Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**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 Лабораторних Робіт № 5, 7-8

Практичних Робіт до блоку № 6

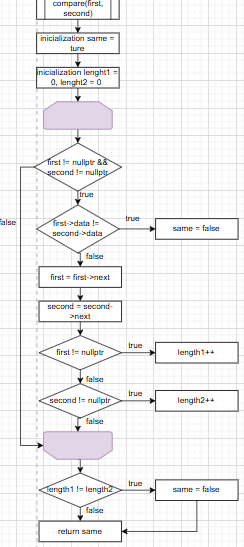
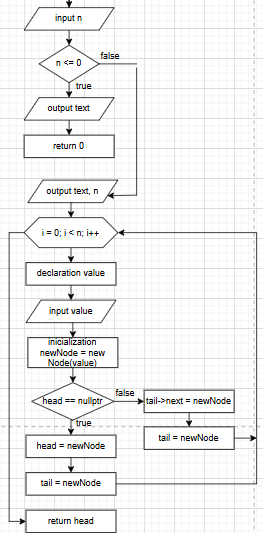
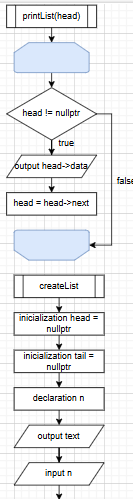
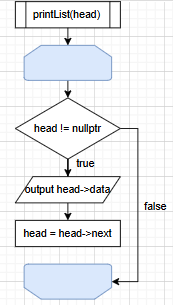
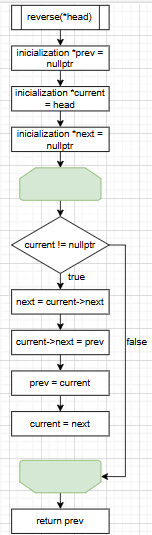
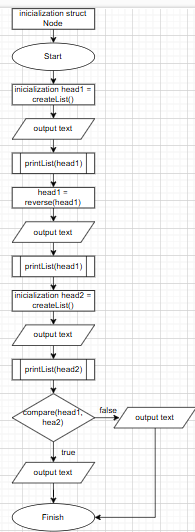
**Виконала:**

Cтудентка групи ШІ-12

Смачило Іванна

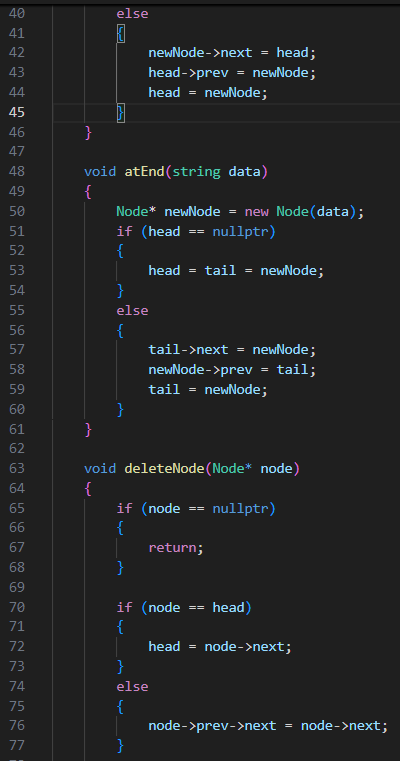
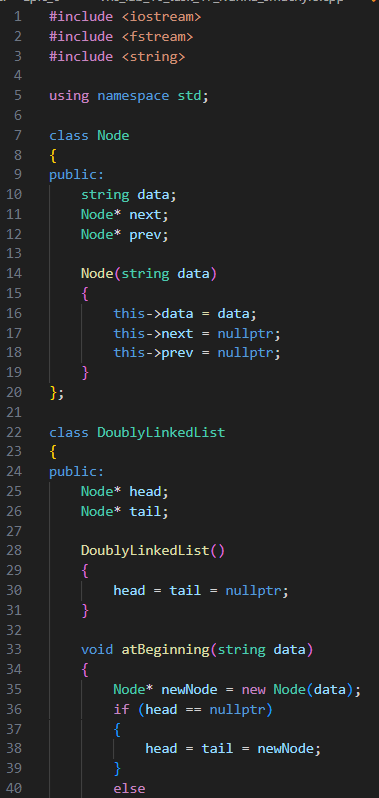
Львів – 2024

