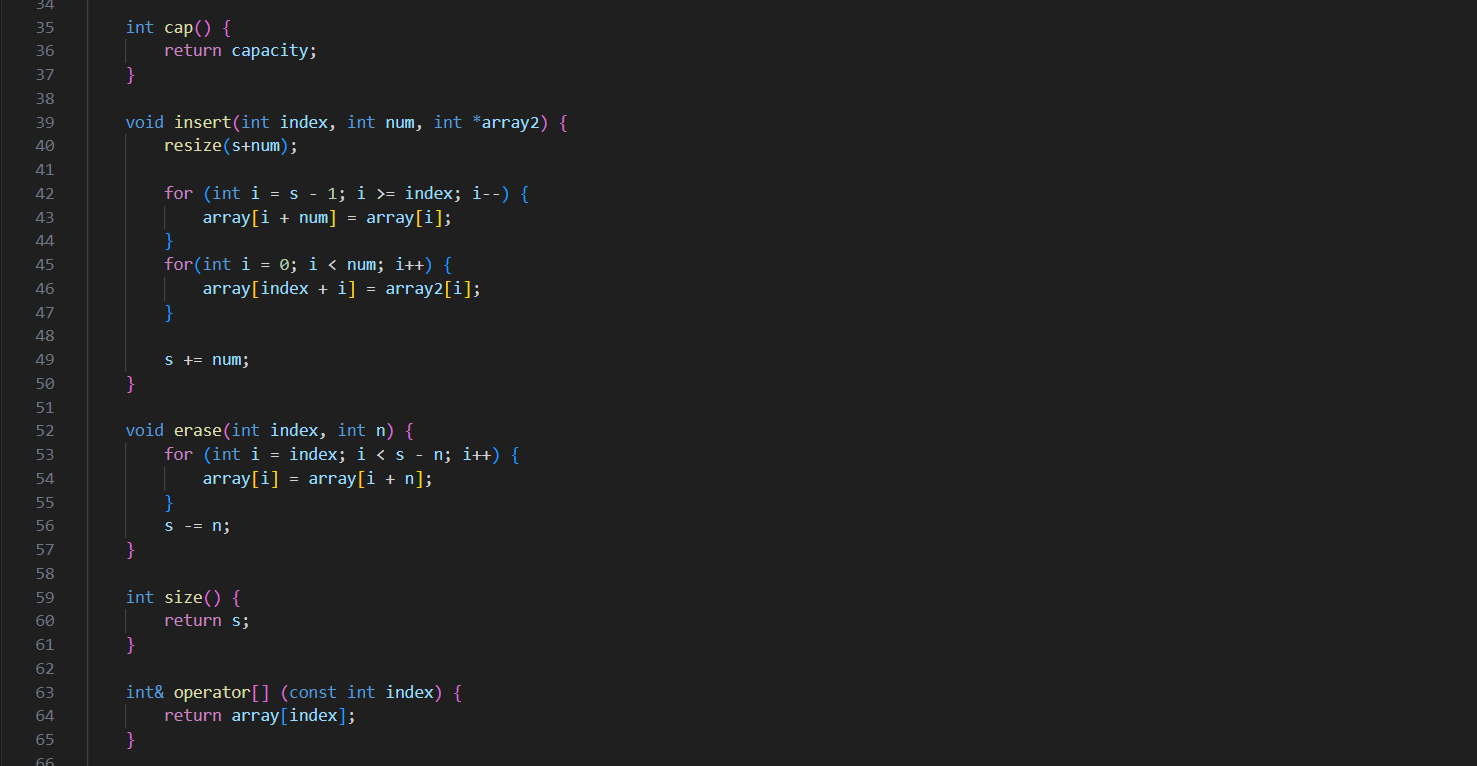
На дану задачу я витратив 30 хв.

