Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

**про виконання лабораторних та практичних робіт блоку № 5**

На тему:  «Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.»

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

до:

Практичних Робіт до блоку № 5

**Виконала:**

Студентка групи ШІ-12

Лазаревич Юлія Дмитрівна

Львів 2024

**Тема роботи:**

Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

**Мета роботи:**

Ознайомитись з файлами, бінарними файлами, символами і рядковими змінними та текстовими файлами, стандартною бібліотекою та деталями/методами роботи з файлами, створенням та використанням бібліотек.

**Теоретичні відомості:**

Тема №1: Вступ до Роботи з Файлами.

* Джерела Інформації:
* [Приклади використання засобів C++ для роботи з файлами](https://www.bestprog.net/uk/2019/09/11/examples-of-using-c-tools-for-working-with-files-ua/)
* [Файлова система C++. Загальні принципи роботи. Приклади. Відкриття/закриття файлу](https://www.bestprog.net/uk/2019/09/07/c-file-system-general-principles-of-work-examples-open-close-file-ua/)
* Що опрацьовано:
* У C++ використовується стандартна бібліотека <fstream> для роботи з файлами. Для текстових файлів використовуються ifstream (читання), ofstream (запис) і fstream (читання-запис). Для коректного завершення роботи важливо закривати файли методом .close(), щоб уникнути втрати даних через буферизацію.
* Файлові дескриптори – низькорівневі ідентифікатори файлів, які працюють безпосередньо з системними викликами, наприклад, через функції open, read, write в стилі C.
* Важливо перевіряти стан файлу за допомогою методів, таких як .fail(), .eof() (кінець файлу), .bad() для обробки помилок.
* Статус: ознайомлена.
* Початок опрацювання теми: 22.11.24
* Завершення опрацювання теми 26.11.24

Тема №2: Символи і Рядкові Змінні.

* Джерела Інформації:
* Лекції О. Пшеничного
* Практичні М. Фаріон
* Що опрацьовано:
* Робота з char і std::string: Рядки типу char обробляються як масиви символів, тоді як std::string є класом із зручними методами, такими як append, substr, find.
* Використовується оператор + для об'єднання рядків, а для роботи з окремими символами застосовується індексація або метод at().
* Статус: ознайомлена.
* Початок опрацювання теми: 22.11.24
* Завершення опрацювання теми 25.11.24

Тема №3: Текстові Файли.

* Джерела Інформації:
* [Приклади роботи з текстовими файлами. Модифікація файлів. Сортування даних у файлах. Конвертування даних файлу в масив](https://www.bestprog.net/uk/2020/05/14/c-examples-of-working-with-text-files-modification-of-files-sort-data-in-files-convert-file-data-to-list-ua/)
* Що опрацьовано:
* Для текстових файлів використовуються функції getline, ignore, peek для обробки вводу/виводу.
* Методи setw, setfill, setprecision дозволяють форматувати текст у файлах, наприклад, створювати таблиці або відображати дані з потрібною кількістю знаків після коми.
* Парсинг тексту застосовується для обробки файлів, наприклад, розділення рядків на слова через std::istringstream або пошук патернів.
* Статус: ознайомлена.
* Початок опрацювання теми: 27.11.24
* Завершення опрацювання теми 28.11.24

Тема №4: Бінарні Файли.

* Джерела Інформації:
* [Робота з файлами в C++. Частина 1 – бібліотека fstream](https://purecodecpp.com/uk/archives/2751).
* Що опрацьовано:
* Бінарні файли зберігають дані у вигляді послідовності байтів, без форматування тому їх зручно використовувати для великих обсягів даних.
* Робота з позиціонуванням: методи seekg та seekp дозволяють переміщувати покажчик файлу для читання чи запису з певної позиції.
* Серіалізація: використовується для збереження складних структур у файл і відновлення їх.
* Статус: ознайомлена.
* Початок опрацювання теми: 28.11.24
* Завершення опрацювання теми 29.11.24

Тема №5: Стандартна бібліотека та робота з файлами.

* Джерела Інформації:
* [Стандартна бібліотека та файли](https://codefinity.com/ua/courses/v2/59a3e4eb-a3a1-49a4-ba70-da98e8fee510/6d02884f-1f0f-4c9f-af75-25adad41655a/48a66c94-d684-44d7-a316-31b6e1176430)
* Що опрацьовано:
* Потоки вводу/виводу:
* ifstream: Використовується для читання даних з файлів. Підтримує методи, такі як open() для відкриття файлу, .good() для перевірки стану, і >> для читання даних.
* ofstream: Призначений для запису даних у файл. Підтримує методи форматування, такі як setprecision, setw, і .put() для запису символів.
* fstream: Комбінує функції обох потоків, дозволяючи читання і запис в одному об'єкті. Підтримує методи для позиціонування (seekg, seekp), читання і запису рядків (getline, write).
* Всі ці класи успадковують базові функції від класу std::ios.
* Основні методи обробки помилок при роботі з файлами:
* .fail(): Повертає true, якщо відбулася помилка під час роботи з файлом.
* .eof(): Визначає, чи досягнуто кінця файлу.
* .bad(): Виявляє критичні помилки (наприклад, пошкодження файлу).
* Приклади помилок:
* Невдале відкриття файлу: можна перевірити через if (file.fail()).
* Помилки читання: обробляються через блоки try-catch, якщо використовується std::exception.
* Статус: ознайомлена.
* Початок опрацювання теми: 28.11.24
* Завершення опрацювання теми 29.11.24

Тема №6: Створення й використання бібліотек.

* Джерела Інформації:

* [Статичні та динамічні бібліотеки](https://acode.com.ua/statychni-i-dynamichni-biblioteky/)
* Що опрацьовано:
* Коди зазвичай поділяють на файли .h (оголошення) та .cpp (реалізація).
* Типи бібліотек:
* Статичні (.lib): Включаються в компіляцію програми.
* Динамічні (.dll, .so): Завантажуються під час виконання.
* Інтерфейс бібліотеки має бути чітко визначений та задокументований для зручного використання іншими розробниками.
* Статичні бібліотеки проти динамічних:

 Статичні бібліотеки:

* Включаються в основний файл під час компіляції.
* Переваги: швидкість виконання, немає залежності від зовнішніх файлів.
* Недоліки: збільшується розмір виконуваного файлу.

