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 № 6**

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

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

до:

ВНС Лабораторної Роботи № 10

Алготестер Лабораторної Роботи № 5

Алготестер Лабораторної Роботи № 7-8

Практичних Робіт № 6

**Виконав:**

Студент групи ШІ-11

Хайлов Володимир Олексійович

# **Тема роботи:**

Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки Дерев.

# **Мета роботи:**

Ознайомитись та навчатись використовувати динамічні структури даних, такі як Черга, Стек, Списки, та Дерево, а також в освоєнні алгоритмів обробки даних в дереві. Особливості застосування кожної з структур та їх використання для кращої ефективності конкретної задачі.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Динамічні структури (Черга, Стек, Списки, Дерево).
* Тема №2: Алгоритми обробки Дерев.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Динамічні структури (Черга, Стек, Списки, Дерево).
  + Джерела Інформації
    - Присутність на лекціях, опрацювання надоного матеріалу.
    - <https://www.geeksforgeeks.org/data-structures/>
    - <https://www.boardinfinity.com/blog/guide-to-5-data-structures-in-c/>
  + Що опрацьовано:
    - Ознайомився з різними структурами даних стеки, чергою, зв’язними списками та деревами.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 11.29.2023
  + Звершення опрацювання теми: 12.4.2023
* Тема №2: Алгоритми обробки Дерев.
  + Джерела Інформації:
    - Присутність на лекціях, опрацювання надоного матеріалу.
    - <https://www.programiz.com/dsa/trees>
  + Що опрацьовано:
    - Ознайомився з алгоритмами обробки дерев, властивості бінарного дерева, сортування ним та бінарним пошуком.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 11.29.2023
  + Звершення опрацювання теми: 12.4.2023

# **Виконання роботи:**

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

Завдання №\_1 VNS Lab 10 Task 1

* Варіант завдання: 11
* Написати програму, у якій створюються динамічні структури й виконати їхню обробку у відповідності зі своїм варіантом:

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

Завдання №\_2 Algotester Lab 5

* Варіант завдання: 2

Використовуючи функції, розв’язати зазначене у варіанті завдання. Масив

повинен передаватися у функцію як параметр.

Завдання №\_3 Algotester Lab 7-8

* Варіант завдання: 3
* Розв’язати задачу про положення двох дронів.

Завдання №\_4 Class Practice Task 1

* Реверс списку (Reverse list).
* Порівняння списків.
* Додавання великих чисел.

Завдання №\_5 Class Practice Task 2

* Віддзеркалення дерева.
* Записати кожному батьківському вузлу суму підвузлів

Завдання №\_6 Self Practice Task 1

* Задача Впадина

Завдання №\_7 Self Practice Task 2

* Задача Розклад на прості множники

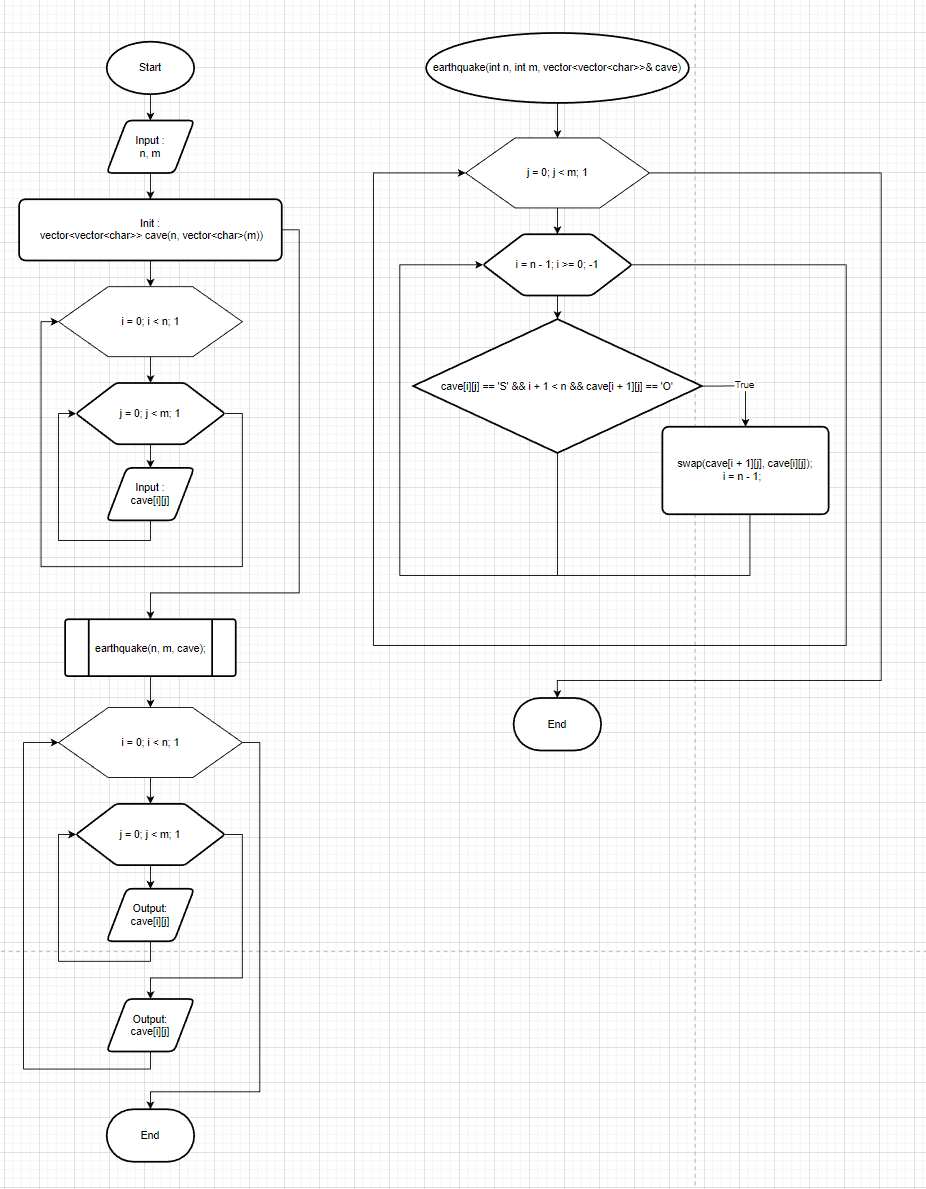
## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма №\_1 VNS Lab 10 Task 1

* Планований час на реалізацію 10 хв.

