Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 6**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки динамічних структур.»

***Виконала:***

студентка групи ШІ-12

Онаць Максим Володимирович

# **Тема роботи:**

Написання лабораторних робіт на мові С++, ознайомлення з такими поняттями як: черга, стек, однозв’язний список, двозв’язний список, бінарне дерево; їх практичне застосування.

# **Мета роботи:**

Написати лабораторні роботи на мові С++; ознайомитись з такими поняттями як: черга, стек, однозв’язний список, двозв’язний список, бінарне дерево; та їх практично застосувати.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Однозв’язний список
* Тема №2: Двозв’язний список
* Тема №3: Бінарне дерево

1. Індивідуальний план опрацювання теорії:
   * Тема №1: Однозв’язний список
     + Відео: <https://www.youtube.com/watch?v=-25REjF_atI>
     + Стаття: <https://prometheus.org.ua/cs50/sections/section6.html>
   * Що опрацьовано:
     + Опрацьовано поняття однозв’язного списку, його використання, переваги і недоліки
   * Статус: Ознайомлений
   * Початок опрацювання теми: 04.12.2023
   * Звершення опрацювання теми: 07.12.2023

* Тема №2: Двозв’язний список
  + Джерела Інформації:
    - Відео: <https://www.youtube.com/watch?v=QLzu2-_QFoE>
  + Що опрацьовано:
    - Опрацьовано поняття двозв’язного списку, його використання, переваги і недоліки
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.12.2023
  + Звершення опрацювання теми: 10.12.2023
* Тема №3: Бінарне дерево
  + Джерела Інформації:
    - Відео: <https://www.youtube.com/watch?v=qBFzNW0ALxQ>
  + Що опрацьовано:
    - Опрацьовано поняття бінарного дерева, його структуру і використання
  + Статус: Ознайомлений
  + Початок опрацювання теми: 10.12.2023
  + Звершення опрацювання теми: 14.12.2023

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 ”Class Practice\_1”

* Варіант: Загальний варіант
* Деталі завдання: Реалізувати метод реверсу списку: Node\* reverse(Node \*head);
* Умови задачі:
* - використовувати цілочисельні значення в списку;
* - реалізувати метод реверсу;
* - реалізувати допоміжний метод виведення вхідного і обернутого списків;
* Важливі деталі для врахування в імплементації програми: правильно записати і реалізувати структуру списку, його вузол

Завдання №2 ”Class Practice\_2”

* Варіант завдання: Загальний варіант
* Деталі завдання:
* Порівняння списків
* bool compare(Node \*h1, Node \*h2);
* 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.
* Важливі деталі для врахування в імплементації програми: правильно записати і реалізувати структуру списку, його вузол

Завдання №3 ”Class Practice\_3”

* Варіант завдання: Загальний варіант
* Деталі завдання:
* Додавання великих чисел
* Node\* add(Node \*n1, Node \*n2);
* 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.
* Важливі деталі для врахування в імплементації програми: правильно записати і реалізувати структуру списку, його вузол

Завдання №4 ”Class Practice\_4”

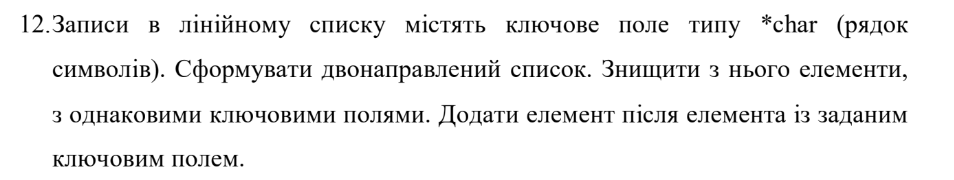
* Варіант завдання: Загальний варіант
* Деталі завдання:
* Віддзеркалення дерева
* TreeNode \*create\_mirror\_flip(TreeNode \*root);
* Умови задачі:
* - використовувати цілі числа для значень у вузлах дерева
* - реалізувати функцію, що проходить по всіх вузлах дерева і міняє місцями праву і ліву вітки дерева
* - функція повертає нове дерево, передане в функцію дерево не модифікується.
* Важливі деталі для врахування в імплементації програми: правильно записати і реалізувати структуру дерева, його вузол, віддзеркалення

Завдання №5 ”Class Practice\_5”

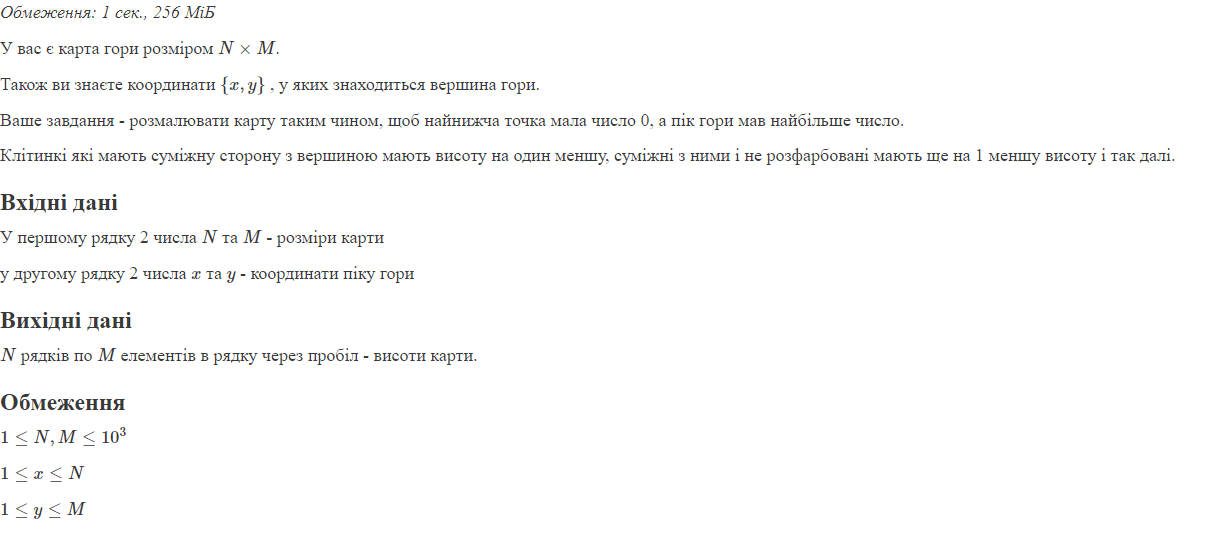
* Варіант завдання: Загальний варіант
* Деталі завдання:
* Записати кожному батьківському вузлу суму підвузлів
* void tree\_sum(TreeNode \*root);
* Умови задачі:
* - використовувати цілочисельні значення у вузлах дерева;
* - реалізувати функцію, яка ітеративно проходить по бінарному дереві і записує у батьківський вузол суму значень підвузлів
* - вузол-листок не змінює значення
* - значення змінюються від листків до кореня дерева.
* Важливі деталі для врахування в імплементації програми: правильно записати і реалізувати структуру дерева, його вузол, додавання вузлів

Завдання №6 “VNS\_LAB\_10”

