

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 ІІІ-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](#)
  - [Файлова система C++. Загальні принципи роботи. Приклади. Відкриття/закриття файлу](#)
- 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](#)
- Що опрацьовано:
  - Для текстових файлів використовуються функції `getline`, `ignore`, `peek` для обробки вводу/виводу.
  - Методи `setw`, `setfill`, `setprecision` дозволяють формувати 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`.](#)
- 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 `seekr` 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](#)
- Що опрацьовано:
  - Потоки вводу/виводу:
    - ifstream: Використовується для читання даних з файлів. Підтримує методи, такі як open() для відкриття файлу, .good() для перевірки стану, i >> для читання даних.
    - ofstream: Призначений для запису даних у файл. Підтримує методи форматування, такі як setprecision, setw, i .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](#)
- 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 вміст, виконати знищення й додавання елементів 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 а з N цілих чисел.

Спочатку видаліть масиву а усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4].

Після цього оберніть посортовану версію масиву а на К, тобто при К=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 sudoku.

Гра виглядає так:

Є поле розміром  $N \times N$ , 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 sudoku! 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 row $i$  яка складається з  $N$  цифр -  $i$ -й рядок.

Ціле число  $Q$  - кількість запитань

У наступних  $Q$  рядках 2 цілих числа  $x_j, y_j$  - координати клітинок  $j$ -го запиту

Вихідні дані

$Q$  разів відповідь у наступному форматі:

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

$M$  цифр розділених пробілом - можливі цифри

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

Реалізувати функцію створення файлу 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 файлу.

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 навички роботи з файлами з використанням



стандартної бібліотеки C++. 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. Імплементация цієї функції наочно демонструє створення алгоритмів у програмуванні.

Освоїти навички роботи з текстовими стрічками: завдання допоможе освоїти роботу з C стрічка, які є масивами з нульовим символом в кінці. Типові концепції при роботі з C 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 навичок програмування. Реалізація функції копіювання вмісту файла допоможе освоїти практичні навички роботи з файлами з використанням стандартної бібліотеки C++. 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 списку детальних кроків. Імплементация цієї функції наочно демонструє створення алгоритмів у програмуванні.

**Освоїти навички роботи з потоком даних:** завдання допоможе освоїти роботу з потоками даних (концепція реалізована в STL як набір класів `*stream*` - `fstream`, `stringstream`, `stringstreambuf` 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 файлу 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  $a_i$  - 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  $a_i$  - елементи масиву.

Вихідні дані

У першому рядку  $M$  - кількість чисел у масиву

У другому рядку  $M$  посортованих за умовою чисел.

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

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

```

G+ vns_lab_6_task_10_yuliia_lazarevych.cpp > ...
1  #include <iostream>
2  #include <cstring>
3  #include <vector>
4  #include <cctype>
5
6  using namespace std;
7
8  bool isDigitOnly(const char* word) {
9      for (int i = 0; word[i]; i++) {
10         if (!isdigit(word[i])) return false;
11     }
12     return true;
13 }
14
15 bool isAlphaOnly(const char* word) {
16     for (int i = 0; word[i]; i++) {
17         if (!isalpha(word[i])) return false;
18     }
19     return true;
20 }
21
22 int main() {
23     char s[256];
24     cout << "Введіть текст (максимум 255 символів): ";
25     gets(s);
26
27     vector<string> digits_only, letters_only, mixed;
28
29     char* word = strtok(s, " ");
30     while (word != nullptr) {
31         int len = strlen(word);
32         if (word[len - 1] == '.') {
33             word[len - 1] = '\0';
34         }
35
36         if (isDigitOnly(word)) {
37             digits_only.push_back(word);
38         } else if (isAlphaOnly(word)) {
39             letters_only.push_back(word);
40         } else {
41             mixed.push_back(word);
42         }
43
44         word = strtok(nullptr, " ");
45     }
46
47     cout << "Результат: ";
48     for (int i=0; i<digits_only.size(); i++) cout << digits_only[i] << " ";
49     for (int i=0; i<letters_only.size(); i++) cout << letters_only[i] << " ";
50     for (int i=0; i<mixed.size(); i++) cout << mixed[i] << " ";
51
52     return 0;
53 }

```

Введіть текст (максимум 255 символів): hjvjv jvvjv 5616 695 bjh6656  
Результат: 5616 695 hjvjv jvvjv bjh6656

