**Міністерство освіти і науки України**

**Національний університет "Львівська Політехніка"**

**Кафедра систем штучного інтелекту**

**Епік №6**

з дисципліни

«Основи програмування»

**Виконав:**

студент групи ШІ-11

Гнатюк Ярослав

Львів – 2024 р.

**Епік №6**

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

**Мета роботи:** Дослідити основи динамічних структур даних (черга, стек, списки, дерево) та розробити алгоритми їхньої ефективної обробки для розв'язання задач із зберігання, пошуку, сортування та маніпулювання даними.

**Теоретичні відомості:**

* Однозв’язний список: <https://www.youtube.com/watch?v=-25REjF_atI&t=53s&pp=ygUm0LHQu9C-0LPQsNC9INC30LLRj9C30L3RliDRgdC_0LjRgdC60Lg%3D>
* Бінарні дерева: <https://www.youtube.com/watch?v=qBFzNW0ALxQ&pp=ygUs0LHRltC90LDRgNC90ZYg0LTQtdGA0LXQstCwIGMrKyDQsdC70L7Qs9Cw0L0%3D>
* Стек (LIFO): <https://www.w3schools.com/cpp/cpp_stacks.asp>
* Черга (FIFO): <https://www.w3schools.com/cpp/cpp_queues.asp>

**Виконання роботи**

**Частина 1**

**Завдання №1**

**Назва:** VNS Lab 10 Variant 5

**Опис:** 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.

Записи в лінійному списку містять ключове поле типу int. Сформувати

однонаправлений список. Знищити з нього К елементів, починаючи із

заданого номера, додати К елементів, починаючи із заданого номера.

**Завдання №2**

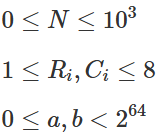
**Назва:** Algotester Lab 5 Variant 1

**Опис:** У світі Атод сестри Ліна і Рілай люблять грати у гру. У них є дошка із 8-ми рядків і 8-ми стовпців. На перетині i -го рядка і j -го стовпця лежить магічна куля, яка може світитись магічним світлом (тобто у них є 64 кулі). На початку гри деякі кулі світяться, а деякі ні... Далі вони обирають N куль і для кожної читають магічне заклинання, після чого всі кулі, які лежать на перетині стовпця і рядка обраної кулі змінюють свій стан (ті що світяться - гаснуть, ті, що не світяться - загораються). Також вони вирішили трохи Вам допомогти і придумали спосіб як записати стан дошки одним числом a із 8-ми байт, а саме (див. Примітки): Молодший байт задає перший рядок матриці; Молодший біт задає перший стовпець рядку;

Значення біту каже світиться куля чи ні (0 - ні, 1 - так);

Тепер їх цікавить яким буде стан дошки після виконання N заклинань і вони дуже просять Вас їм допомогти.

**Вимоги:**

****

**Завдання №3**

**Назва:** Algotester Lab 78 Variant 2

**Опис:** Ваше завдання - власноруч реалізувати структуру даних "Динамічний масив". Ви отримаєте Q запитів, кожен запит буде починатися зі слова-ідентифікатора, після якого йдуть його аргументи.

Вам будуть поступати запити такого типу:

- Вставка: ідентифікатор - insert. Ви отримуєте ціле число index елемента, на місце якого робити вставку. Після цього в наступному рядку рядку написане число N - розмір масиву; який треба вставити.

У третьому рядку N цілих чисел - масив, який треба вставити на позицію indeх.

- Видалення: ідентифікатор - erase. Ви отримуєте 2 цілих числа - index - індекс елемента, з якого почати видалення та N - кількість елементів, яку треба видалити.

- Визначення розміру: ідентифікатор - size. Ви не отримуєте аргументів. Ви виводите кількість елементів у динамічному масиві.

- Визначення кількості зарезервованої пам'яті: Ідентифікатор - capacity Ви не отримуєте аргументів. Ви виводите кількість зарезервованої пам'яті у динамічному масиві. Ваша реалізація динамічного масиву має мати фактор росту рівний 2.

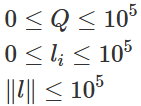
- Отримання значення і-го елементу: Ідентифікатор - get

Ви отримуєте ціле число - index, індекс елемента. Ви виводите значення елемента за індексом.

- Модифікація значення і-го елементу: Ідентифікатор - set. Ви отримуєте 2 цілих числа - індекс елемента, який треба змінити, та його нове значення.

- Вивід динамічного масиву на екран: Ідентифікатор - print. Ви не отримуєте аргументів. Ви виводите усі елементи динамічного масиву через пробіл.

**Вимоги:**

****

**Завдання №4**

**Назва:** Algotester Lab 78 Variant 3

**Опис:** Ваше завдання - власноруч реалізувати структуру даних "Двійкове дерево пошуку". Ви отримаєте Q запитів, кожен запит буде починатися зі слова-ідентифікатора, після якого йдуть його параметри. Вам будуть поступати запити такого типу:

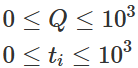
Вставка: Ідентифікатор - insert Ви отримуєте ціле число value - число, яке треба вставити в дерево.

Пошук: Ідентифікатор - contains Ви отримуєте ціле число value - число, наявність якого у дереві необхідно перевірити. Якщо value наявне в дереві - ви виводите Yes , у іншому випадку No.

Визначення розміру: Ідентифікатор - size Ви не отримуєте аргументів. Ви виводите кількість елементів у дереві.

Вивід дерева на екран Ідентифікатор - print Ви не отримуєте аргументів. Ви виводите усі елементи дерева через пробіл. Реалізувати використовуючи перегрузку оператора <<

**Вимоги:**

****

**Завдання №5**

**Назва:** Practice Work Task 1

**Опис:**

1. ***Реалізувати метод реверсу списку:***

*Умови задачі:*

- використовувати цілочисельні значення в списку;

- реалізувати метод реверсу;

- реалізувати допоміжний метод виведення вхідного і обернутого списків;

## ***Порівняння списків***

*Умови задачі:*

- використовувати цілочисельні значення в списку;

- реалізувати функцію, яка ітеративно проходиться по обох списках і порівнює дані в кожному вузлі;

- якщо виявлено невідповідність даних або якщо довжина списків різна (один список закінчується раніше іншого), функція повертає ***false***.

## ***Додавання великих чисел***

*Умови задачі:*

- використовувати цифри від 0 до 9 для значень у списку;

- реалізувати функцію, яка обчислює суму двох чисел, які збережено в списку; молодший розряд числа записано в голові списка (напр. 379 ⟹ 9→7→3);

- функція повертає новий список, передані в функцію списки не модифікуються.