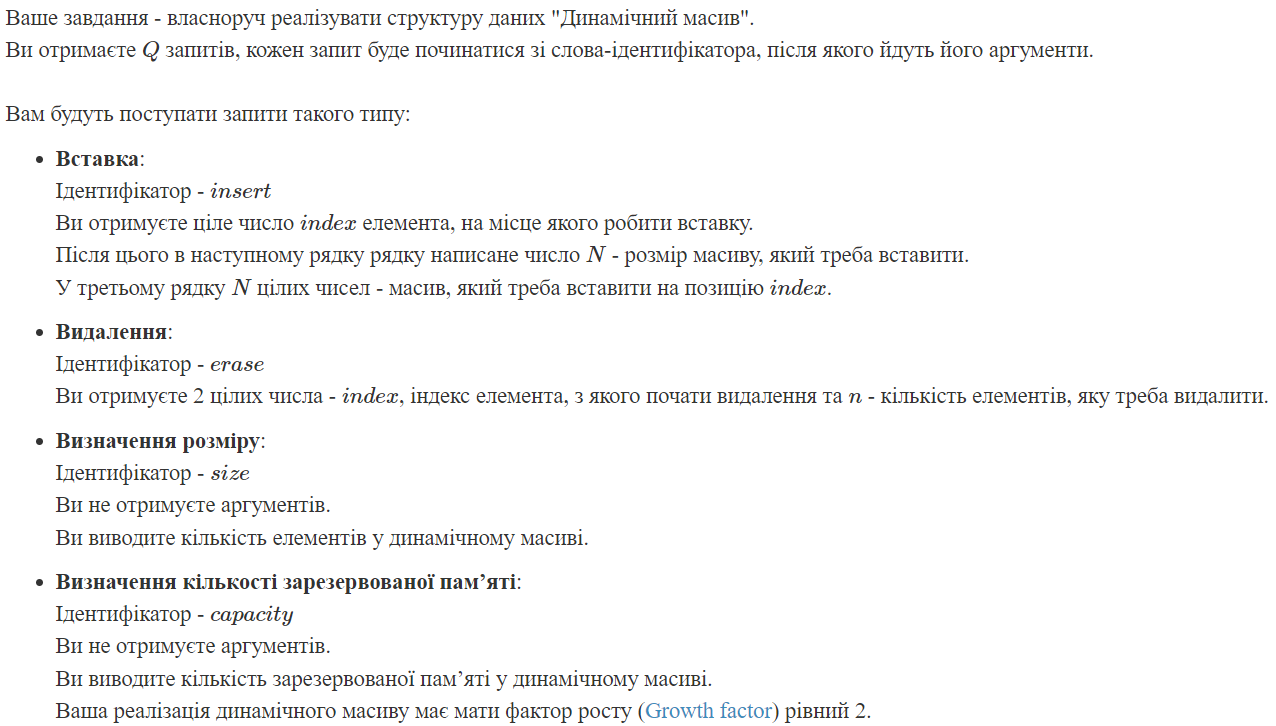
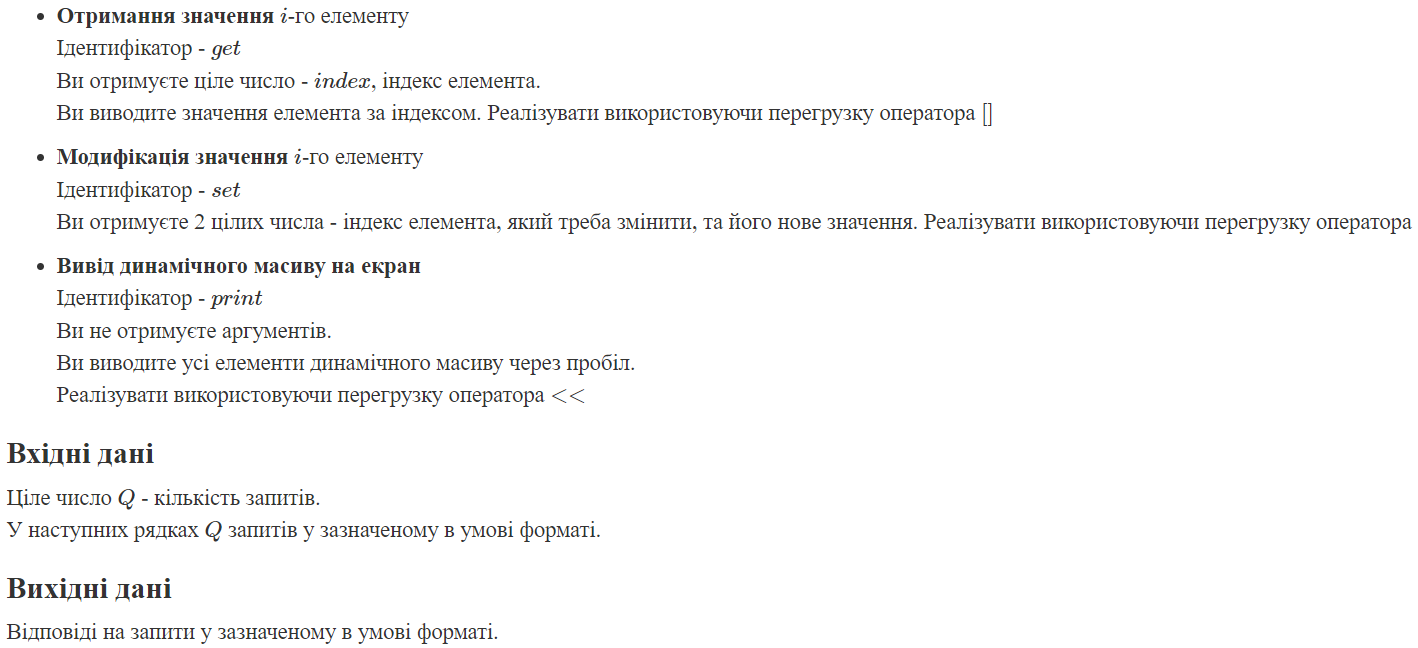
* Варіант завдання:
* Деталі завдання:



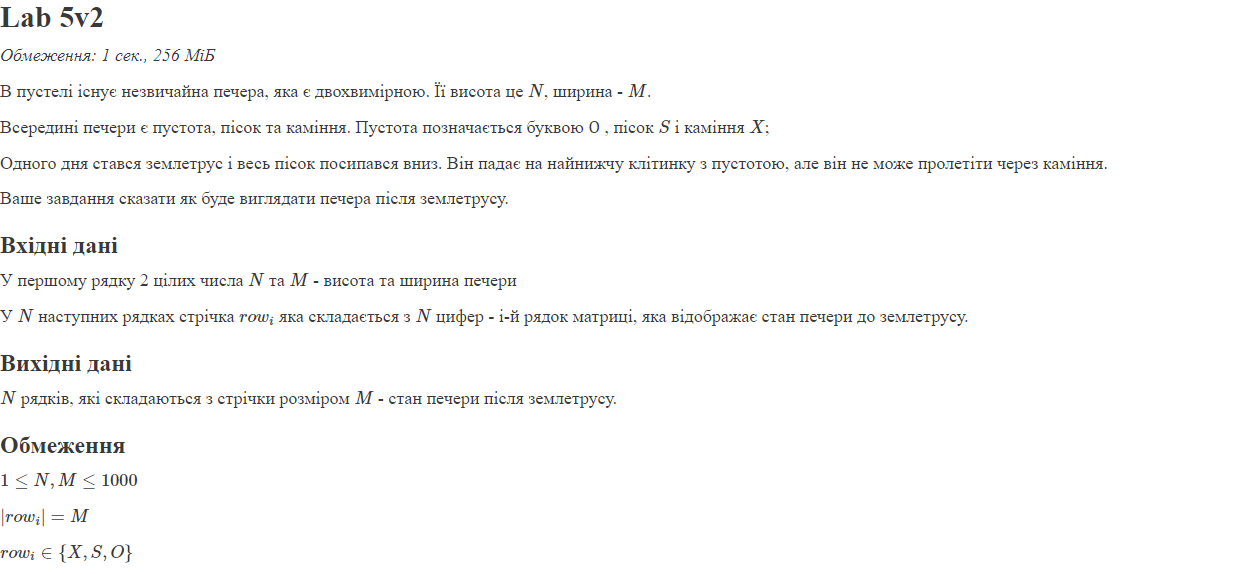
Завдання №7 “Algotester\_lab\_5”