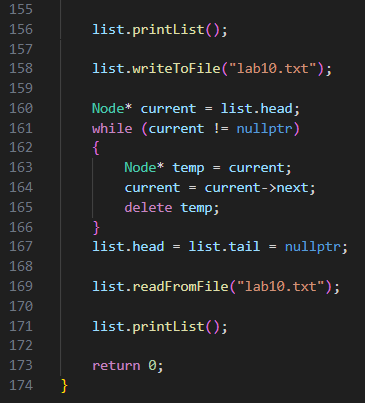
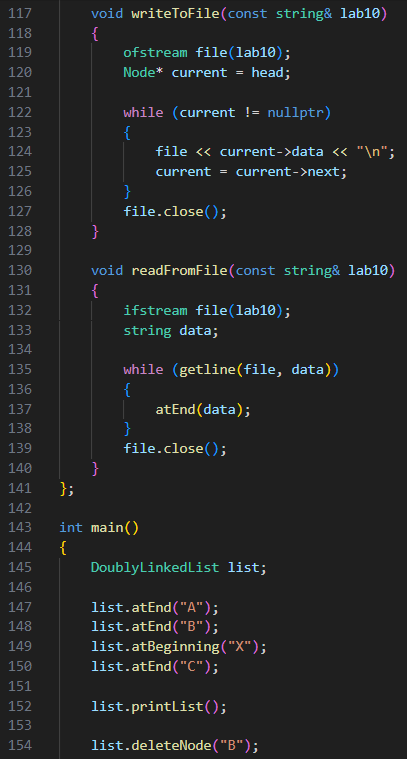
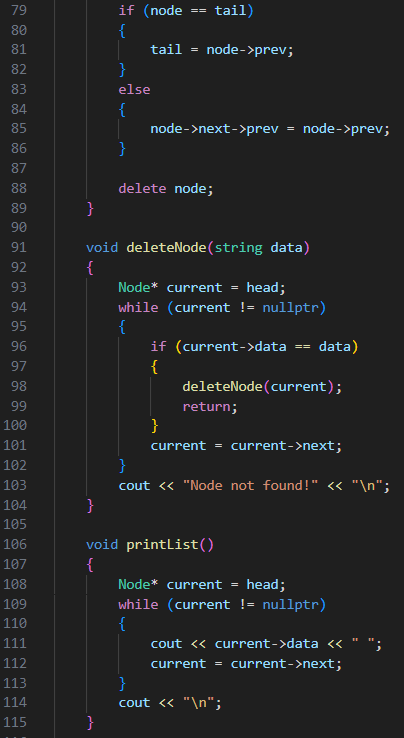
**Тема роботи**

1. Динамічні структури, види динамічних структур, їх використання, алгоритми їх обробки.
2. **Мета роботи**
3. 1. Навчитись створювати та використовувати динамічні структури, такі як: Черга, Стек, Списки, Дерево.
4. 2. Навчитись виконувати алгоритми обробки динамічний структур.
5. **Теоретичні відомості**
6. 1. Стек і купа: <https://www.bestprog.net/uk/2019/09/18/c-the-concept-of-stack-operations-on-the-stack-an-example-implementation-of-the-stack-as-a-dynamic-array-ua/#google_vignette>
7. <https://acode.com.ua/urok-111-stek-i-kupa/>
8. <https://dystosvita.org.ua/mod/page/view.php?id=888>
9. 2. Динамічні масиви: <https://acode.com.ua/urok-90-dynamichni-masyvy/>
10. 3. Черга: <https://itproger.com/ua/spravka/cpp/queue>
11. <https://www.bestprog.net/uk/2019/09/26/c-queue-general-concepts-ways-to-implement-the-queue-implementing-a-queue-as-a-dynamic-array-ua/>
12. 4. Зв’язні списки: <https://prometheus.org.ua/cs50/sections/section6.html>
13. 5. Список list: <https://codelessons.dev/ru/spisok-list-v-s-polnyj-material/>
14. 6. Circular linked list: <https://www.geeksforgeeks.org/circular-linked-list-in-cpp/>
15. 7. Бінарні дерева: <https://purecodecpp.com/uk/archives/2483#google_vignette>
16. 8. Перевантаження операторів вводу і виводу: <https://acode.com.ua/urok-141-perevantazhennya-operatoriv-vvodu-i-vyvodu/#toc-0>
17. 9. Reverse in C++: <https://www.geeksforgeeks.org/stdreverse-in-c/>
18. 10. Compare in C++: <https://www.geeksforgeeks.org/stdstringcompare-in-c/>
19. 11. C++ Program For Inserting A Node In A Linked List: <https://www.geeksforgeeks.org/cpp-program-for-inserting-a-node-in-a-linked-list/>
20. **Виконання роботи**
21. **Task 2 - Requirements management (understand tasks) and design activities (draw flow diagram for Class Practice Task)**
22. ****

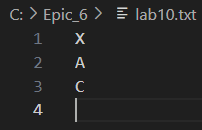
**Task 3 - Lab# programming: VNS Lab 10**

Завдання: 17.Записи в лінійному списку містять ключове поле типу \*char (рядок символів). Сформувати двонаправлений список. Знищити елемент із заданим номером. Додати К елементів у початок списку.

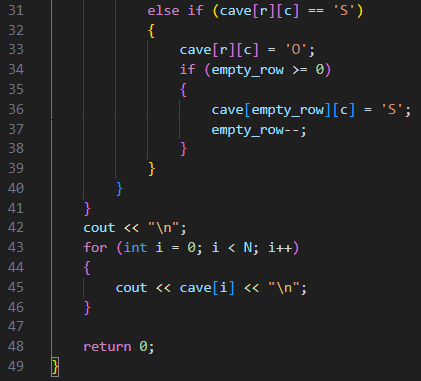
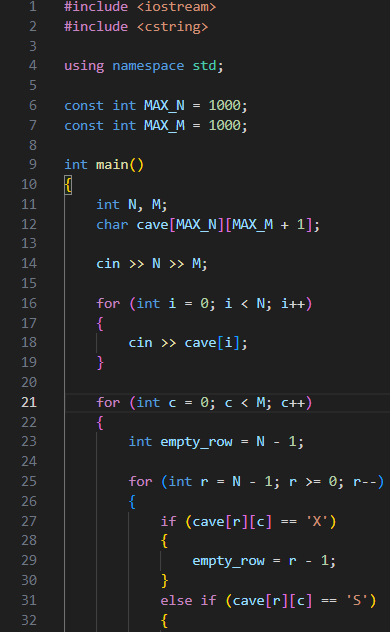
****