**Завдання №6**

**Назва:** Practice Work Task 1

**Опис:**

## **Віддзеркалення дерева**

*Умови задачі:*

- використовувати цілі числа для значень у вузлах дерева

- реалізувати функцію, що проходить по всіх вузлах дерева і міняє місцями праву і ліву вітки дерева

- функція повертає нове дерево, передане в функцію дерево не модифікується

## **Записати кожному батьківському вузлу суму підвузлів**

*Умови задачі:*

- використовувати цілочисельні значення у вузлах дерева;

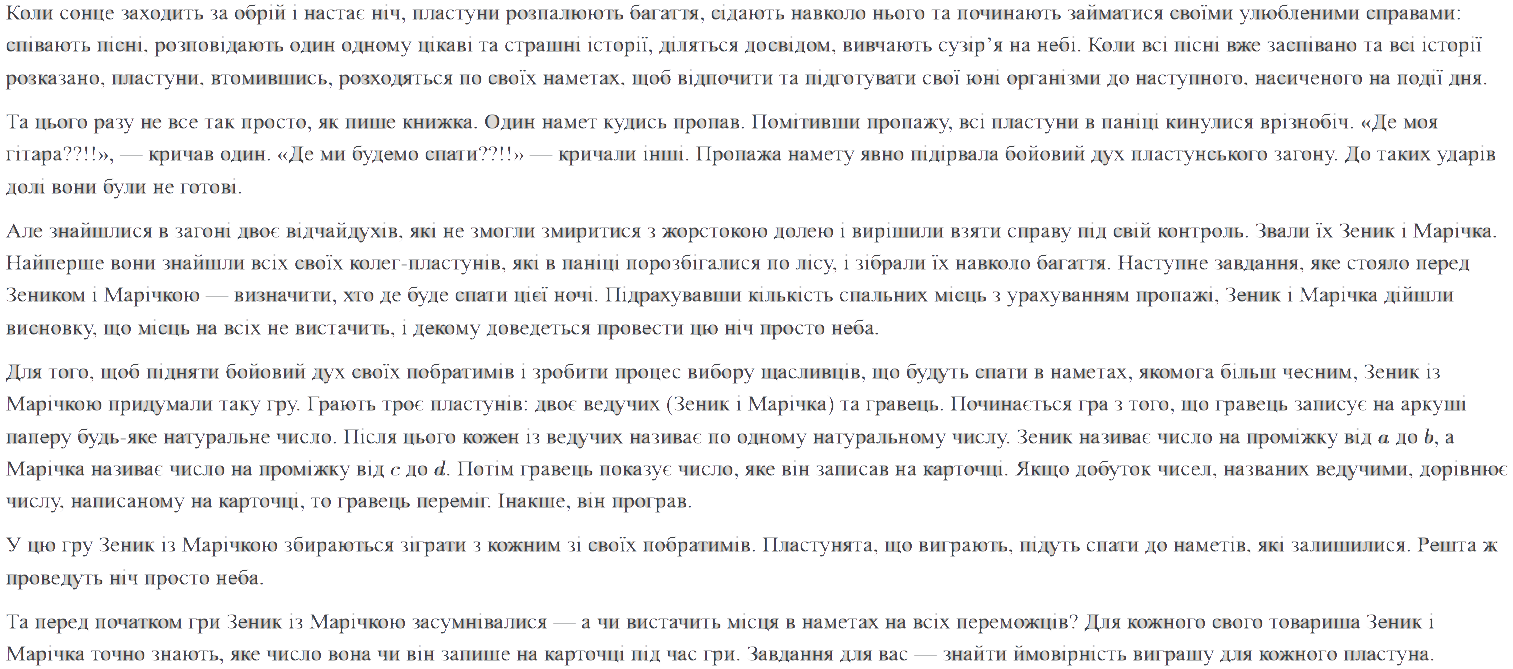
- реалізувати функцію, яка ітеративно проходить по бінарному дереві і записує у батьківський вузол суму значень підвузлів