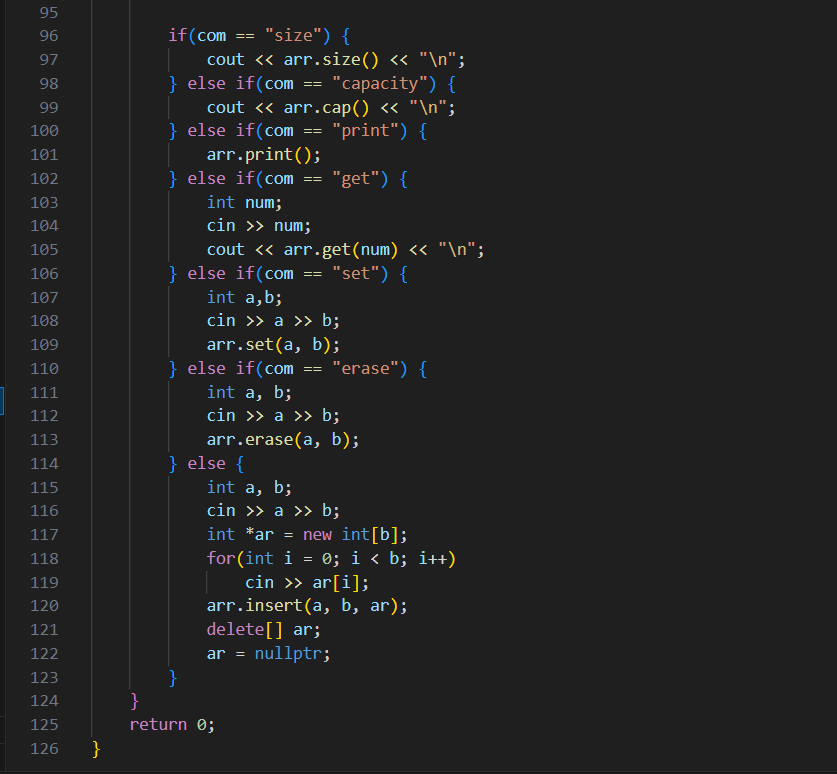
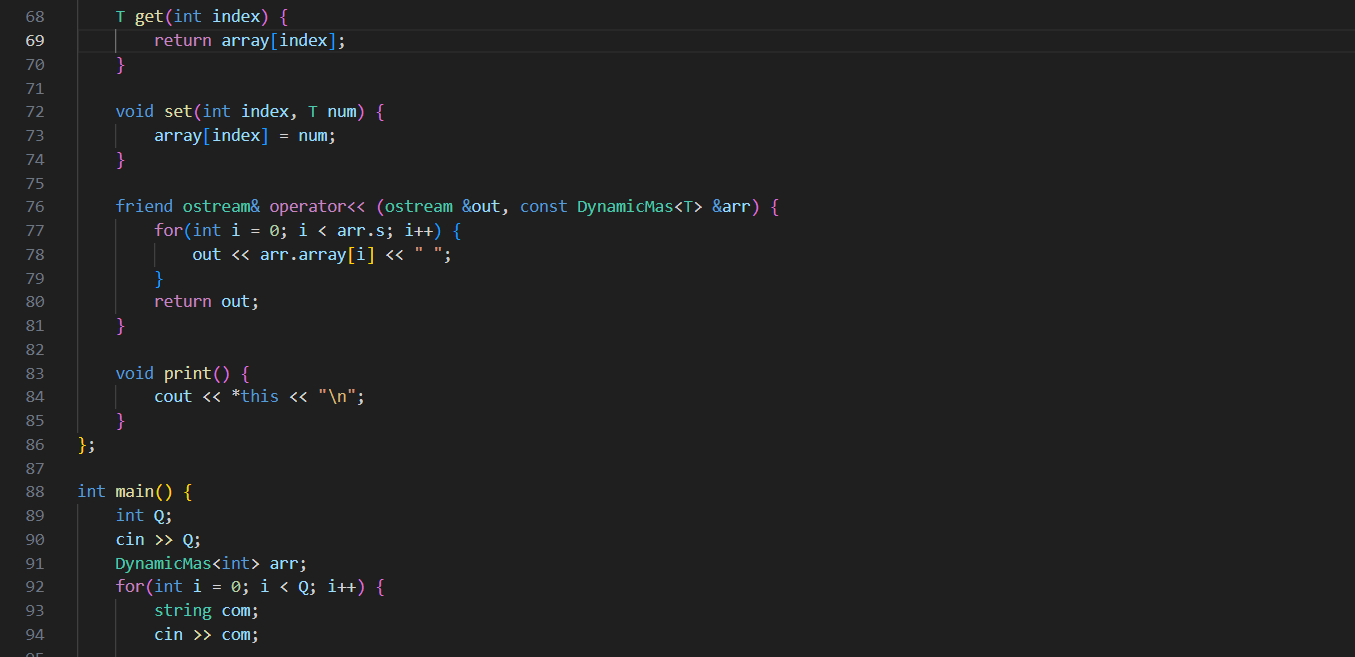
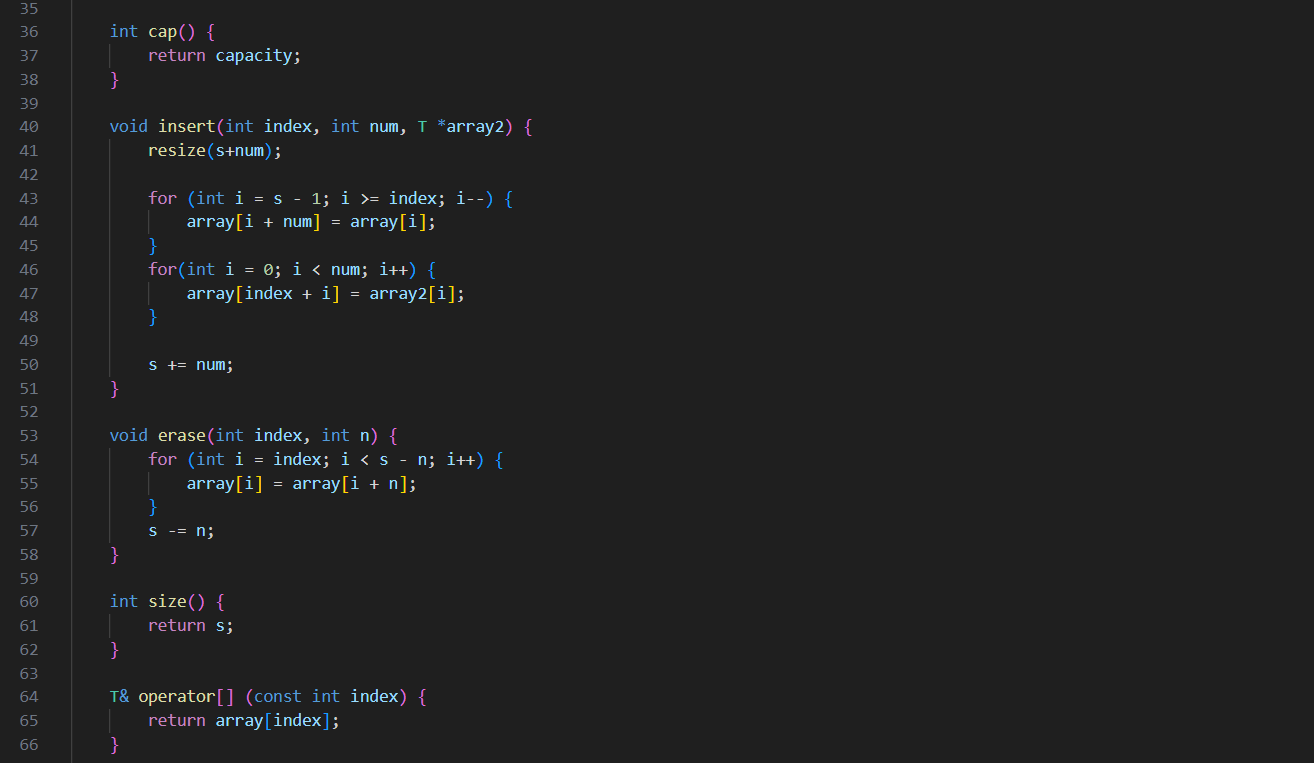
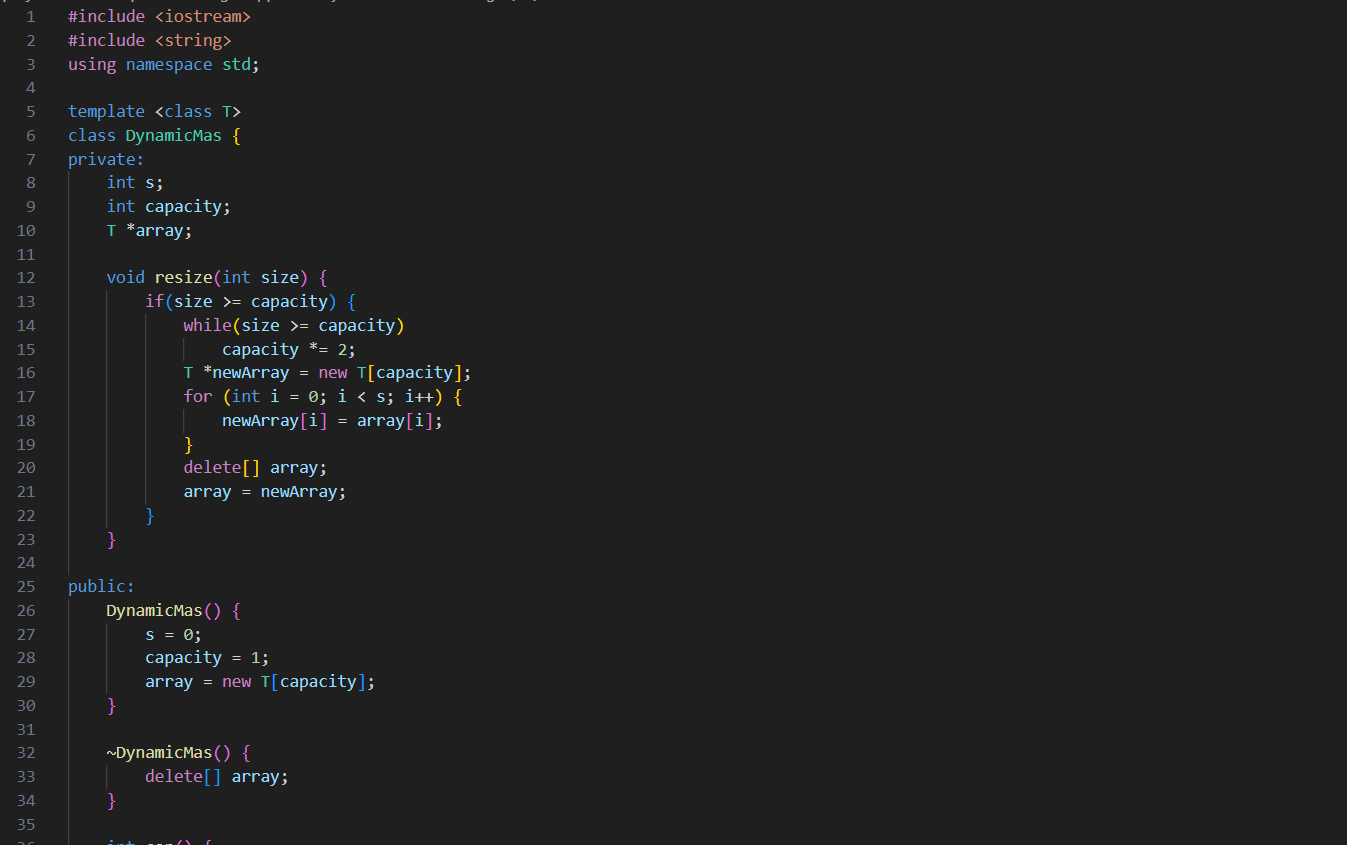
*Task 5 - Lab# programming: Algotester Lab 7-8*