****

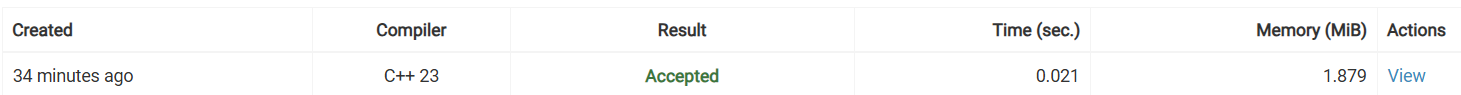
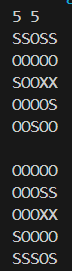
**Результат:**

****

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

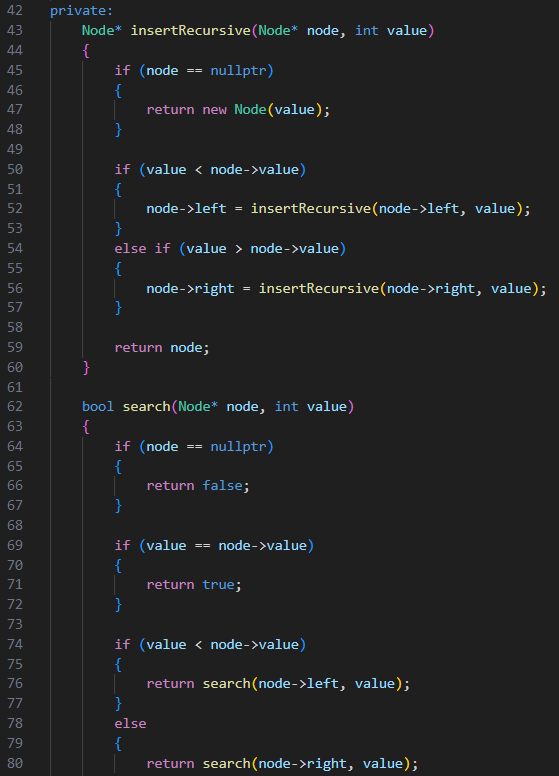
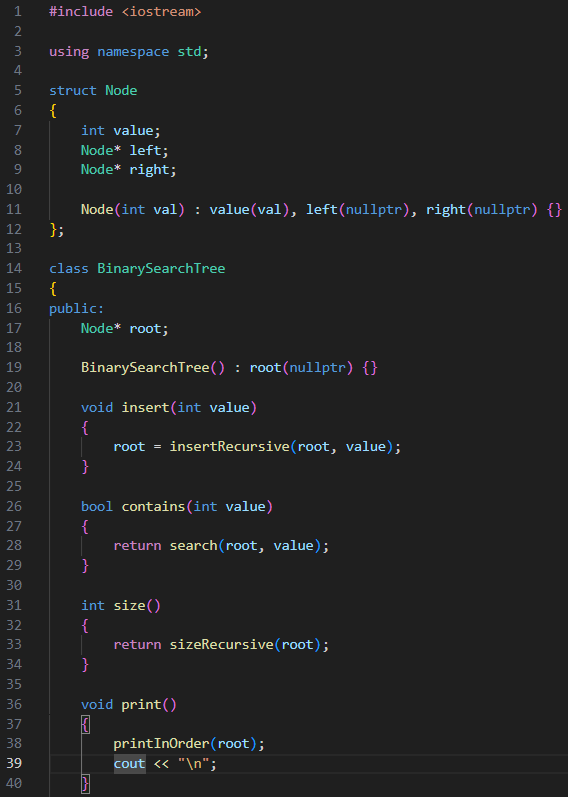
****

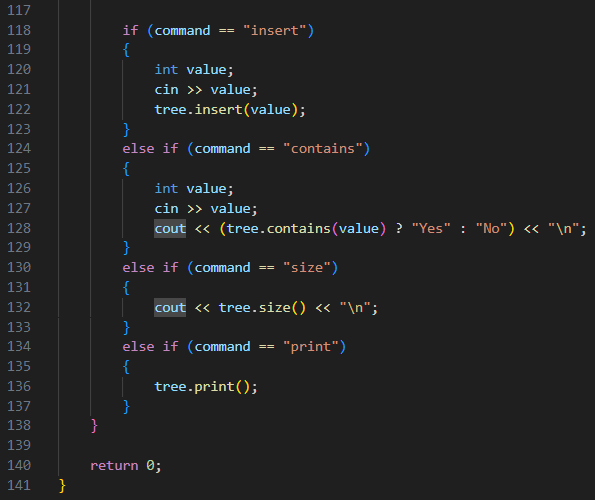
**Результат:**

****

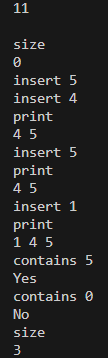
**Task 5 - Lab# programming: Algotester Lab 7-8**