- вузол-листок не змінює значення

- значення змінюються від листків до кореня дерева

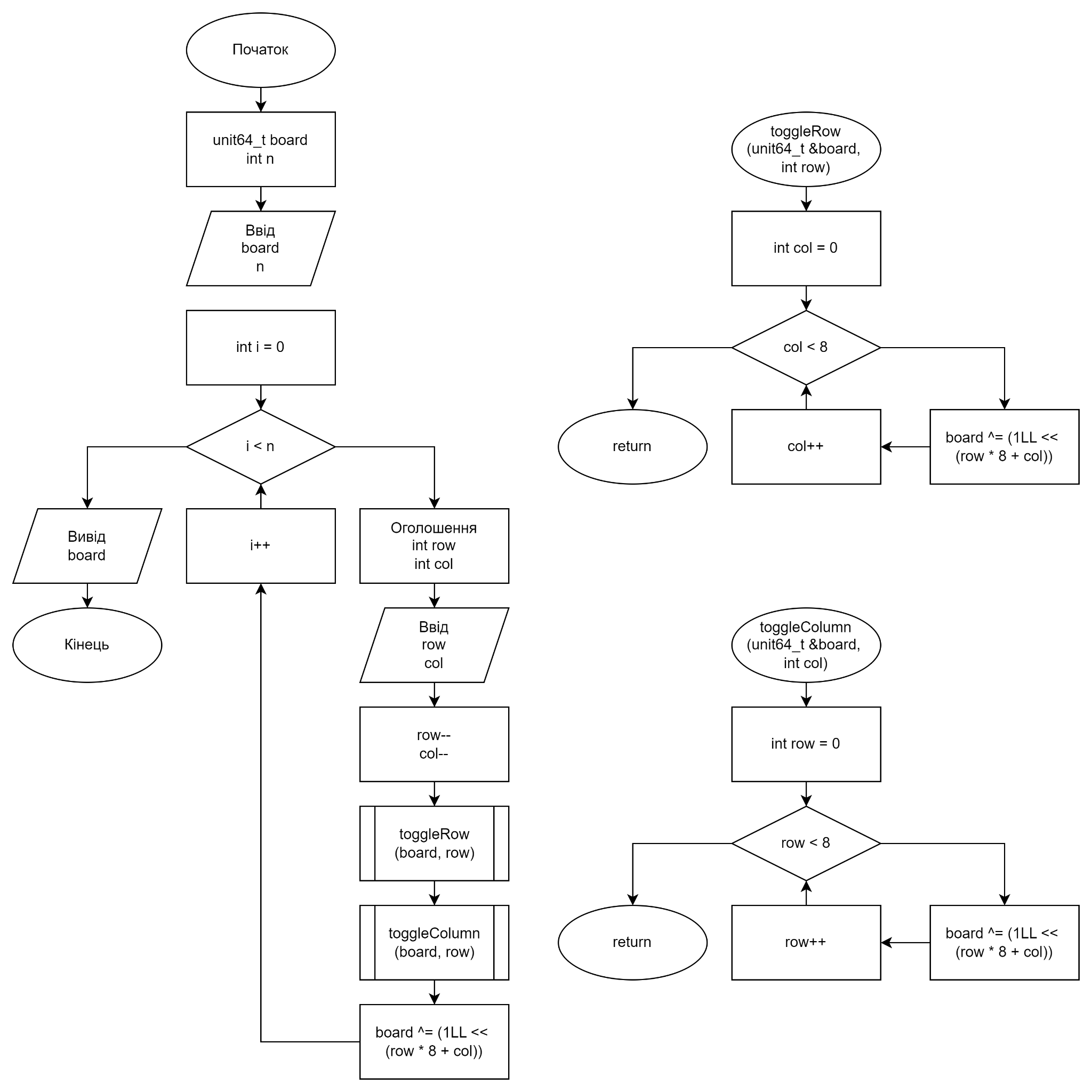
**Завдання №7**

**Назва:** Self Practice Work (Зрада)

**Опис:   
**

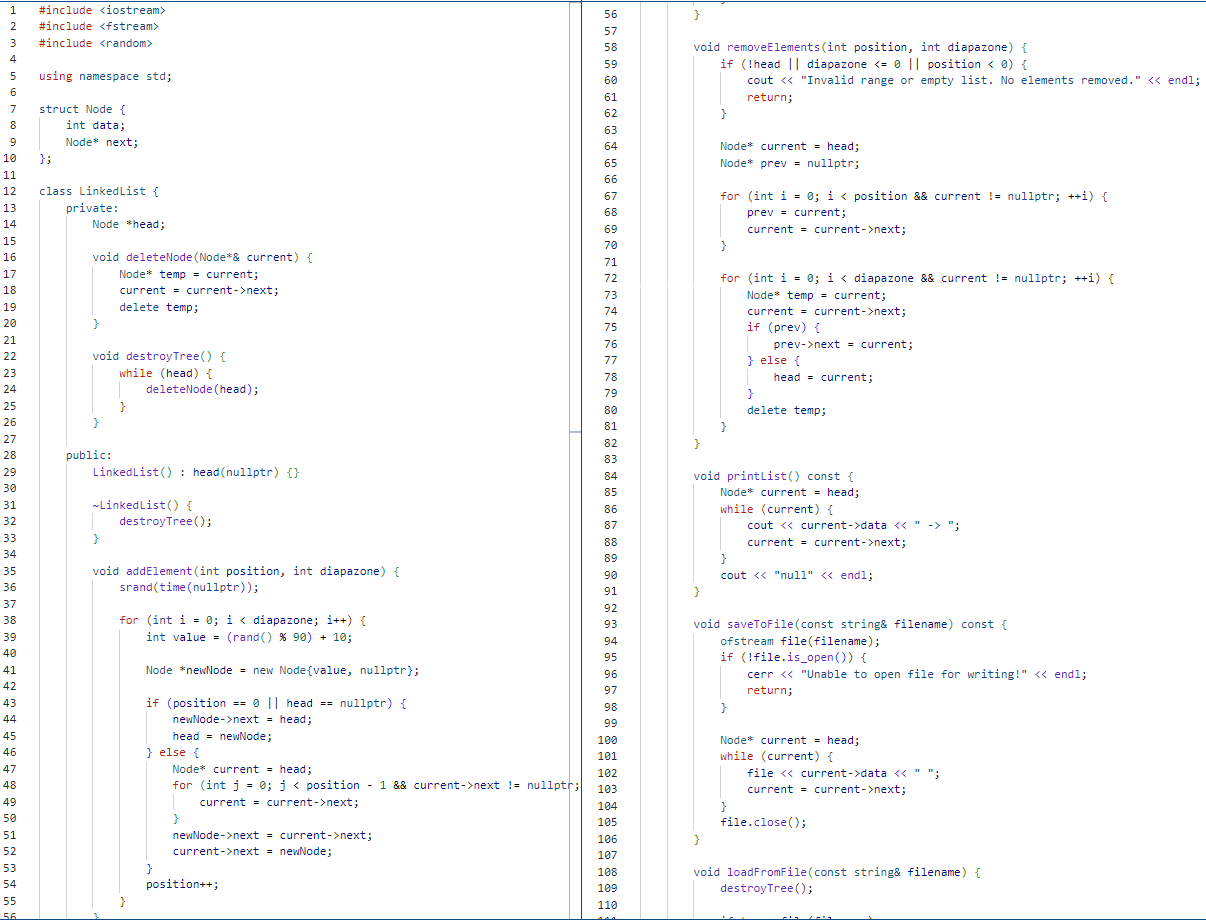