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

```
vns_lab_8_task_10_yuliia_lazarevych.cpp > ...
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <cstring>
5  #include <string>
6
7  using namespace std;
8
9  //структура "Information"
10 struct Information {
11     char medium[50];
12     int volume;
13     char title[100];
14     char author[50];
15 };
16
17 //допоміжна функція для створення об'єктів в 'Інформації'
18 Information create_information(const char* medium, int volume, const char* title, const char* author) {
19     Information info;
20     strncpy(info.medium, medium, sizeof(info.medium) - 1);
21     info.medium[sizeof(info.medium) - 1] = '\0';
22
23     info.volume = volume;
24
25     strncpy(info.title, title, sizeof(info.title) - 1);
26     info.title[sizeof(info.title) - 1] = '\0';
27
28     strncpy(info.author, author, sizeof(info.author) - 1);
29     info.author[sizeof(info.author) - 1] = '\0';
30
31     return info;
32 }
33
34 //створюємо бінарний файл
35 void create_file(const string& filename, const vector<Information>& elements) {
36     ofstream file(filename, ios::binary);
37     if (!file) {
38         cerr << "Error: Cannot open file for writing.\n";
39         return;
40     }
41     for (const auto& element : elements) {
42         file.write(reinterpret_cast<const char*>(&element), sizeof(Information));
43     }
44     file.close();
45 }
```

```

47 //виводимо складові файлу
48 void print_file(const string& filename) {
49     ifstream file(filename, ios::binary);
50     if (!file) {
51         cerr << "Error: Cannot open file for reading.\n";
52         return;
53     }
54     Information element;
55     while (file.read(reinterpret_cast<char*>(&element), sizeof(Information))) {
56         cout << "Medium: " << element.medium
57             << ", Volume: " << element.volume
58             << ", Title: " << element.title
59             << ", Author: " << element.author << "\n";
60     }
61     file.close();
62 }
63
64 int main() {
65     string filename = "information.bin";
66
67     //ініціалізуємо вектор використовуючи допоміжну функцію
68     vector<Information> elements = {
69         create_information("CD", 700, "Learning C++", "Bjarne Stroustrup"),
70         create_information("USB", 16000, "C++ Primer", "Stanley Lippman"),
71         create_information("HDD", 500000, "Effective C++", "Scott Meyers")
72     };
73
74     create_file(filename, elements);
75     cout << "Contents of the initial file:\n";
76     print_file(filename);
77
78     return 0;
79 }

```

Contents of the initial file:

```

Medium: CD, Volume: 700, Title: Learning C++, Author: Bjarne Stroustrup
Medium: USB, Volume: 16000, Title: C++ Primer, Author: Stanley Lippman
Medium: HDD, Volume: 500000, Title: Effective C++, Author: Scott Meyers