**Variant 3:**

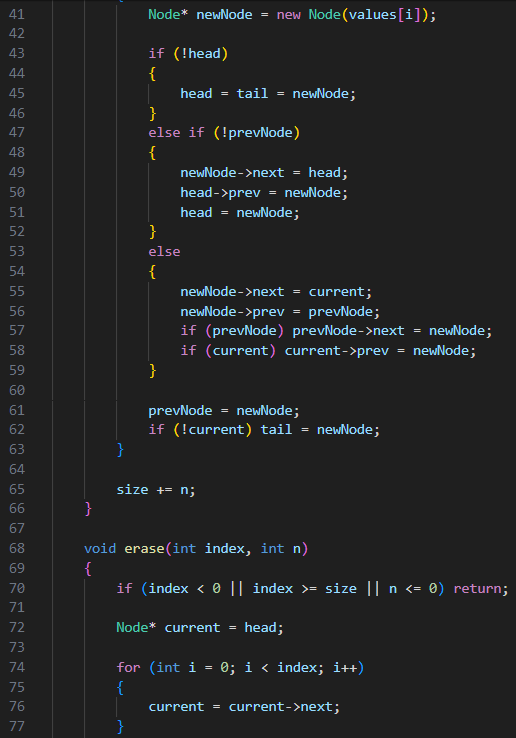
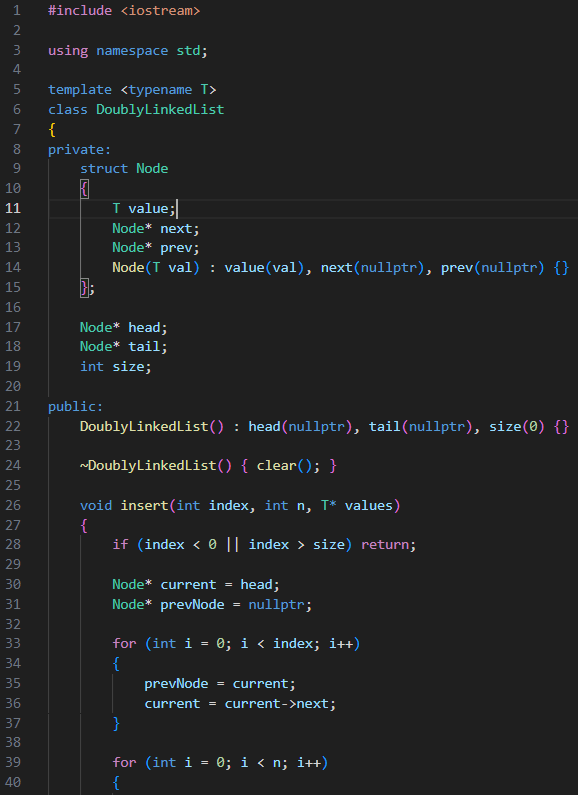
****

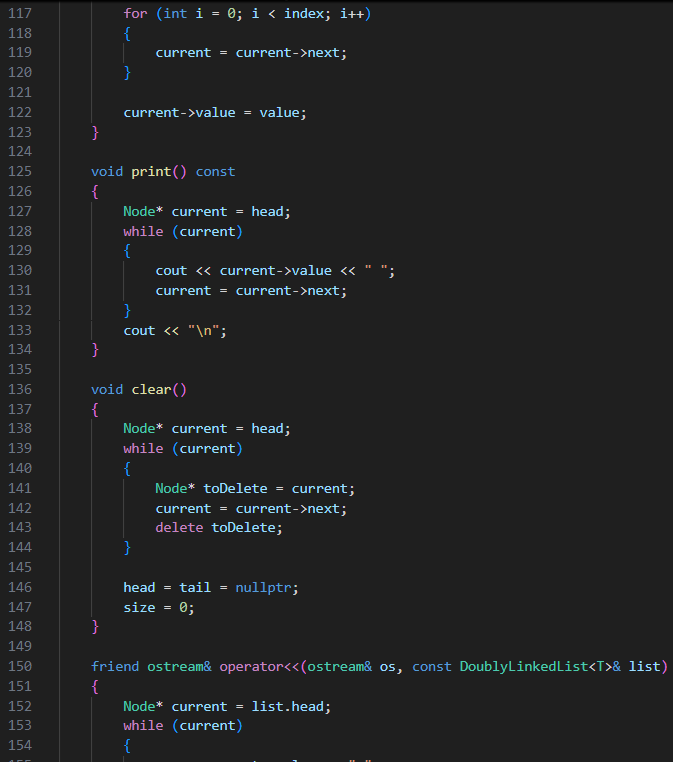
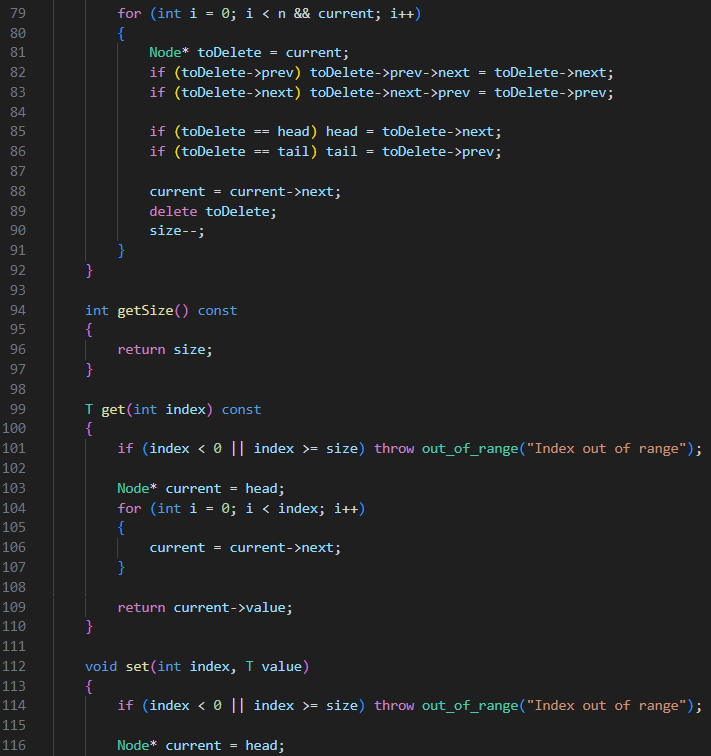
****