**Частина 2**

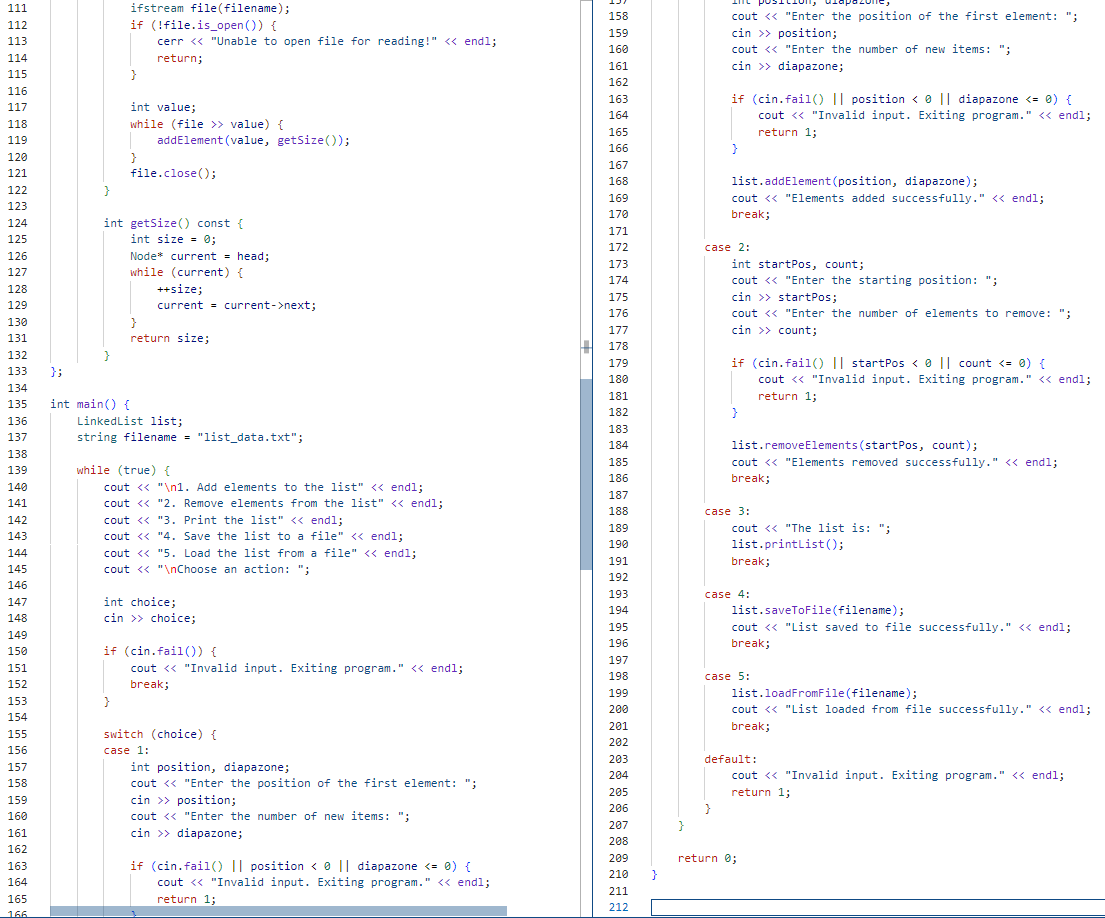
Завдання №2



**Частина 3**

**Завдання №1**

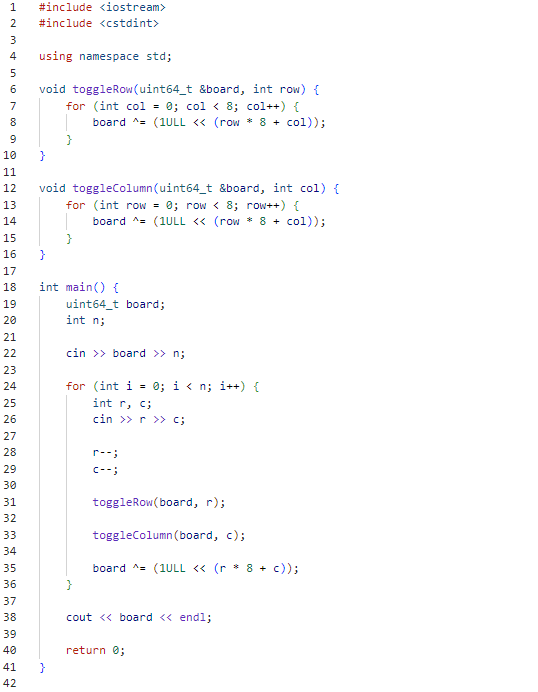
****

****

**Орієнтовний час виконання: 2 год**

**Реальний час