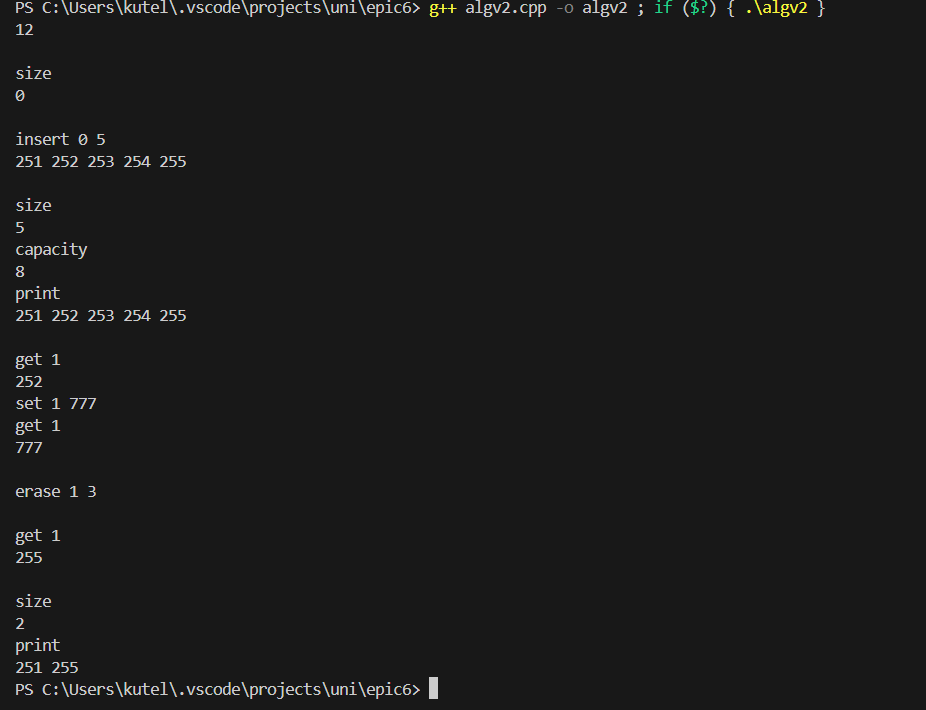
Умова:



Програма(з використанням структури):

