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 № 4**

На тему:  «Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.»

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

до:

ВНС Лабораторної Роботи №4

ВНС Лабораторної Роботи №5

Алготестер Лабораторної Роботи №2

Алготестер Лабораторної Роботи №3

Практичних Робіт до блоку №4

**Виконав:**

Студент групи ШІ-11

Саух Богдан Володимирович

Львів 2024

**Тема роботи:** Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами

**Мета роботи:** Навчитися ефективно працювати з одновимірними та двовимірними масивами, використовувати вказівники та посилання для оптимізації роботи з динамічними масивами, розібратися зі структурами даних і вкладеними структурами для зберігання складних даних, а також освоїти алгоритми обробки та маніпуляції масивами та структурами.

**Теоретичні відомості:**

**1)Перелік тем:**

1. Класи пам'яті у C++
2. Вступ до Масивів і Вказівників
3. Одновимірні Масиви
4. Вказівники та Посилання
5. Двовимірні Масиви
6. Динамічні Масиви
7. Структури Даних
8. Вкладені Структури
9. Використання структур
10. Алгоритми обробки та робота з масивами та структурами

**2)Індивідуальний план опрацювання теорії:**

1. **Класи пам'яті у C++**
   * [C++ Storage Classes](https://www.tutorialspoint.com/cplusplus/cpp_storage_classes.htm)
   * [Memory Classes in C++](https://www.geeksforgeeks.org/storage-classes-in-c/)
2. **Вступ до Масивів і Вказівників**
   * [Introduction to Arrays in C++](https://www.w3schools.com/cpp/cpp_arrays.asp)
   * [Pointers in C++](https://www.cplusplus.com/doc/tutorial/pointers/)
3. **Одновимірні Масиви**
   * [C++ One-Dimensional Arrays](https://www.w3schools.com/cpp/cpp_arrays.asp)
   * [Array Basics in C++](https://www.geeksforgeeks.org/arrays-in-c-cpp/)
4. **Вказівники та Посилання**
   * [Pointers and References in C++](https://www.cplusplus.com/doc/tutorial/pointers/)
   * [References in C++](https://www.geeksforgeeks.org/references-in-c/)
5. **Двовимірні Масиви**
   * [Two-Dimensional Arrays in C++](https://www.tutorialspoint.com/cplusplus/cpp_multi_dimensional_arrays.htm)
   * [2D Arrays in C++](https://www.geeksforgeeks.org/multidimensional-arrays-c-cpp/)
6. **Динамічні Масиви**
   * [Dynamic Arrays in C++](https://www.geeksforgeeks.org/dynamic-arrays-in-c-cpp/)
   * [C++ Dynamic Memory](https://www.tutorialspoint.com/cplusplus/cpp_dynamic_memory.htm)
7. **Структури Даних**
   * [Structures in C++](https://www.w3schools.com/cpp/cpp_structs.asp)
   * [Introduction to Data Structures](https://www.geeksforgeeks.org/data-structures/)
8. **Вкладені Структури**
   * [Nested Structures in C++](https://www.geeksforgeeks.org/nested-structures-in-c/)
   * [C++ Nested Structures](https://www.tutorialspoint.com/structure-within-structure-in-cplusplus)
9. **Використання структур**
   * [C++ Structs and their Use](https://www.w3schools.com/cpp/cpp_structs.asp)
   * [Uses of Structures in C++](https://www.geeksforgeeks.org/structures-in-cpp/)
10. **Алгоритми обробки та робота з масивами та структурами**
    * [Algorithms for Arrays in C++](https://www.geeksforgeeks.org/array-data-structure/)
    * [Working with Arrays and Structures in C++](https://www.tutorialspoint.com/cplusplus/cpp_structures.htm)

**Виконання роботи:**

**1)Перелік завдань:**

* John Black - Epic 4 Task 1 - Theory Education Activities
* John Black - Epic 4 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-8)
* John Black - Epic 4 Task 3 - Lab# programming: VNS Lab 4(варіант 22)
* John Black - Epic 4 Task 4 - Lab# programming: VNS Lab 5(варіант 22)
* John Black - Epic 4 Task 5 - Lab# programming: Algotester Lab 2(варіант 3)
* John Black - Epic 4 Task 6 - Lab# programming: Algotester Lab 3(варіант 3)
* John Black - Epic 4 Task 7 - Practice# programming: Class Practice Task
* John Black - Epic 4 Task 8  - Practice# programming:  Self Practice Task
* John Black - Epic 4 Task 9  - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)
* John Black - Epic 4 Task 10 - Results Evaluation and Release

**2)Умови завдань:**

**Task 3:**

Реалізувати з використанням масиву стек (перший прийшов, останній пішов), для чого організувати додавання, знищення елементів з масиву і друк масиву після кожної операції.

**Task4: Використовуючи функції, розв’язати зазначене у варіанті завдання. Масив повинен передаватися у функцію як параметр.**

У двовимірному масиві знайти середнє арифметичне першого стовпця й кількість елементів у кожному з наступних стовпців, що перевищують середнє арифметичне попереднього стовпця.

**Task5:**

Вам дано масив цілих чисел розміром N, на першій та останній клітинці розміщено по дрону.

Вони одночасно взлітають.

На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться.

Тобто лівий дрон у першу секунду з клітинки з індексом 1 перелетить у клітинку з індексом a1, тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону.

Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якшо 2 дрони опиняються в одній клітинці - ви виводите **Collision**.

Якщо лівий дрон опиниться справа від правого - це **Miss**

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках ai та ai+1 - виведіть **Stopped**

Врахуйте, що перевіряти треба також до взльоту.

# **Task6:**

Вам дана стрічка s.

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

# **Task7: Перевірка чи слово або число є паліндромом**

# **Задача**

## Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

## Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

## **Мета Задачі**

Навчитися користуватися механізмами перевантаження функції та використовувати рекурсію для вирішення задач обчислення.

### Вимоги:

1. Визначення функції:

a.Реалізуйте рекурсивну функцію *isPalindrome*, яка перевіряє, чи заданий рядок є паліндромом.

1. Приклад визначення функції:

*a. bool isPalindrome(const string& str, int start, int end);*

1. Перевантаження функцій:

a. Перевантажте функцію *isPalindrome* для роботи з цілими значеннями.

b. *bool isPalindrome*(ціле число);

1. Рекурсія:

a.Рекурсивна функція для рядків перевірить символи в поточній початковій і кінцевій позиціях. Якщо вони збігаються, він буде рекурсивно перевіряти наступні позиції, поки початок не перевищить кінець, після чого рядок буде визначено як паліндром.

### Кроки реалізації

* Визначте та реалізуйте рекурсивну функцію isPalindrome для рядків.
* Визначте та реалізуйте перевантажену функцію isPalindrome для цілих чисел. Використати математичний підхід щоб перевірити чи число є паліндромом.

**Task 8:**

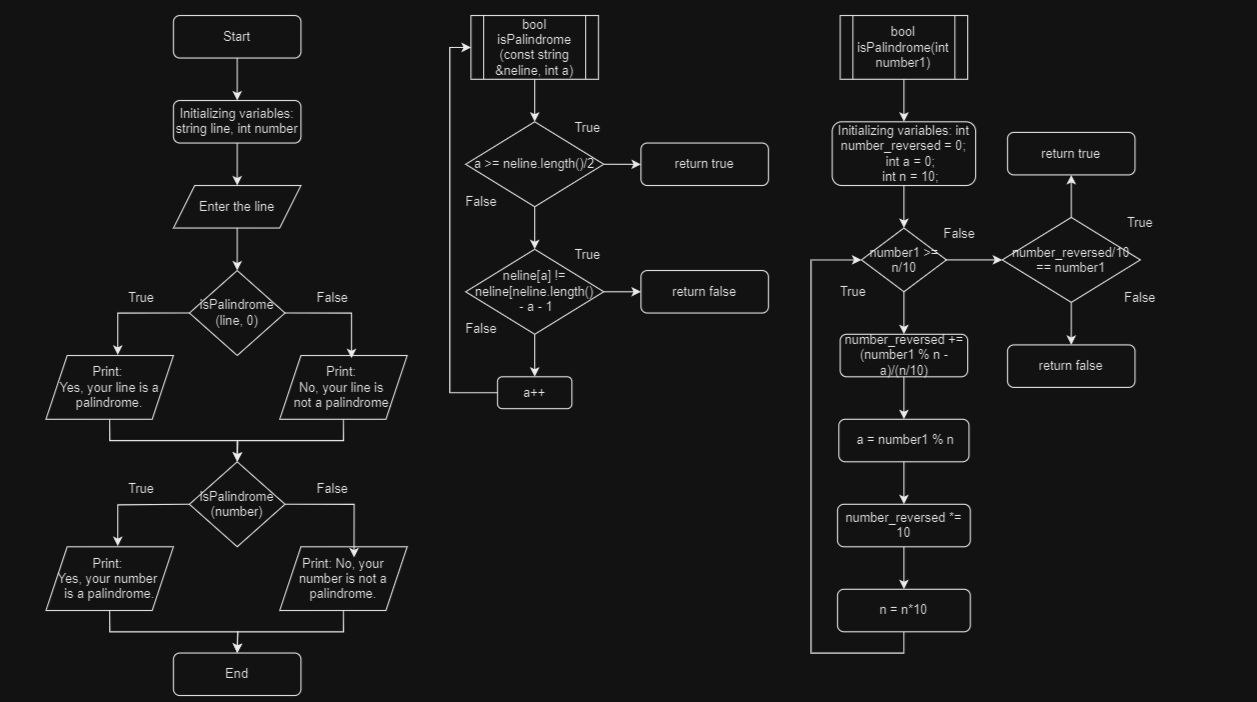
Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елементу.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