* Варіант завдання: 3
* Деталі завдання:

Завдання №8 “Algotester\_lab\_78”

* Варіант завдання: 2
* Деталі завдання:
*  

Завдання №9 ”Self Practice”

* Деталі завдання: 

## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма №1 ”Class Practice\_1”

* Планований час на реалізацію: 40 хв

Програма №2 ”Class Practice\_2”

* Планований час на реалізацію: 45 хв

Програма №3 ”Class Practice\_3”

* Планований час на реалізацію: 35 хв

Програма №4 ”Class Practice\_4”

* Планований час на реалізацію: 50 хв

Програма №5 ”Class Practice\_5”

* Планований час на реалізацію: 1 год

Програма №6 “VNS\_LAB\_10”

Блок-схема

Рисунок 1: Блок схема до програми №6

* Планований час на реалізацію: 2 год

Програма №7 “Algotester\_lab\_5”

* Планований час на реалізацію: 1 год

Програма №8 “Algotester\_lab\_78”

* Планований час на реалізацію: 2 год

Програма №9 “Self Practice”

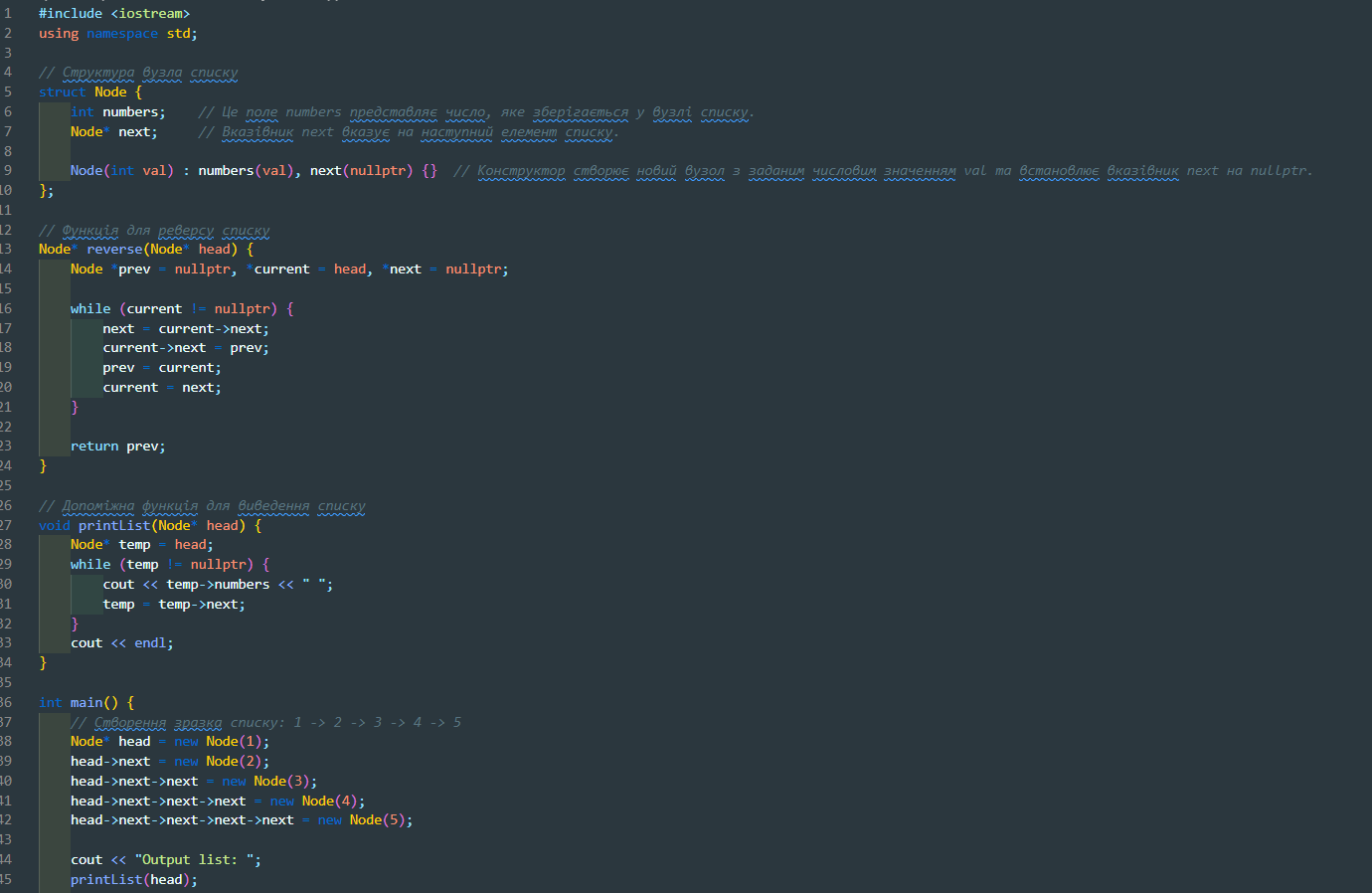
* Планований час на реалізацію: 40 хв

## **3. Конфігурація середовища до виконання завдань:**

Додаткова конфігурація середовища до виконання завдань не потрібна.

## **4. Код програм з посиланням на зовнішні ресурси:**

Завдання №1 ”Class Practice\_1”



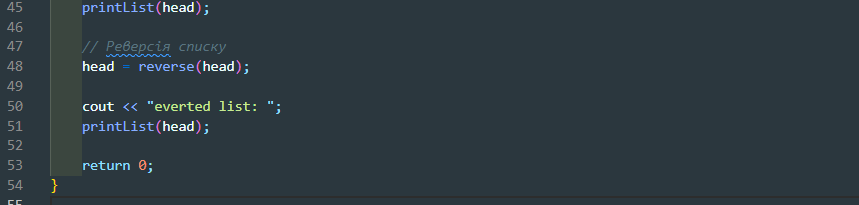
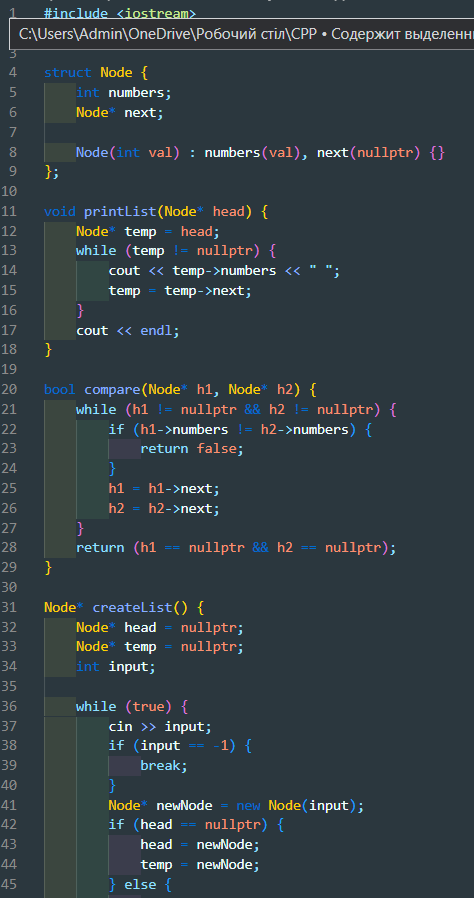