Програма №\_2 Algotester Lab 5

* Блок-схема



*Figure 1. Flowchart №\_2 Algotester Lab 5.*

* Планований час на реалізацію 15 хв.

Програма №\_3 Algotester Lab 7-8

* Планований час на реалізацію 20 хв.

Програма №\_4 Class Practice Task 1

* Планований час на реалізацію 15 хв.

Програма №\_5 Class Practice Task 2

* Планований час на реалізацію 15 хв.

Програма №\_6 Self Practice Task 1

* Планований час на реалізацію 15 хв.

Програма №\_7 Self Practice Task 2

* Планований час на реалізацію 25 хв.

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

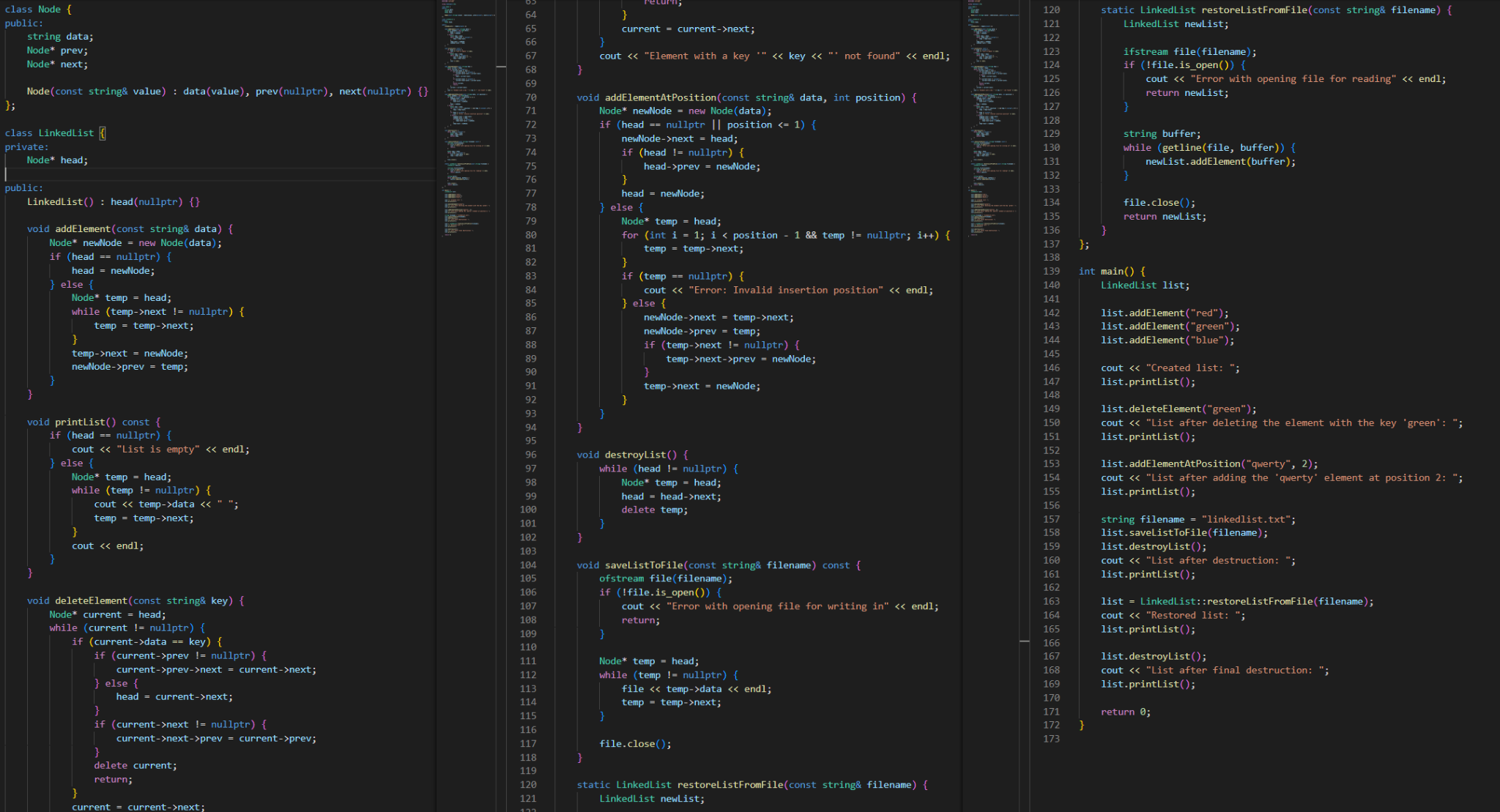
Встановлено термінал MSYS2, та вньому встановлено C++.

Скачано VSCode та встановлено extensions.

У всіх інші необхідних середовищах Git, Git Hub, Algotester, Trello, Draw.io - зареєстрований та сконфігуровав всі потрібні налаштування.