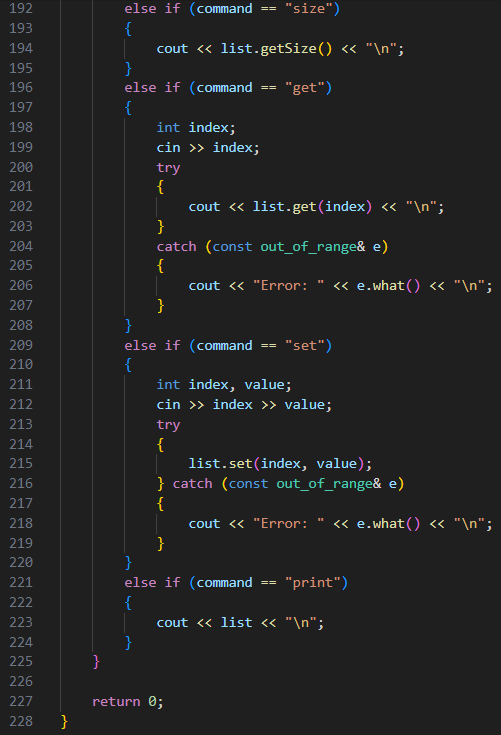
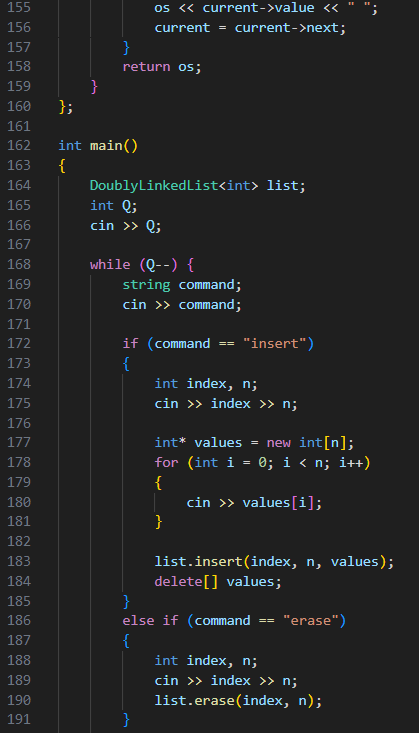
**Результат:**

****

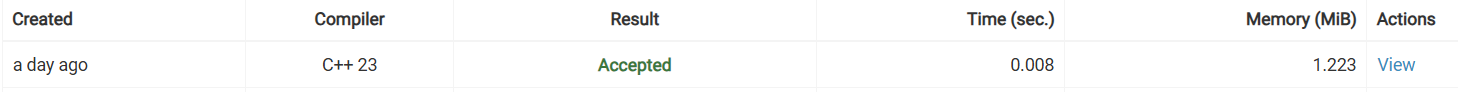
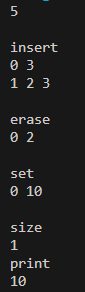
**Variant 1:**

****

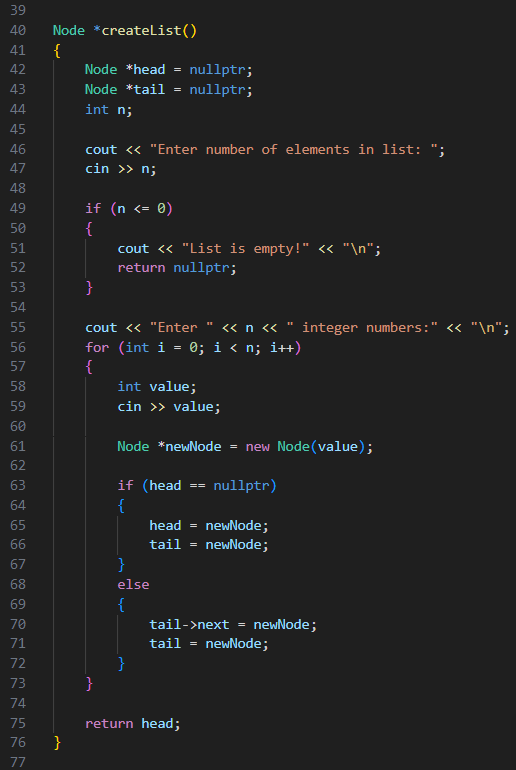
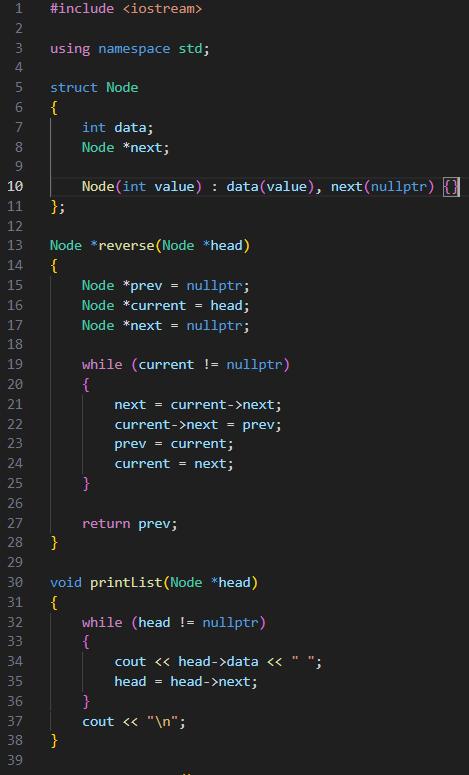
****