**3)Дизайн та планова оцінка часу виконання завдань:**

Task 7 - Practice# programming: Class Practice Task

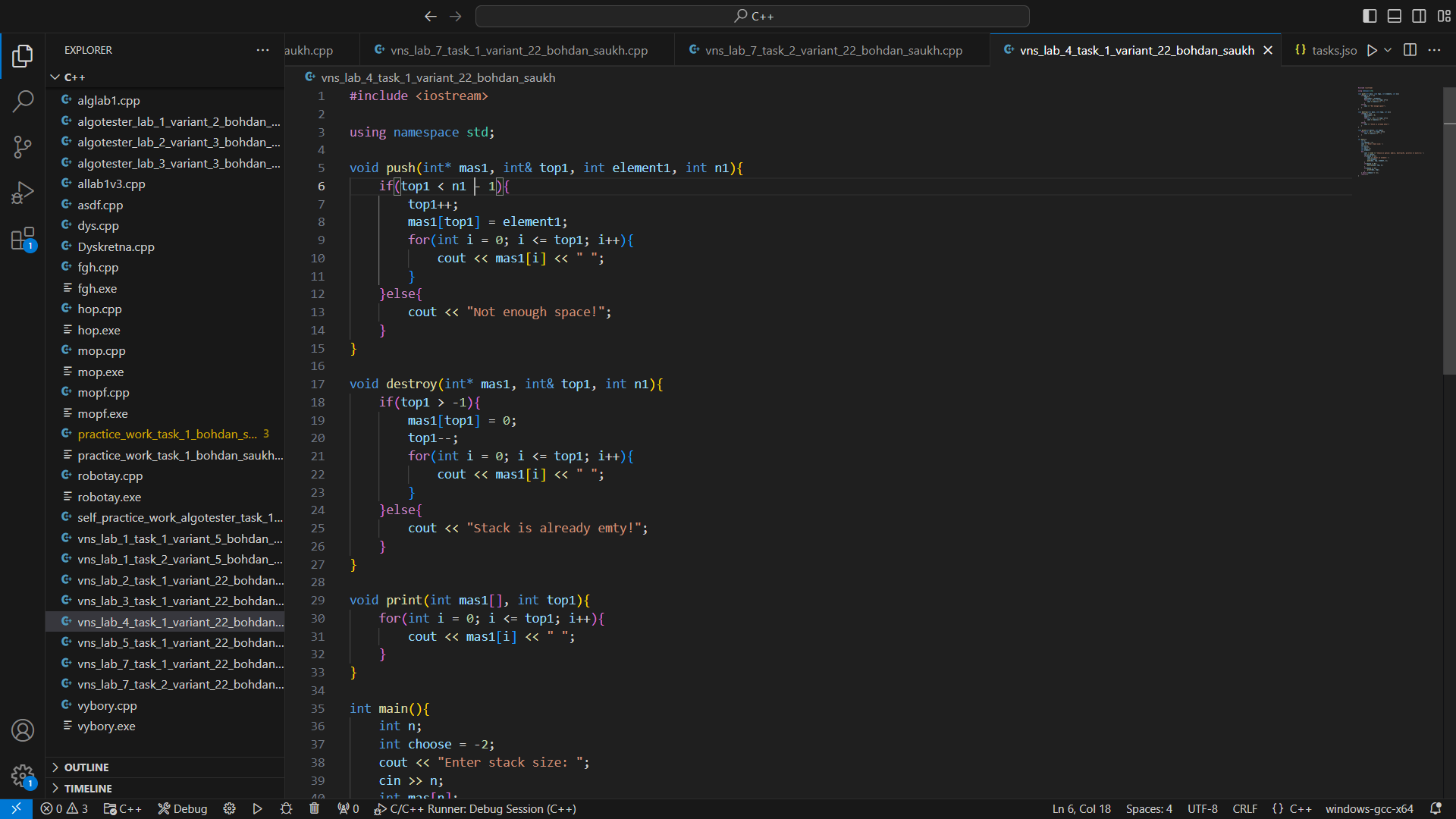


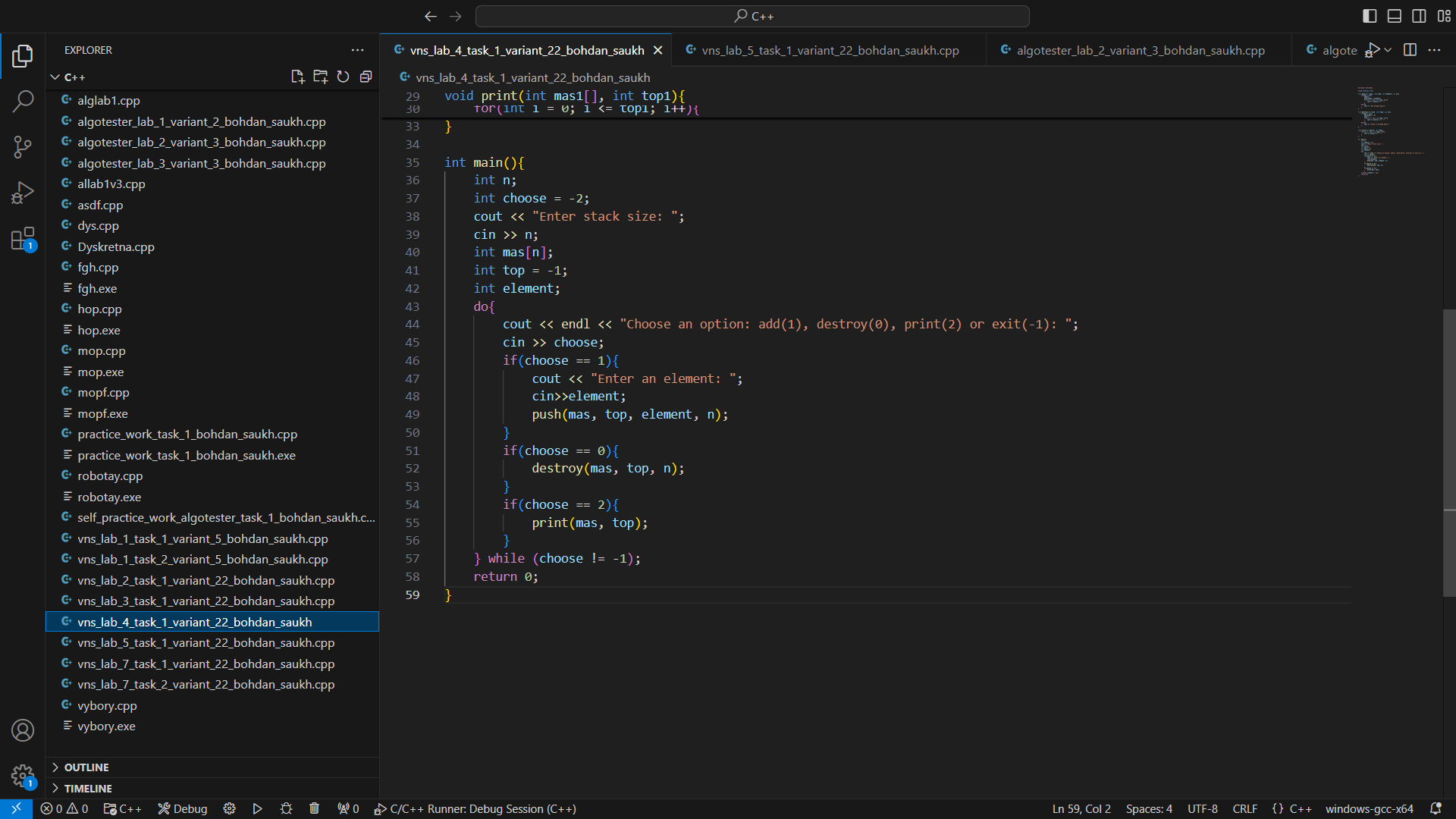
Орієнтовний час виконання: 45хв

**4)Код програм з посиланням на зовнішні ресурси:**

Task 3 - Lab# programming: VNS Lab 4(варіант 22)