виконання: 2.5 год**

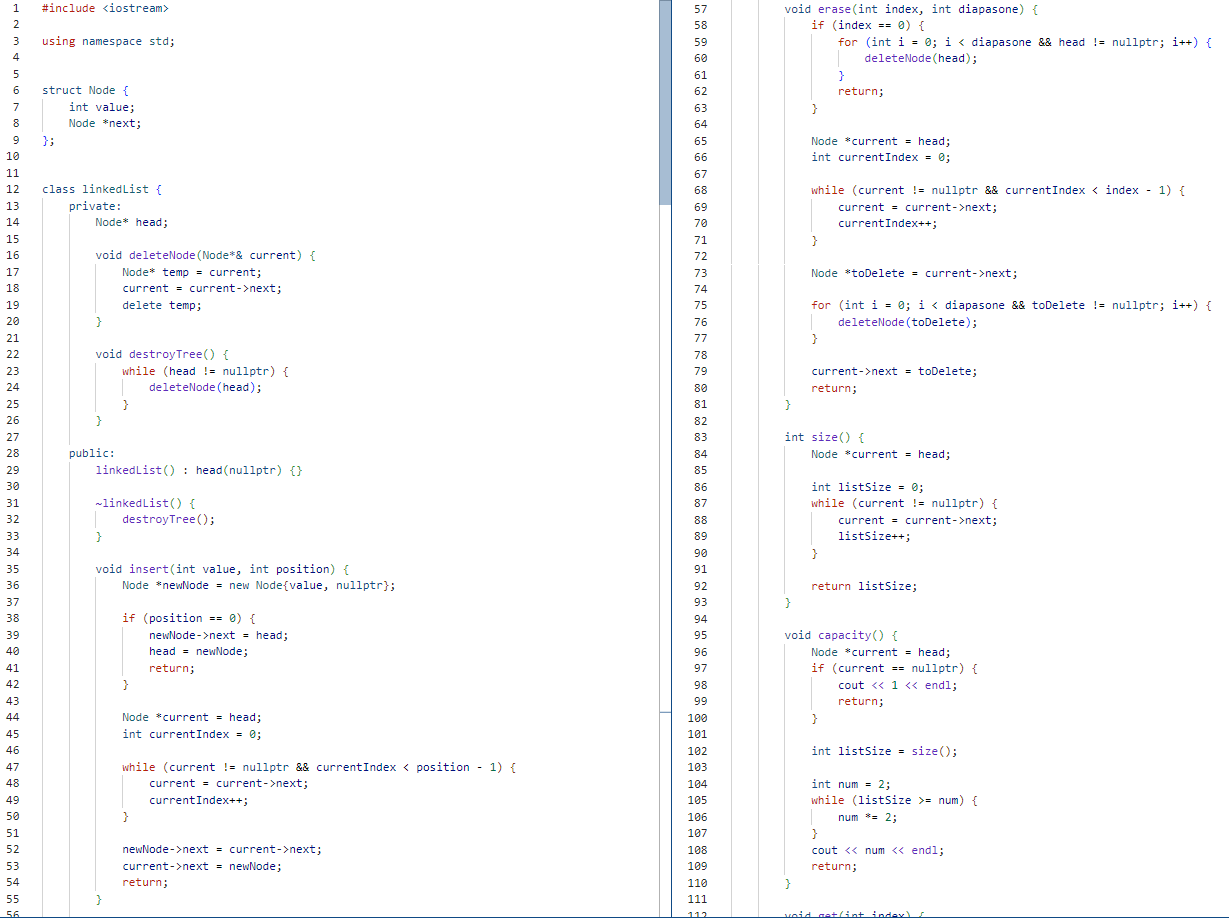
**Завдання №2**

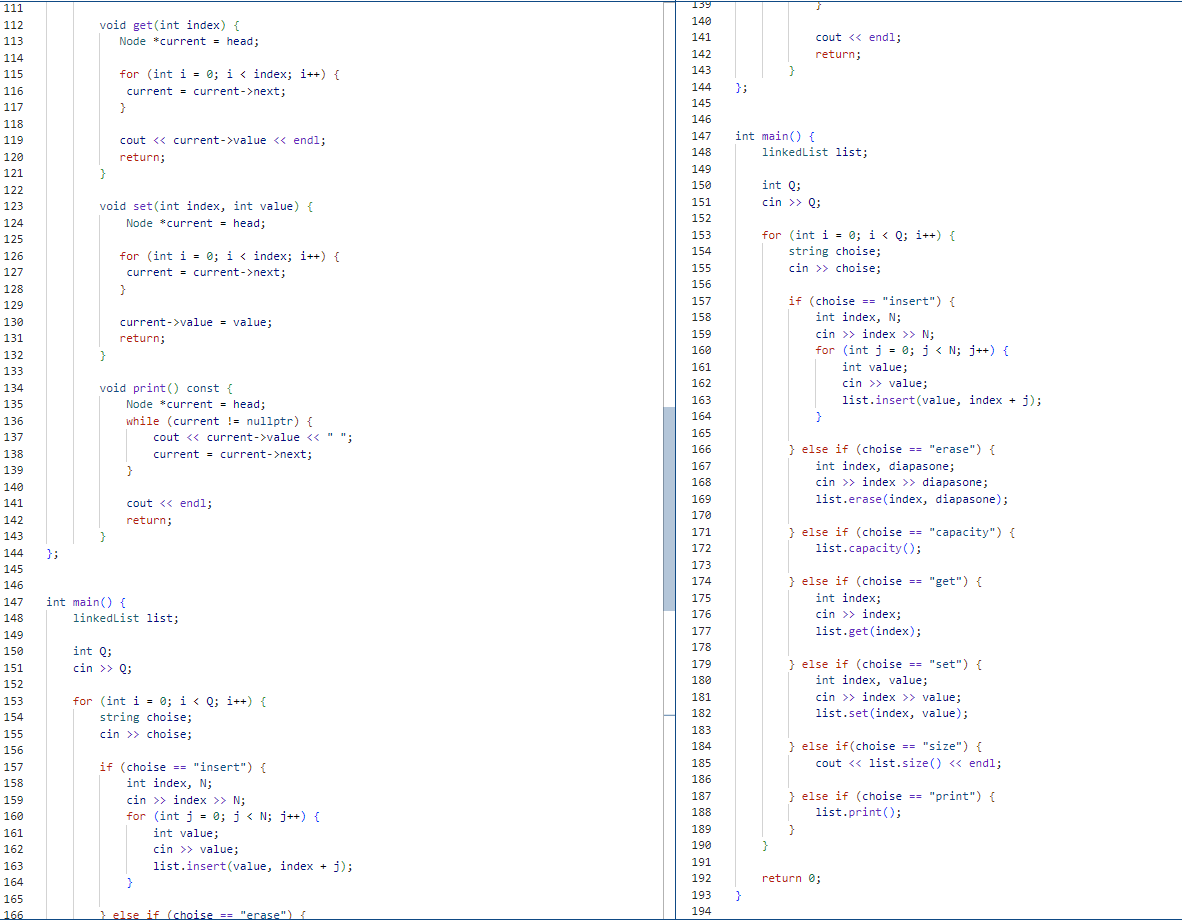
****

**Орієнтовний час виконання: 3 год**

**Реальний час виконання: 3.5 год**

**Завдання №3**