 Динамічні бібліотеки:

* Завантажуються під час виконання програми.
* Переваги: менший розмір виконуваного файлу, зручність оновлення бібліотеки.
* Недоліки: можливі проблеми з версіями (DLL Hell).
* Щоб використати сторонню бібліотеку у своєму проєкті треба використати #include для оголошення функцій із бібліотеки, або додати бібліотеку до компілятора через ключі, наприклад, -l для GCC.
* Компілікація бібліотек:
* Статичні бібліотеки компілюються в .lib (Windows) або .a (Linux).
* Динамічні бібліотеки створюють .dll або .so файли.
* Використовуйте директиви #pragma once або #ifndef/#define для уникнення дублювань при інклюдах.
* Статус: ознайомлена.
* Початок опрацювання теми: 22.11.24
* Завершення опрацювання теми 23.11.24

**Виконання роботи:**

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

Завдання №1 – VNS Lab 6 – Task 1 – Variant 10.

Задано рядок, що складається із символів. Символи поєднуються в слова. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів.

Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом.

Перетворити рядок таким чином, щоб на його початку були записані слова, що містять тільки цифри, потім слова, що містять тільки букви, а потім слова, які містять і букви і цифри.

Завдання №2 – VNS Lab 8 – Task 1 – Variant 10.

Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вмістиме, виконати знищення й додавання елементі у відповідності зі своїм варіантом, використовуючи для пошуку елементів що знищуються чи додаються, функцію. Формування, друк, додавання й знищення елементів оформити у вигляді функцій. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

Структура "Інформація":

- носій;

- об’єм;

- назва;

- автор.

Знищити перший елемент із заданим об’ємом інформації, додати елемент перед елементом із зазначеним номером.

Завдання №3 – VNS Lab 9 – Task 1 – Variant 10.

Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію. Виконати завдання.

1) Скопіювати з файлу F1 у файл F2 всі рядки, які не містять слова, що починаються на одну букву.

2) Знайти найкоротше слово у файлі F2.

Завдання №4.1 – Algotester Lab 4 – Variant 2.

Вам дано масив a з N цілих чисел.  
Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4].  
Після цього оберніть посортовану версію масиву a на K, тобто при K=3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3].  
Виведіть результат.

# Вхідні дані

У першому рядку цілі числа N та K

У другому рядку N цілих чисел - елементи масиву a

# Вихідні дані

У першому рядку ціле число N - розмір множини a  
У наступному рядку N цілих чисел - множина a

Завдання №4.2 – Algotester Lab 4 – Variant 2.

Вам дано масив a з N цілих чисел.  
Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4].  
Після цього оберніть посортовану версію масиву a на K, тобто при K=3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3].  
Виведіть результат.

# Вхідні дані

У першому рядку цілі числа N та K

У другому рядку N цілих чисел - елементи масиву a

# Вихідні дані

У першому рядку ціле число N - розмір множини a  
У наступному рядку N цілих чисел - множина a

Завдання №5 – Algotester Lab 6 – Variant 3.

У Клінта в черговий раз виключилось світло і йому немає чим зайнятися. Так як навіть це не заставить його подивитися збережені відео про програмування на ютубі - він вирішив придумати свою гру на основі судоку.

Гра виглядає так:

Є поле розміром N×N, в якому частина клітинок заповнена цифрами, а частина клітинок пусті (позначаються нулем). Також у нього є Q пар координат X та Y.

Завданням гри є написати до кожної координати скільки чисел туди можна вписати (якщо вона пуста) і які це числа (обов’язково в посортовані по зростанню!). В клітинку можна вписати лише ті числа, які не зустрічаються в рядку та стовбці, які перетинаються у цій клітинці.

Під час гри поле не міняється!

Також необовязково, щоб це було валідне судоку! Якщо є клітинка, в яку не можна вписати ніяку цифру - виведіть 0.

Також допускаються рядки та стовпці, в яких цифра записана кілька разів.

# Вхідні дані

У першому рядку ціле число N - розмір поля для гри

У N наступних рядках стрічка rowi яка складається з N цифер - i-й рядок.

Ціле число Q - кількість запитань

У наступних Q рядках 2 цілих числа xj,yj - координати клітинок j-го запитання

# Вихідні дані

Q разів відповідь у наступному форматі:

Натуральне число M - кількість цифр, які можна вписати в клітинку

M цифер розділених пробілом - можливі цифри

Завдання №6**–** Class Practice Work 1 – Запис текстової стрічки у файл із заданим ім’ям.

Реалізувати функцію створення файла і запису в нього даних:

enum FileOpResult { Success, Failure, … };

FileOpResult write\_to\_file(char \*name, char \*content);

Умови задачі:

- створити файл із заданим ім’ям; якщо файл існує – перезаписати його вміст

- написати код стійкий до різних варіантів вхідних параметрів

- name – ім’я, може не включати шлях

- записати у файл вміст стрічки content, прочитати content із стандартного вводу

- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.

Мета задачі

Розуміння методів роботи з файлами: Робота з файлами є одним з базових навиків програмування. Реалізація функції створення та запису в файл допоможе освоїти практичні навики роботи з файлами з використанням стандартної бібліотеки С++. Для виконання завдання студент має навчитись використовувати методи відкриття файла, запису масиву даних у файл, закриття файла та обробки помилок чи станів операції на кожному з етапів.

Розвиток алгоритмічне мислення: Запис у файл включає набір операції, які якнайкраще вкладаються в концепцію алгоритма, як списка детальних кроків. Імплементація цієї функції наочно демонструє створення алгоритмів у програмуванні.

Освоїти навики роботи з текстовими стрічками: завдання допоможе освоїти роботу з С стрічка, які є масивами з нульовим символом в кінці. Типові концепції при роботі з С стрічками це арифметика вказівників, ітерація по стрічці, копіювання частини стрічки, розбиття на токени по заданому символу.

Розвинути навички розв’язувати задачі: Запис у файл може супроводжуватись набором станів (немає доступу на створення, недостатньо місця, ін.), які необхідно передбачити у алгоритмі. Аналіз цих станів дозволяє розвинути навик розв’язання інженерних задач у програмуванні.

Завдання №7**–** Class Practice Work 2 – Копіювання вмісту файла у інший файл.

***Реалізувати функцію створення файла і запису в нього даних:***

enum FileOpResult { Success, Failure, … };

FileOpResult copy\_file(char \*file\_from, char \*file\_to);

*Умови задачі:*

-       копіювати вміст файла з ім’ям file\_from у файл з ім’ям file\_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів

-       file\_from, file\_to – можуть бути повним або відносним шляхом

-       повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

**Мета задачі**

***Розуміння методів роботи з файлами:*** Робота з файлами є одним з базових навиків програмування. Реалізація функції копіювання вмісту файла допоможе освоїти практичні навики роботи з файлами з використанням стандартної бібліотеки С++. Для виконання завдання студент має навчитись використовувати методи відкриття файла, читання вмісту файла, запису масиву даних у файл, закриття файла та обробки помилок чи станів операції на кожному з етапів.