Рисунок 2: Код до програми №1

Завдання №2 ”Class Practice\_2”



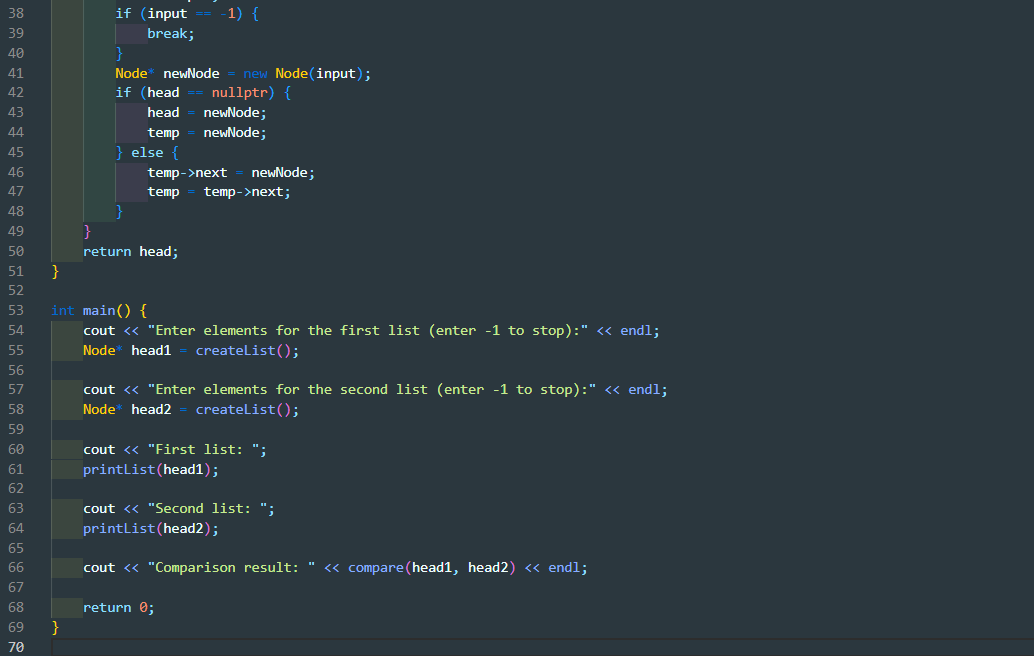


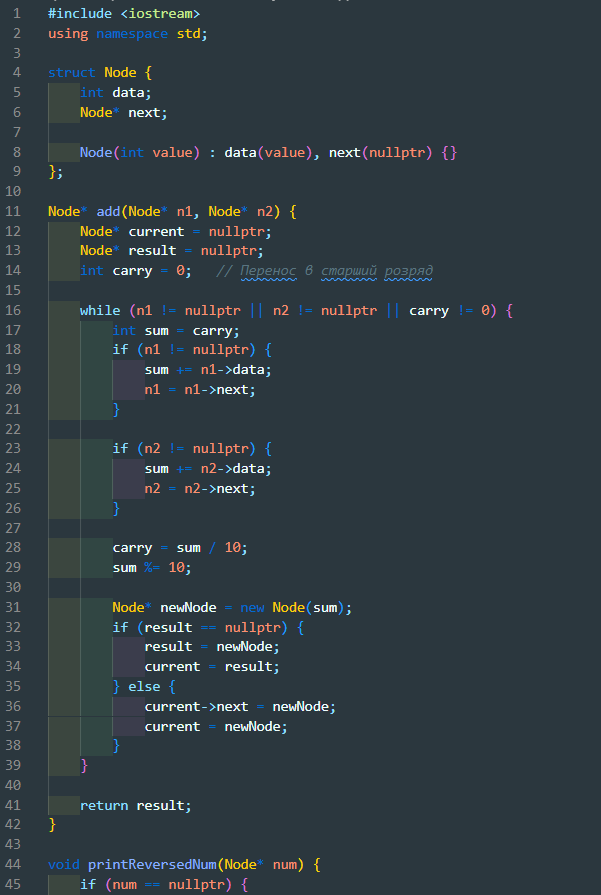
Рисунок 3: Код до програми №2

Деталі завдання:

* Виконано порівняння списків
* bool compare(Node \*h1, Node \*h2);
* 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.
* Важливі деталі для врахування в імплементації програми: правильно записано і реалізовано структуру списку, його вузол

Посилання на файл програми у пул-запиті GitHub

Завдання №3 ”Class Practice\_3”



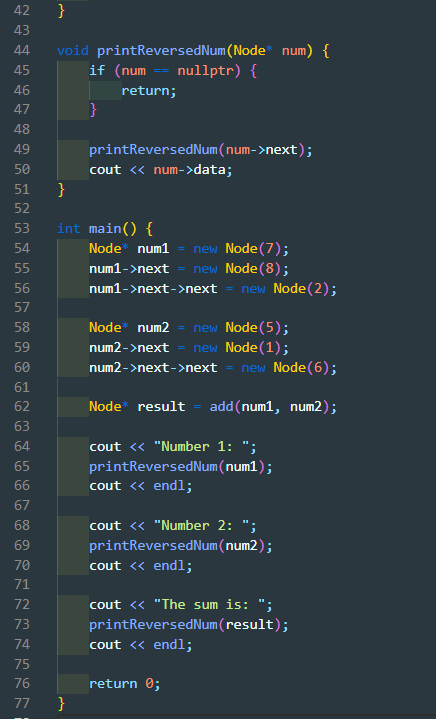


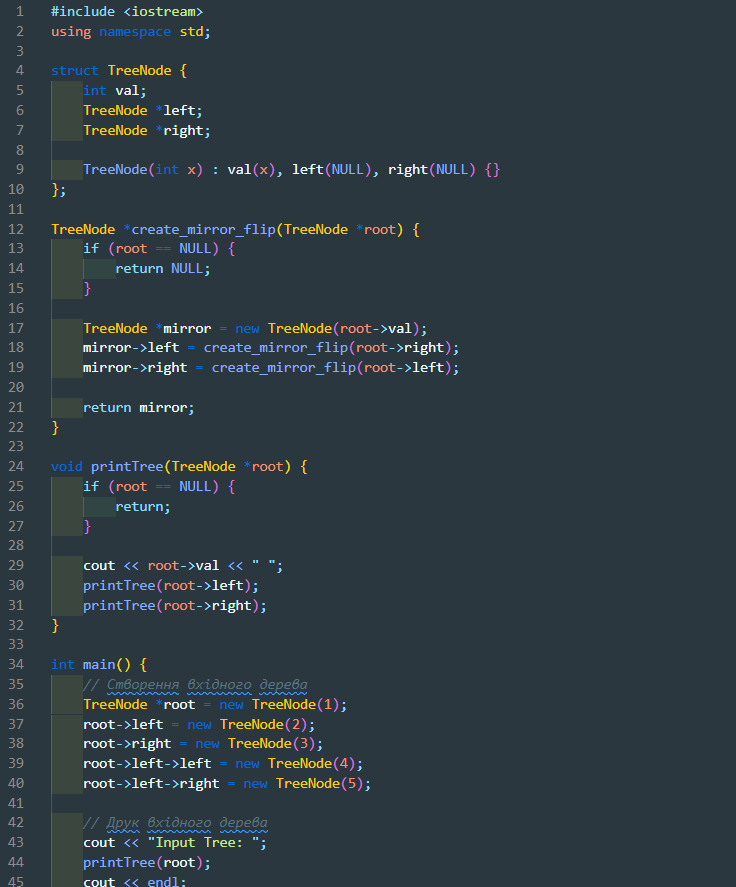
Рисунок 4: Код до програми №3

* Деталі завдання:
* Реалізовано додавання великих чисел
* Node\* add(Node \*n1, Node \*n2);
* 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.