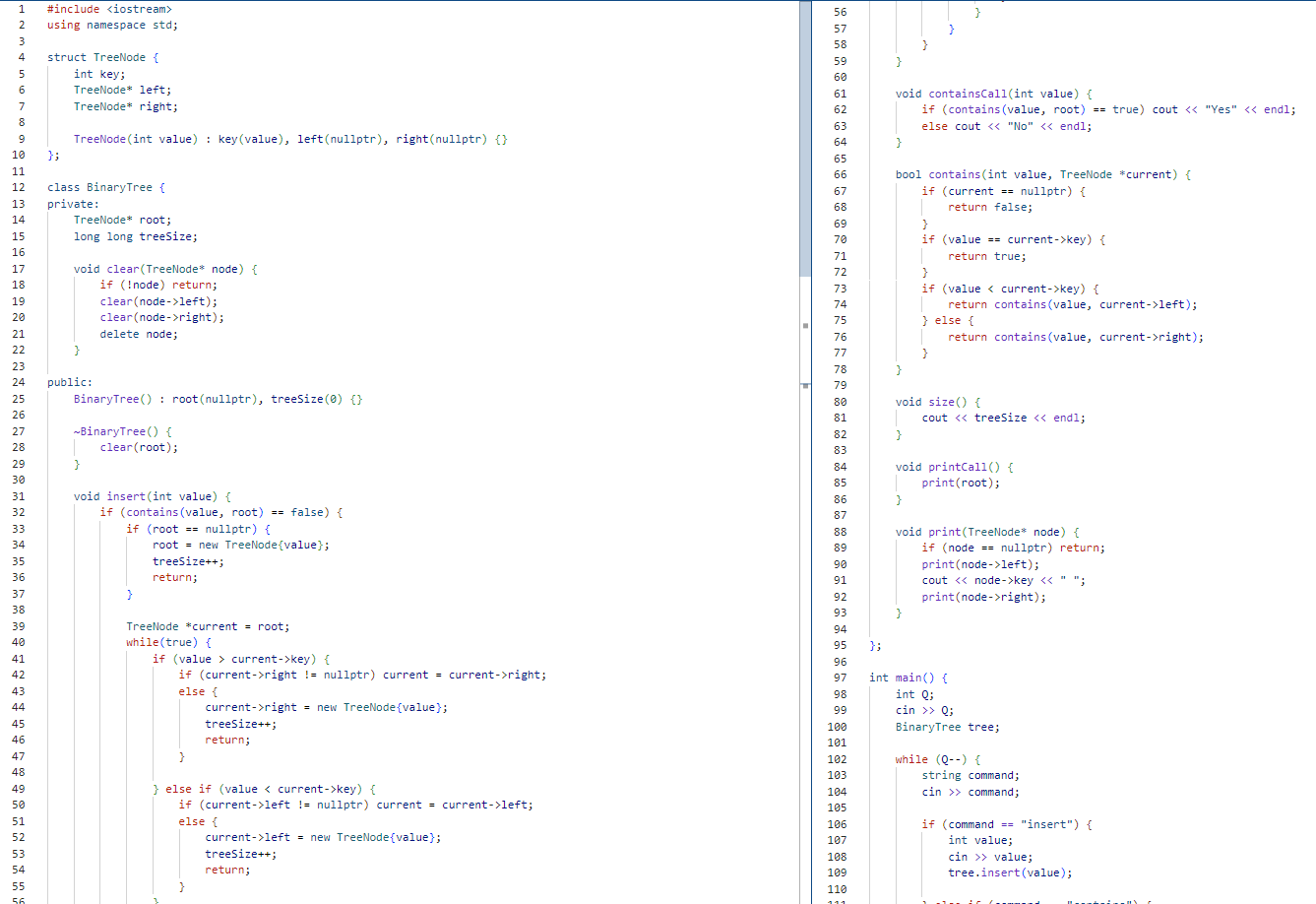
****

****

**Орієнтовний час виконання: 3 год**

**Реальний час виконання: 3 год**

**Завдання №4**

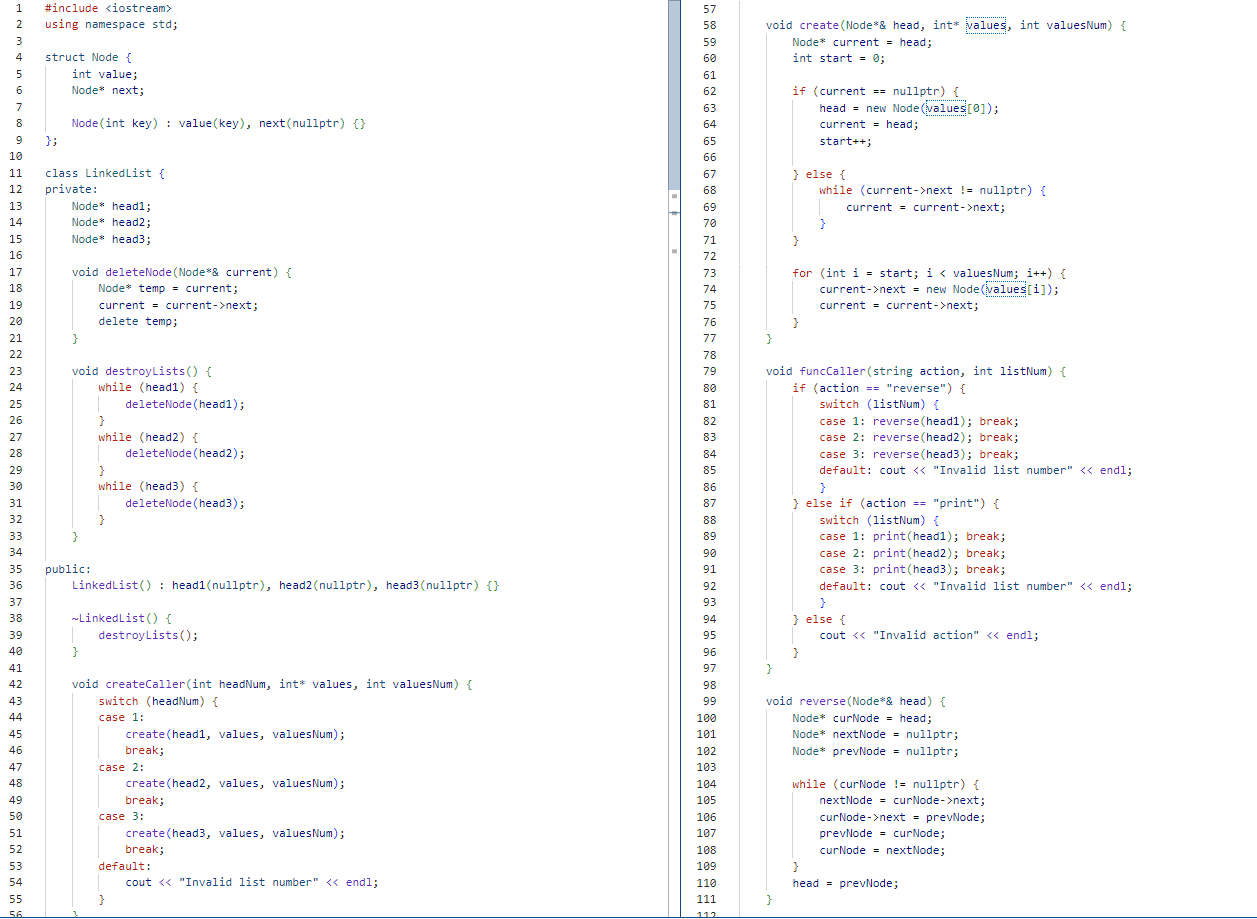
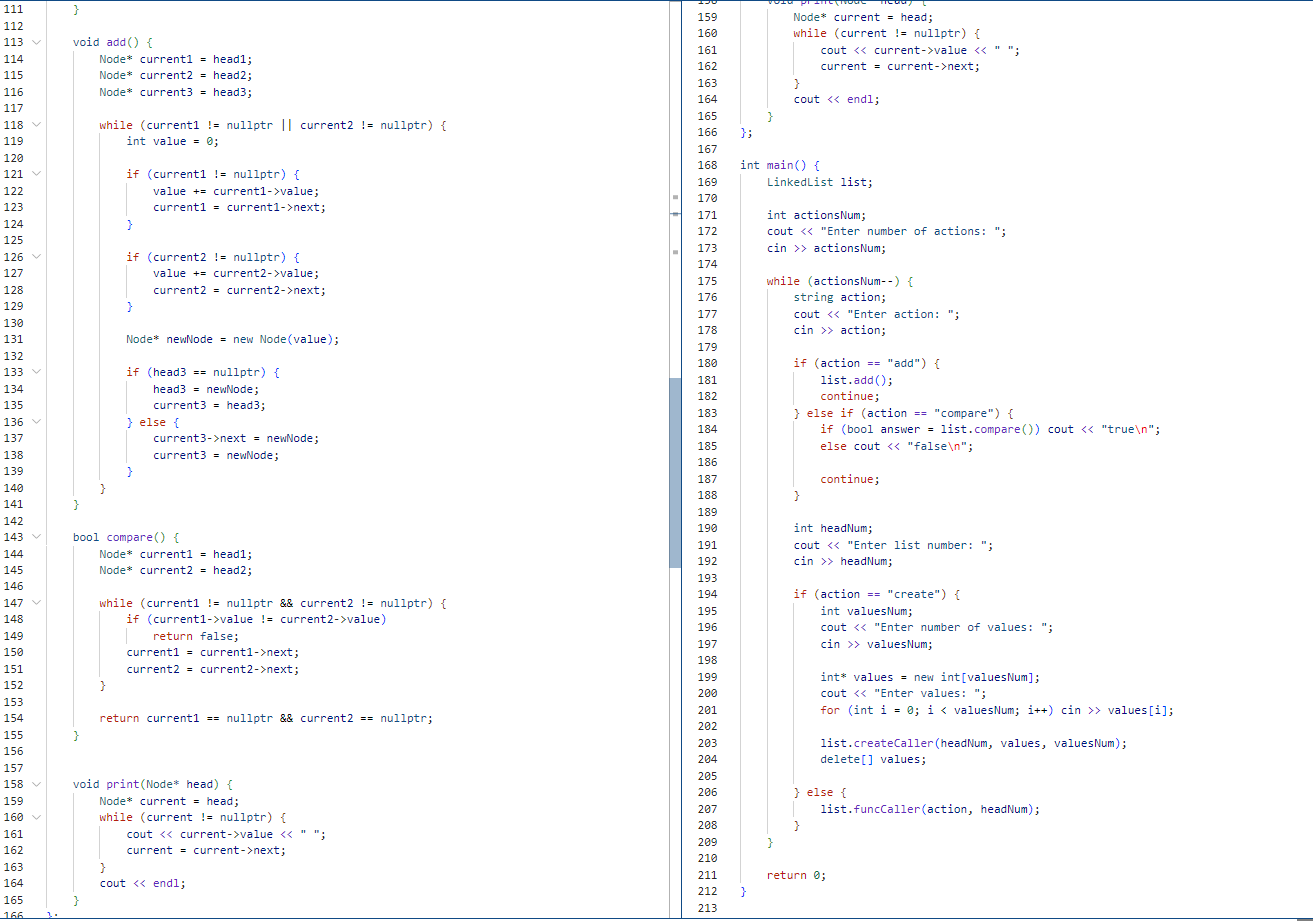
****