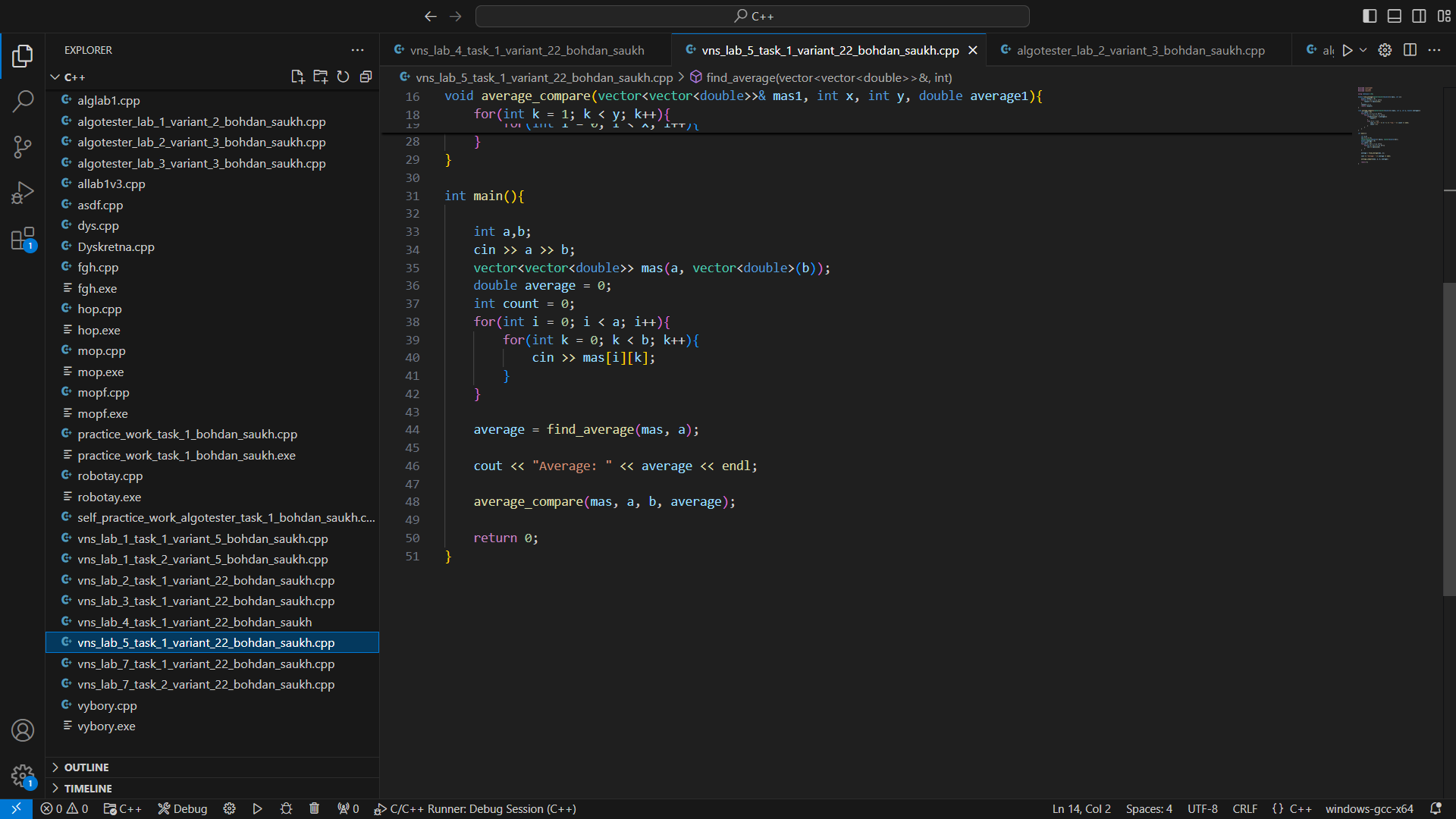
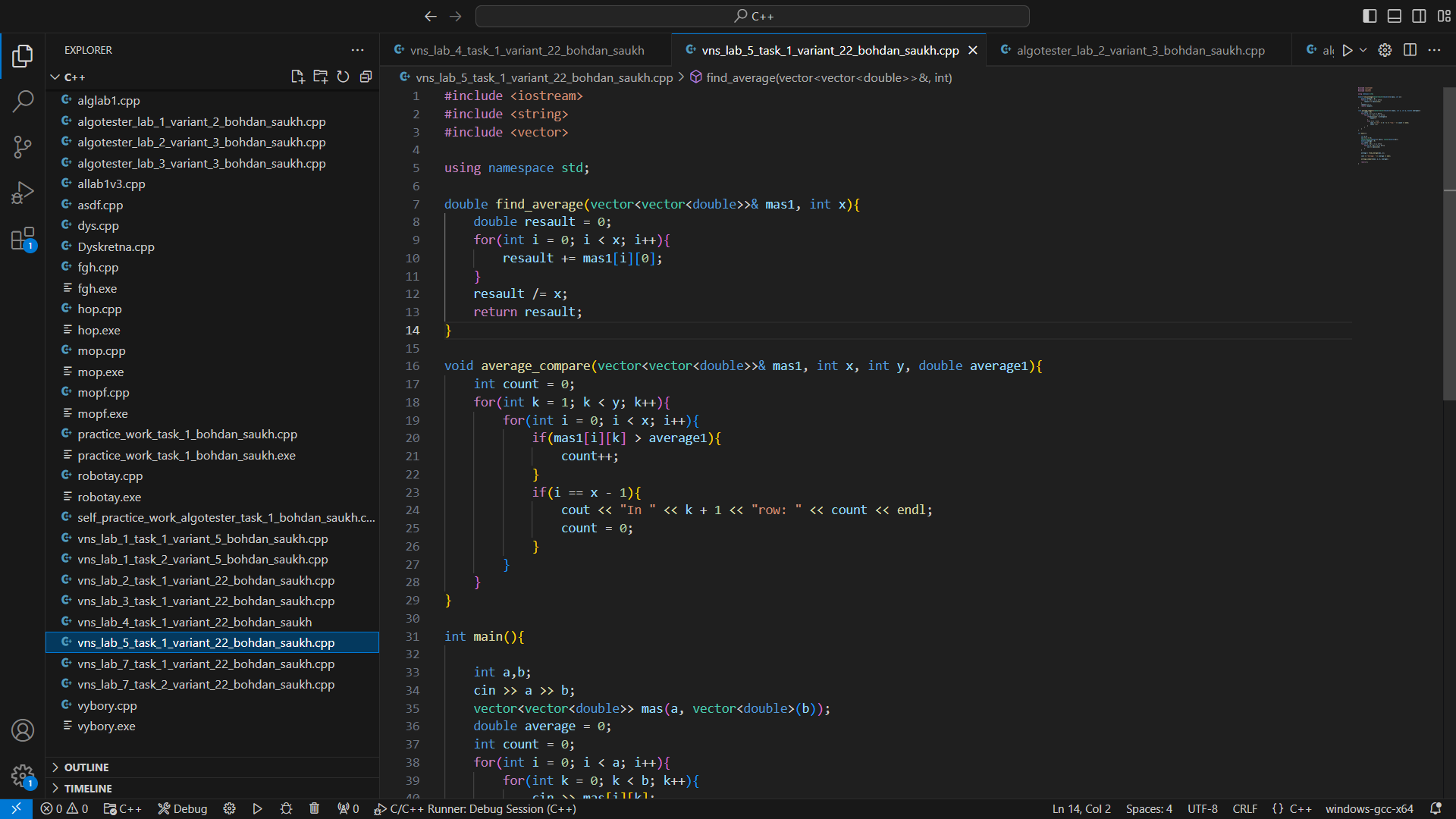
Посилання на файл програми:





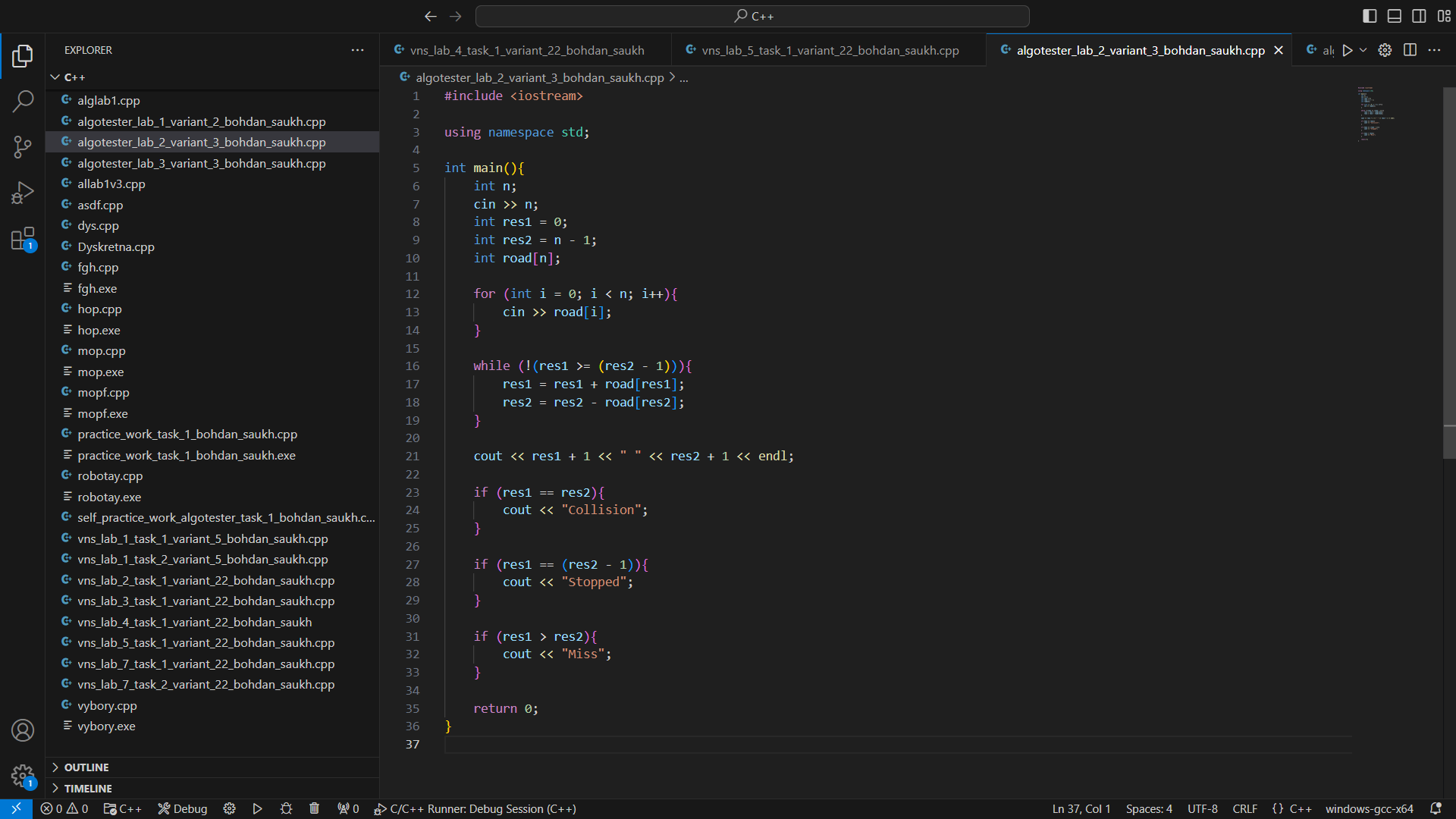
Task 4 - Lab# programming: VNS Lab 5(варіант 22)