Важливі деталі для врахування в імплементації програми: правильно записано і реалізовано структуру списку, його вузол

Посилання на файл програми у пул-запиті GitHub

Завдання №4 ”Class Practice\_4”



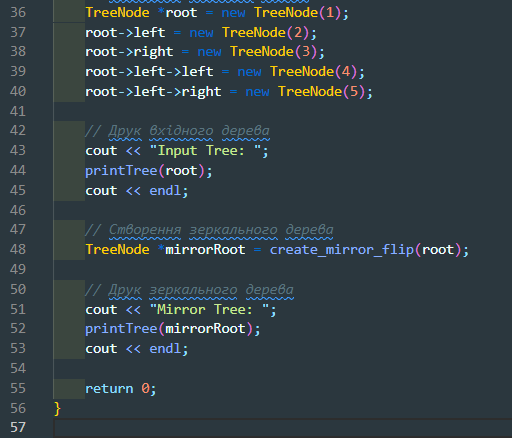
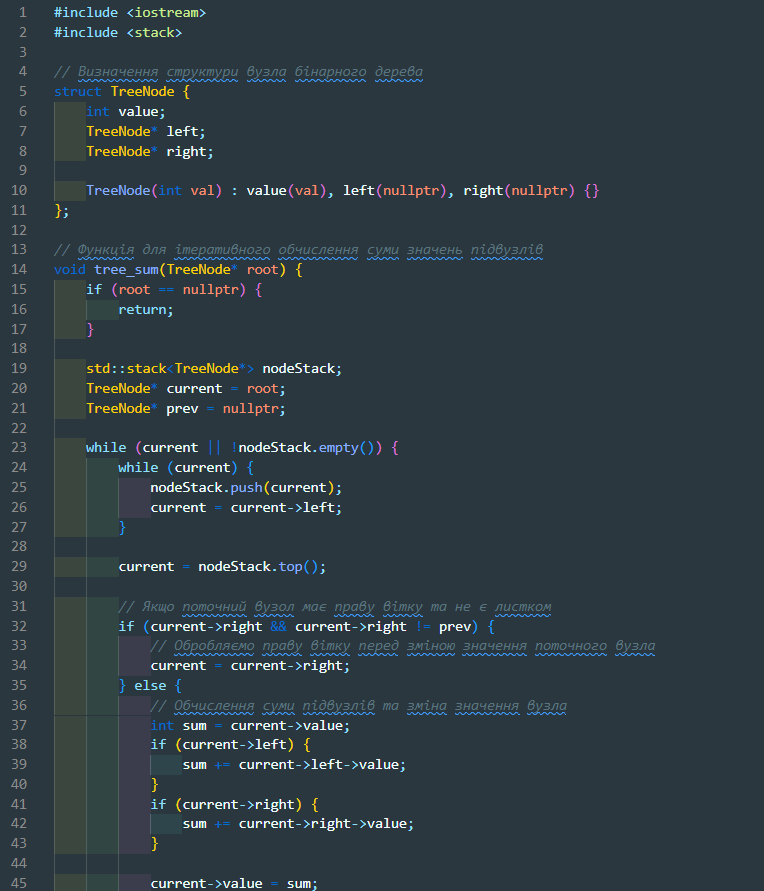


Рисунок 5: Код до програми №4

* Деталі завдання:
* Реалізовано віддзеркалення дерева
* TreeNode \*create\_mirror\_flip(TreeNode \*root);
* 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.
* Важливі деталі для врахування в імплементації програми: правильно записано і реалізовано структуру дерева, його вузол, віддзеркалення

Посилання на файл програми у пул-запиті GitHub

Завдання №5 ”Class Practice\_5”



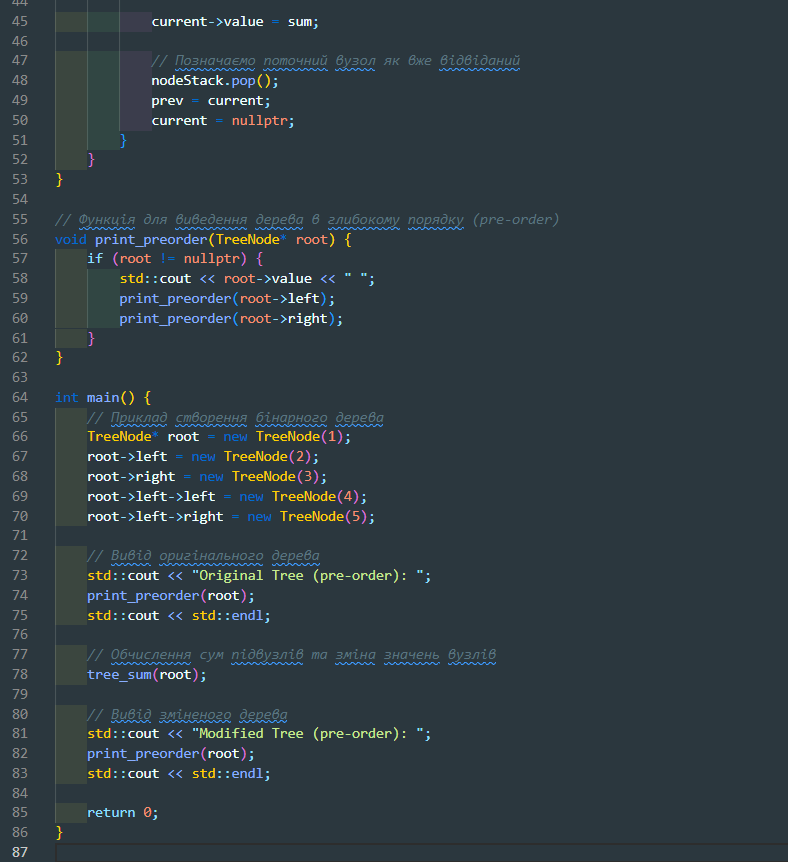
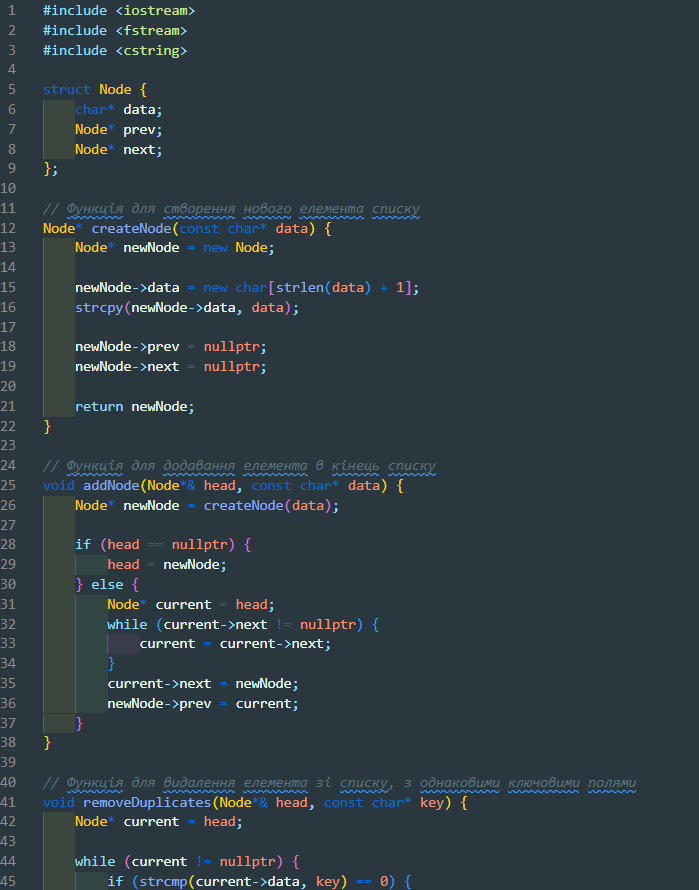