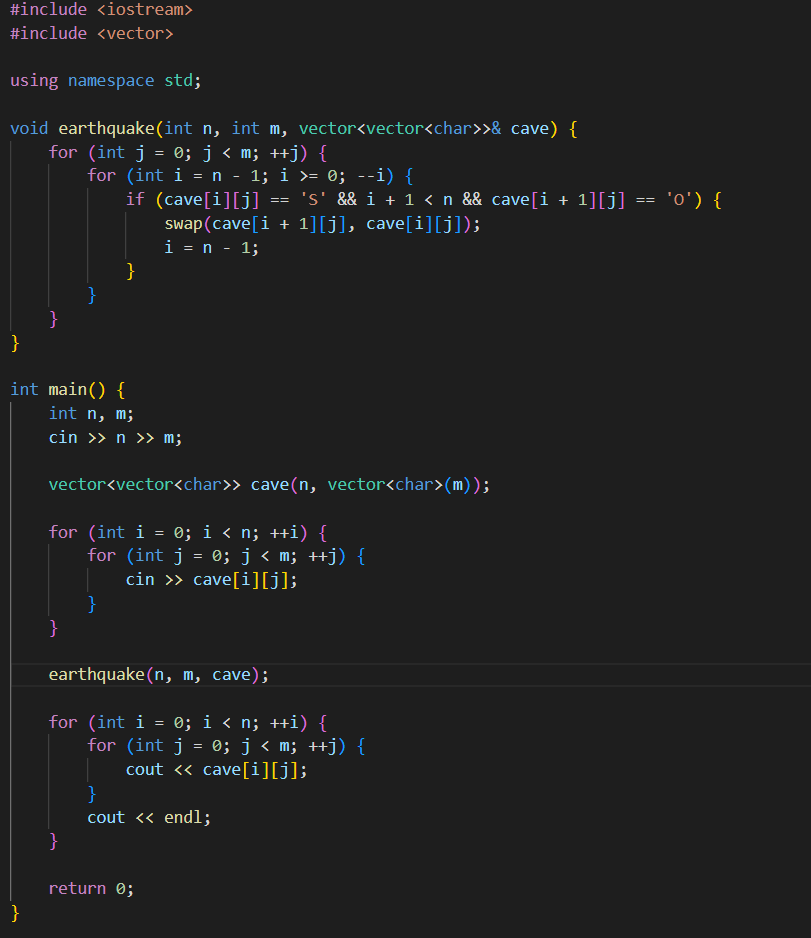
## **4. Код програм з посиланням на зовнішні ресурси:**