***Розвиток алгоритмічне мислення:*** Читання та запис у файл включає набір операцій, які якнайкраще вкладаються в концепцію алгоритма, як списка детальних кроків. Імплементація цієї функції наочно демонструє створення алгоритмів у програмуванні.

***Освоїти навики роботи з потоком даних:*** завдання допоможе освоїти роботу з потоками даних (концепція реалізована в STL як набір класів \*stream\* - fstream, stringstream, streambuf та ін.). Концепція потоку даних дозволяє абстрагувати роботу з джерелами та приймачами даних та писати з її допомогою високорівневий код.

***Розвинути навички розв’язувати задачі:*** Операції читання з файла та запис у файл можуть супроводжуватись набором різних станів (немає доступу на читання чи створення, недостатньо місця, ін.), які необхідно передбачити у алгоритмі. Аналіз цих станів дозволяє розвинути навик розв’язання інженерних задач у програмуванні.

Завдання №8- Self Practice Work – Algotester Lab 4 – Variant 3.1.

Вам дано масив, який складається з N додатніх цілих чисел.

Ваше завдання - розділити його на три частини, по остачі від ділення на 3, по зростанню остачі (тобто спочатку йдуть числа, у яких остача 0, далі числа з остачею 1 і тоді нарешті числа з остачею 2).

Далі необхідно ті елементи, остача від ділення на 3 яких парна посортувати по зростанню, а ті, у яких остача 1 - по спаданню.

Після цього видаліть усі дублікати з масиву.

Виведіть результуючий масив.

# Вхідні дані

У першому рядку N - кількість чисел.

У другому рядку N чисел ai - елементи масиву.

# Вихідні дані

У першому рядку M - кількість чисел у масиву

У другому рядку M посортованих за умовою чисел.

Додаткове завдання №9– Algotester Lab 4 – Variant 3.2.

Вам дано масив, який складається з N додатніх цілих чисел.

Ваше завдання - розділити його на три частини, по остачі від ділення на 3, по зростанню остачі (тобто спочатку йдуть числа, у яких остача 0, далі числа з остачею 1 і тоді нарешті числа з остачею 2).

Далі необхідно ті елементи, остача від ділення на 3 яких парна посортувати по зростанню, а ті, у яких остача 1 - по спаданню.

Після цього видаліть усі дублікати з масиву.

Виведіть результуючий масив.

# Вхідні дані

У першому рядку N - кількість чисел.

У другому рядку N чисел ai - елементи масиву.

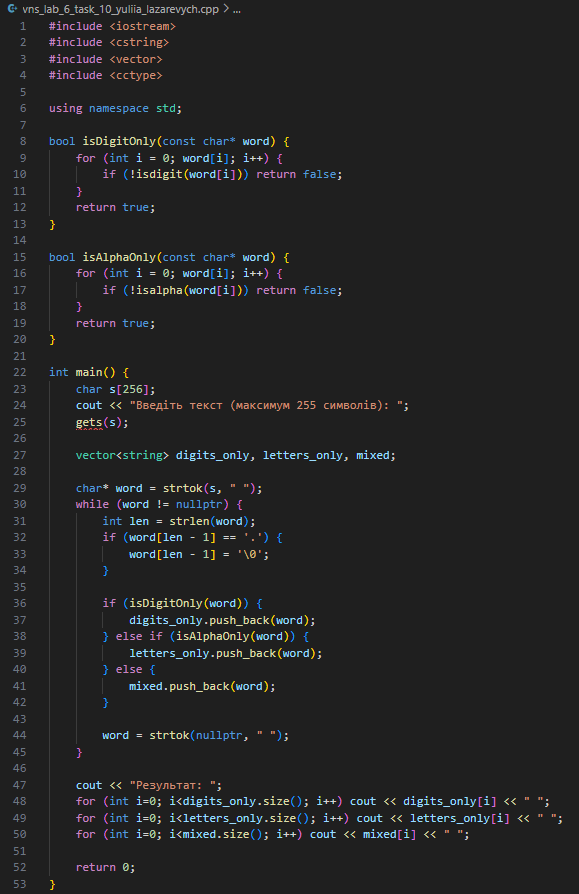
# Вихідні дані

У першому рядку M - кількість чисел у масиву

У другому рядку M посортованих за умовою чисел.

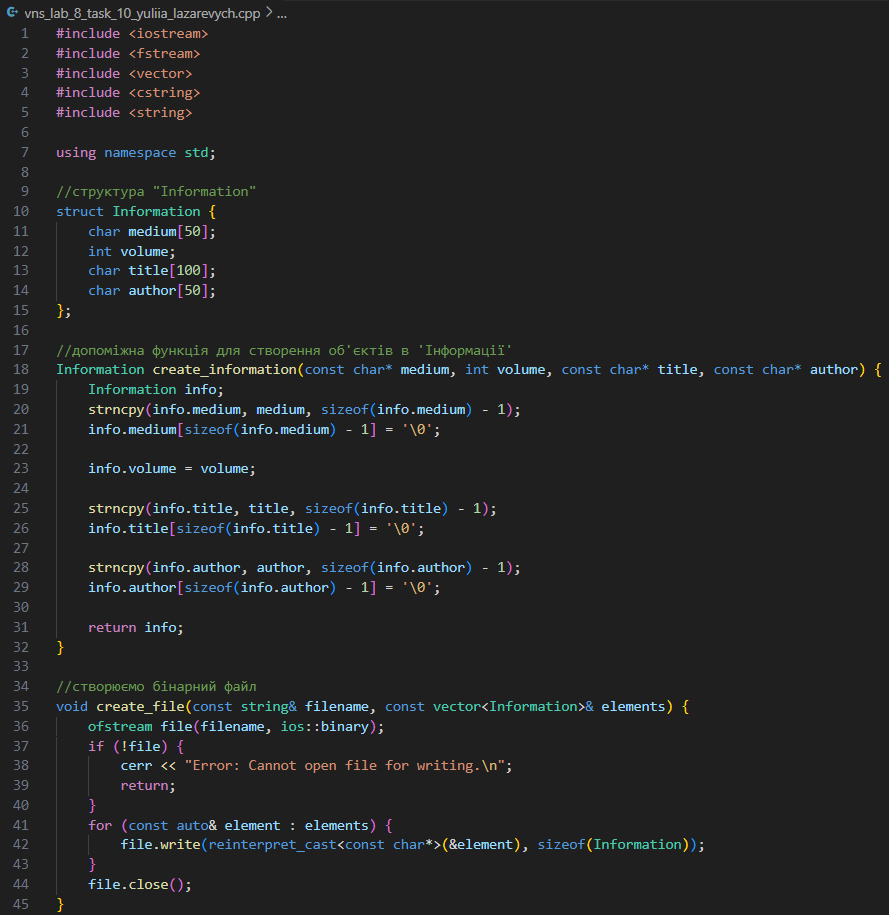
1. **Код програм з посиланням на зовнішні ресурси:**