```

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

```
vns_lab_9_task_10_yuliia_lazarevych.cpp > ...
1  #include <iostream>
2  #include <fstream>
3  #include <sstream>
4  #include <string>
5  #include <vector>
6  #include <limits>
7
8  using namespace std;
9
10 //функція перевіряє, чи містить рядок слово, що починається на певну букву
11 bool containsWordStartingWith(const string& line, char letter) {
12     stringstream stream(line);
13     string word;
14     while (stream >> word) {
15         if (!word.empty() && word[0] == letter) {
16             return true;
17         }
18     }
19     return false;
20 }
21
22 //функція знаходить найкоротше слово у файлі
23 string findShortestWordInFile(const char* filename) {
24     ifstream file(filename);
25     if (!file) {
26         cerr << "Помилка відкриття файлу " << filename << endl;
27         return "";
28     }
29
30     string line, word, shortestWord;
31     size_t shortestLength = numeric_limits<size_t>::max();
32
33     while (getline(file, line)) {
34         stringstream stream(line);
35         while (stream >> word) {
36             if (word.length() < shortestLength) {
37                 shortestLength = word.length();
38                 shortestWord = word;
39             }
40         }
41     }
42
43     file.close();
44     return shortestWord;
45 }
46
47 int main() {
48     const char* file1 = "F1.txt";
49     const char* file2 = "F2.txt";
50     char targetLetter = 'A'; //'цільова' літера
51
52     //створюємо текстовий файл F1
53     ofstream f1(file1);
54     if (!f1) {
55         cerr << "Помилка створення файлу " << file1 << endl;
```

```

56         return 1;
57     }
58     f1 << "Apple banana cherry\n";
59     f1 << "Apricot melon peach\n";
60     f1 << "Grapes pineapple mango\n";
61     f1 << "Plum orange pear\n";
62     f1 << "Strawberry blueberry\n";
63     f1 << "Avocado kiwi lychee\n";
64     f1 << "Tomato cucumber carrot\n";
65     f1 << "Watermelon papaya sashahvostic\n";
66     f1 << "Cherry raspberry fig\n";
67     f1 << "Pear pomegranate grapefruit\n";
68     f1.close();
69
70     //копіюємо рядки у файл F2
71     ifstream input(file1);
72     ofstream output(file2);
73     if (!input || !output) {
74         cerr << "Помилка відкриття файлу для копіювання!" << endl;
75         return 1;
76     }
77
78     string line;
79     while (getline(input, line)) {
80         if (!containsWordStartingWith(line, targetLetter)) {
81             output << line << endl;
82         }
83     }
84
85     input.close();
86     output.close();
87
88     //шукаємо найкоротше слово у файлі F2
89     string shortestWord = findShortestWordInFile(file2);
90     if (!shortestWord.empty()) {
91         cout << "Найкоротше слово у файлі " << file2 << ": " << shortestWord << endl;
92     } else {
93         cout << "Файл " << file2 << " порожній і не містить слів!" << endl;
94     }
95
96     return 0;
97 }

```

Найкоротше слово у файлі F2.txt: fig  
PS D:\shi\epics\epic\_5>

Завдання №4.1 – Algotester Lab 4 – Variant 2.



```

algotester_lab_4.1_task_2_yuliia_lazarevych.cpp > ...
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main() {
8      int N, K;
9      cin >> N >> K;
10
11     vector<int> a(N);
12     for (int i = 0; i < N; i++) {
13         cin >> a[i];
14     }
15
16     //сортуємо масив
17     sort(a.begin(), a.end());
18
19     //видаляємо дублікати
20     auto uniqueEnd = unique(a.begin(), a.end());
21     a.erase(uniqueEnd, a.end());
22
23     //зсуваємо масив на K
24     K %= a.size();
25     rotate(a.begin(), a.begin() + K, a.end());
26
27     cout << a.size() << endl;
28     for (int num : a) {
29         cout << num << " ";
30     }
31     cout << endl;
32
33     return 0;
34 }

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.203	<a href="#">Перегляд</a>

## Завдання №4.2 – Algotester Lab 4 – Variant 2.

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  //використовуємо функцію для видалення дублікатів
7  void removeDuplicates(vector<int>& a) {
8      for (size_t i = 0; i < a.size(); i++) {
9          for (size_t j = i + 1; j < a.size(); j++) {
10             if (a[i] == a[j]) {
11                 a.erase(a.begin() + j); // Видалення дублікату
12             } else {
13                 j++;
14             }
15         }
16     }
17 }
18
19 //використовуємо функцію для сортування масиву вибором
20 void sortArray(vector<int>& a) {
21     for (size_t i = 0; i < a.size() - 1; i++) {
22         size_t minIndex = i;
23         for (size_t j = i + 1; j < a.size(); j++) {
24             if (a[j] < a[minIndex]) {
25                 minIndex = j;
26             }
27         }
28         swap(a[i], a[minIndex]);
29     }
30 }
31
32 //використовуємо функцію для зсуву масиву на K елементів
33 void rotateArray(vector<int>& a, int K) {
34     K %= a.size(); //уникаємо зайвих обертань
35     vector<int> temp(a.begin(), a.begin() + K);
36     for (size_t i = 0; i < a.size() - K; i++) {
37         a[i] = a[i + K];
38     }
39     for (size_t i = 0; i < temp.size(); i++) {
40         a[a.size() - K + i] = temp[i];
41     }
42 }
43
44 int main() {
45     int N, K;

```

```

46     cin >> N >> K;
47
48     vector<int> a(N);
49     for (int i = 0; i < N; i++) {
50         cin >> a[i];
51     }
52
53     //видаляємо дублікати
54     removeDuplicates(a);
55
56     sortArray(a);
57
58     //зсуваємо на K
59     rotateArray(a, K);
60
61     cout << a.size() << endl;
62     for (int num : a) {
63         cout << num << " ";
64     }
65     cout << endl;
66
67     return 0;
68 }

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.430	<a href="#">Перегляд</a>

## Завдання №5 – Algotester Lab 6 – Variant 3.