Завдання №\_1 VNS Lab 10 Task 1



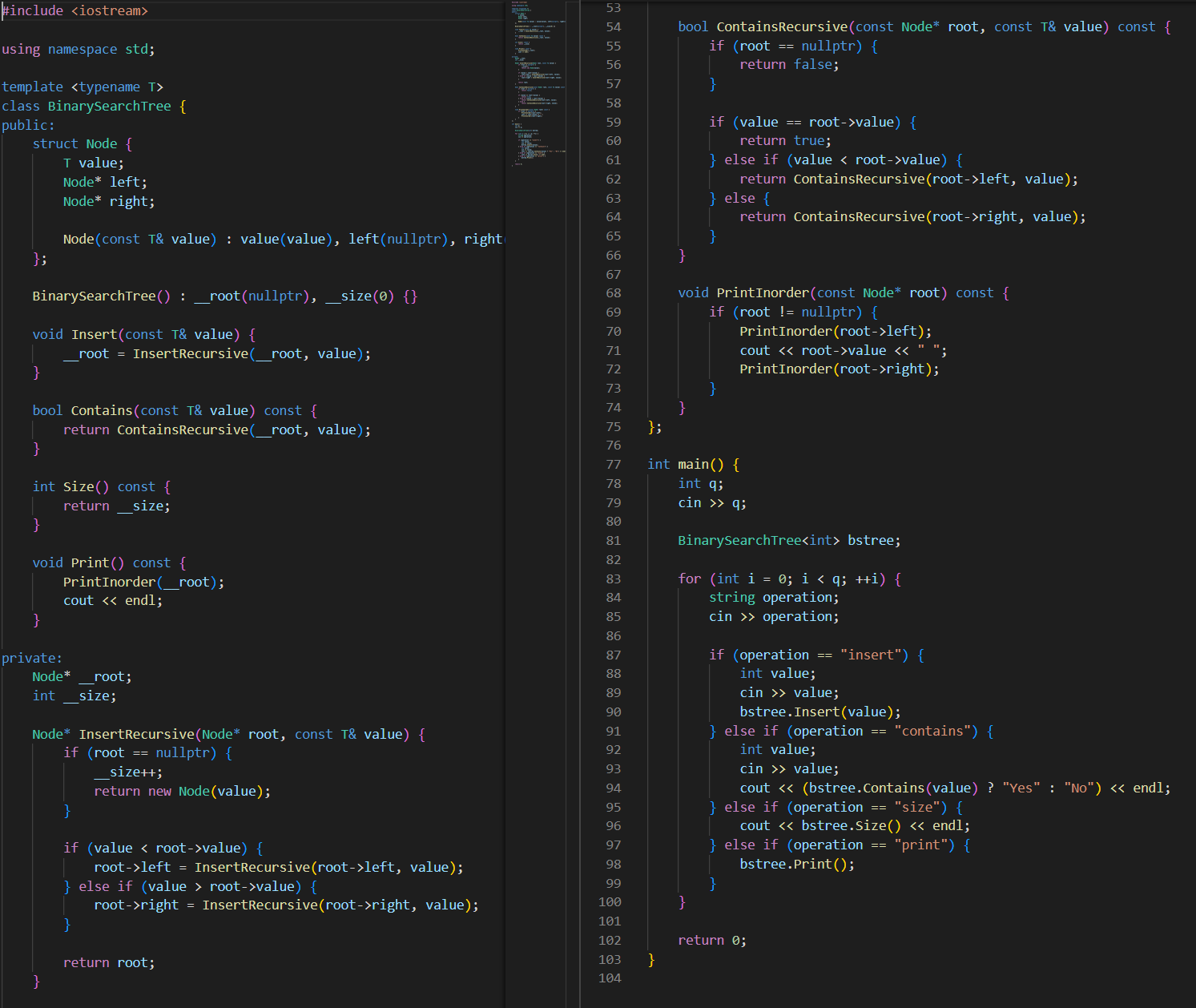
*Figure 2. №\_1 VNS Lab 10 Task 1.*

Завдання №\_2 Algotester Lab 5



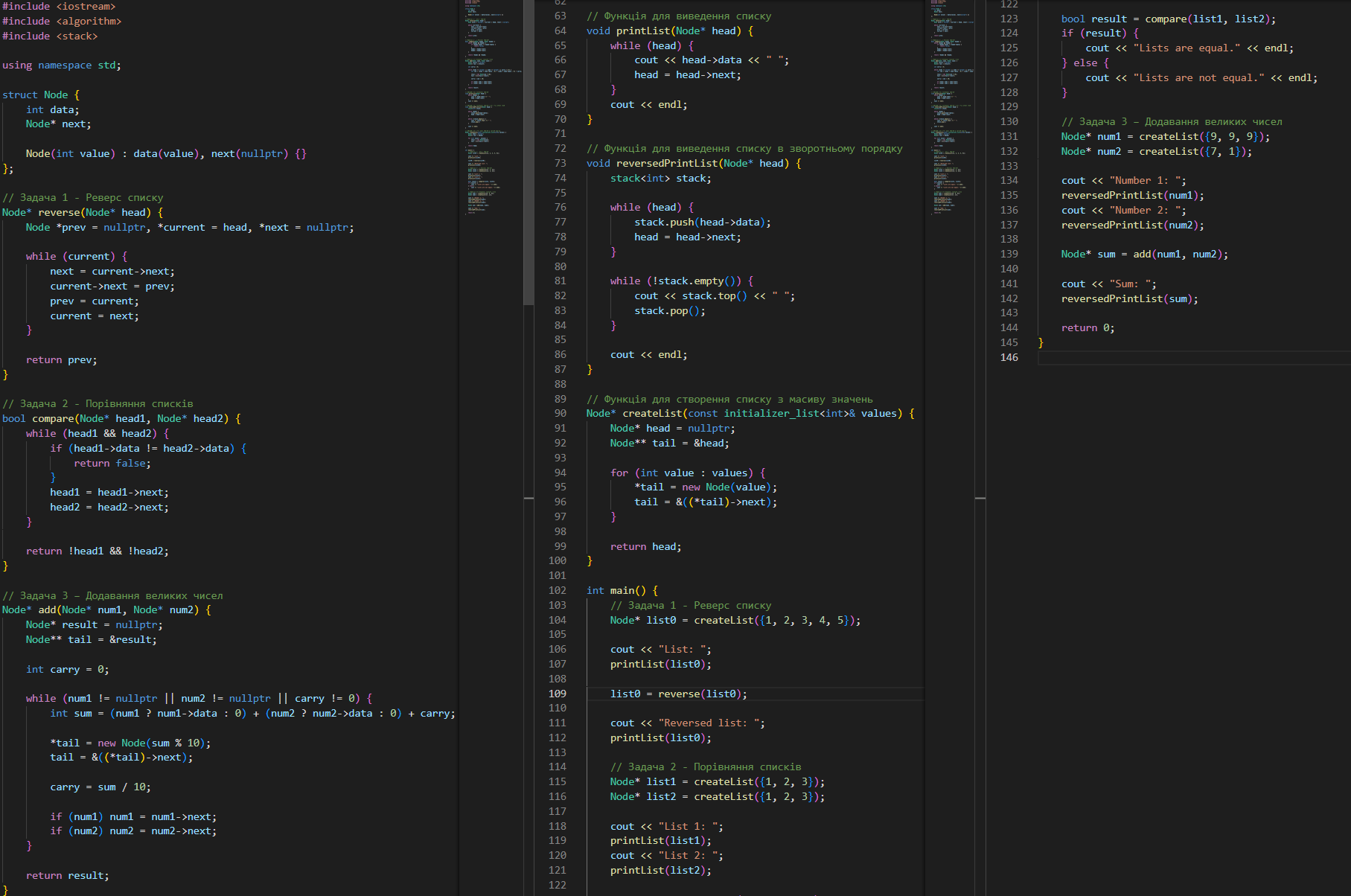
*Figure 3. №\_2 Algotester Lab 5.*

Завдання №\_3 Algotester Lab 7-8



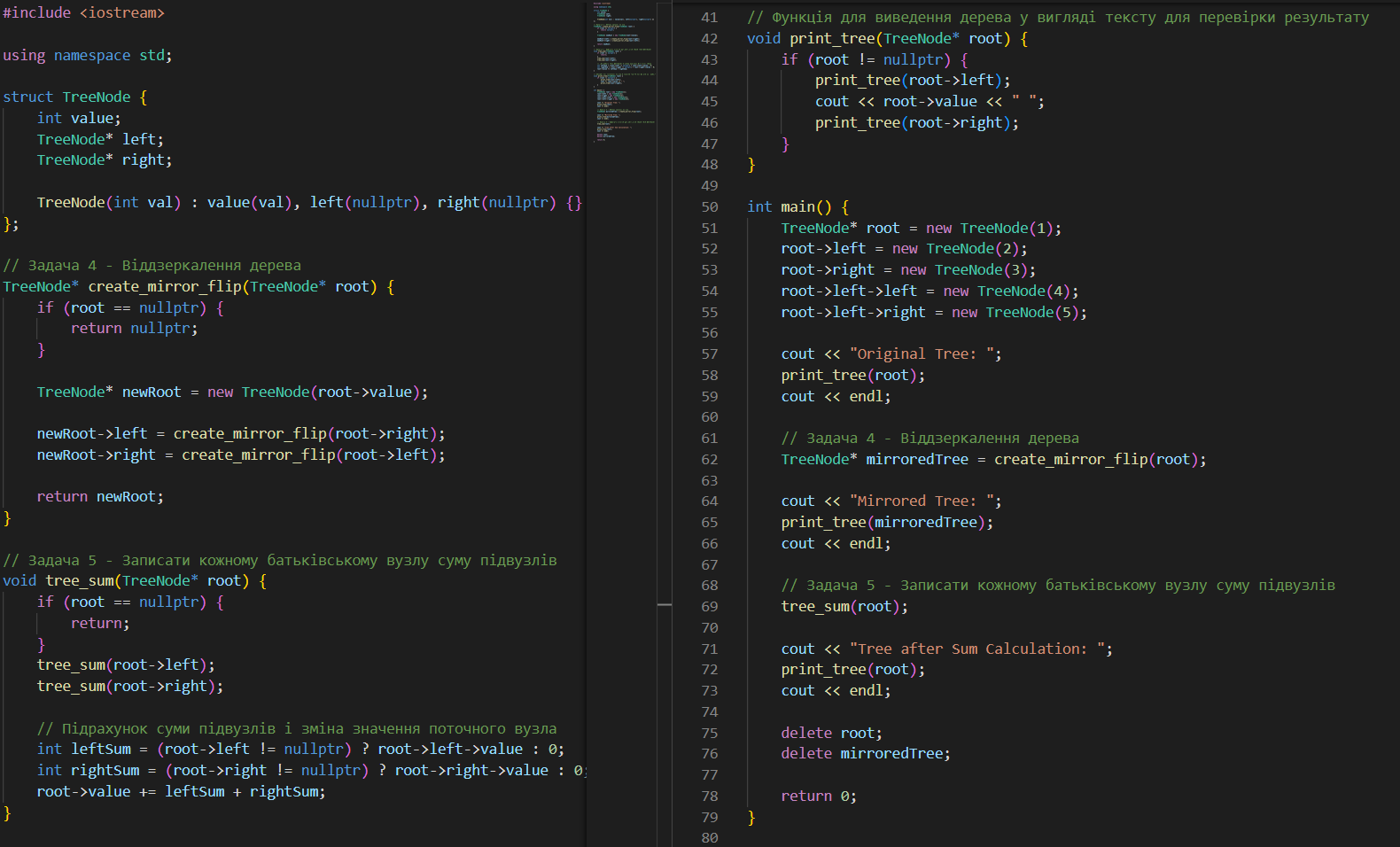
*Figure 4. №\_3 Algotester Lab 7-8.*

Завдання №\_4 Class Practice Task 1



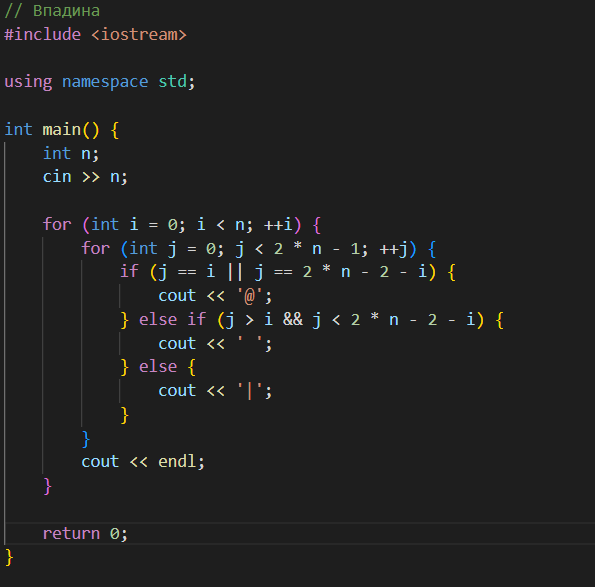
*Figure 5. №\_4 Class Practice Task 1.*

Завдання №\_5 Class Practice Task 2



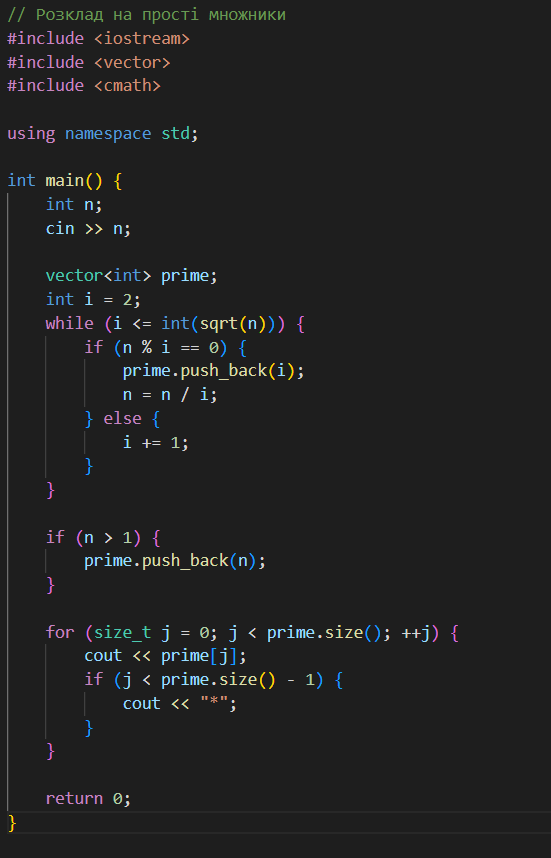
*Figure 6. №\_5 Class Practice Task 2.*

Завдання №\_6 Self Practice Task 1



*Figure 7. №\_6 Self Practice Task 1.*

Завдання №\_7 Self Practice Task 2



*Figure 8. №\_7 Self Practice Task 2.*

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

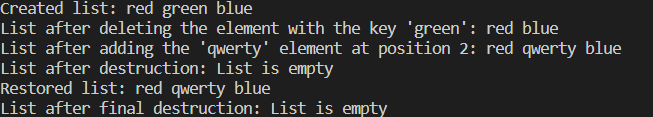
Завдання №\_1 VNS Lab 10 Task 1

* Варіант завдання: 11

Записи в лінійному списку містять ключове поле типу \*char(рядоксимволів). Сформувати двонаправлений список. Знищити з нього елемент із заданим ключем, додати елемент із зазначеним номером.