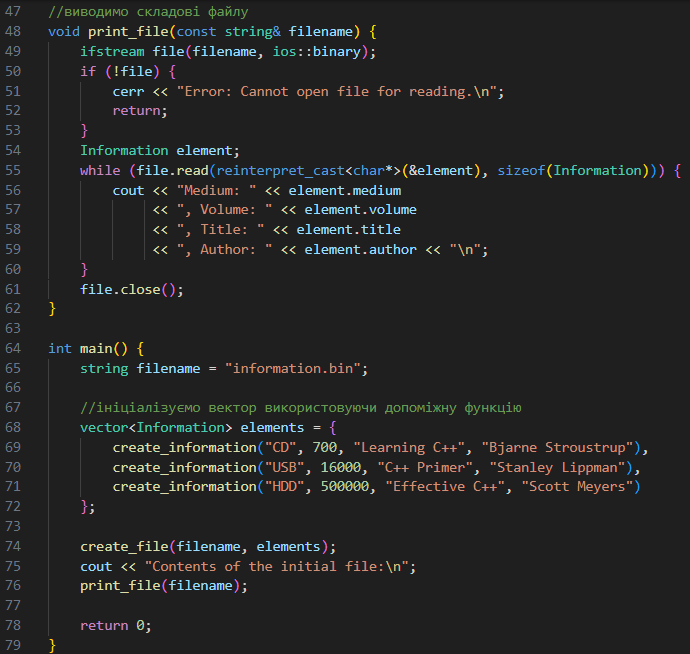
Завдання №1 – VNS Lab 6 – Task 1 – Variant 10.

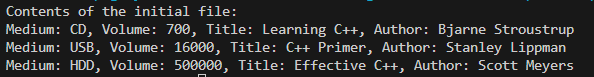




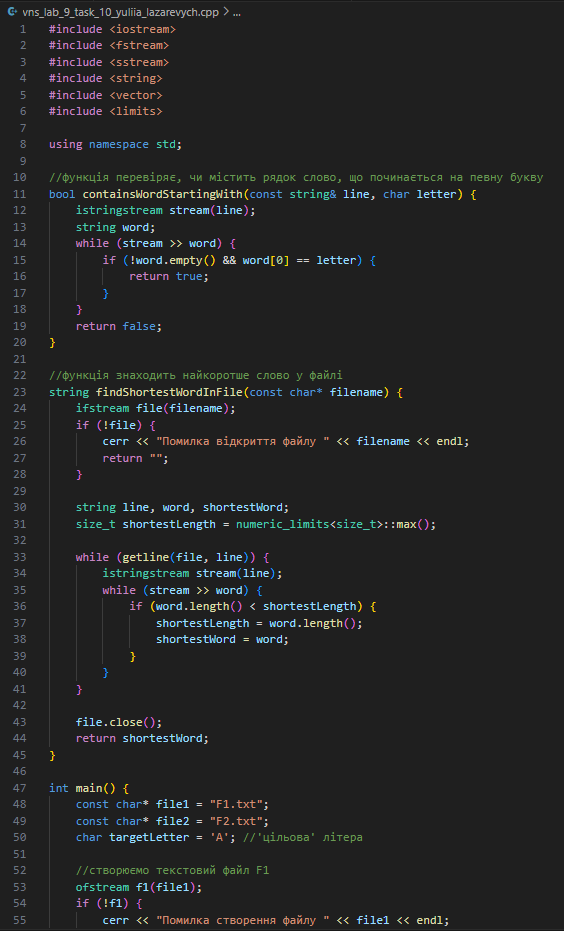
Завдання №2 – VNS Lab 8 – Task 1 – Variant 10.

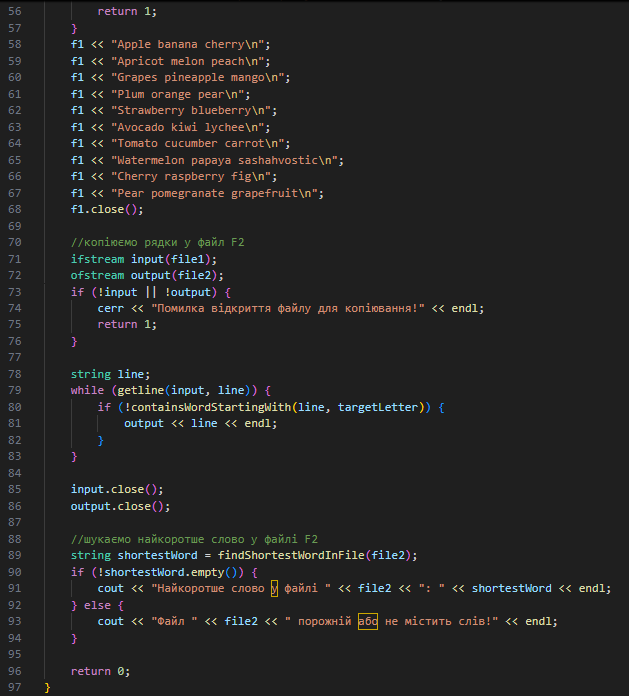






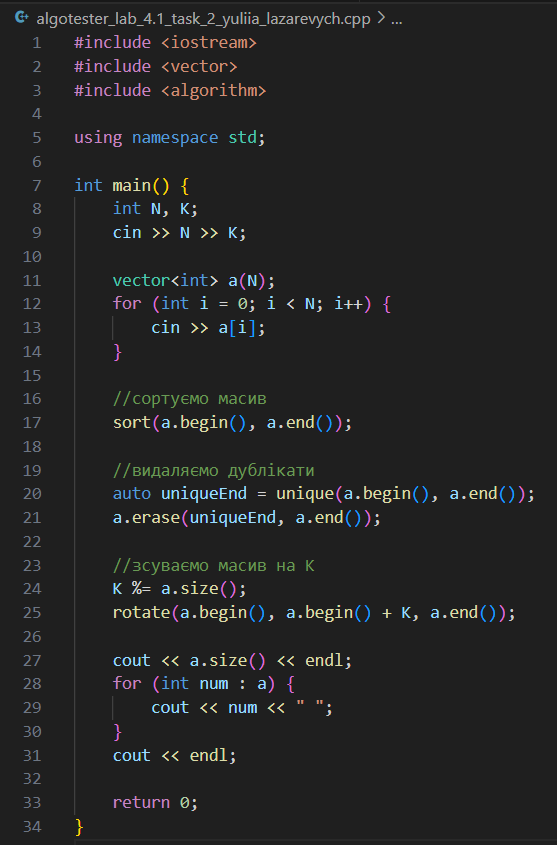
Завдання №3 – VNS Lab 9 – Task 1 – Variant 10.