```

G algotester_lab_6_task_3_yuliia_lazarevych.cpp > ...
1  #include <iostream>
2  #include <vector>
3  #include <set>
4  #include <string>
5
6  using namespace std;
7
8  vector<int> findNumbers(const vector<string> &sudoku, int x, int y) {
9      vector<int> possibleNumbers;
10     int N = sudoku.size();
11     if (sudoku[x][y] != '0') {
12         possibleNumbers.push_back(sudoku[x][y] - '0');
13         return possibleNumbers;
14     }
15     set<int> usedNumbers;
16     for (int i = 0; i < N; i++) {
17         if (sudoku[x][i] != '0') usedNumbers.insert(sudoku[x][i] - '0');
18         if (sudoku[i][y] != '0') usedNumbers.insert(sudoku[i][y] - '0');
19     }
20     for (int num = 1; num <= N; num++) {
21         if (usedNumbers.find(num) == usedNumbers.end()) {
22             possibleNumbers.push_back(num);
23         }
24     }
25     return possibleNumbers;
26 }
27
28 int main() {
29     int N;
30     cin >> N;
31     vector<string> sudoku(N);
32     for (int i = 0; i < N; i++) {
33         cin >> sudoku[i];
34     }
35     int Q;
36     cin >> Q;
37     vector<pair<int, int>> queries(Q);
38
39     for (int i = 0; i < Q; i++) {
40         int x, y;
41         cin >> x >> y;
42         queries[i] = {x - 1, y - 1};
43     }
44
45     for (const auto &query : queries) {
46         int x = query.first;
47         int y = query.second;
48         vector<int> result = findNumbers(sudoku, x, y);
49         cout << result.size() << endl;
50         for (int num : result) {
51             cout << num << " ";
52         }
53         cout << endl;
54     }
55     return 0;
56 }

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.242	<a href="#">Перегляд</a>

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

```
practice_work_team_tasks_1_yuliia_lazarevych.cpp > ...
1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4
5  using namespace std;
6
7  // Перелік статусів операцій з файлами
8  enum FileOpResult { Success, Failure, InvalidInput, WriteError, CloseError };
9
10 //використовую функцію для роботи з текстовими файлами
11 FileOpResult write_to_file(const char *name, const char *content) {
12     //використовую перевірку вхідних параметрів для стійкості до різних вхідних даних
13     if (name == nullptr || content == nullptr || *name == '\0') {
14         cerr << "Помилка: Недійсні вхідні дані.\n";
15         return InvalidInput;
16     }
17
18     //відкриваю файл для запису в текстовий файл
19     ofstream file(name, ios::out | ios::trunc);
20     if (!file) {
21         cerr << "Помилка: Не вдалося створити або відкрити файл.\n";
22         return Failure;
23     }
24
25     //записую вмісту у файл
26     file << content;
27     if (!file) {
28         cerr << "Помилка: Не вдалося записати дані у файл.\n";
29         return WriteError;
30     }
31
32     //закриваю файл
33     file.close();
34     if (!file) {
35         cerr << "Помилка: Не вдалося коректно закрити файл.\n";
36         return CloseError;
37     }
38
39     return Success;
40 }
41
42 //використовую функцію для роботи з бінарними файлами
43 FileOpResult write_binary_file(const char *name, const char *data, size_t size) {
44     //перевіряю вхідні параметри для стійкості до різних вхідних даних
45     if (name == nullptr || data == nullptr || *name == '\0') {
```

```

46     cerr << "Помилка: Недійсні вхідні дані.\n";
47     return InvalidInput;
48 }
49
50 //відкриваю файл для запису в бінарний файл
51 ofstream file(name, ios::out | ios::binary | ios::trunc);
52 if (!file) {
53     cerr << "Помилка: Не вдалося створити або відкрити файл.\n";
54     return Failure;
55 }
56
57 //записую бінарний вміст у файл
58 file.write(data, size);
59 if (!file) {
60     cerr << "Помилка: Не вдалося записати дані у файл.\n";
61     return WriteError;
62 }
63
64 //закриваю файл
65 file.close();
66 if (!file) {
67     cerr << "Помилка: Не вдалося коректно закрити файл.\n";
68     return CloseError;
69 }
70
71 return Success;
72 }
73
74 int main() {
75     //робота з текстовими файлами
76     char fileName[256];
77     string content;
78
79     cout << "Введіть ім'я текстового файлу: ";
80     cin.getline(fileName, sizeof(fileName));
81
82     cout << "Введіть вміст для запису у текстовий файл: ";
83     getline(cin, content);
84
85     //записую у текстовий файл
86     FileOpResult result = write_to_file(fileName, content.c_str());
87     if (result == Success) {
88         cout << "Текстовий файл успішно записаний.\n";