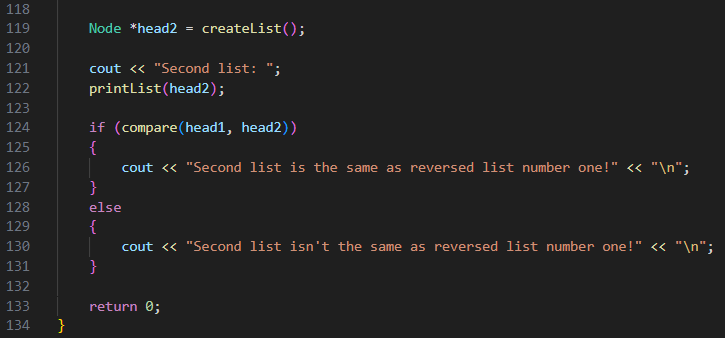
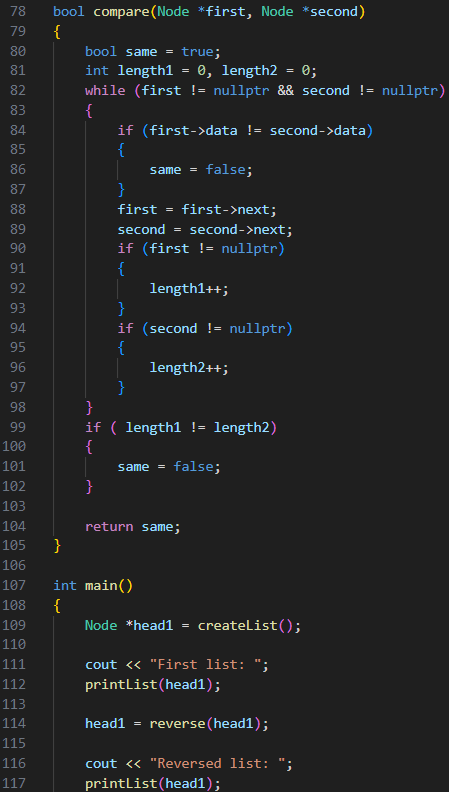
****

**Результат:**

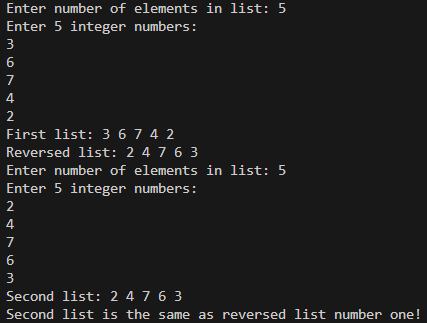
****

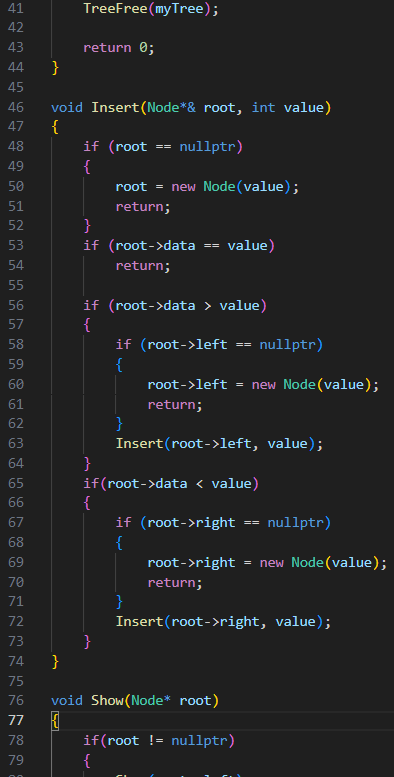
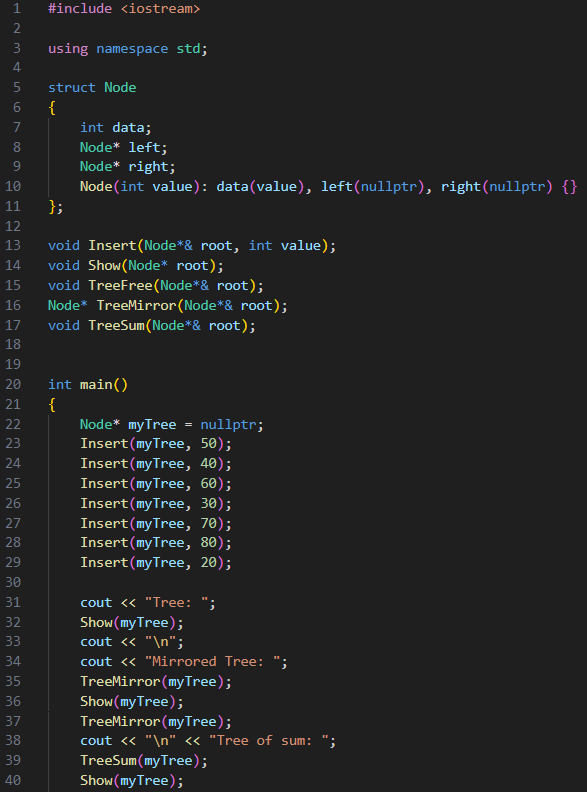
**Task 6 - Practice# programming: Class Practice Task**