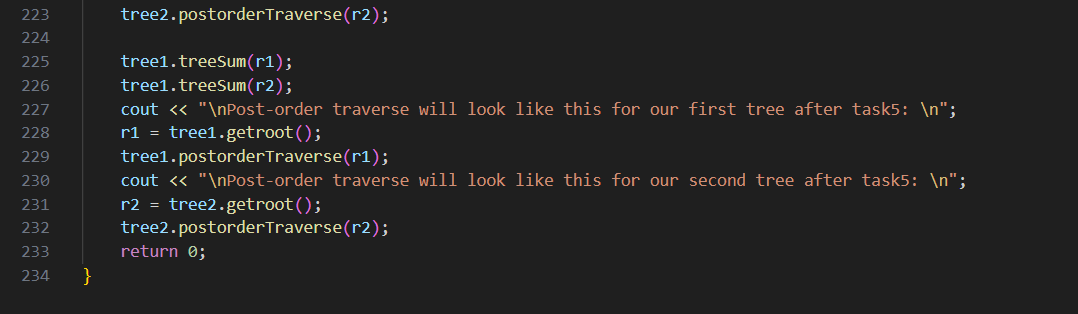
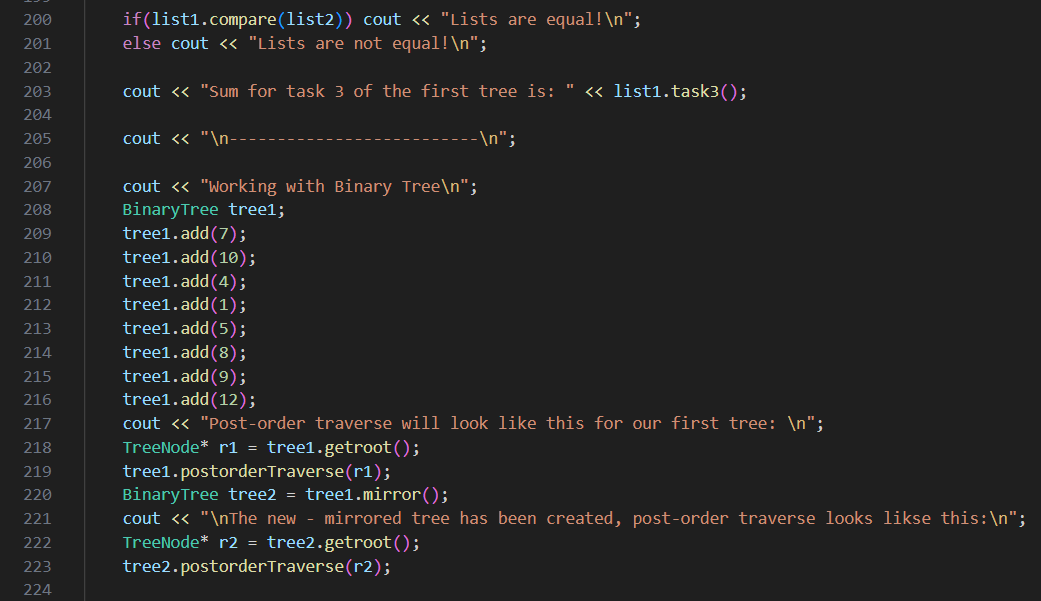
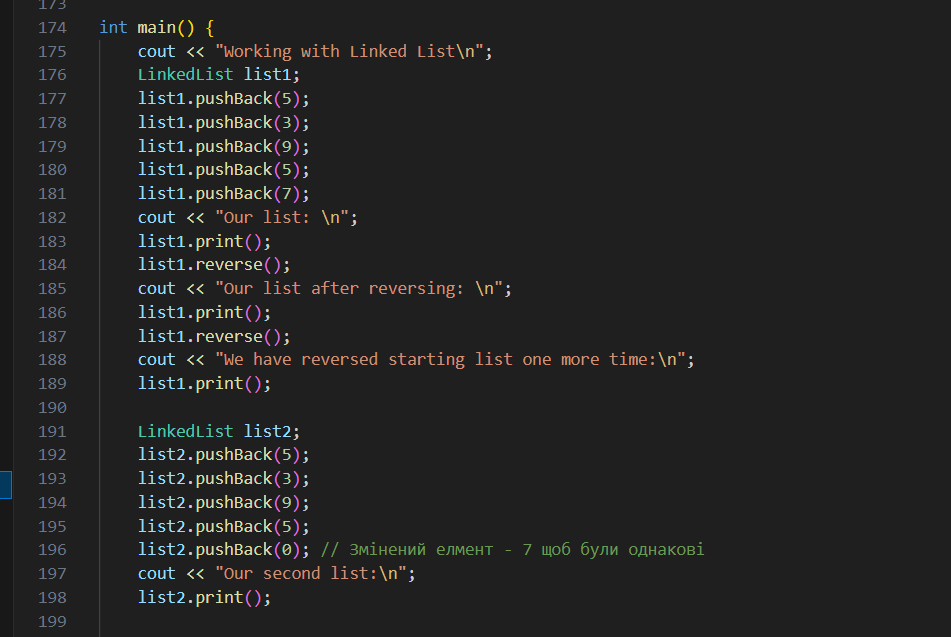
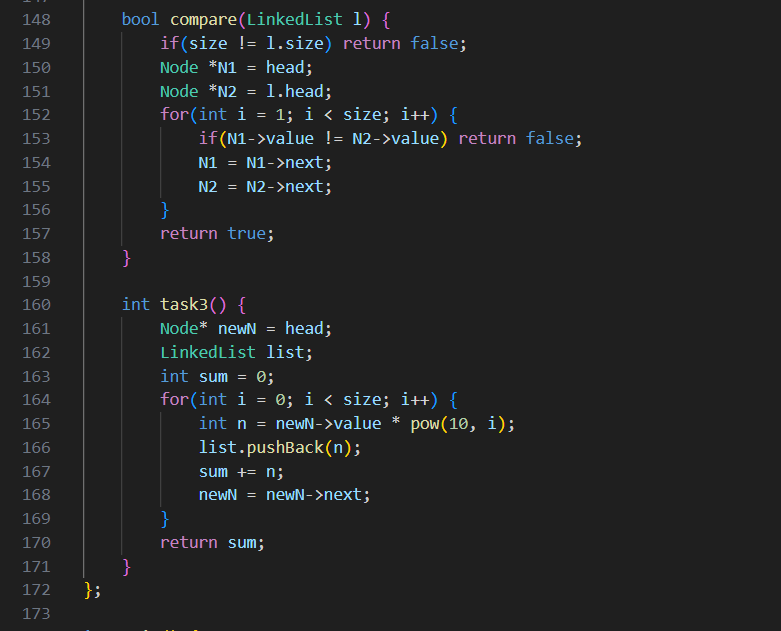
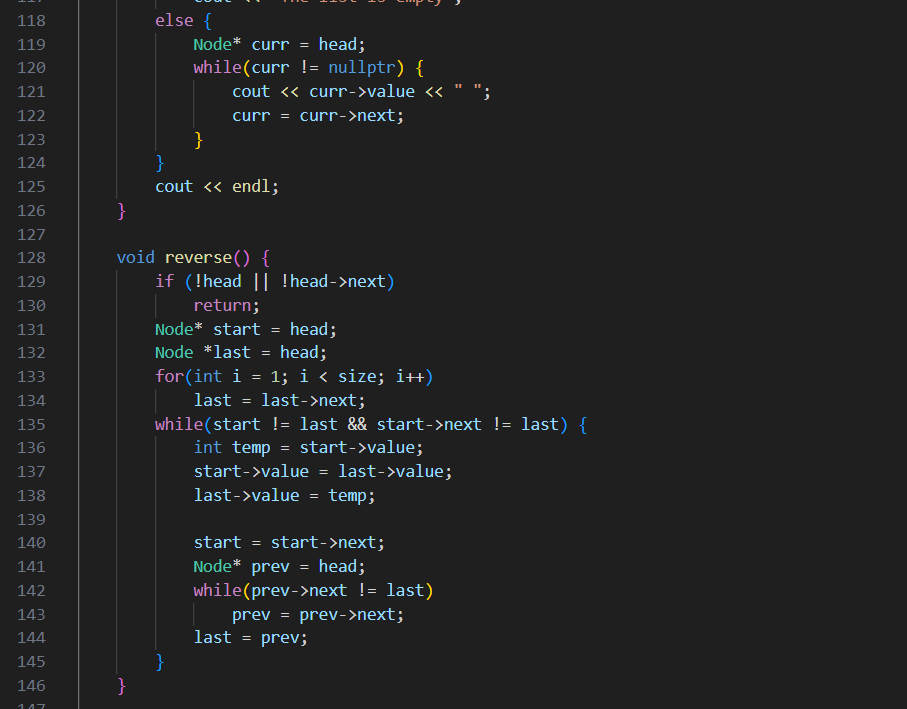
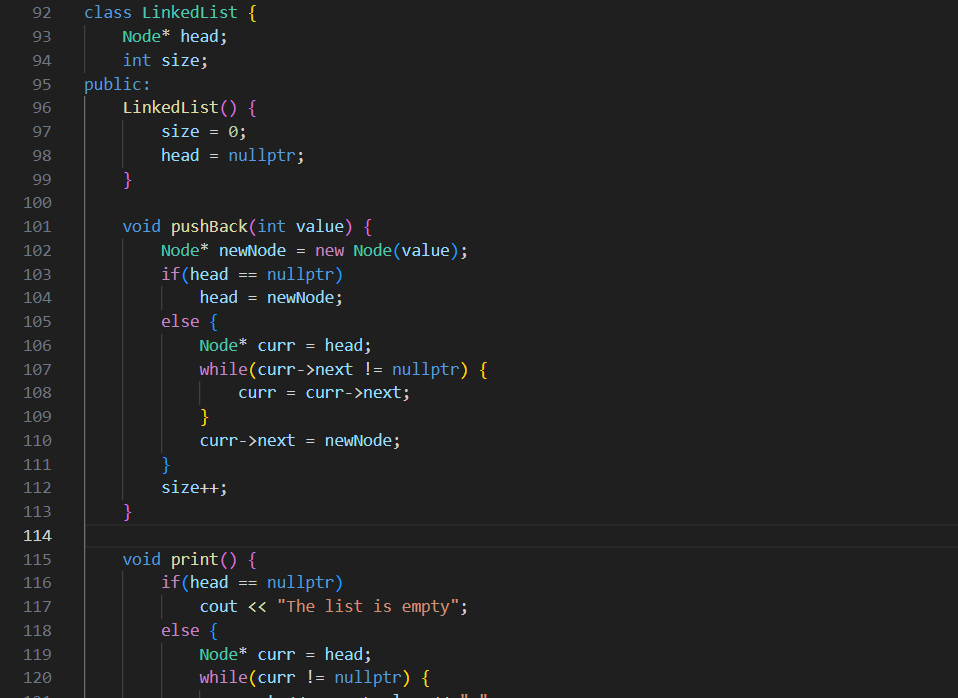
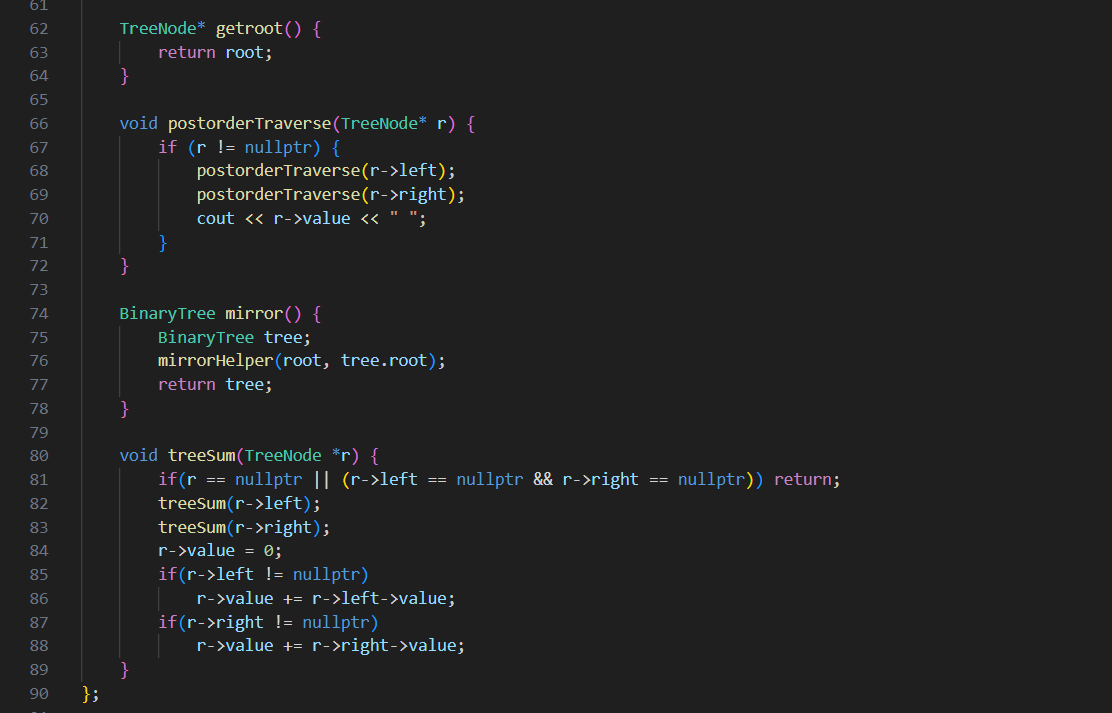
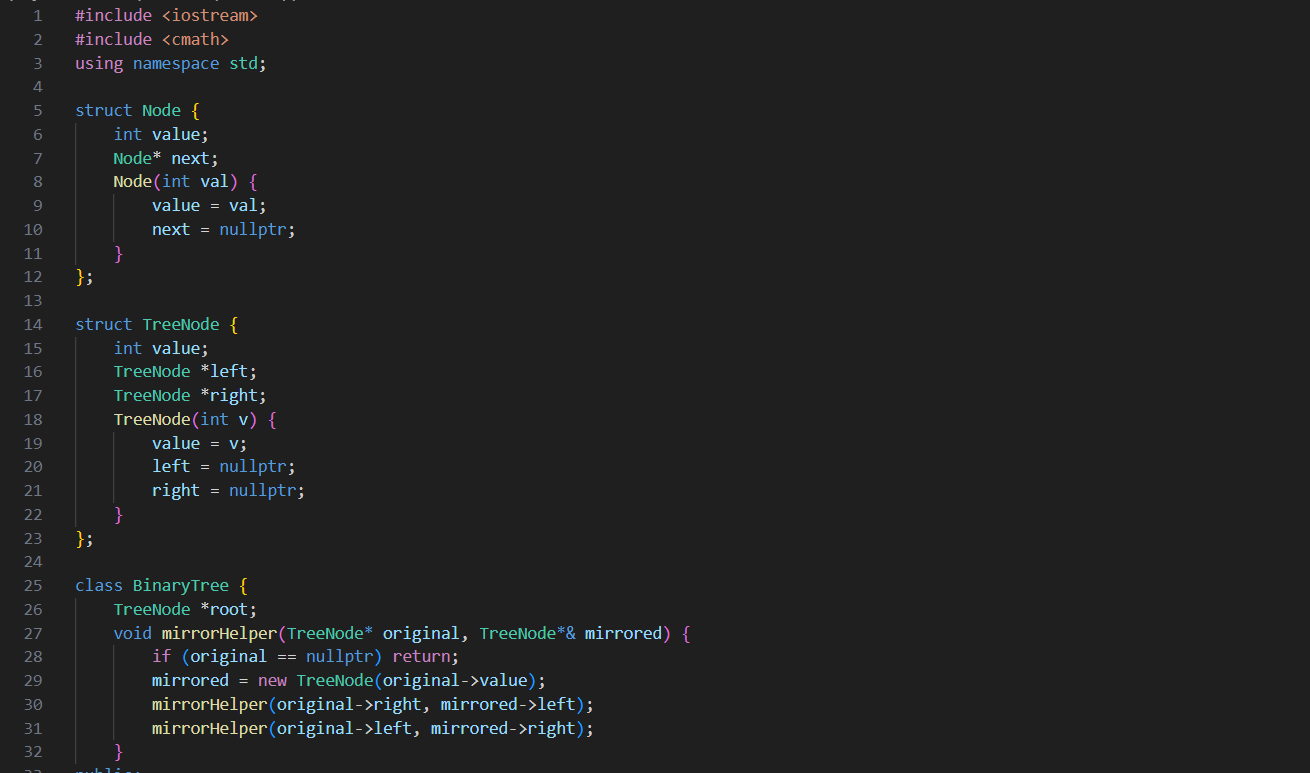
Програма(з використанням шаблону класу):