Посилання на файл програми:



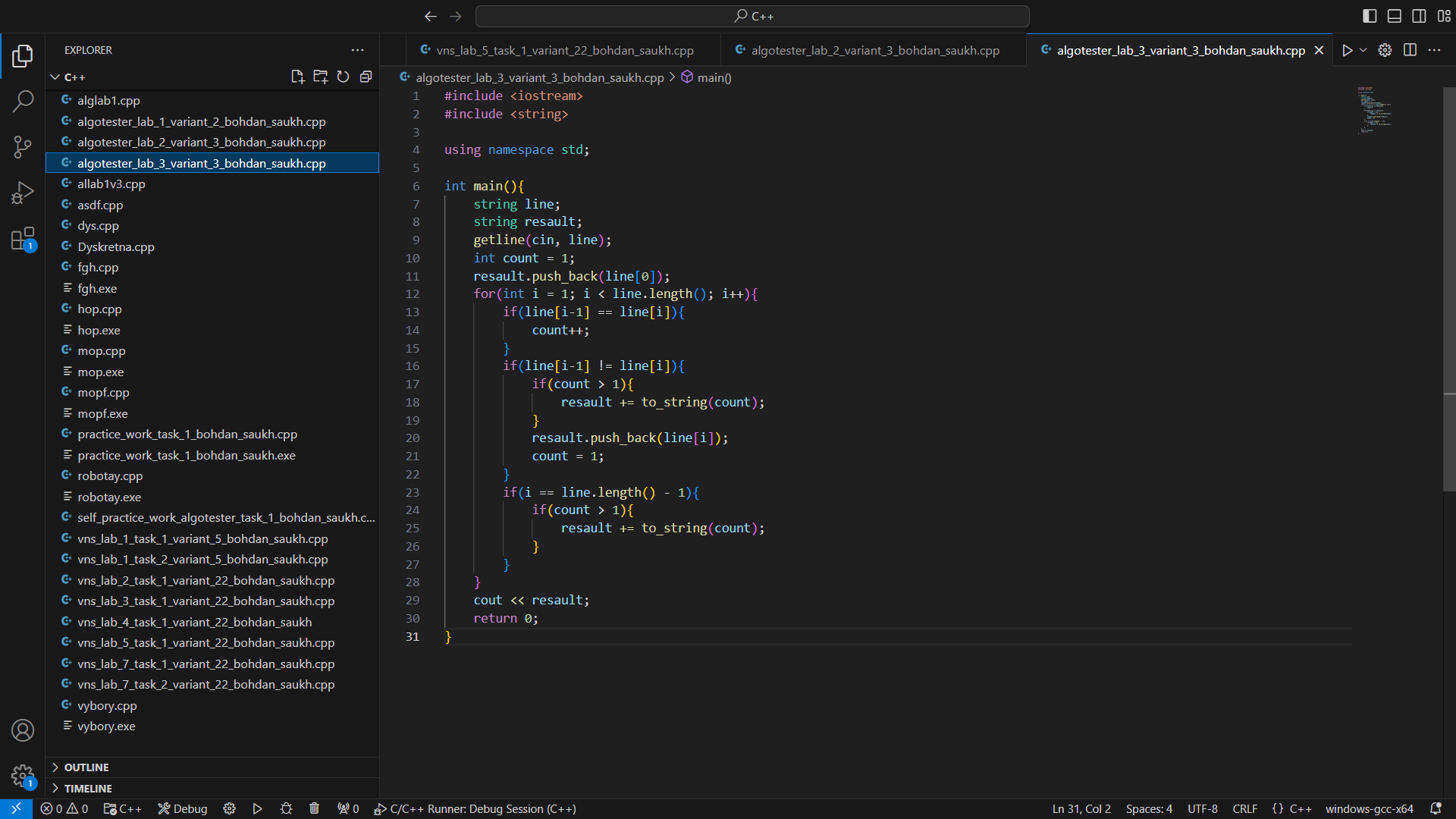
Task 5 - Lab# programming: Algotester Lab 2(варіант 3)

Посилання на файл програми:



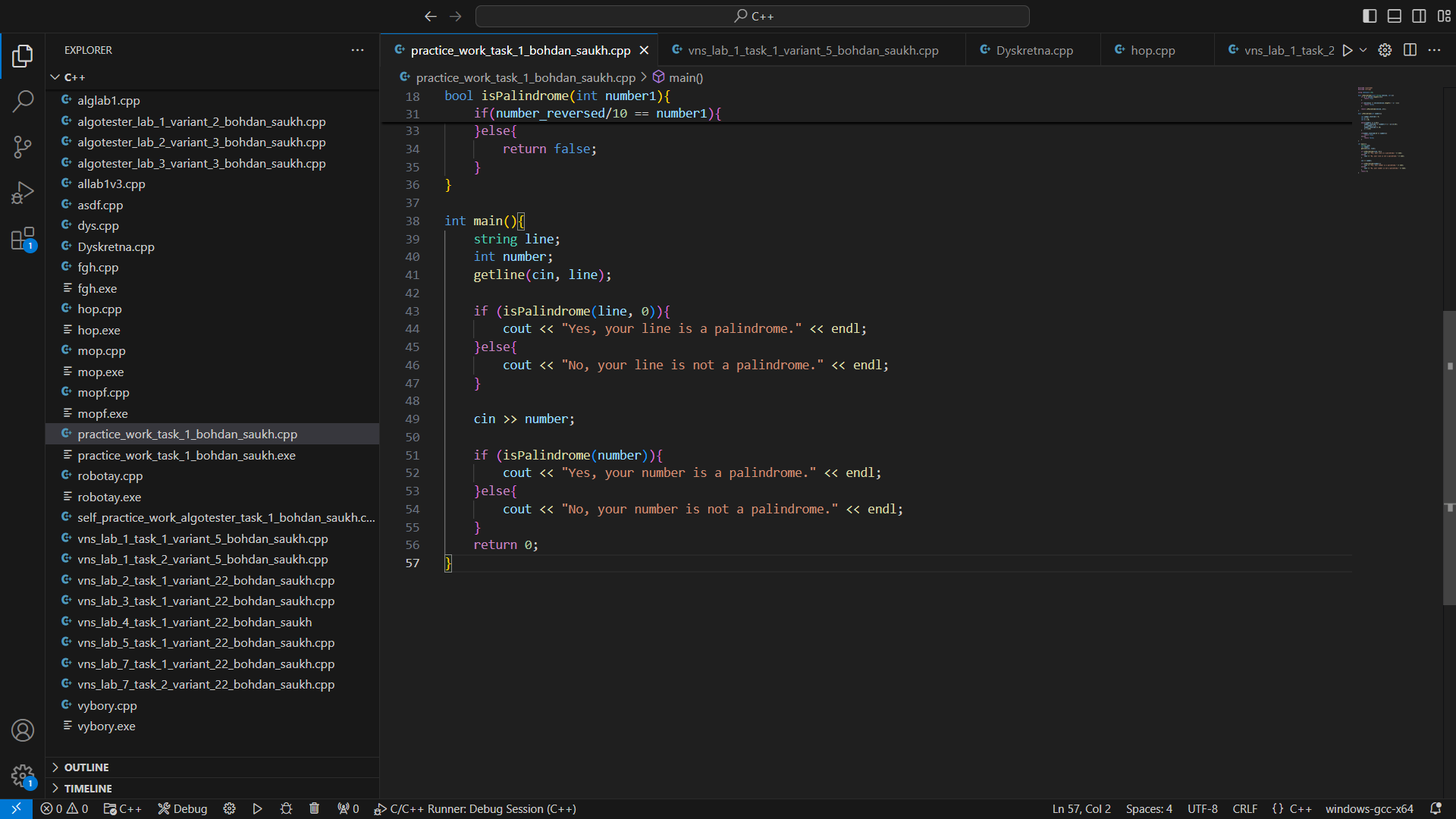
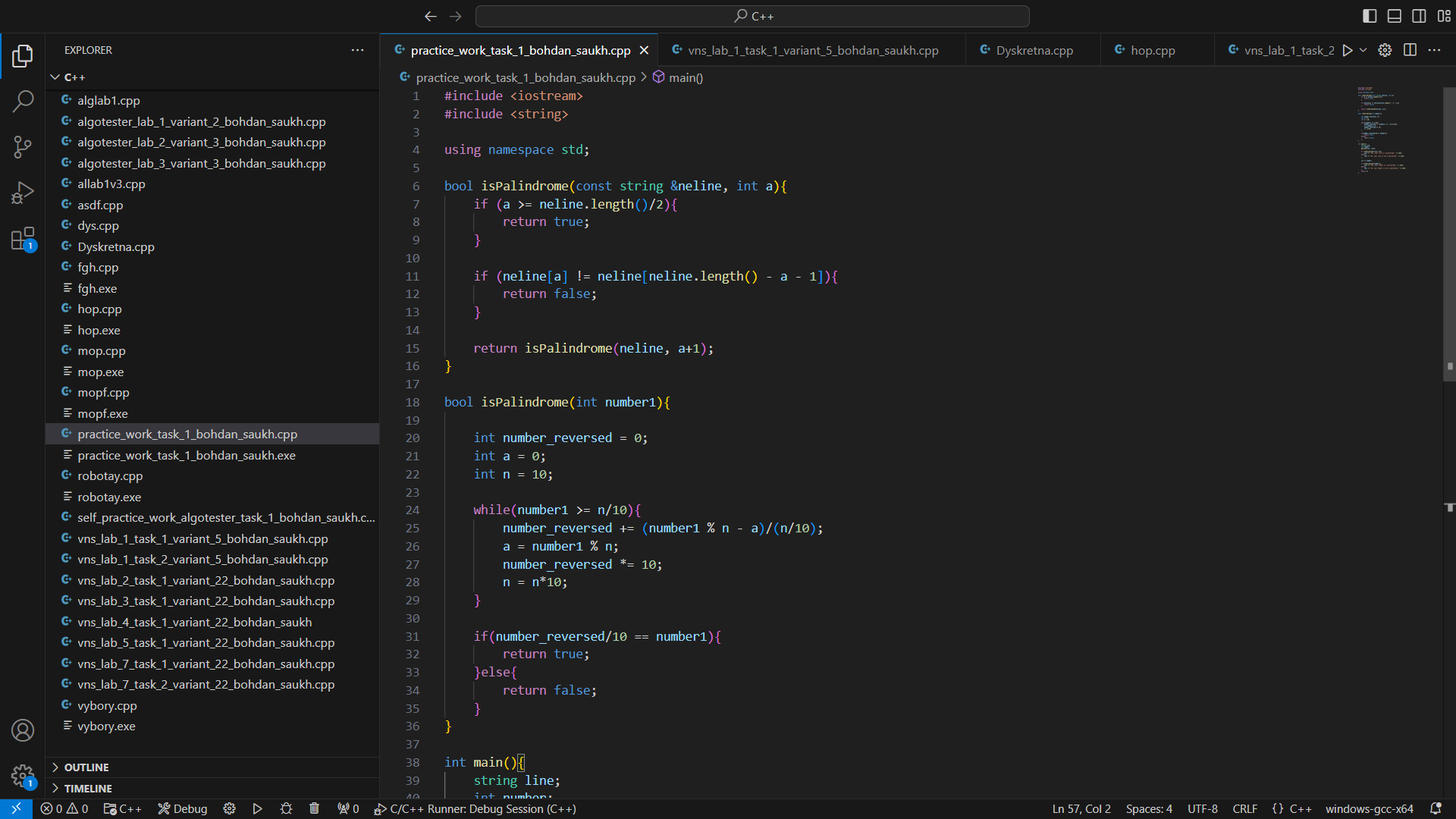
Task 6 - Lab# programming: Algotester Lab 3(варіант 3)