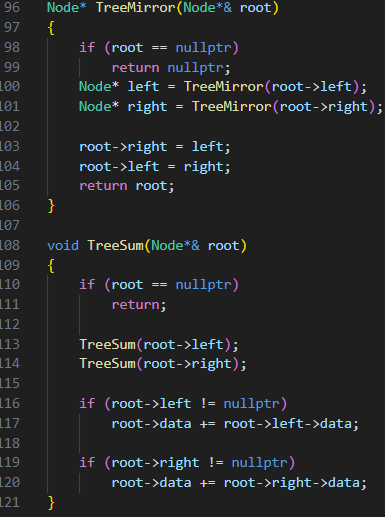
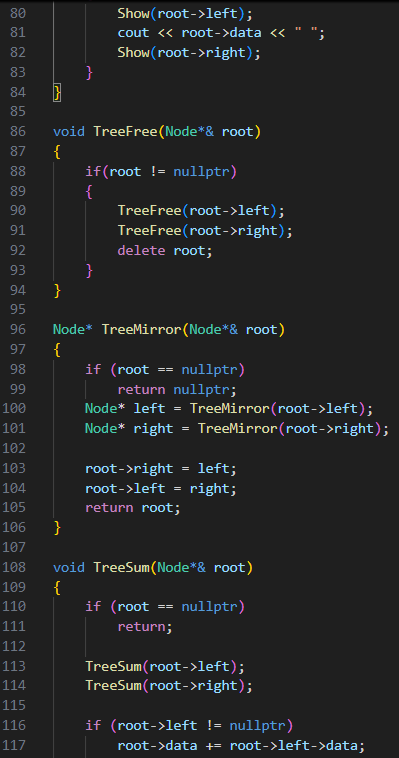
****

****

**Результат:**

****

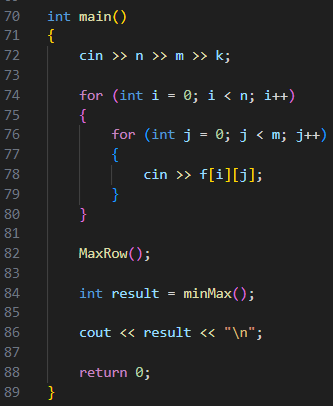
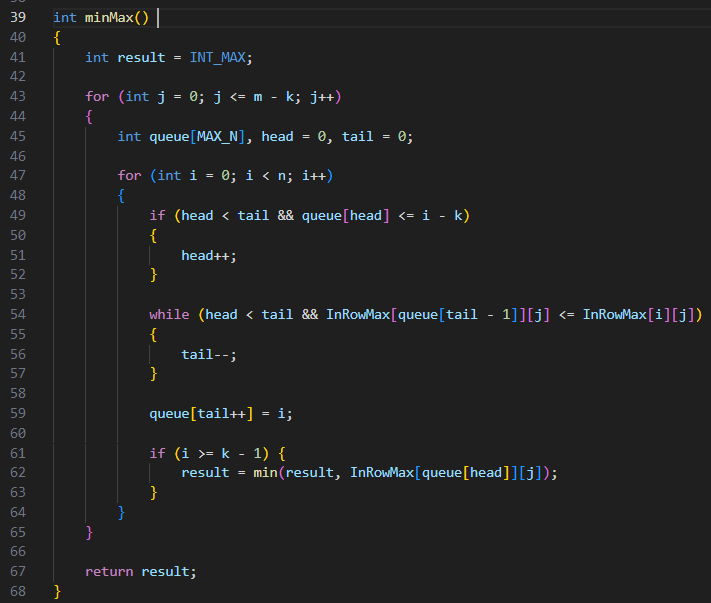
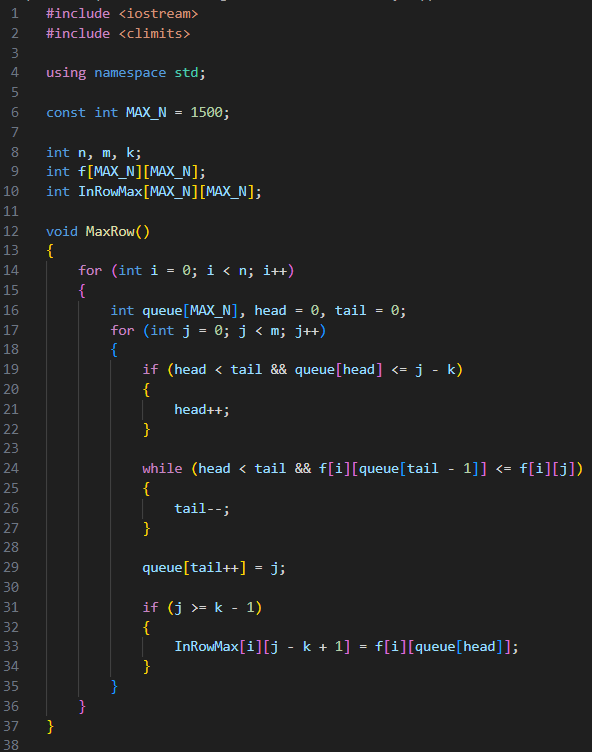
****