```

```

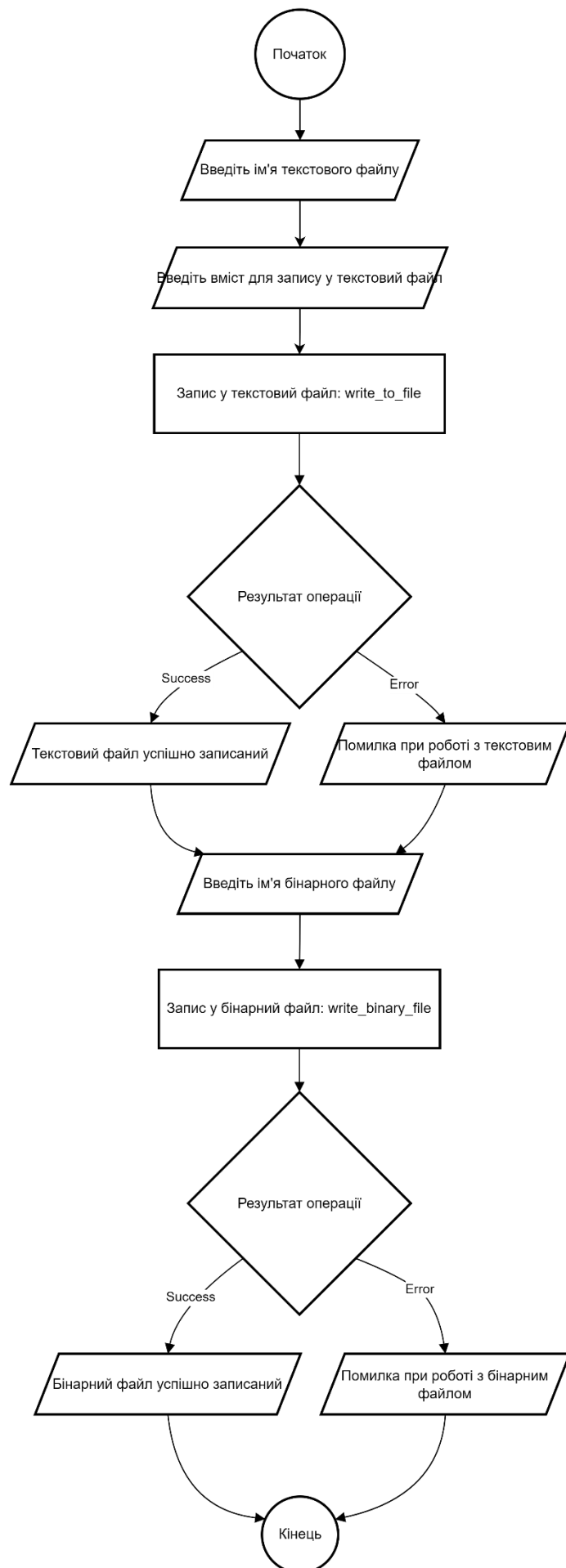
89     } else {
90         cout << "Помилка при роботі з текстовим файлом.\n";
91     }
92
93     //робота з бінарними файлами
94     cout << "Введіть ім'я бінарного файлу: ";
95     cin.getline(fileName, sizeof(fileName));
96
97     char binaryData[] = {0x01, 0x02, 0x03, 0x04, 0x05};
98     size_t binarySize = sizeof(binaryData);
99
100    //записую у бінарний файл
101    result = write_binary_file(fileName, binaryData, binarySize);
102    if (result == Success) {
103        cout << "Бінарний файл успішно записаний.\n";
104    } else {
105        cout << "Помилка при роботі з бінарним файлом.\n";
106    }
107
108    return 0;
109 }

```

```

Введіть ім'я текстового файлу: pisnya
Введіть вміст для запису у текстовий файл: sigma sigma boy sigma boy sigma boy
Текстовий файл успішно записаний.
Введіть ім'я бінарного файлу: pisnya
Бінарний файл успішно записаний.