Посилання на файл програми:



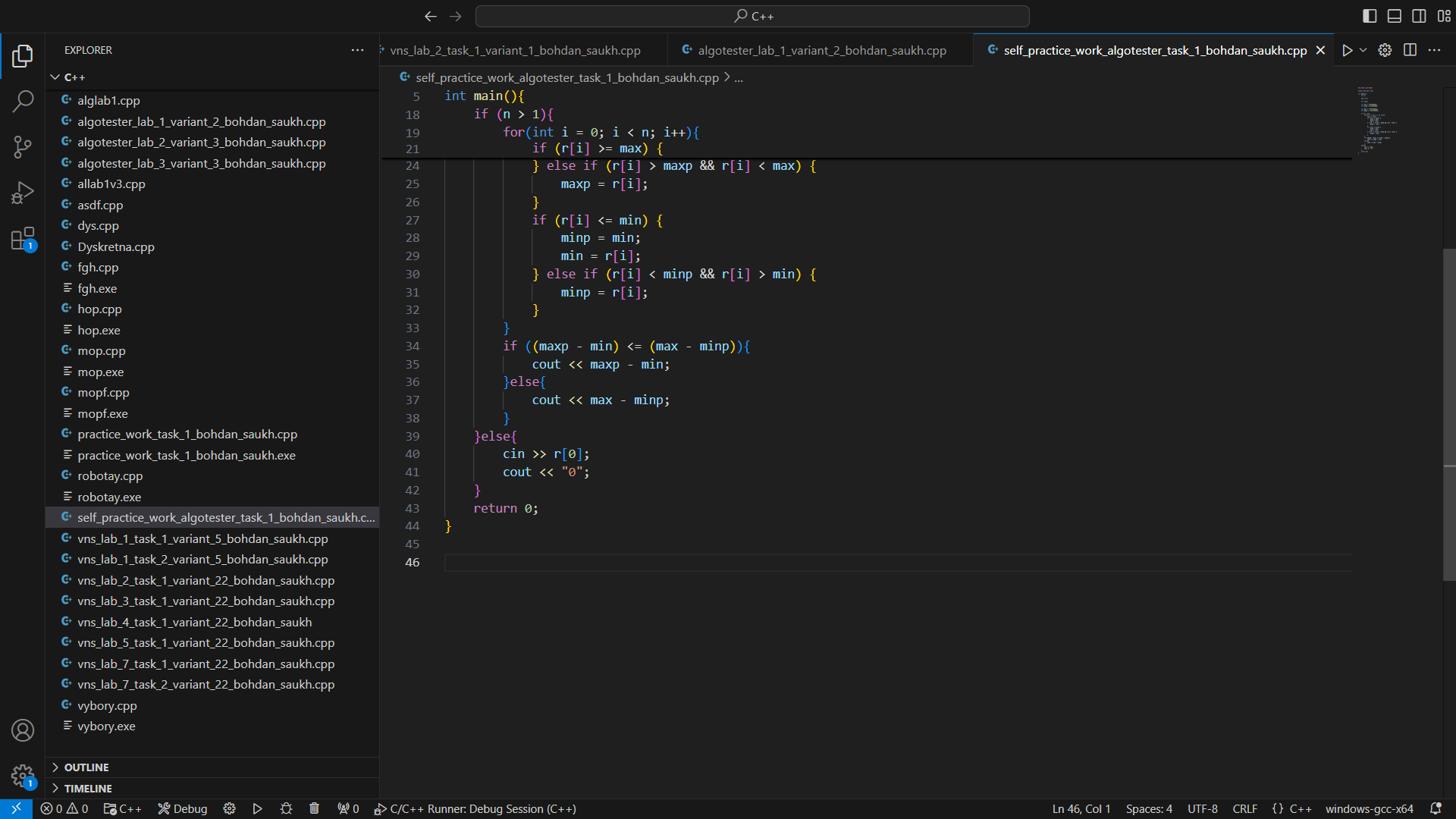
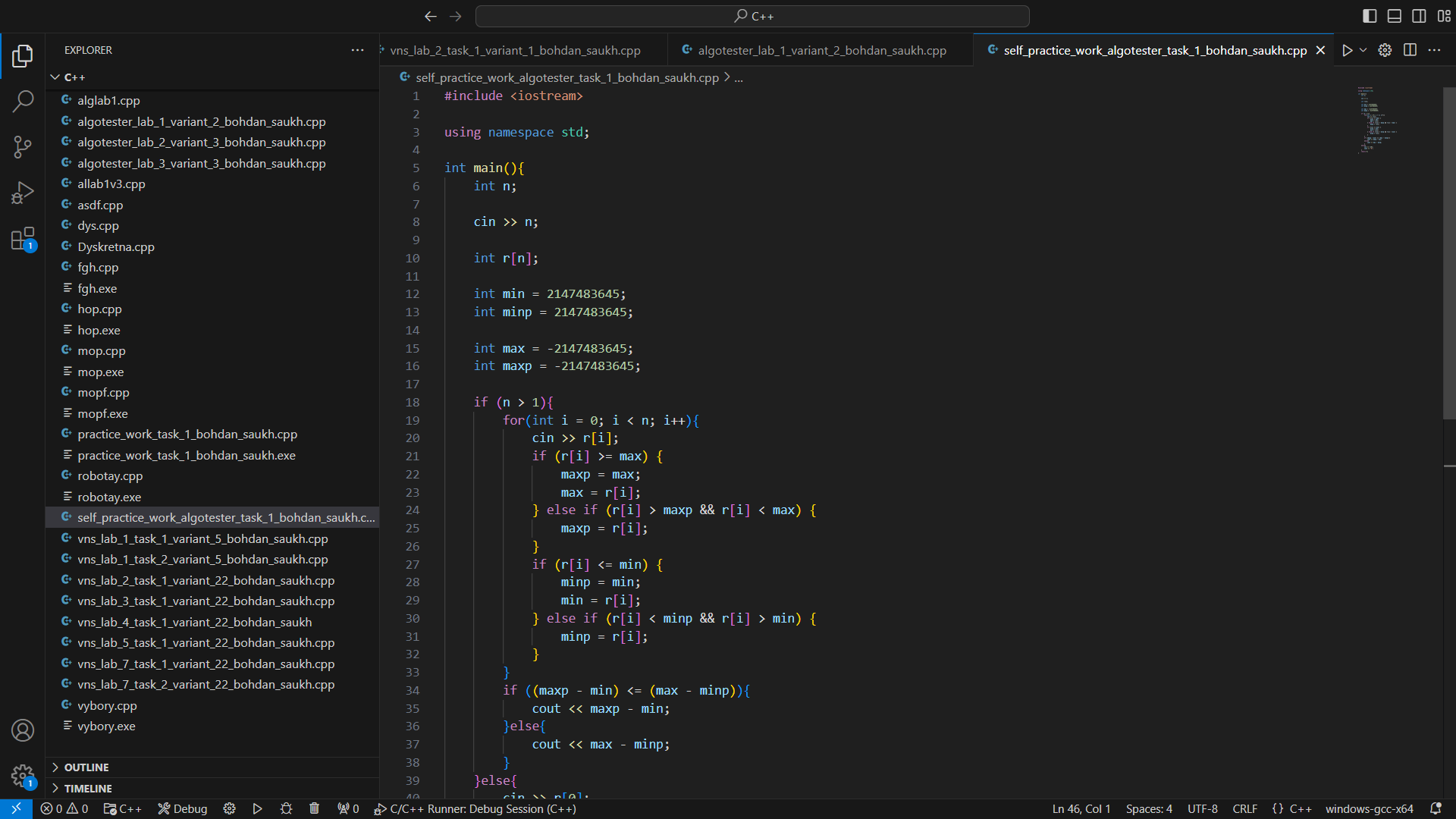
Task 7 - Practice# programming: Class Practice Task