Рисунок 6: Код до програми №5

* Деталі завдання:
* Записано кожному батьківському вузлу суму підвузлів
* void tree\_sum(TreeNode \*root);
* 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.
* Важливі деталі для врахування в імплементації програми: правильно записано і реалізовано структуру дерева, його вузол, додавання вузлів

Посилання на файл програми у пул-запиті GitHub

Завдання №6 “VNS\_LAB\_10”



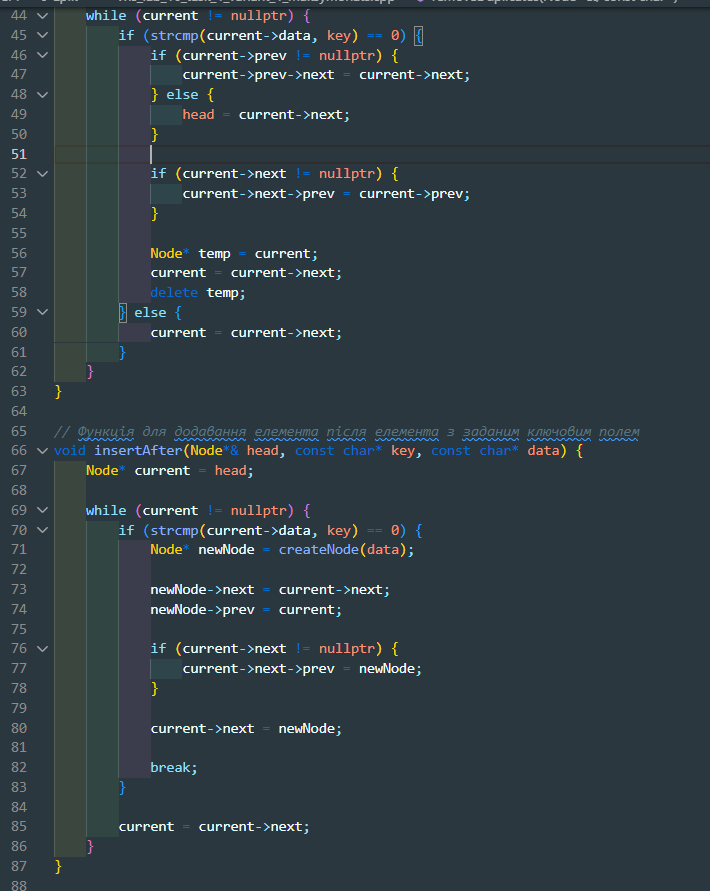


Рисунок 7: Код до програми №6

* Деталі завдання: Написано програму, у якій створюються динамічні структури й виконано їхню обробку у відповідності з моїм варіантом.

Посилання на файл програми у пул-запиті GitHub

Завдання №7 “Algotester\_lab\_5”

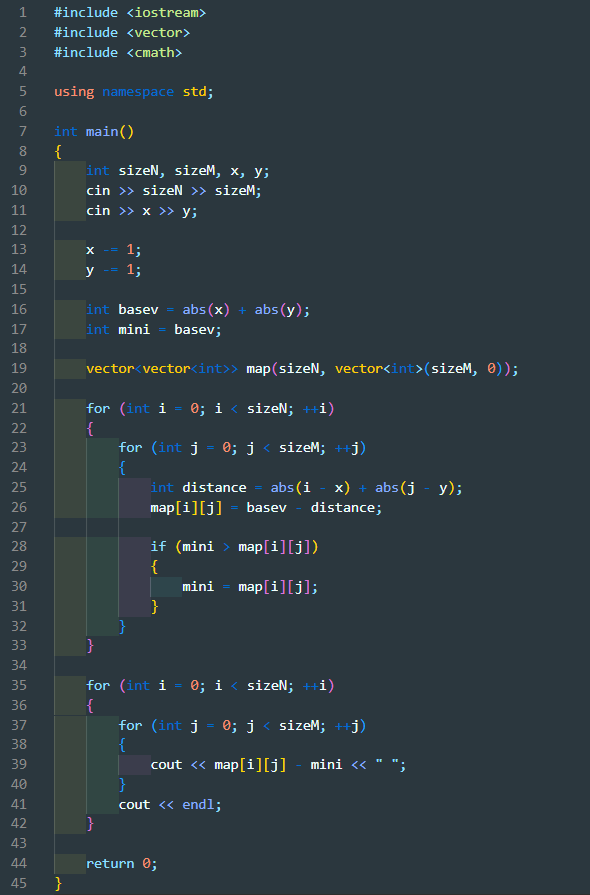
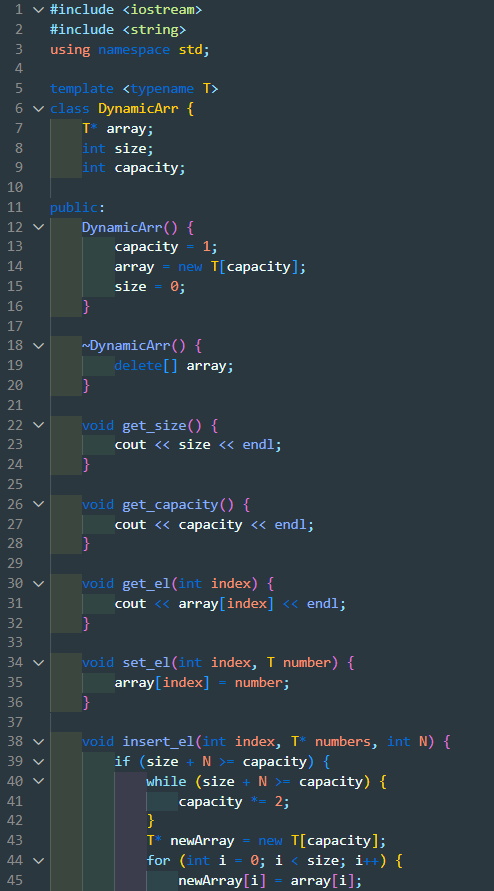
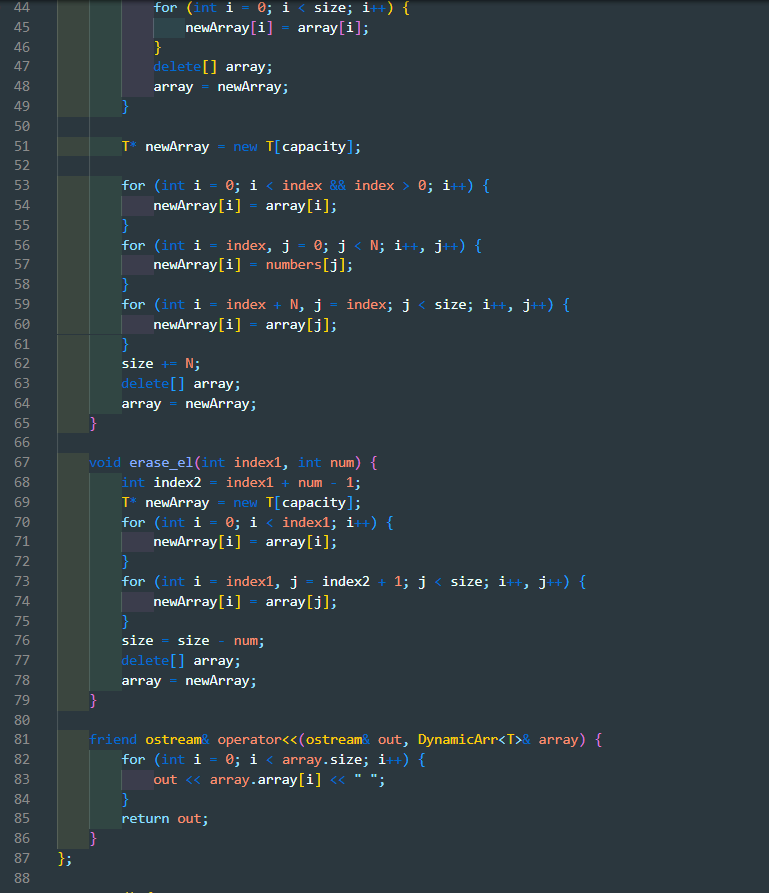