****

**Орієнтовний час виконання: 1 год**

**Реальний час виконання: 1.5 год**

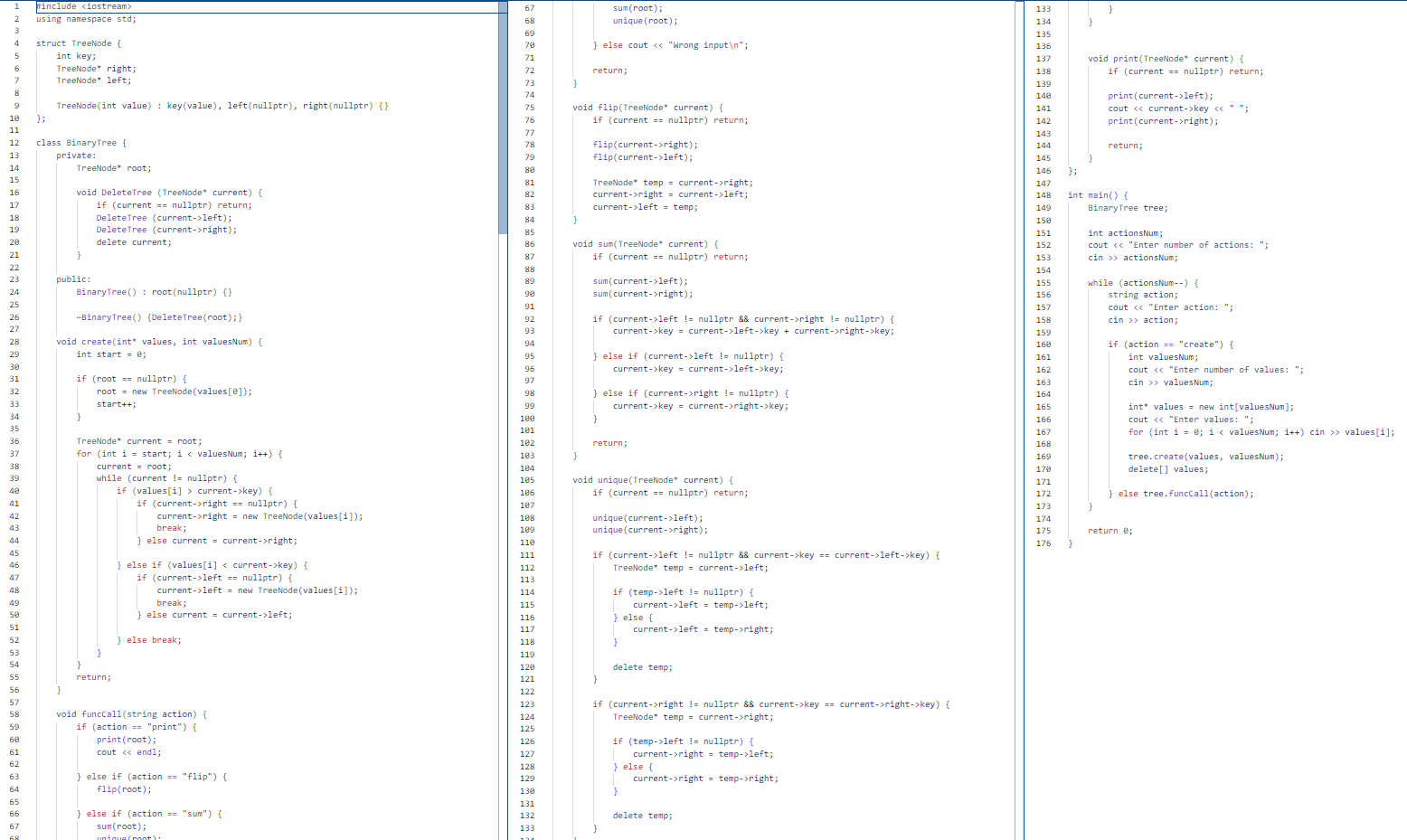
**Завдання №5**

**  
**

**Орієнтовний час виконання: 1 год**

**Реальний час виконання: 1.5 год**

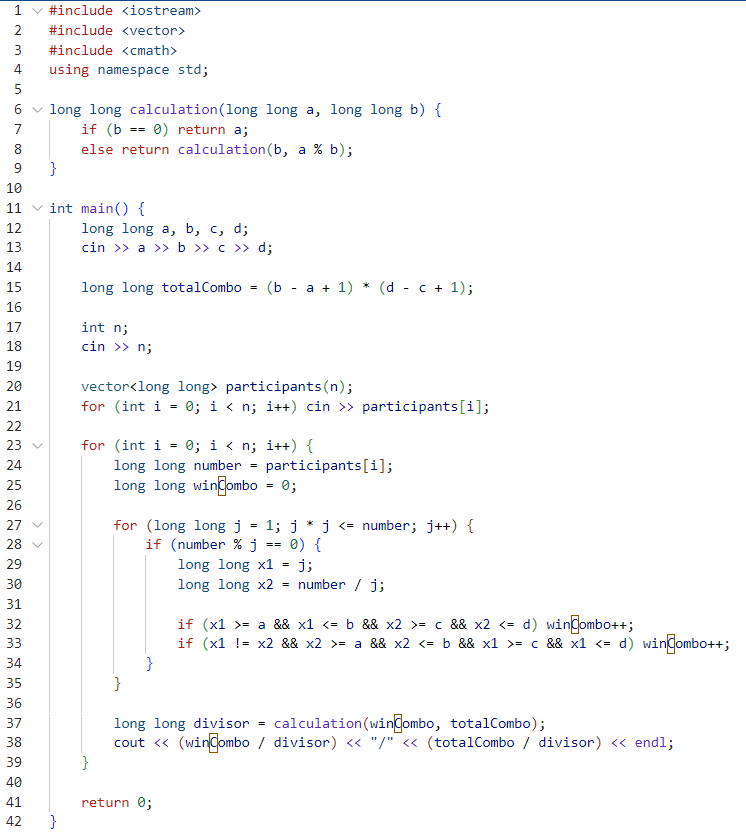
**Завдання №6**

****

**Орієнтовний час виконання: 1 год**

**Реальний час виконання: 1 год**

**Завдання №7**

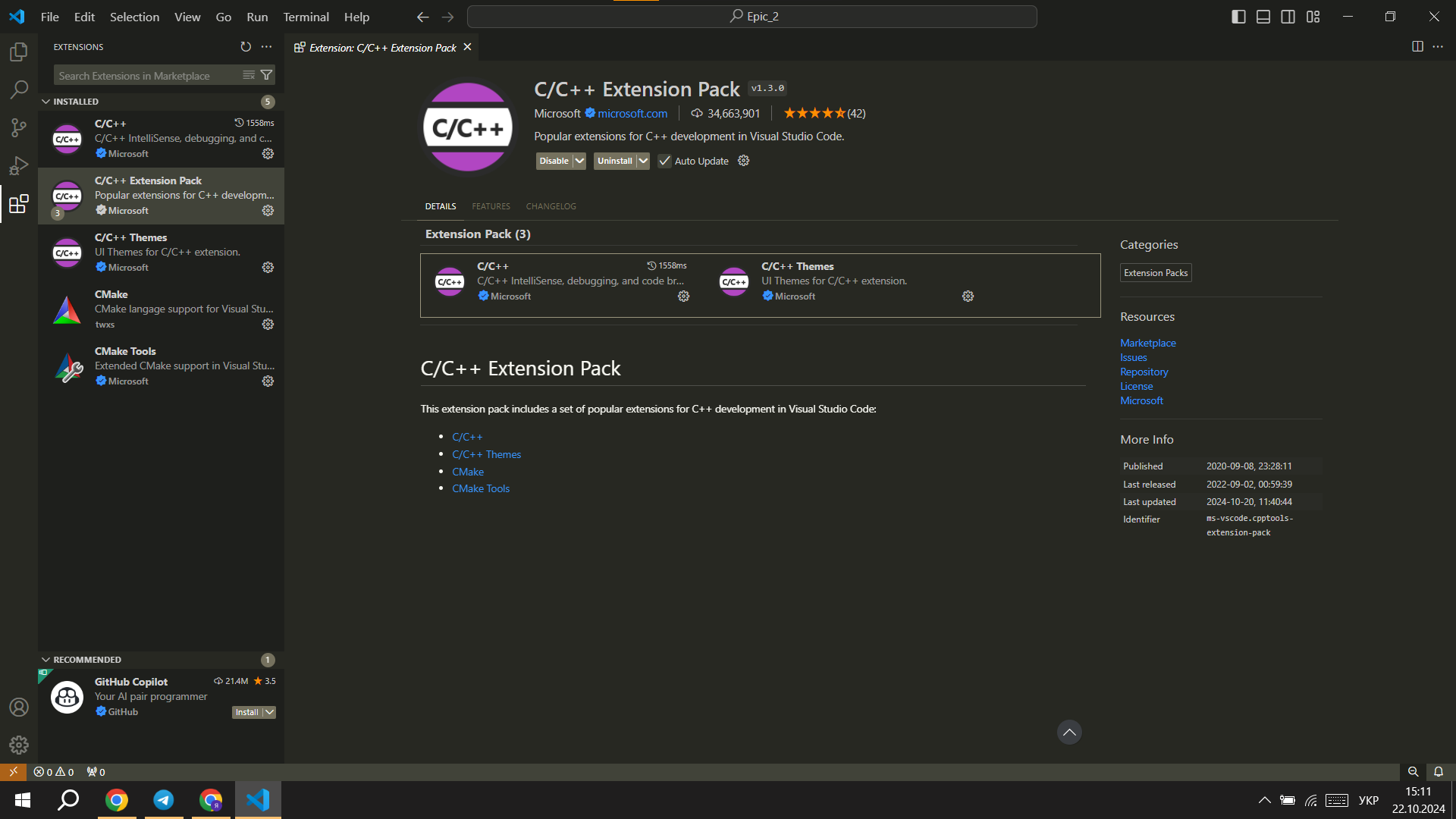
****

**Орієнтовний час виконання: 1 год**

**Реальний час виконання: 1 год**

**Частина 3**

Для виконання роботи використовується середовище **Visual Studio Code** зі встановленим розширенням **С/C++ Extension Pack.**

****

**Використані бібліотеки:**

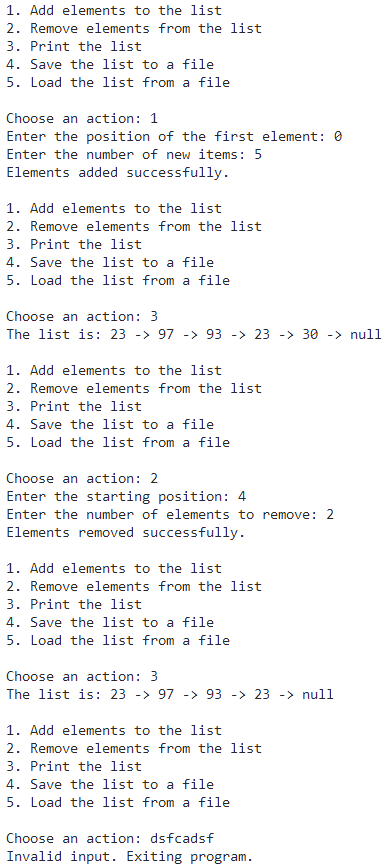
* **iostream**
* **vector**
* **cmath**