Посилання на файл програми:



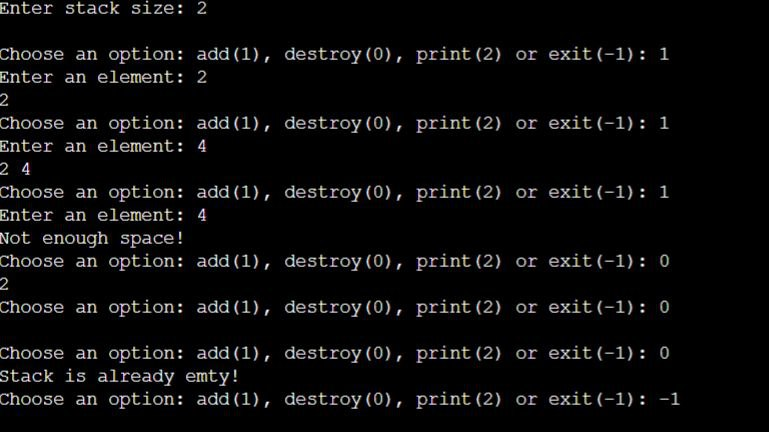
Task 8  - Practice# programming:  Self Practice Task

Посилання на файл програми:



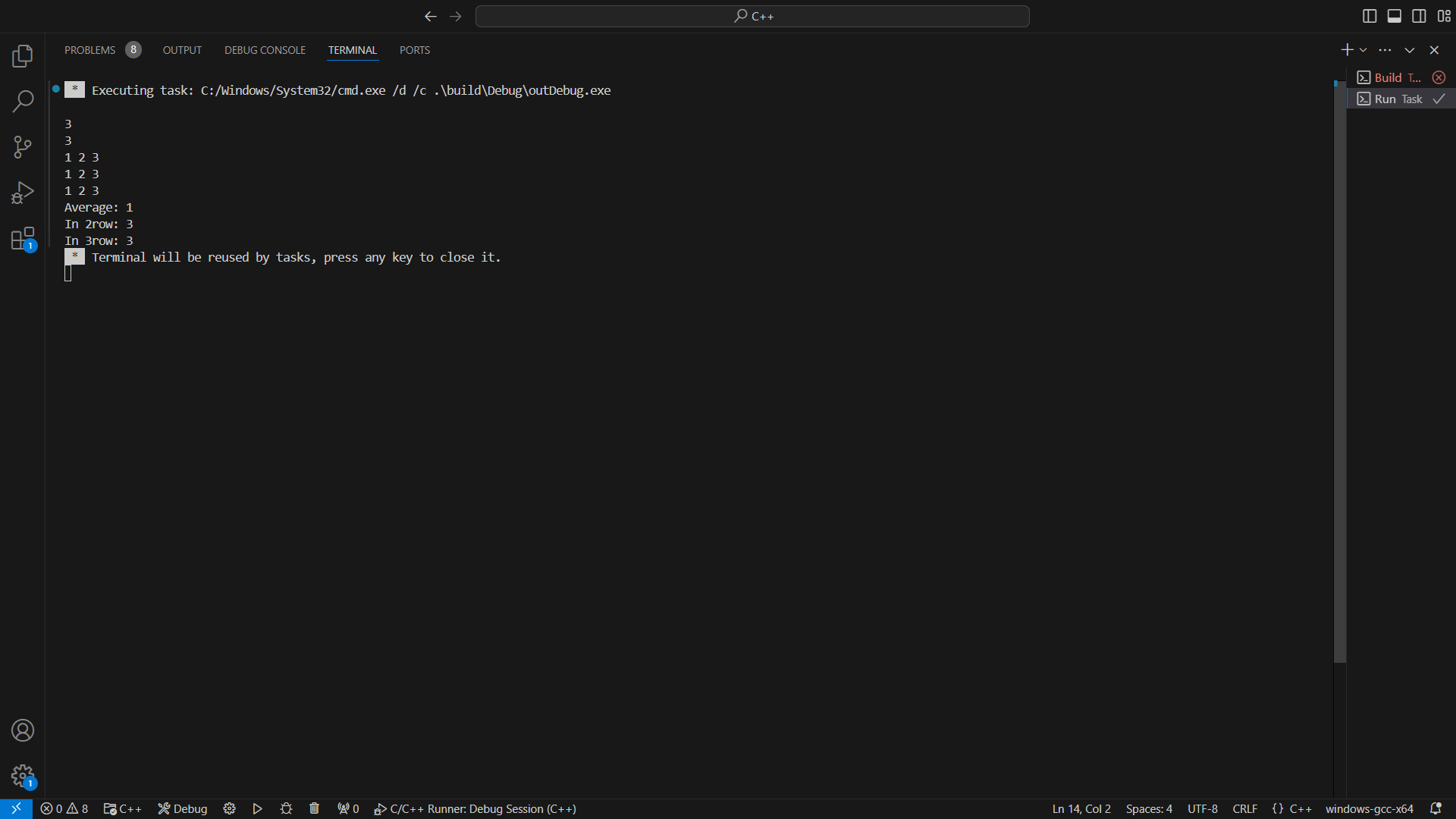
**5)Результати виконання завдань та фактично затрачений час**

Task 3 - Lab# programming: VNS Lab 4(варіант 22)