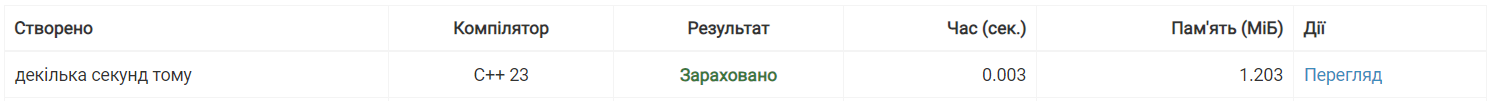




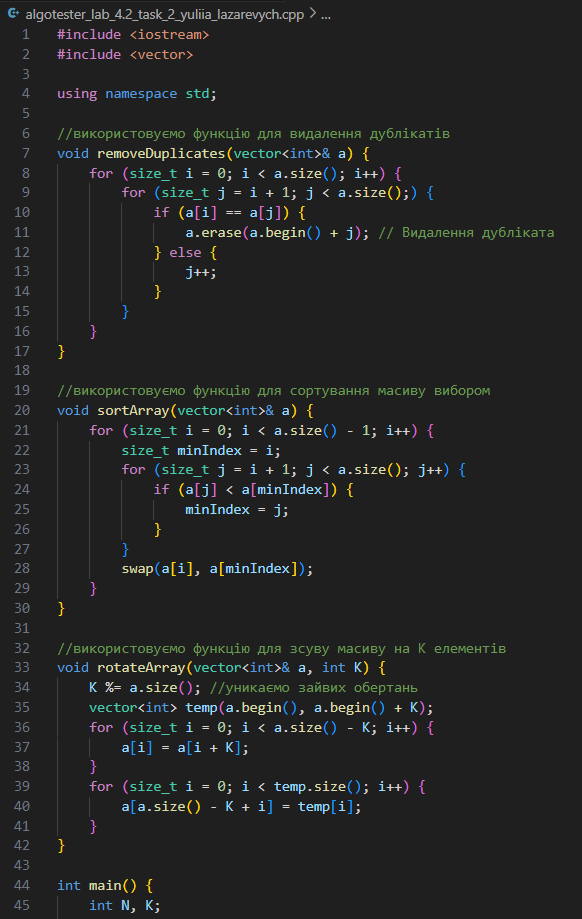


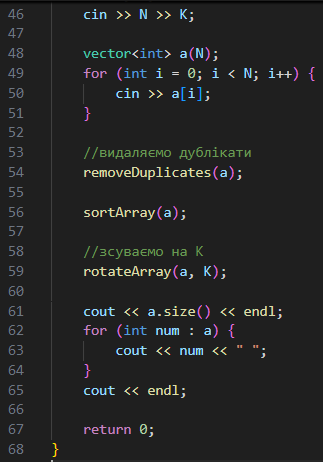
Завдання №4.1 – Algotester Lab 4 – Variant 2.

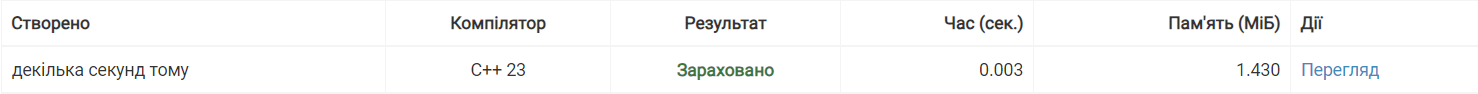




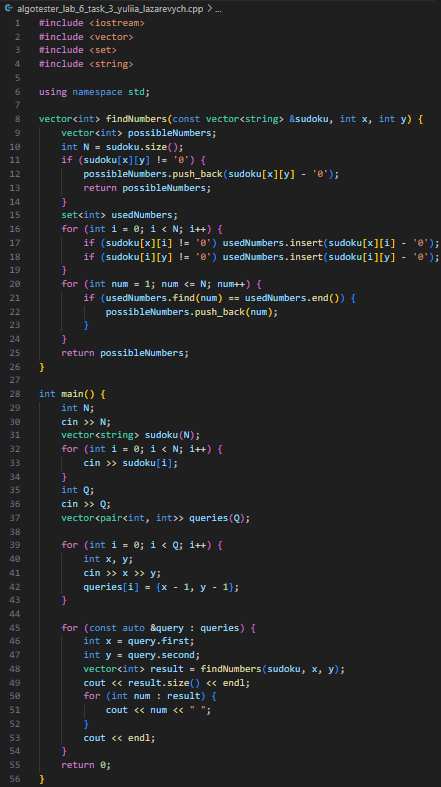
Завдання №4.2 – Algotester Lab 4 – Variant 2.