****

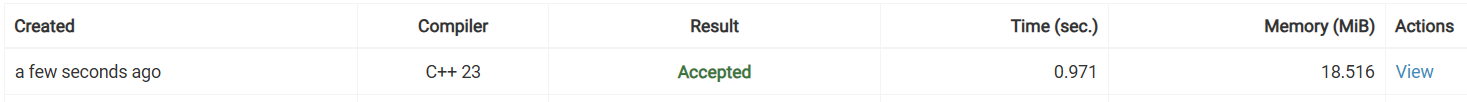
**Результат:**

****

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

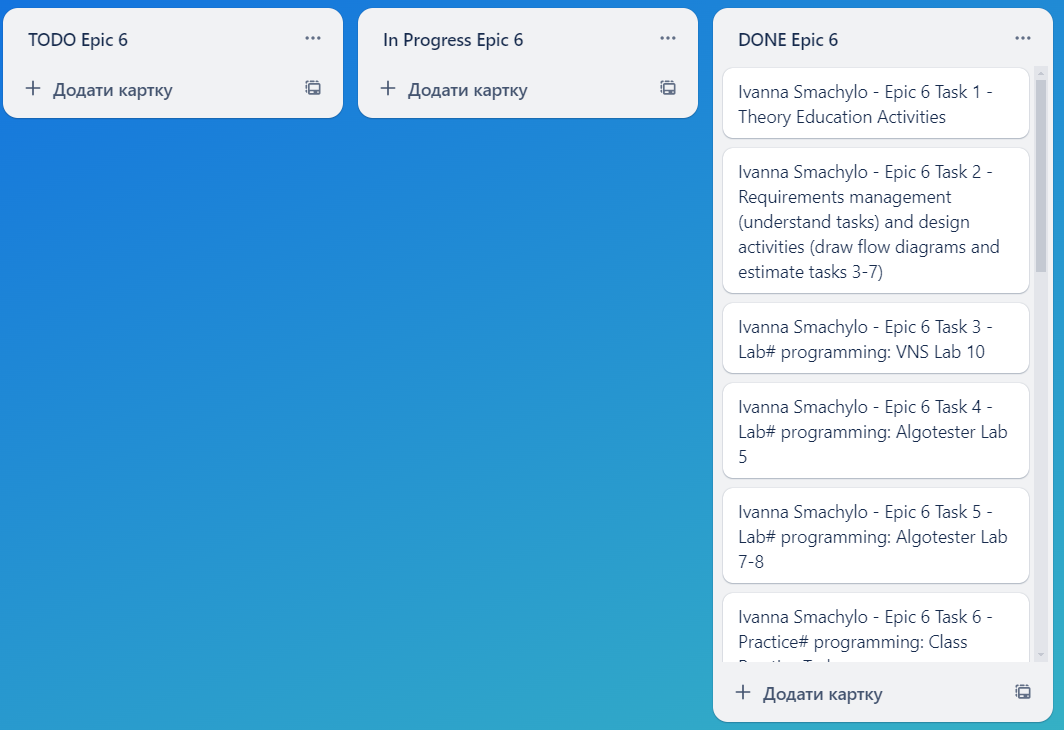


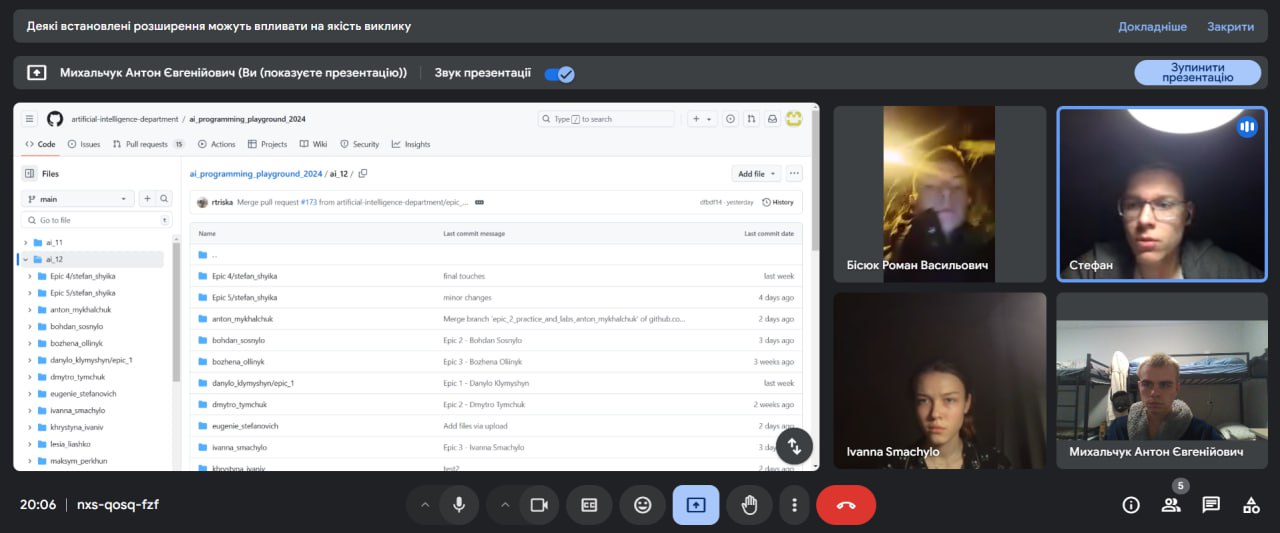
**Результат:**



**Зустрічі з командою:**

Зустрічались двічі для обговорення задач, поставлених в шостому епіку. Створили нову дошку в Trello й бачили прогрес одне одного:





**Висновок:** в ході роботи над даним епіком я навчилась використовувати на практиці нові знання, такі як стек і купа, динамічні масиви, черга, зв’язні списки, список list, бінарні дерева, перевантаження операторів вводу і виводу, reverse, compare in C++.