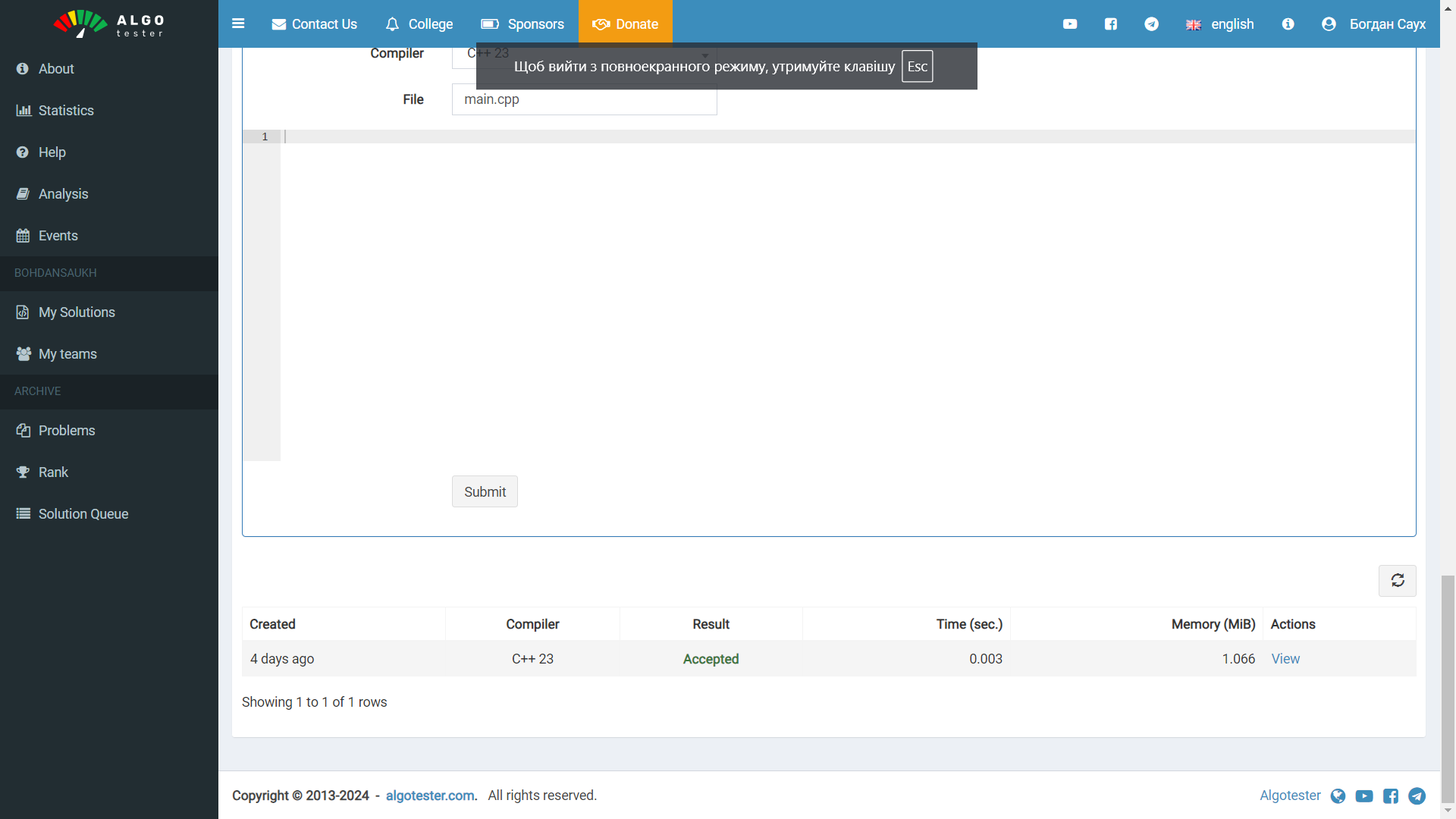
Фактично затрачений час: 1 год

Task 4 - Lab# programming: VNS Lab 5(варіант 22)

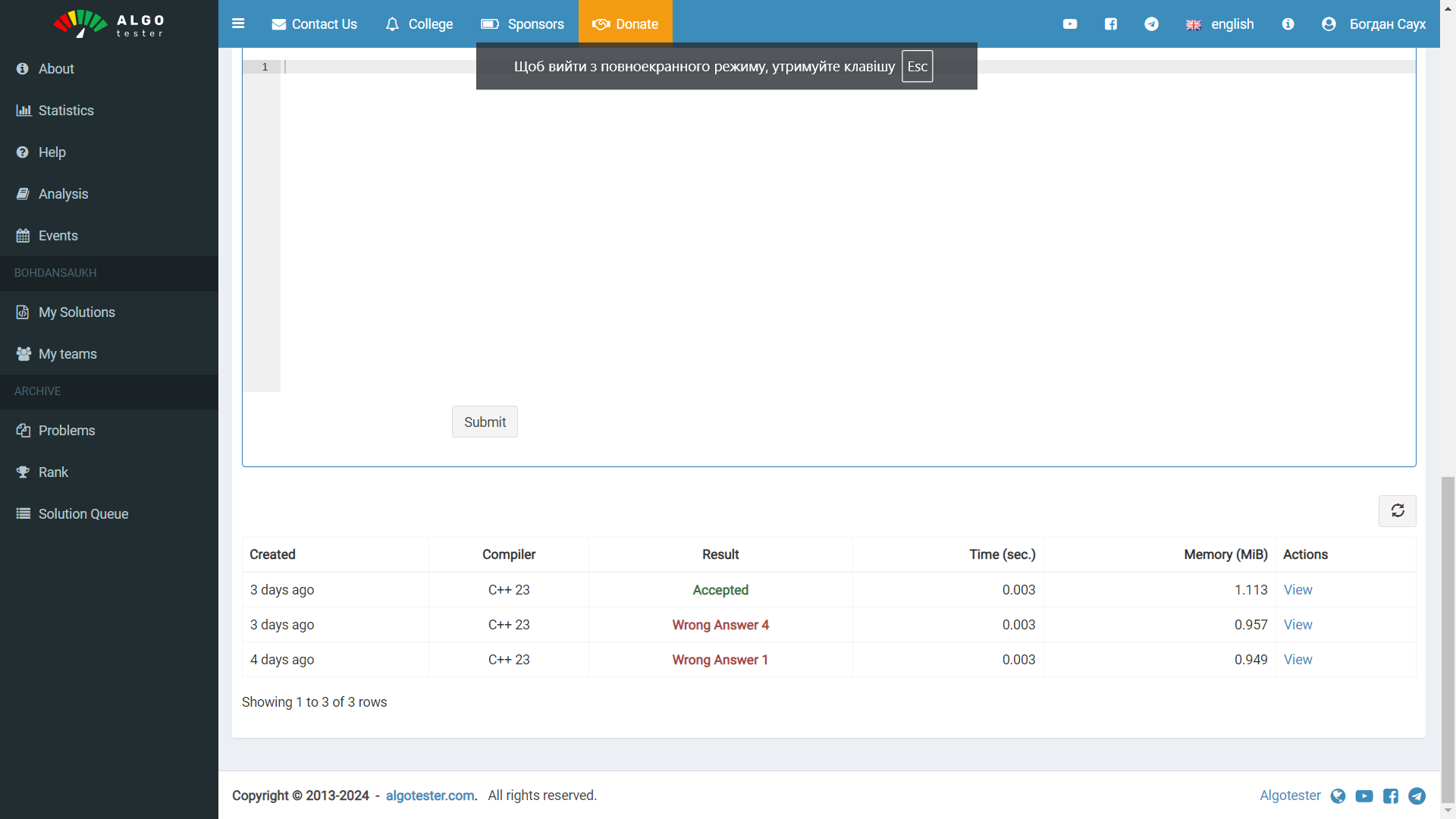


Фактичний час затрачений на виконання: 50хв

Task 5 - Lab# programming: Algotester Lab 2(варіант 3)

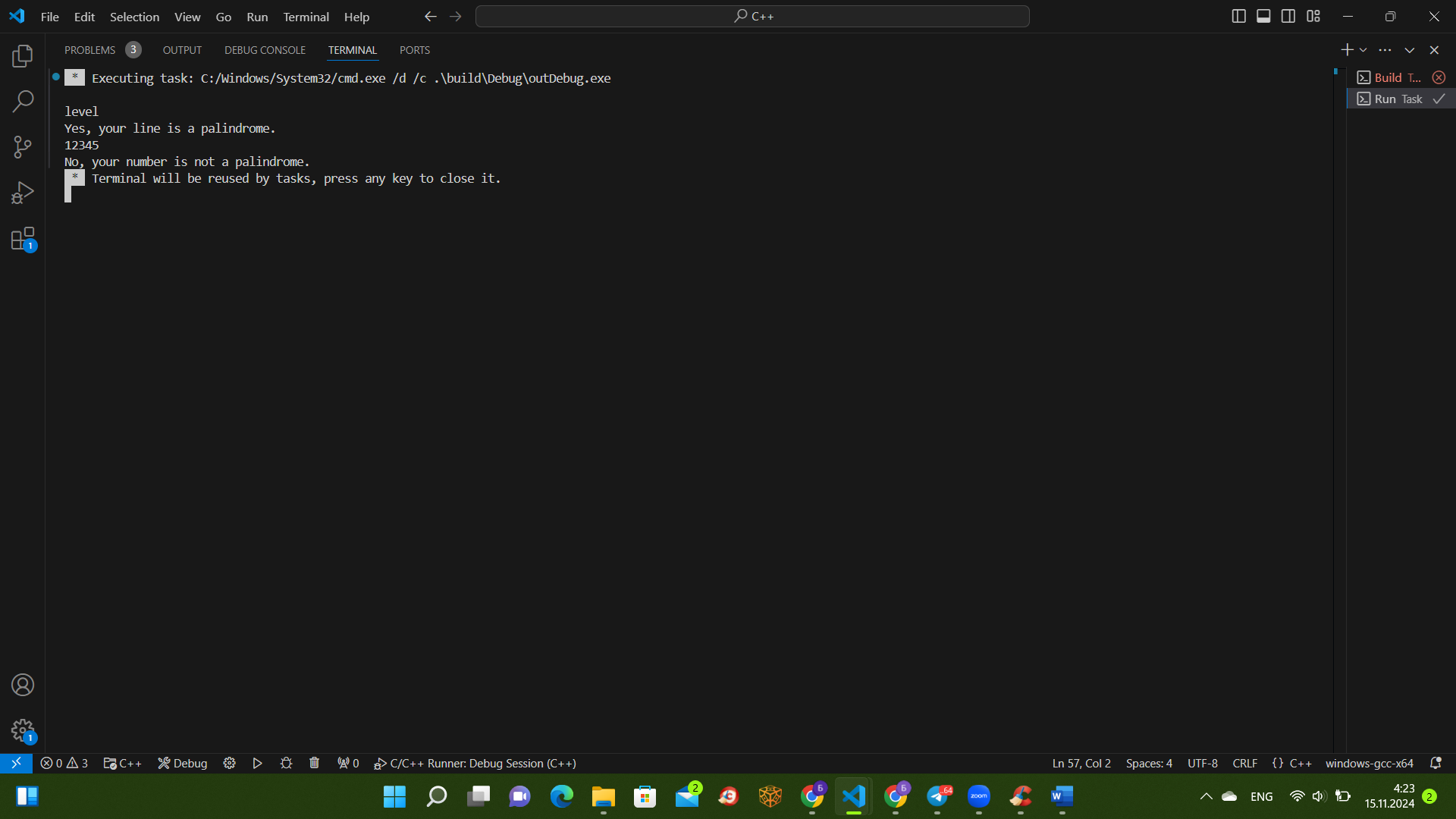
  
Фактичний час затрачений на виконання: 35хв

Task 6 - Lab# programming: Algotester Lab 3(варіант 3)