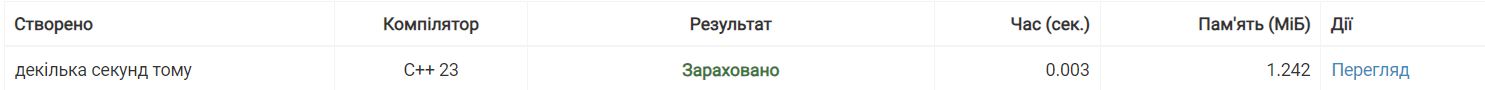




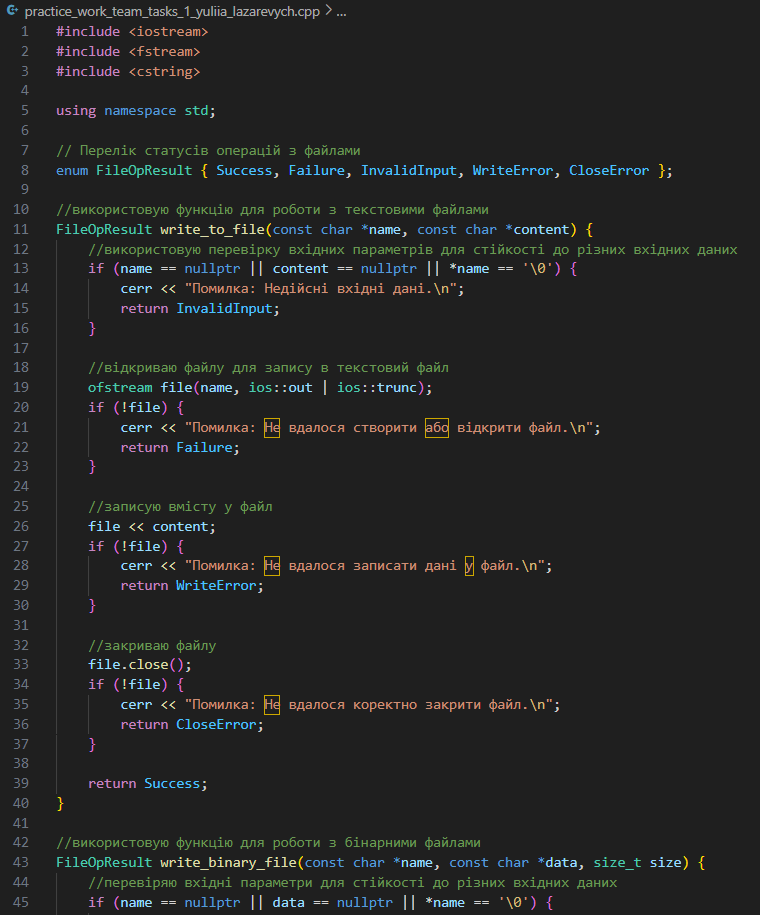


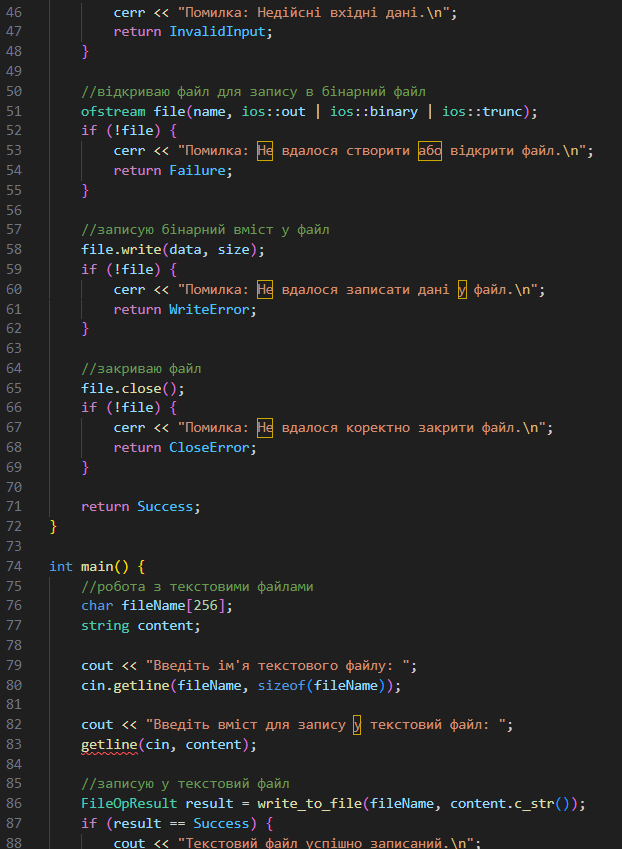
Завдання №5 – Algotester Lab 6 – Variant 3.

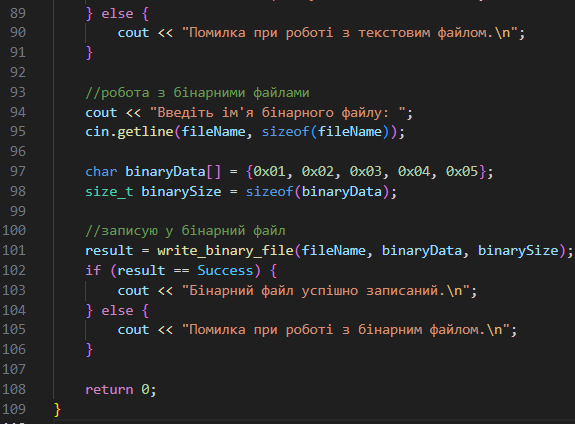


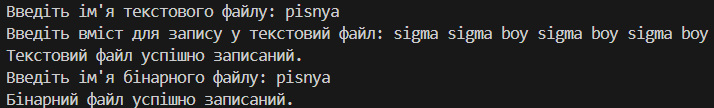


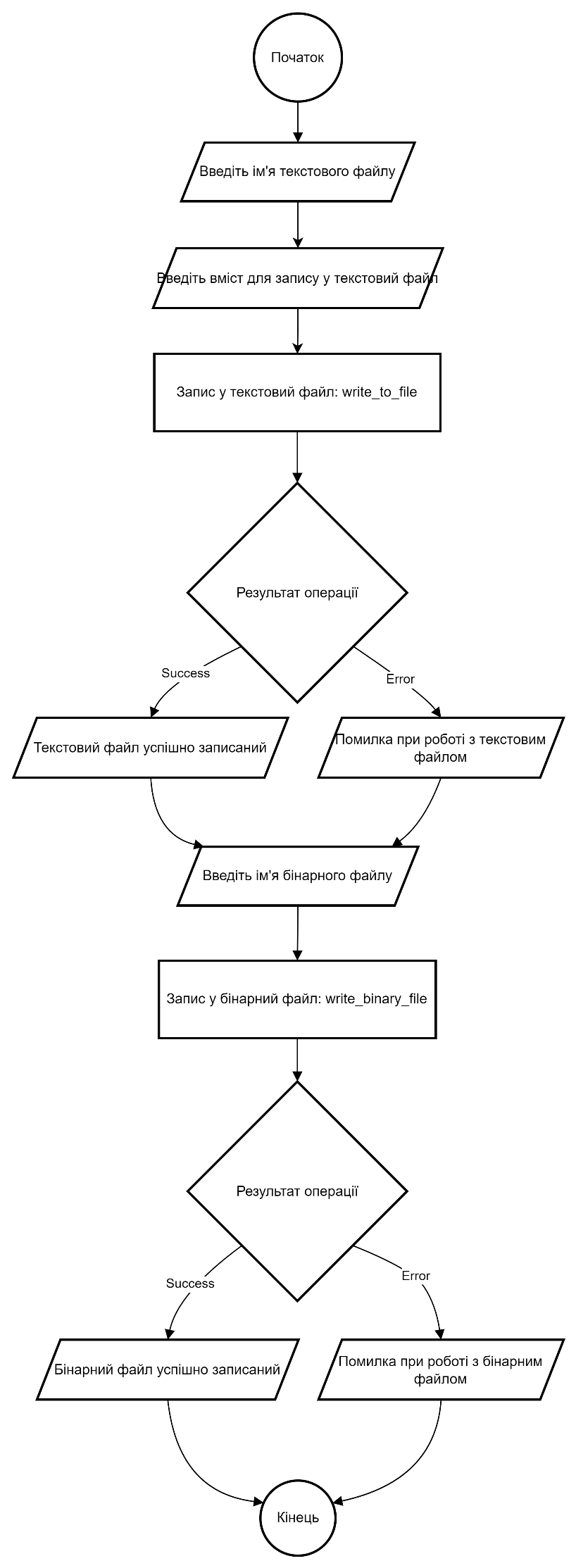
Завдання №6**–** Class Practice Work 1 – Запис текстової стрічки у файл із заданим ім’ям.