```



## Завдання №7 – Class Practice Work 2 – Копіювання вмісту файлу у інший файл.

```
practice_work_team_tasks_2_yuliia_lazarevych.cpp > ...
1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  //перелік статусів операцій із файлами
7  enum FileOpResult { Success, Failure, InvalidInput };
8
9  //використовую функцію копіювання вмісту з одного файлу в інший
10 FileOpResult copy_file(const char *file_from, const char *file_to) {
11     //перевіряю вхідних параметрів
12     if (file_from == nullptr || file_to == nullptr || *file_from == '\0' || *file_to == '\0') {
13         cerr << "Помилка: Недійсні вхідні параметри.\n";
14         return InvalidInput;
15     }
16
17     //відкриваю файл для читання
18     ifstream source(file_from, ios::in | ios::binary);
19     if (!source) {
20         cerr << "Помилка: Не вдалося відкрити файл для читання: " << file_from << "\n";
21         return Failure;
22     }
23
24     //відкриваю файл для запису
25     ofstream destination(file_to, ios::out | ios::binary | ios::trunc);
26     if (!destination) {
27         cerr << "Помилка: Не вдалося створити або відкрити файл для запису: " << file_to << "\n";
28         return Failure;
29     }
30
31     //копіюю вміст
32     destination << source.rdbuf();
33     if (!destination) {
34         cerr << "Помилка: Не вдалося записати дані у файл.\n";
35         return Failure;
36     }
37
38     //закриваю файл
39     source.close();
40     if (!source) {
41         cerr << "Помилка: Не вдалося коректно закрити вхідний файл.\n";
42         return Failure;
43     }
44
45     destination.close();
```

```

46     if (!destination) {
47         cerr << "Помилка: Не вдалося коректно закрити вихідний файл.\n";
48         return Failure;
49     }
50
51     return Success;
52 }
53
54 int main() {
55     char fileFrom[256], fileTo[256];
56
57     cout << "Введіть ім'я файлу для копіювання (file_from): ";
58     cin.getline(fileFrom, sizeof(fileFrom));
59
60     cout << "Введіть ім'я файлу призначення (file_to): ";
61     cin.getline(fileTo, sizeof(fileTo));
62
63     //викликаю функцію копіювання
64     FileOpResult result = copy_file(fileFrom, fileTo);
65
66     if (result == Success) {
67         cout << "Файл успішно скопійовано.\n";
68     } else {
69         cout << "Сталася помилка під час копіювання файлу.\n";
70     }
71
72     return 0;
73 }

```

```

Введіть ім'я файлу для копіювання (file_from): hello
Введіть ім'я файлу призначення (file_to): pisnya
Файл успішно скопійовано.

```

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



```

practice_work_self_algotester_tasks_yuliia_lazarevych.cpp > ...
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <set>
5  using namespace std;
6
7  int main() {
8      int N;
9      cin >> N;
10     vector<int> arr(N);
11     for (int i = 0; i < N; ++i) {
12         cin >> arr[i];
13     }
14
15     //видаляємо дублікати
16     sort(arr.begin(), arr.end()); //спершу сортуємо для застосування std::unique
17     arr.erase(unique(arr.begin(), arr.end()), arr.end()); //видаляємо дублікати
18
19     //створюю три групи за остачею від ділення на 3
20     vector<int> group0, group1, group2;
21     for (int x : arr) {
22         if (x % 3 == 0) group0.push_back(x);
23         else if (x % 3 == 1) group1.push_back(x);
24         else group2.push_back(x);
25     }
26
27     //сортую групи
28     sort(group0.begin(), group0.end()); //за зростанням
29     sort(group1.rbegin(), group1.rend()); //за спаданням
30     sort(group2.begin(), group2.end()); //за зростанням
31
32     //об'єдную всі групи
33     vector<int> result;
34     result.insert(result.end(), group0.begin(), group0.end());
35     result.insert(result.end(), group1.begin(), group1.end());
36     result.insert(result.end(), group2.begin(), group2.end());
37
38     cout << result.size() << endl;
39     for (int x : result) {
40         cout << x << " ";
41     }
42     cout << endl;
43
44     return 0;
45 }

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.422	<a href="#">Перегляд</a>

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