Порядок виконання роботи:

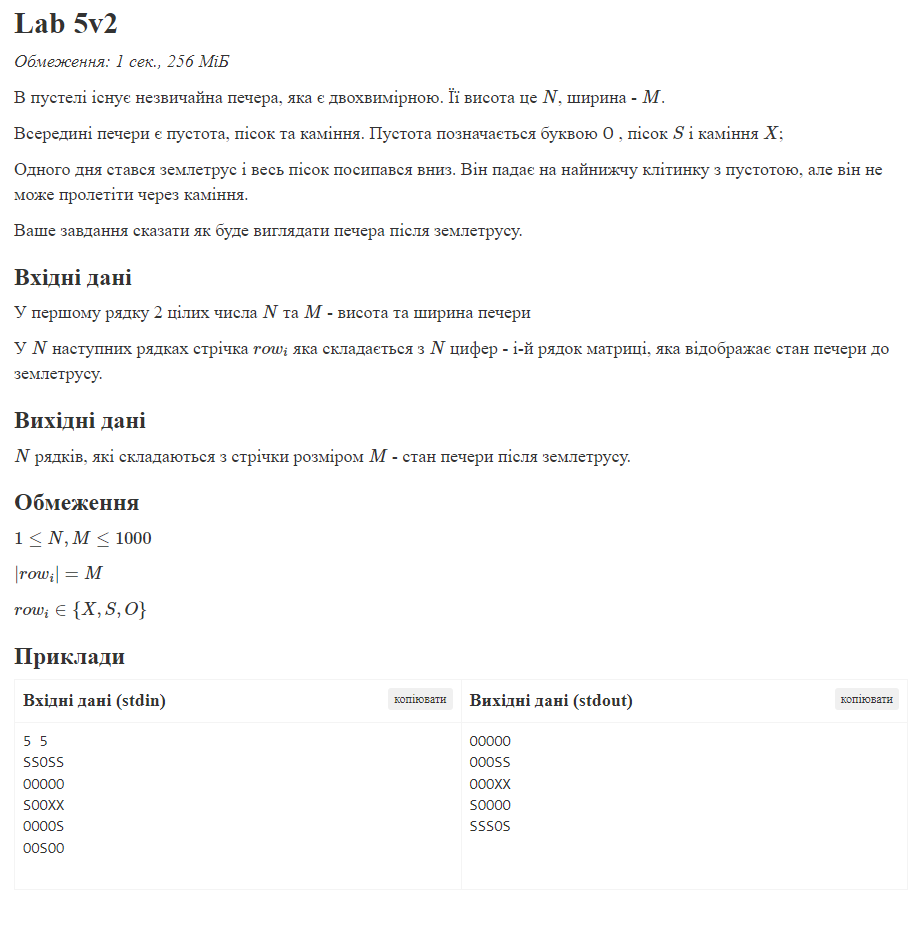
1. Написати функцію для створення списку. Функція може створювати порожній список, а потім додавати в нього елементи.
2. Написати функцію для друку списку. Функція повинна передбачати вивід повідомлення, якщо список порожній.
3. Написати функції для знищення й додавання елементів списку у відповідності зі своїм варіантом.
4. Виконати зміни в списку й друк списку після кожної зміни.
5. Написати функцію для запису списку у файл.
6. Написати функцію для знищення списку.
7. Записати список у файл, знищити його й виконати друк (при друці повинне бути видане повідомлення "Список порожній").
8. Написати функцію для відновлення списку з файлу.
9. Відновити список і роздрукувати його.
10. Знищити список.

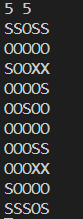


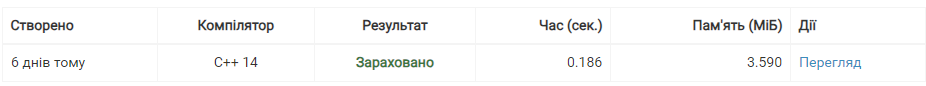
*Figure 9. Result №\_1 VNS Lab 10 Task 1.*

Час затрачений на виконання завдання : 10 хв.