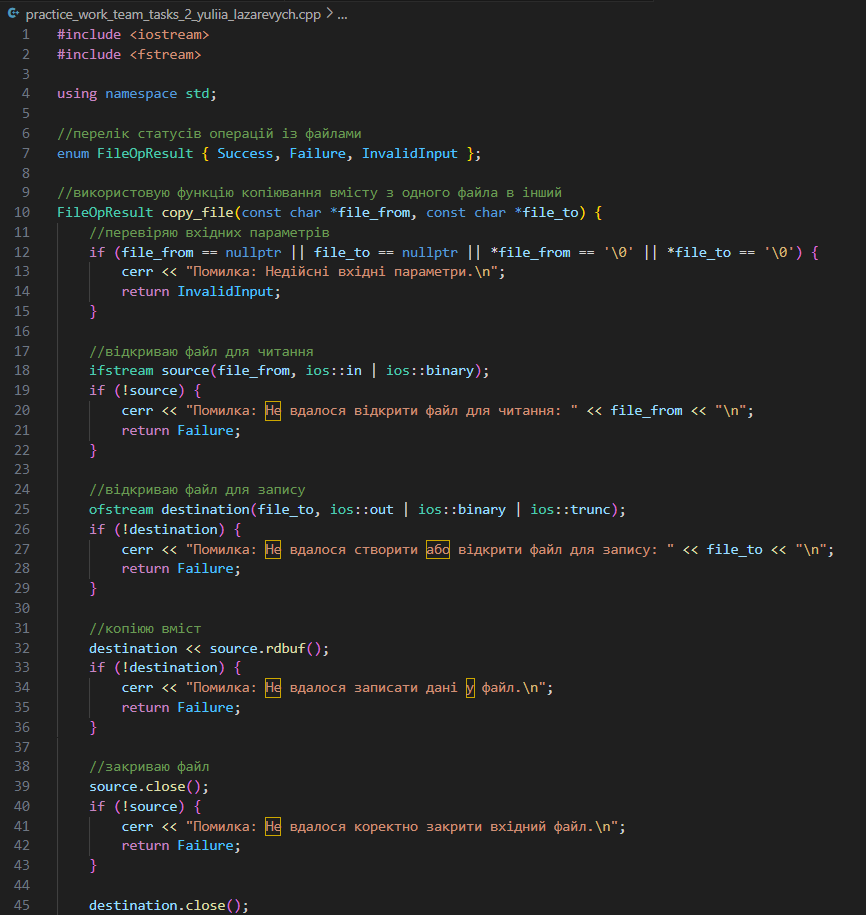


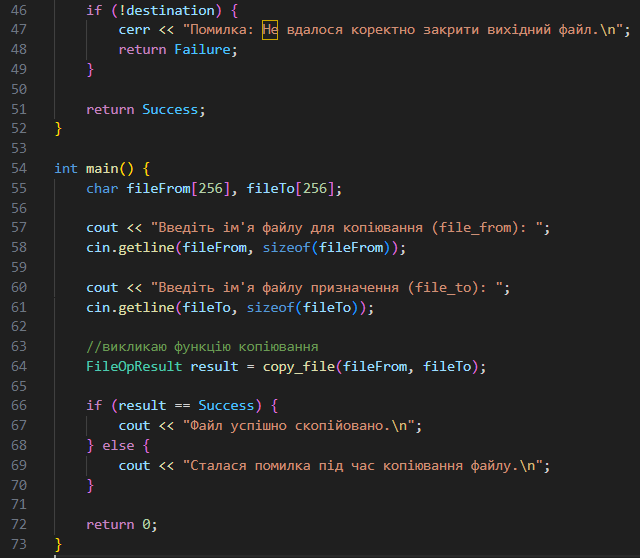






Завдання №7**–** Class Practice Work 2 – Копіювання вмісту файла у інший файл.