Рисунок 8: Код до програми №7

Посилання на файл програми у пул-запиті GitHub

Завдання №8 “Algotester\_lab\_78”





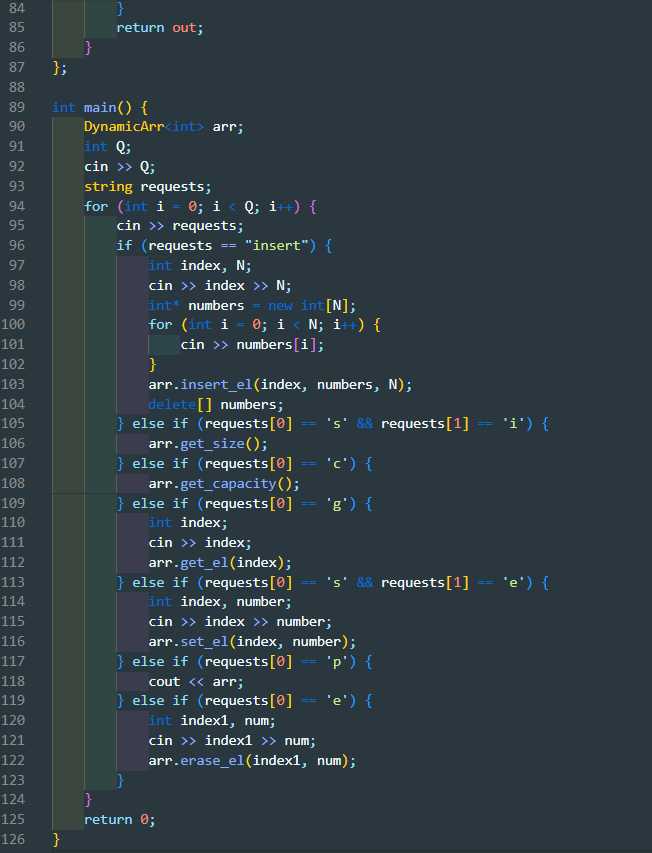


Рисунок 9: Код до програми №8

Посилання на файл програми у пул-запиті GitHub

Завдання №9 ”Self Practice”

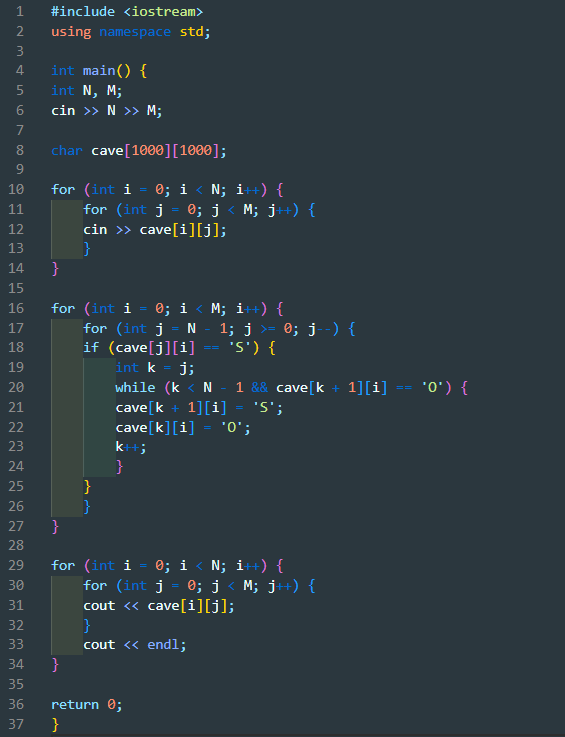


Рисунок 10: Код до програми №9

* Деталі завдання:

Посилання на файл програми у пул-запиті GitHub

## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 ”Class Practice\_1”



Рисунок 11: Результати програми №1

Час затрачений на виконання завдання: 40 хв

Завдання №2 ”Class Practice\_2”

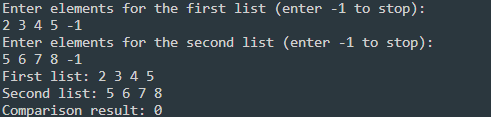


Рисунок 12: Результати програми №2

Час затрачений на виконання завдання: 45 хв

Завдання №3 ”Class Practice\_3”

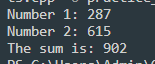


Рисунок 13: Результати програми №3

Час затрачений на виконання завдання: 1 год

Завдання №4 ”Class Practice\_4”



Рисунок 14: Результати програми №4

Час затрачений на виконання завдання: 1,5 год

Завдання №5 ”Class Practice\_5”



Рисунок 15: Результати програми №5

Час затрачений на виконання завдання: 1 год

Завдання №6 ”VNS\_LAB\_10”

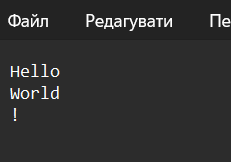


Рисунок 16: Результати програми №6

Час затрачений на виконання завдання: 2,5 год

Завдання №7 ”Algotester\_lab\_5”

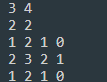
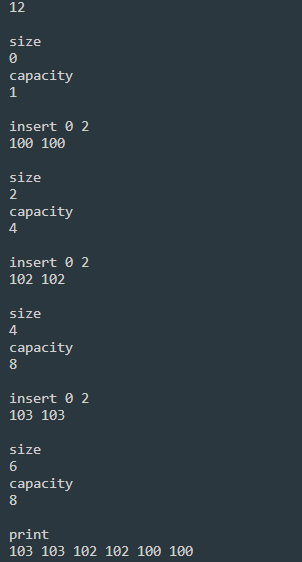


Рисунок 17: Результати програми №7

Час затрачений на виконання завдання: 80 хв

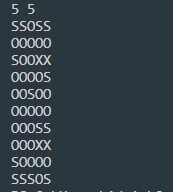
Завдання №8 ”Algotester\_lab\_78”



Час затрачений на виконання завдання: 2 год

Рисунок 18: Результати програми №8

Завдання №9 ”Self practice”