Завдання №\_2 Algotester Lab 5



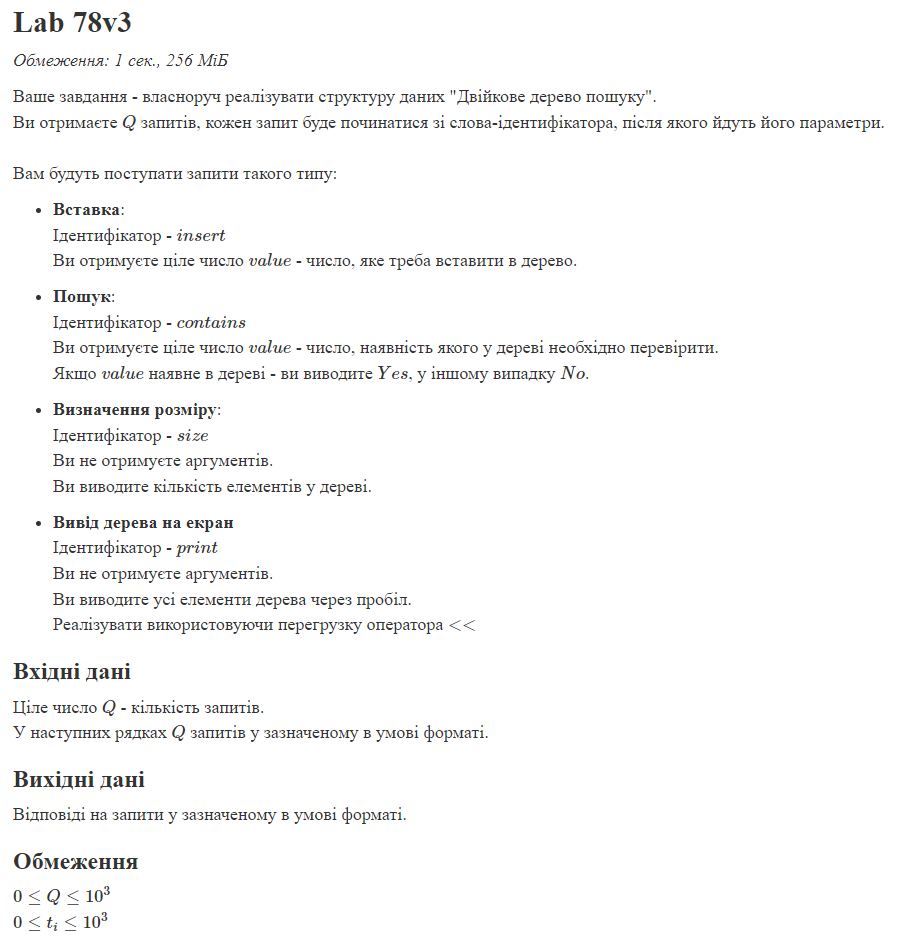


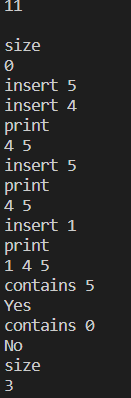


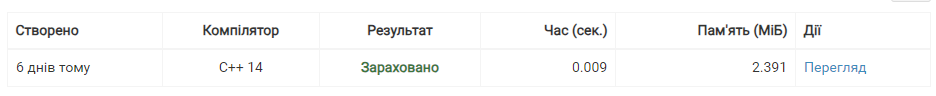
*Figure 10. Result №\_2 Algotester Lab 5.*

Час затрачений на виконання завдання : 15 хв.

Завдання №\_3 Algotester Lab 7-8





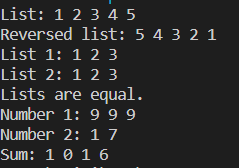


*Figure 11. Result №\_3 Algotester Lab 7-8.*

Час затрачений на виконання завдання : 20 хв.

Завдання №\_4 Class Practice Task 1

* Умови підзадачі 1:
* використовувати цілочисельні значення в списку;
* реалізувати метод реверсу;
* реалізувати допоміжний метод виведення вхідного і обернутого списків;
* Умови підзадачі 2:
* 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.
* Умови підзадачі 3:
* 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.



*Figure 12. Result №\_4 Class Practice Task 1.*

Час затрачений на виконання завдання : 15 хв.

Завдання №\_5 Class Practice Task 2

* Умови підзадачі 4:
* 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
* Умови під задачі 5:
* 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

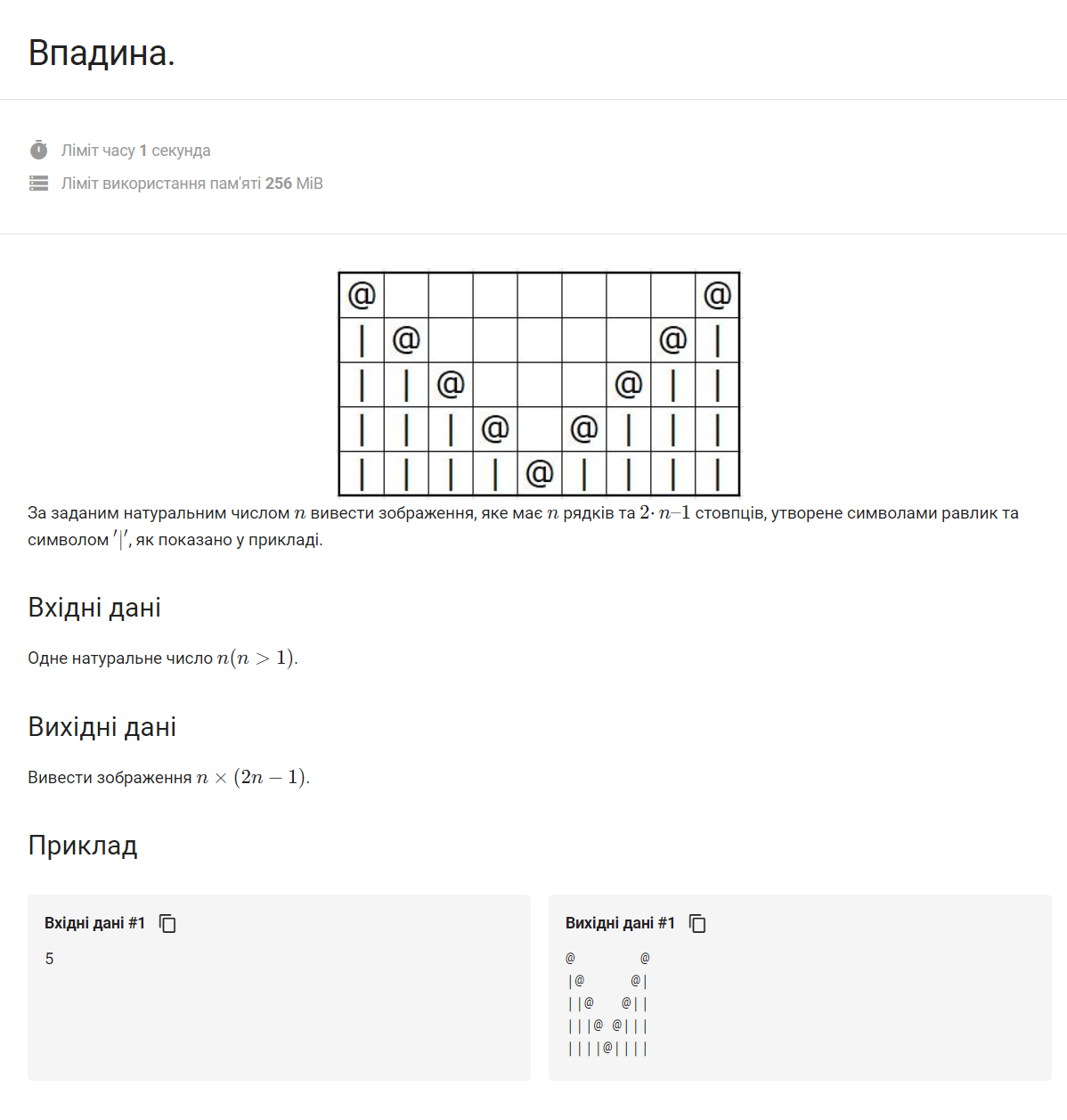


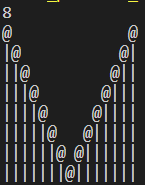
*Figure 13. Result №\_5 Class Practice Task 2.*

Час затрачений на виконання завдання : 15 хв.