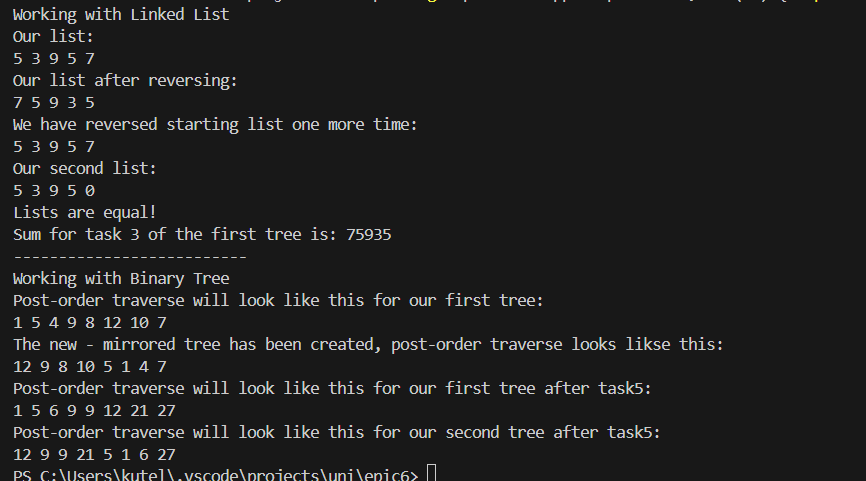
Результат:

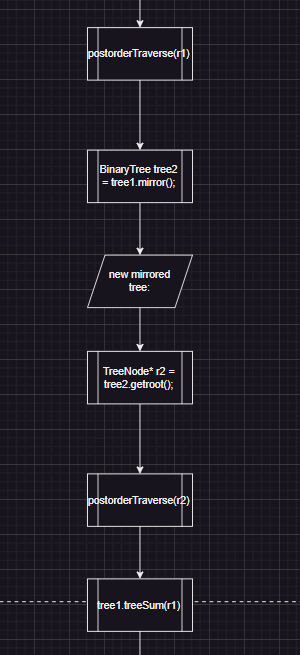
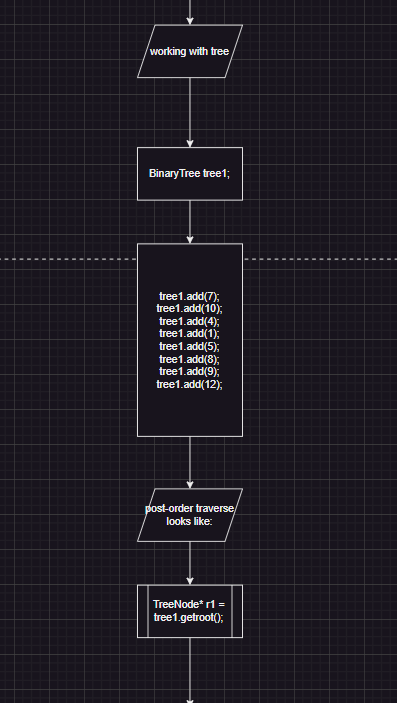
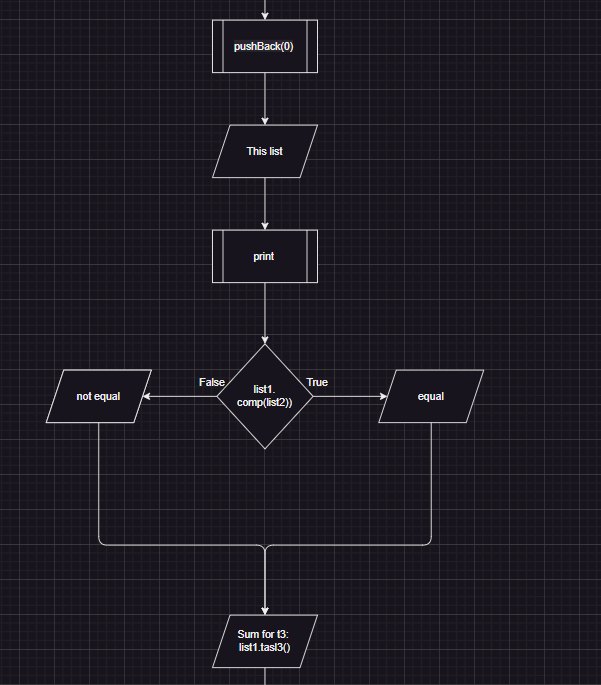
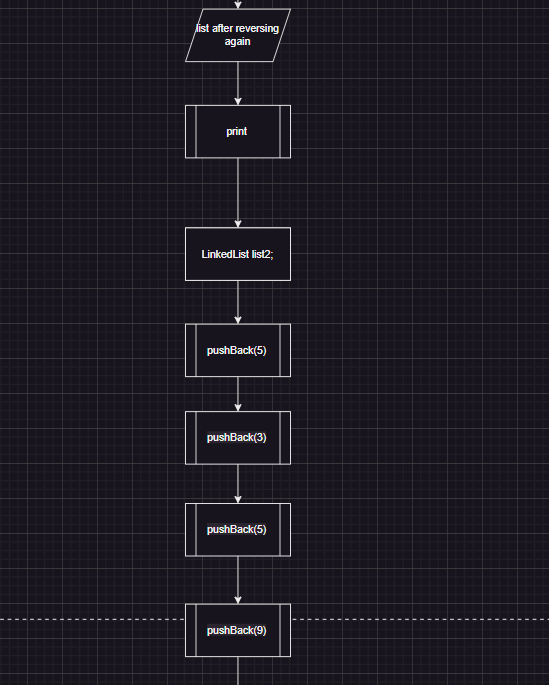
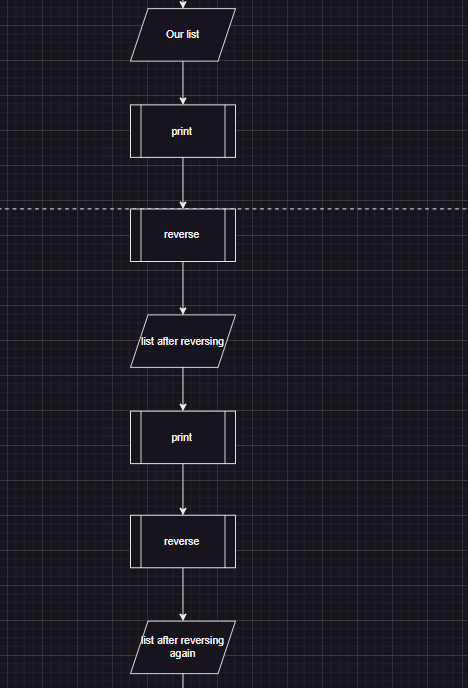
Ця програма була вже на порядок складнішою, тому я написав її за 3,5 години.

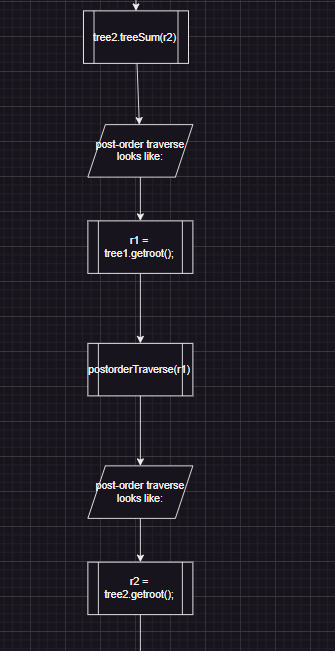
*Task 6 - Practice# programming: Class Practice Task*

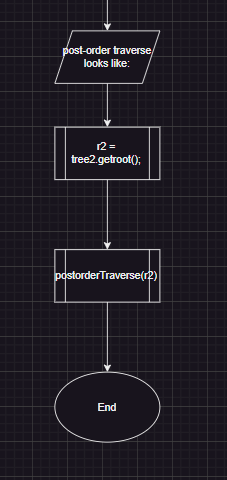
Задача полягала в опрацюванні однозв’язного списку та бінарного дерева, використовуючи різні функції. Програма:

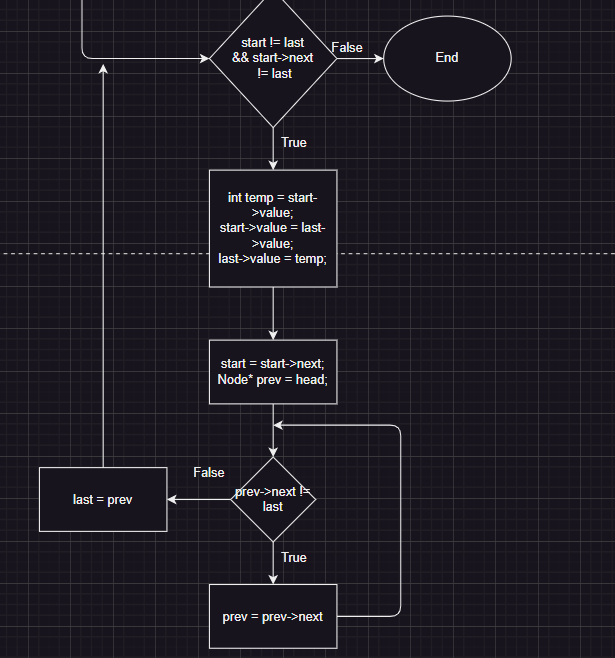
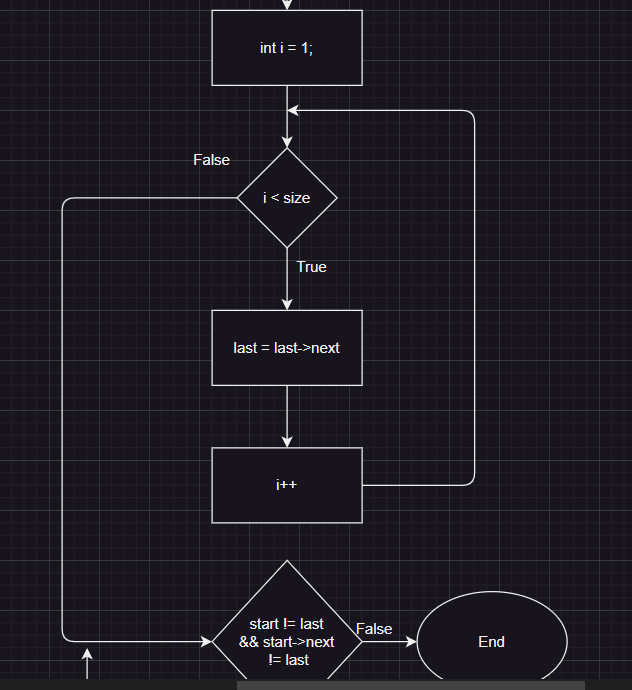
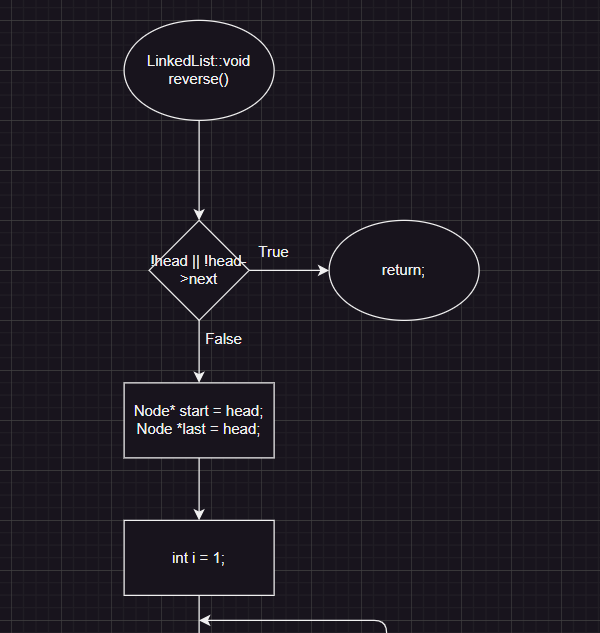
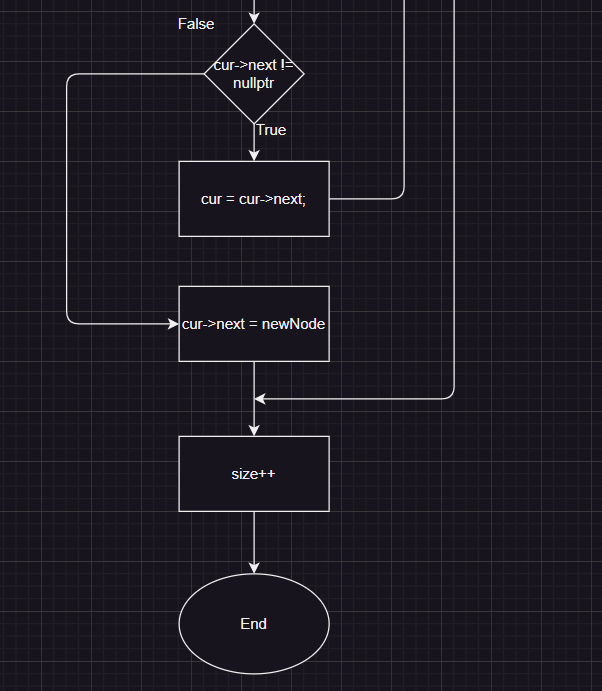
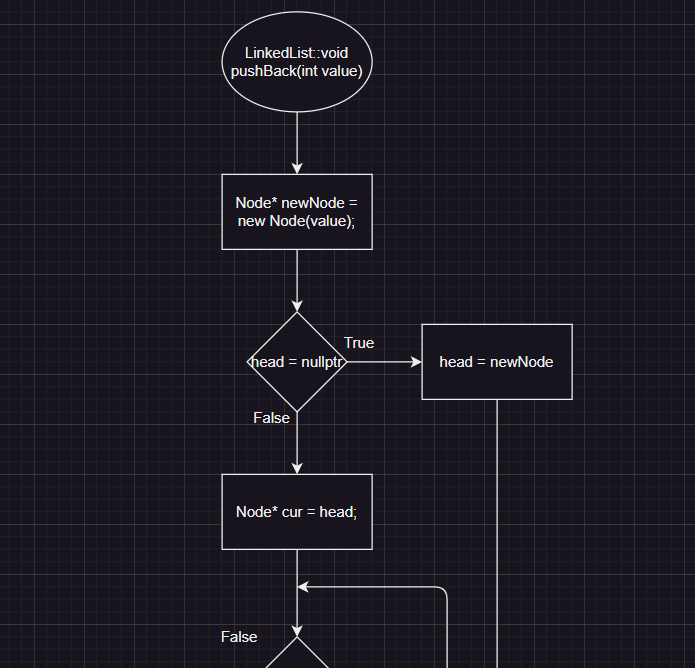
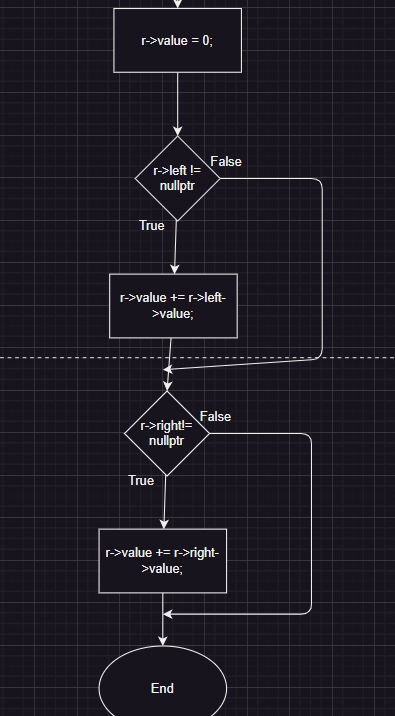
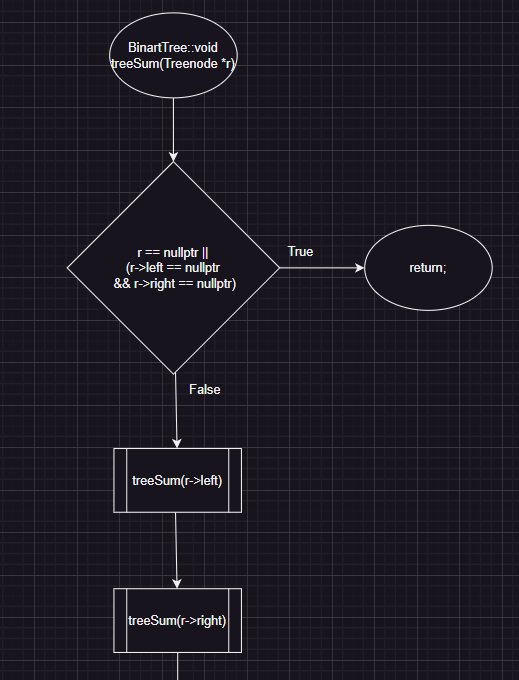
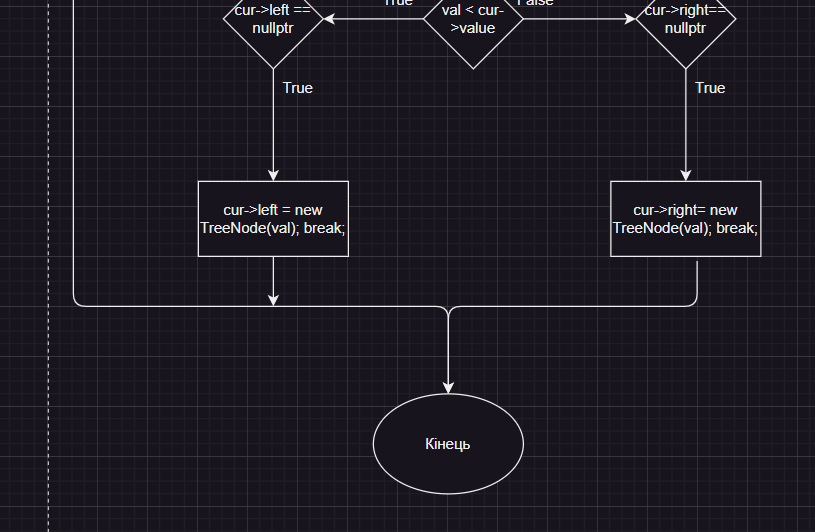
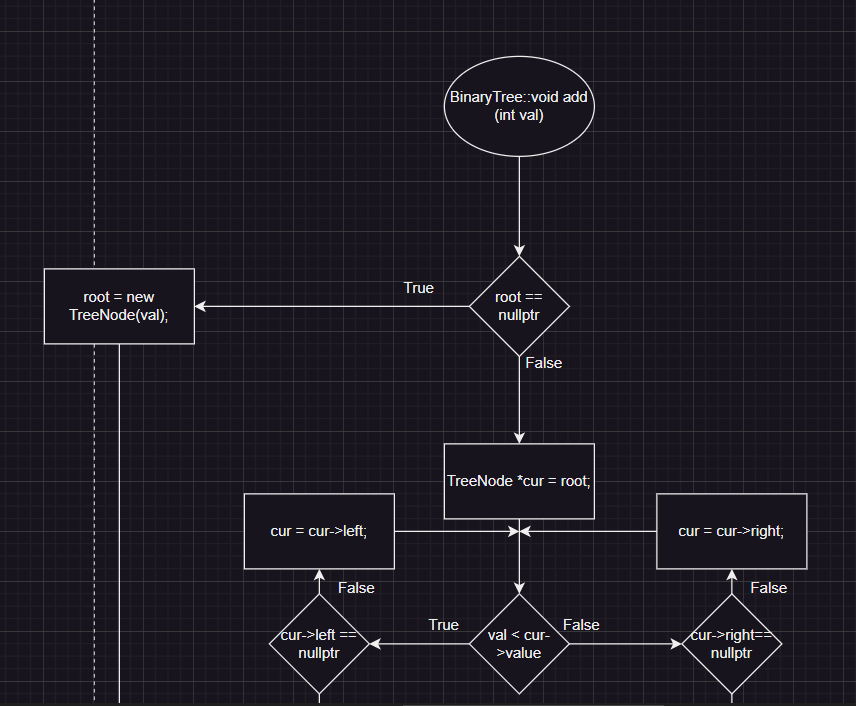
Результат:

Це найскладніша задача з даного епіку, написав я її за 4 години, блок - схема до неї виглядатиме так:









На написання цієї блок-схеми я витратив 2 години.

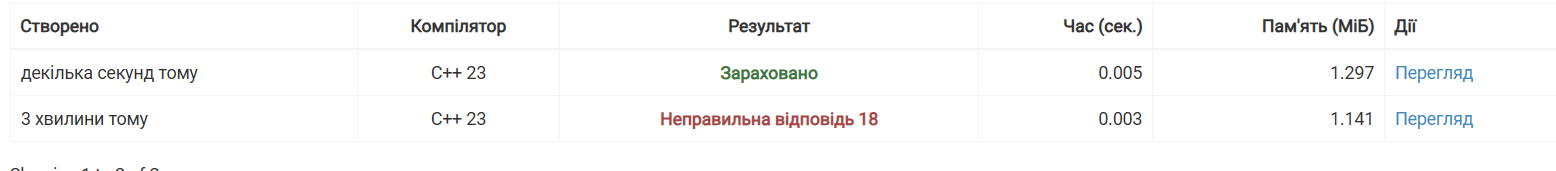
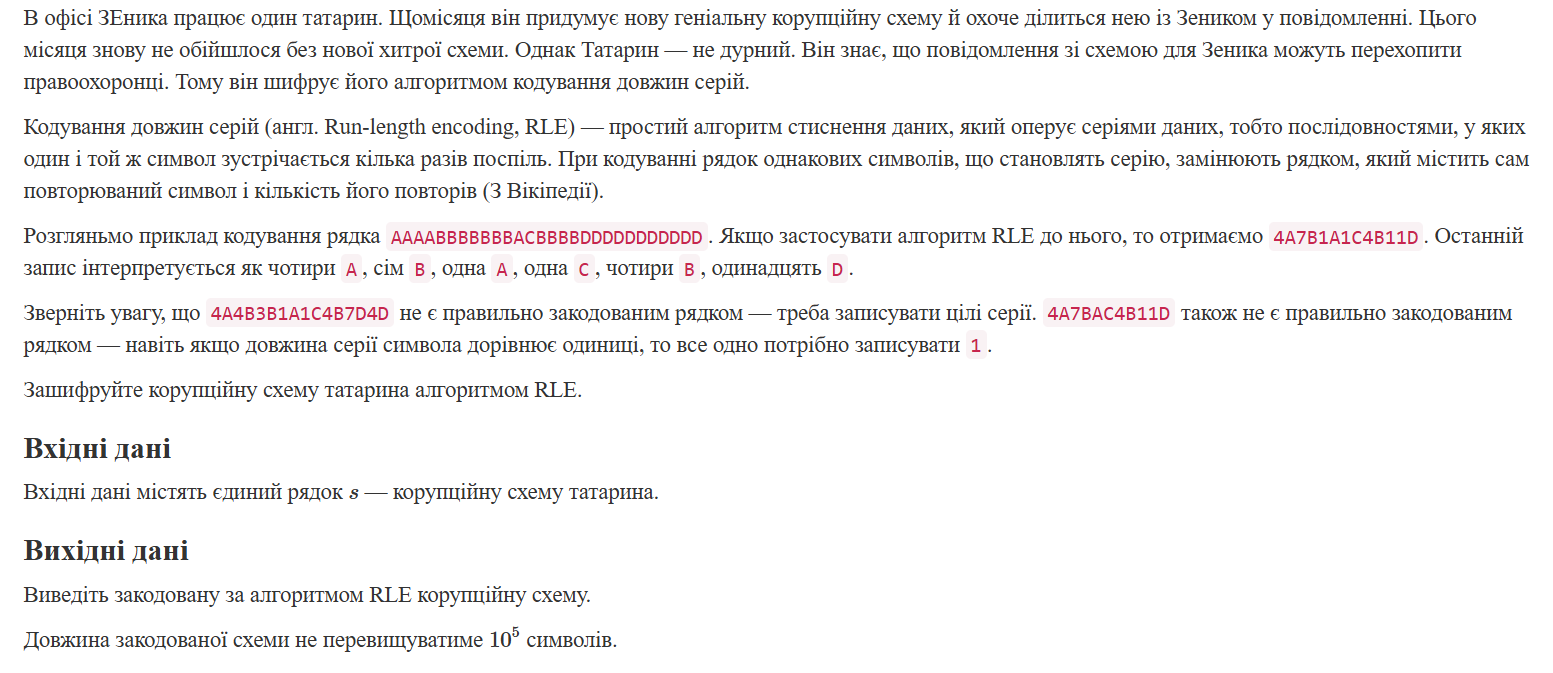
*Task 7 - Practice# programming: Self Practice Task*

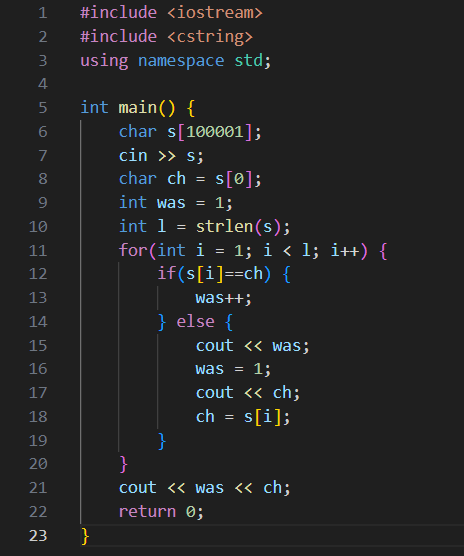
<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40829>

Шифрування корупційних схем

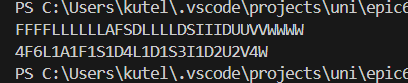
Я не зміг знайти задачі, яка б стосувалась епіка напряму, тому обрав цю цікавеньку задачу, з цікавим описом.

Умова:

 Програма:



Результат:



Загалом задачка не дуже складна, тому я її написав за 15 хв.

**Робота в команді**

Загалом, працюючи в команді, ми обговорили як реалізувати задані нам динамічні структури та допомагали одне одному з труднощами. Ось один із скріншотів наших зустрічей:



**Висновок:** Я отримав практичні навички у роботі з динамічними структурами, алгоритмами обробок динамічних структур