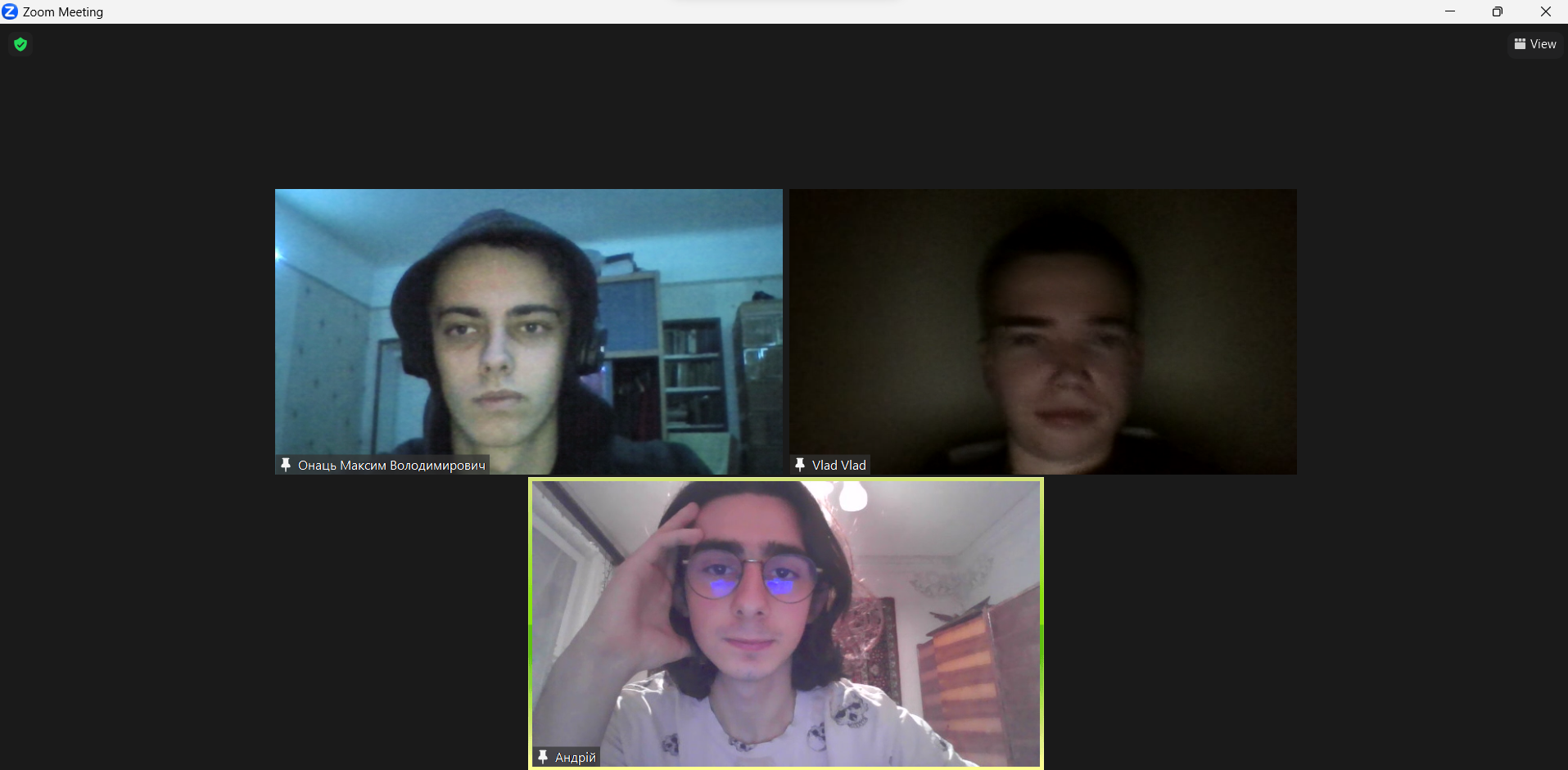


Час затрачений на виконання завдання: 70 хв

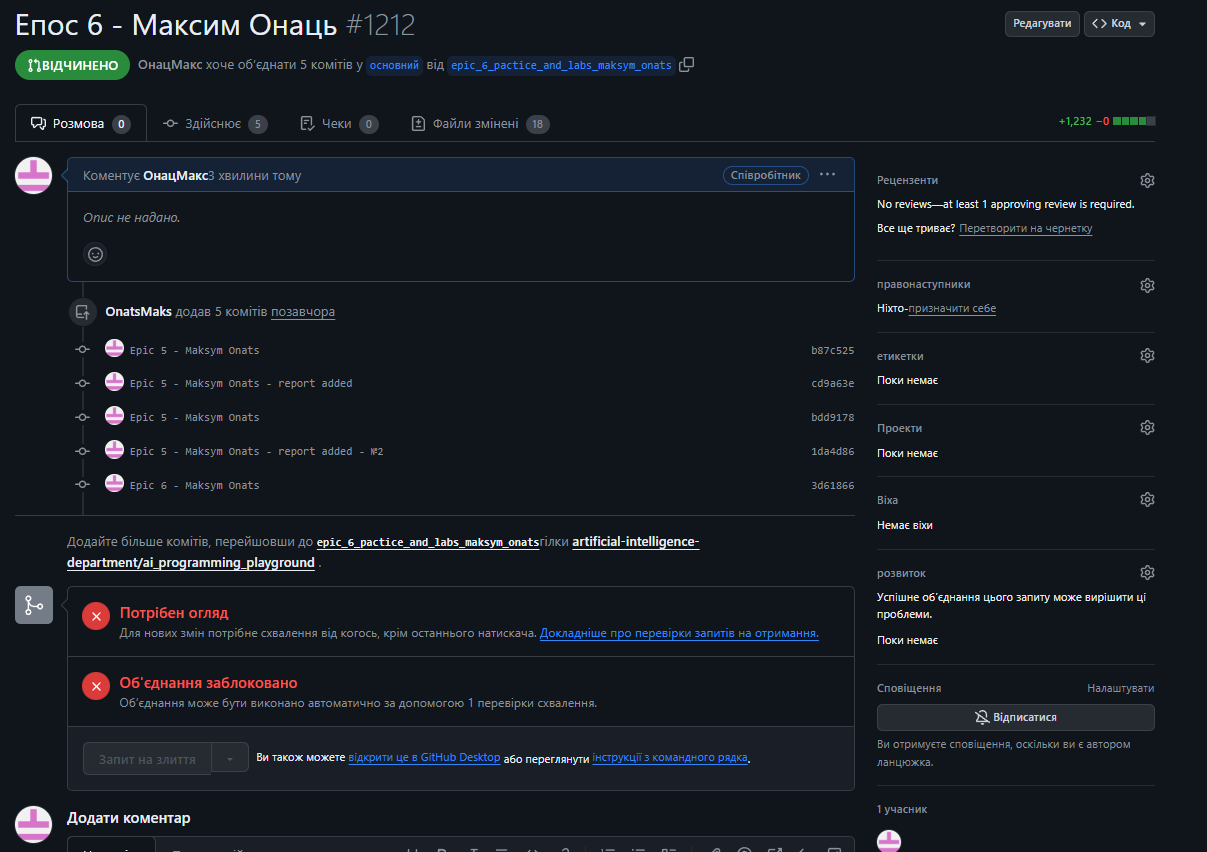
Рисунок 19: Результати програми №9

## **6. Кооперація з командою:**

* Скрін з 1-ї зустрічі по обговоренню задач Епіку



* Скрін пул реквесту



# **Висновки:**

Написано лабораторні роботи на мові С++; ознайомилась з такими поняттями як: черга, стек, однозв’язний список, двозв’язний список, бінарне дерево; та їх практично застосовано.