Завдання №\_6 Self Practice Task 1

* Задача Впадина







*Figure 14. Result №\_6 Self Practice Task 1.*

Час затрачений на виконання завдання : 15 хв.

Завдання №\_7 Self Practice Task 2

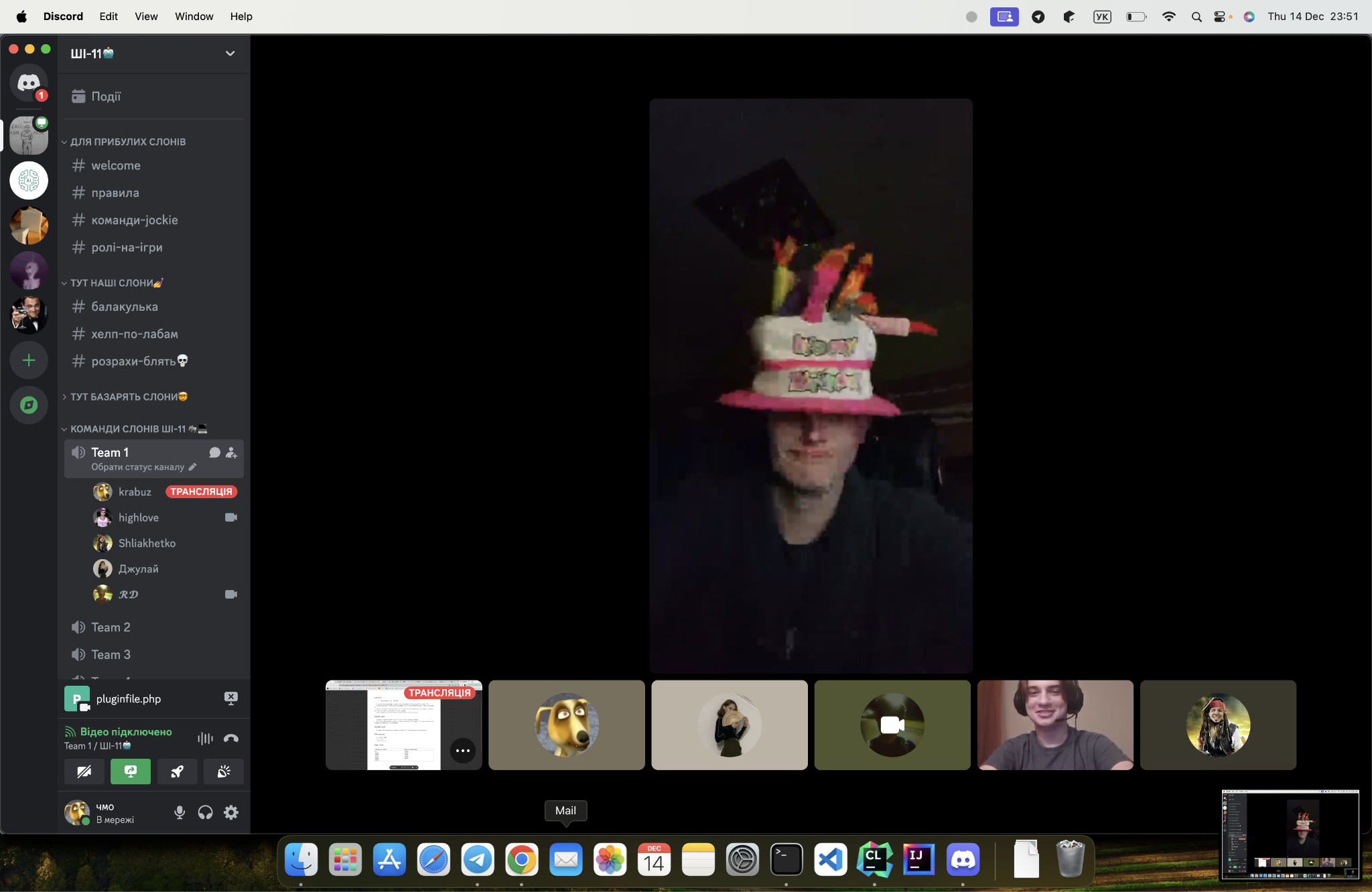
* Задача Розклад на прості множники


*Figure 15. Result №\_7 Self Practice Task 2.*

Час затрачений на виконання завдання : 20 хв.

## **6. Кооперація з командою:**

* Мали оффлайн зустріч 12.1 де обговорили VNS LABS ;
* Мали оффлайн зустріч 12.4 де обговорили Сlass Practice, та Algotester Labs;
* Мали онлайн зустріч 12.5 де поділилися досвідом написання програм.  
  

# **Висновки:**

Отримав практичний досвід роботи з динамічними структурами, реалізував зв’язний список та алгоритми обробки дерев. Отримані навички покращили розуміння алгоритмів та їхню практичну реалізацію в програмах. Процес вирішення завдань розвинув логічне мислення та допоміг оптимізувати код. Практичні задачі, про суму великих чисел за допомогою зв’язних списків, задача про печеру, та алгоритми обробки дерев, тощо - дозволили застосувати теоретичні знання в реальних сценаріях програмування.