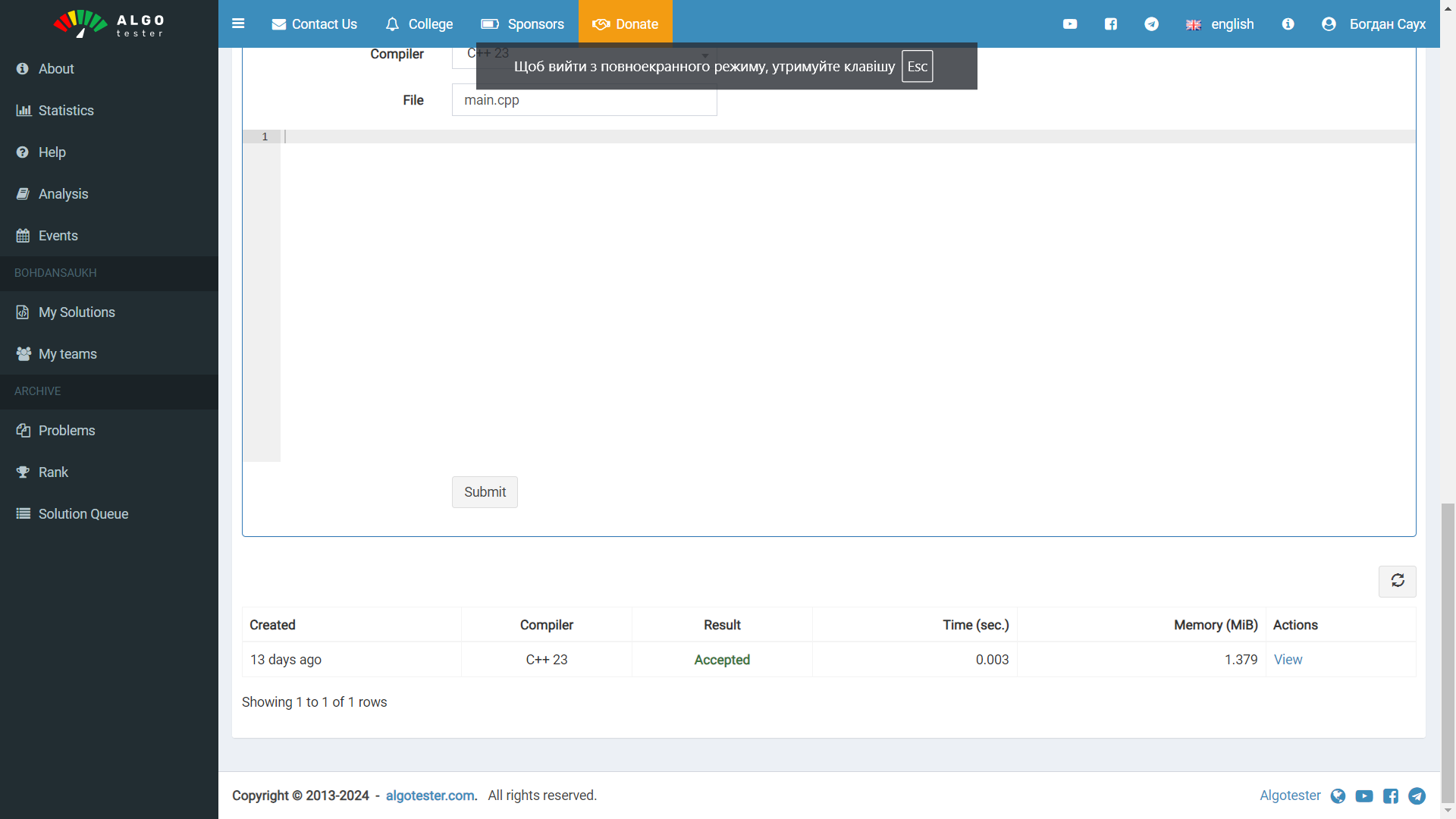
Фактичний час затрачений на виконання: 2 год

Task 7 - Practice# programming: Class Practice Task



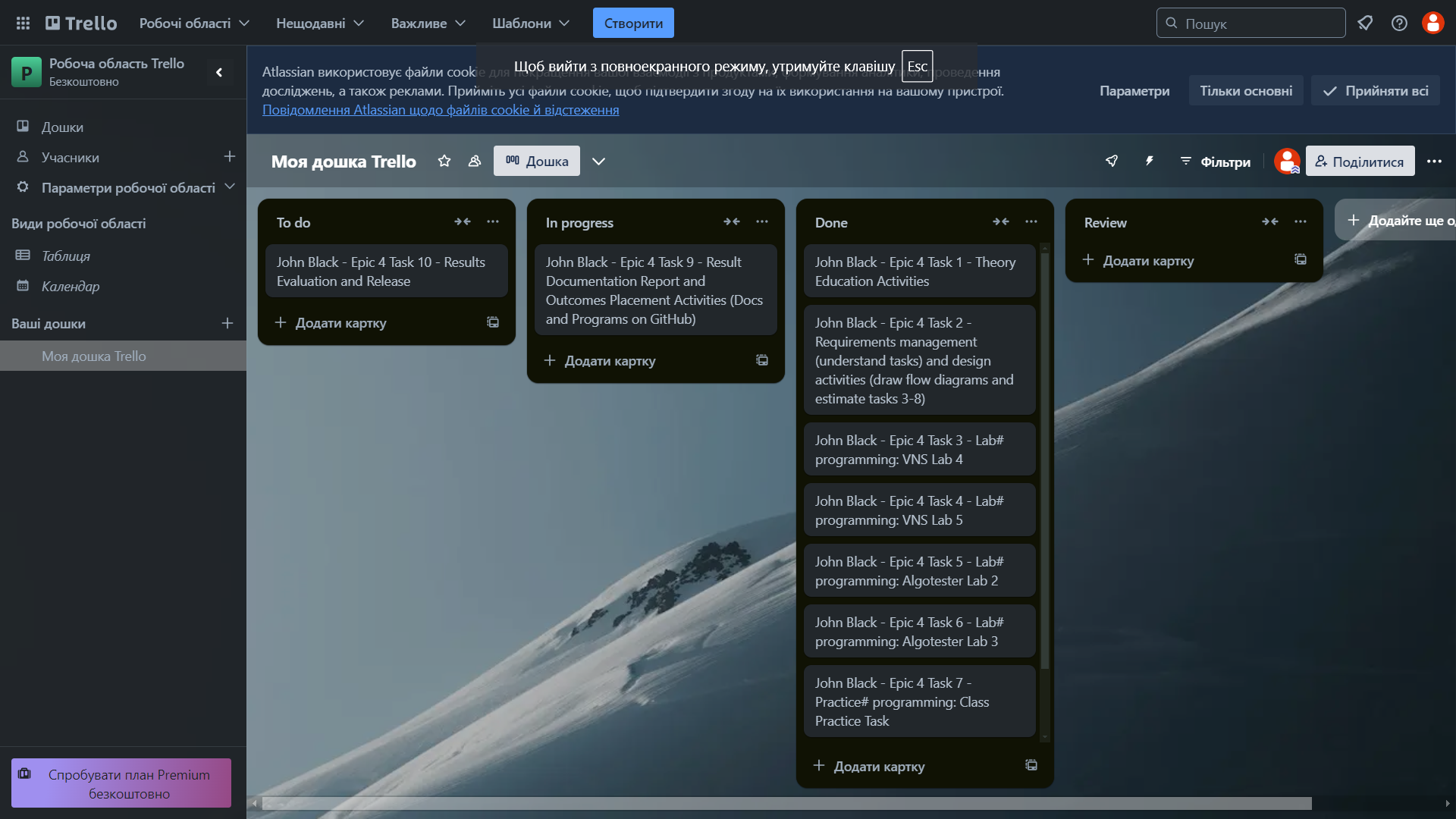
Фактичний час виконання: 1 год

Task 8  - Practice# programming:  Self Practice Task



Фактичний час виконання: 40хв

**6)Робота з комадою**



**Висновок:** Опановуючи роботу з масивами різних типів, вказівниками, посиланнями, динамічними структурами даних та алгоритмами їх обробки, ми набуваємо необхідних знань і навичок для ефективного управління даними та їхньої оптимізації. Це дозволяє будувати більш гнучкі та продуктивні програми, здатні працювати з великими обсягами даних, зберігаючи при цьому структурованість і логічну цілісність інформації.

**Посилання на пул реквест: https://github.com/artificial-intelligence-department/ai\_programming\_playground\_2024/pull/315**