* **fstream**
* **random**
* **csdint**

**Частина 4**

**Завдання №**

**Частина 5**

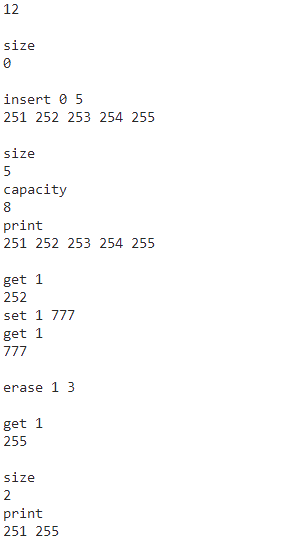
**Завдання №1**



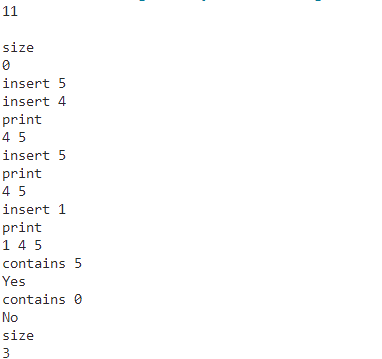
**Завдання №2**



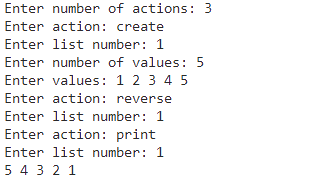
**Завдання №3**



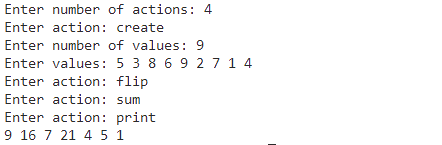
**Завдання №4**



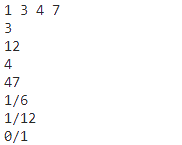
**Завдання №5**

****

**Завдання №6**

****

**Завдання №7**

****

**Висновок**

У цій роботі я здобув практичні навички роботи з лінкованими списками та бінарними деревами в C++. Я навчився створювати, модифікувати, видаляти елементи структур даних, а також реалізовувати алгоритми пошуку, вставки та обходу. Це заклало міцну основу для подальшого вивчення складніших алгоритмів і структур даних.