```

bonus_task.cpp > ...
1 //lab 3 without stl
2 #include <iostream>
3 #include <vector>
4 using namespace std;
5
6 void sortAscending(vector<int>& arr) {
7     for (int i = 0; i < arr.size(); ++i) {
8         for (int j = i + 1; j < arr.size(); ++j) {
9             if (arr[i] > arr[j]) {
10                 swap(arr[i], arr[j]);
11             }
12         }
13     }
14 }
15
16 void sortDescending(vector<int>& arr) {
17     for (int i = 0; i < arr.size(); ++i) {
18         for (int j = i + 1; j < arr.size(); ++j) {
19             if (arr[i] < arr[j]) {
20                 swap(arr[i], arr[j]);
21             }
22         }
23     }
24 }
25
26 int main() {
27     int N;
28     cin >> N;
29
30     vector<int> arr(N);
31     for (int i = 0; i < N; ++i) {
32         cin >> arr[i];
33     }
34
35     //видаляємо дублікати вручну
36     vector<int> uniqueArr;
37     for (int i = 0; i < N; ++i) {
38         for (int j = 0; j < N; ++j) {
39             bool isDuplicate = false;
40             for (int j = 0; j < uniqueArr.size(); ++j) {
41                 if (arr[i] == uniqueArr[j]) {
42                     isDuplicate = true;
43                     break;
44                 }
45             }
46             if (!isDuplicate) {
47                 uniqueArr.push_back(arr[i]);
48             }
49         }
50
51         //поділяємо на групи
52         vector<int> group0, group1, group2;
53         for (int x : uniqueArr) {
54             if (x % 3 == 0) group0.push_back(x);
55             else if (x % 3 == 1) group1.push_back(x);
56             else group2.push_back(x);
57         }
58
59         //сортуємо групи вручну
60         sortAscending(group0);
61         sortDescending(group1);
62         sortAscending(group2);
63
64         //об'єднуємо групи
65         vector<int> result;
66         result.insert(result.end(), group0.begin(), group0.end());
67         result.insert(result.end(), group1.begin(), group1.end());
68         result.insert(result.end(), group2.begin(), group2.end());
69
70         cout << result.size() << endl;
71         for (int x : result) {
72             cout << x << " ";
73         }
74     }
75 }

```

```

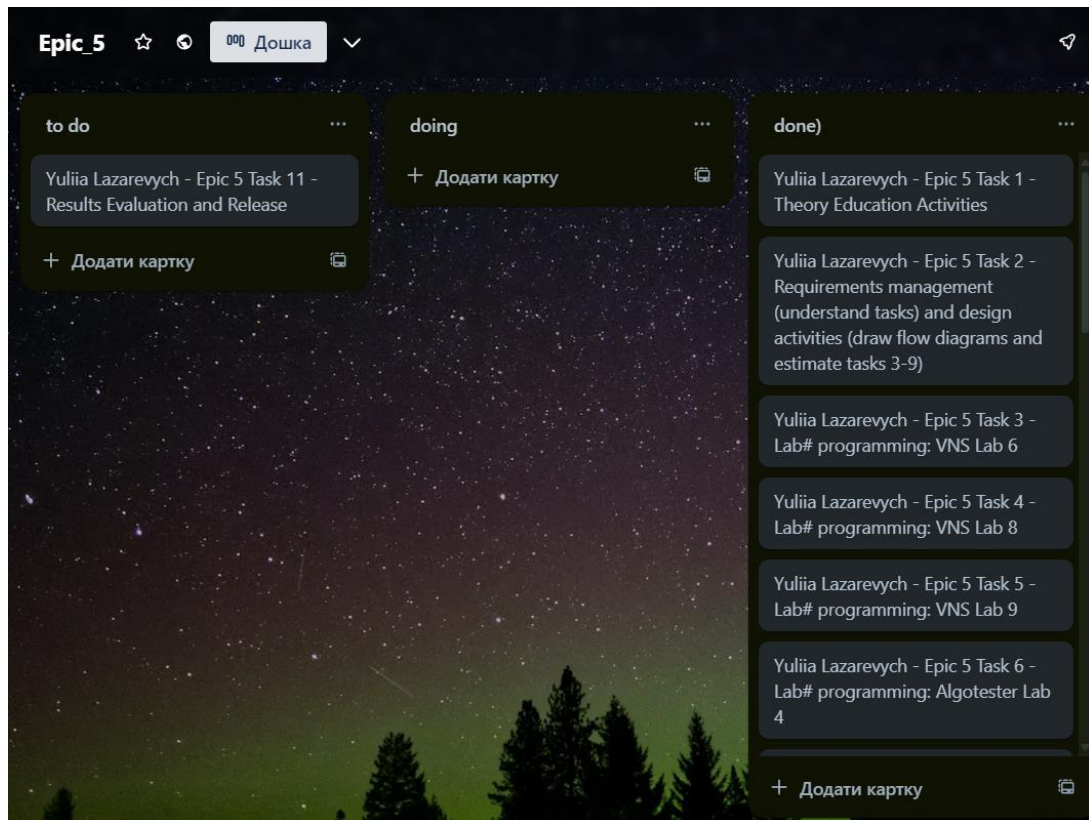
73     cout << endl;
74
75     return 0;
76 }