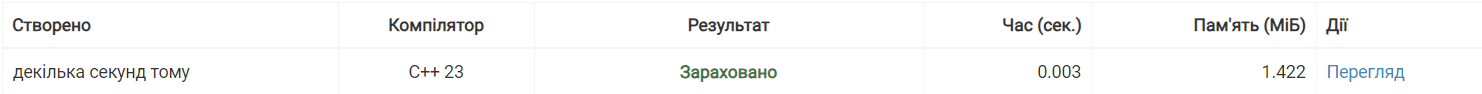




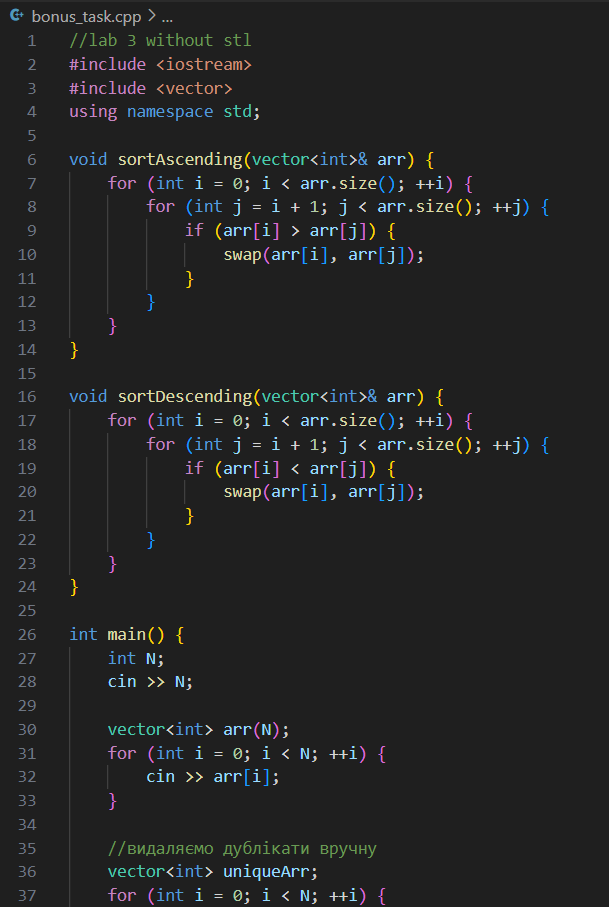


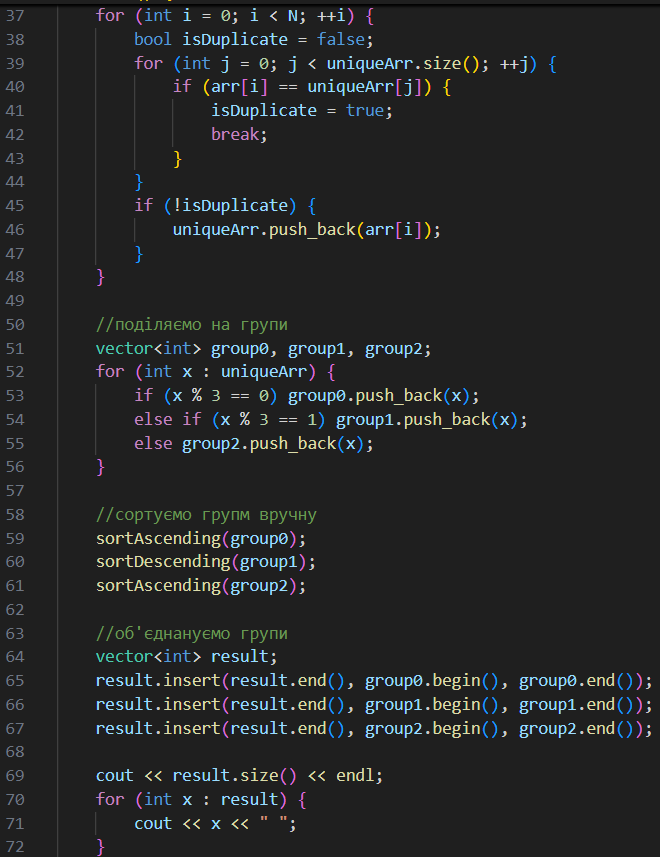
Завдання №8- Self Practice Work – Algotester Lab 4 – Variant 3.1.

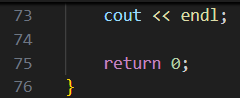


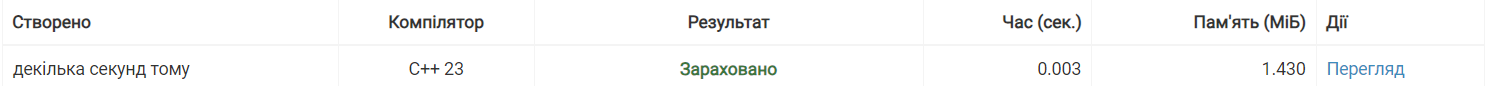


Додаткове завдання №9– Algotester Lab 4 – Variant 3.2.



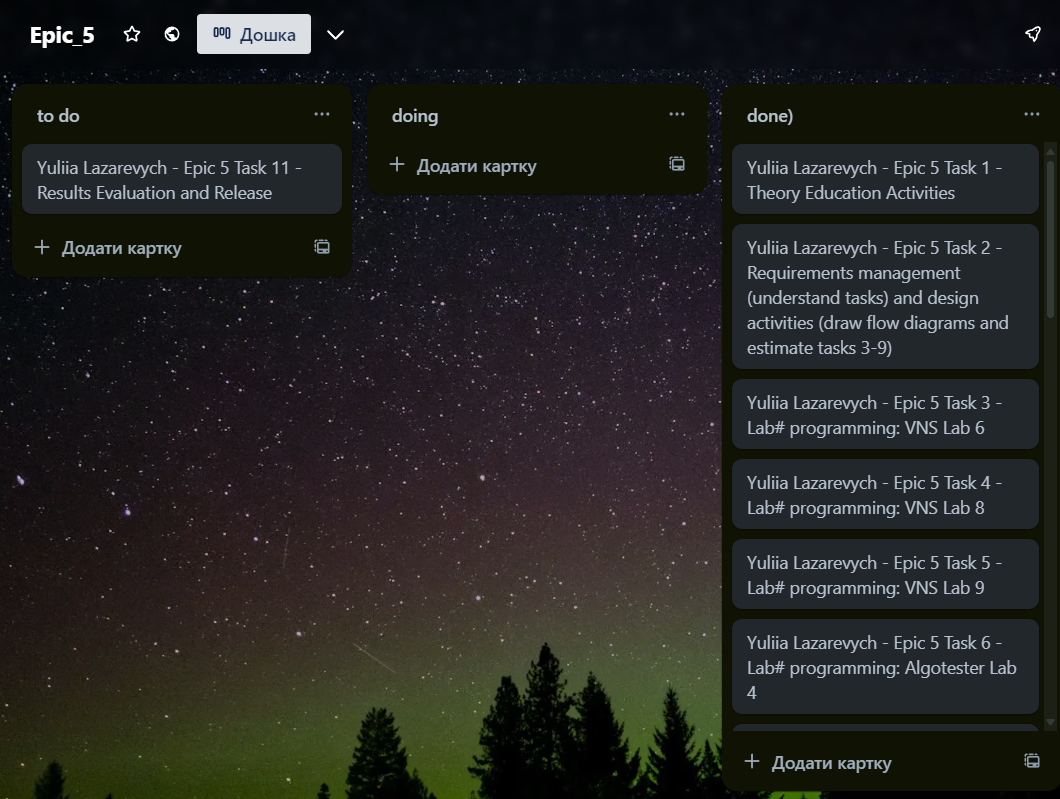






1. **Кооперація з командою:**

* Скрін прогресу по Трелло



Відстежували прогрес всієї команди завдяки дошці Trello

* Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло



Зустрічалися багато разів для обговорення та спільного виконання епіків.

**Висновки:** Виконуючи цей епік я ознайомилась з файлами, бінарними файлами, символами і рядковими змінними та текстовими файлами, стандартною бібліотекою та деталями/методами роботи з файлами, створенням та використанням бібліотек.