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.430	<a href="#">Перегляд</a>

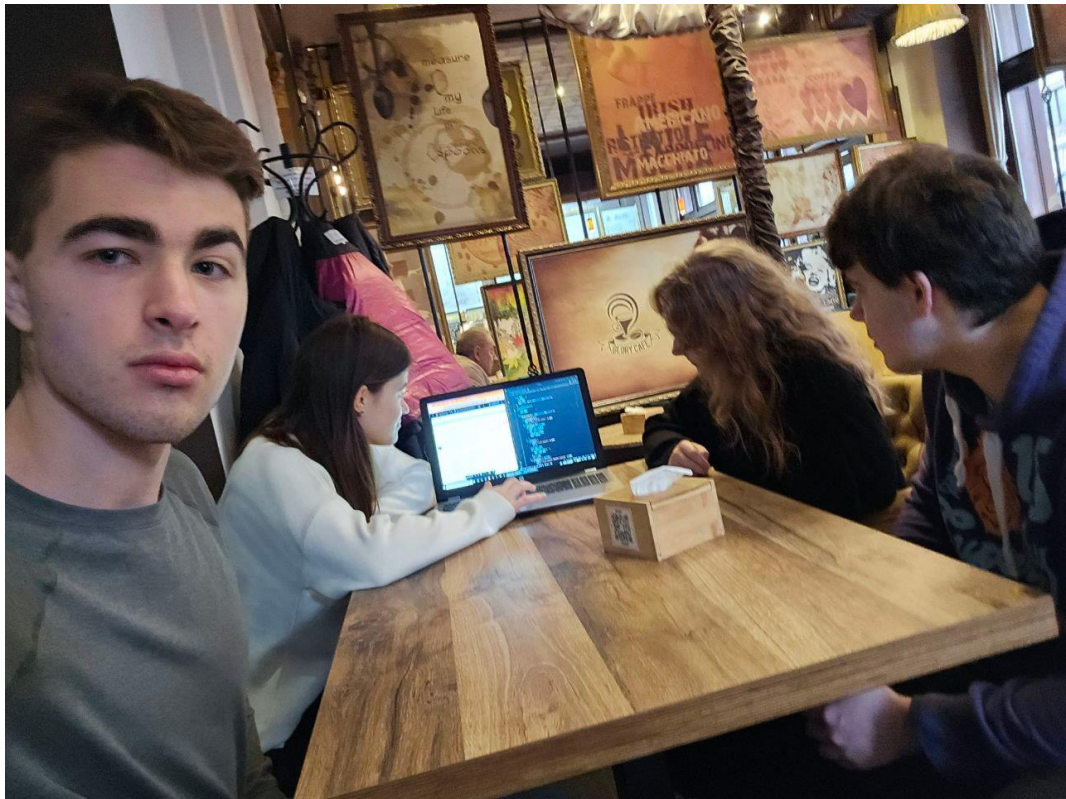
### 3. Кооперація з командою:

- 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.