Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 5 (лабораторні № 6, 8, 9)**

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

***з розділу***: «Файли. Системи числення. Бінарні Файли. Символи та Рядкові Змінні. Текстові Файли. Стандартна бібліотека. Створення й використання бібліотек.»

***Виконав:***

студент групи **ШІ-14**

**Нападайло Микита Олександрович**

# **Тема роботи:**

Робота з текстовими та бінарними файлами, символами та рядковими символами; знайомство з системами числення та стандартною бібліотекою.

# **Мета роботи:**

# Оволодіння роботою з файлами у С++, включаючи зчитування та запис файлів.

# Розуміння основних систем числення та їх використання у програмуванні.

# Навчання роботи з бінарними файлами для зберігання та опрацювання даних.

# Розгляд концепцій символів та рядкових змінних, їх операцій та використання у програмуванні.

# Вивчення обробки текстових файлів та роботи зі стрічками у С++.

# Освоєння стандартної бібліотеки С++, що включає функції для оптимізації обробки файлів та рядків.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Текстові та бінарні файли.
* Тема №2: Символи та рядкові змінні.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Текстові та бінарні файли.
  + Джерела Інформації
    - Відео <https://youtu.be/Cz4fl-TUjVk> .
  + Що опрацьовано:
    - Як записувати інформацію в файли та зчитувати її з них
  + Статус: Ознайомлений
  + Початок опрацювання теми: 07.12.2023
  + Завершення опрацювання теми: 09.12.2023
* Тема №2: Символи та рядкові змінні.
  + - Відео <https://youtu.be/zo6B_eJmeLo>
  + Що опрацьовано:
    - Як реалізовані рядкові змінні в С++, як ними користуватися, корисні функції пов’язані зі string.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 05.12.2023
  + Завершення опрацювання теми: 07.12.2023

# **Виконання роботи:**

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

Завдання №1 **VNS Lab 6 - Task 1-N**

* Варіант завдання – 18
* Деталі завдання: 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, і всі слова рядка, які починаються із цифри, відсортувати за спаданням.

Завдання №2 **VNS Lab 8 - Task 1-N**

* Варіант завдання - 18
* Деталі завдання: Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вмістиме, 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 "Книга":

- назва;

- автор;

- рік видання;

- кількість сторінок.

Знищити 3 елементи з початку файлу, додати елемент перед елементом із зазначеною назвою.

Завдання №3 **VNS Lab 9 - Task 1-N**

* Варіант завдання - 18
* Деталі завдання: Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію. Скопіювати з файлу F1 у файл F2 всі рядки, у яких немає однакових слів. Визначити кількість голосних букв у першому рядку файлу F2.

Завдання №4 **Algotester Lab 4v3 (варіант 1)**

* Варіант завдання – 3
* Деталі завдання: 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.

Завдання №5 **Algotester Lab 4v3 (варіант 2)**

* Варіант завдання – 3
* Деталі завдання: 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.

- Важливі деталі для врахування в імплементації програми: Не використовувати засоби STL

Завдання №6 **Algotester Lab 6v3**

* Варіант завдання – 3
* Деталі завдання: 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.

Завдання №7 **Class Practice Work**

* Деталі завдання: Реалізувати функцію створення файла і запису в нього даних: запис текстової стрічки у файл із заданим ім’ям; копіювання вмісту файла у інший файл.

Завдання №8 **Self-Practice Work 1**

* Варіант завдання: **The Algo Battles 2023 - Етап 7 - А**
* Деталі завдання: Петрик любить грати в ігри, де світ складається з кубів. Сьогодні він вирішив написати свій варіант такої гри. Петрик хоче, щоб у цій грі можна було програмувати нескладні механізми на основі електричного сигналу. Вважається, що сигнал має певну додатну силу, яка позначається цілим числом. Також, сигнал може бути відсутнім, що позначається як сигнал силою 0. Петрик додав в гру три блоки, які по-різному обробляють електричний сигнал:
  + Кабель отримує сигнал силою k і видає сигнал силою max(k−1,0).
  + Повторювач підсилює сигнал до рівня 15 при ненульовому вхідному сигналі, і не видає сигнал при його відсутності. Формально, нехай сила вхідного сигналу k. Тоді повторювач видає сигнал силою 15, якщо k≥1, і сигнал силою 0, якщо k=0.
  + Змінювач не видає сигнал при ненульовому вхідному сигналі та видає сигнал силою 15 при відсутності вхідного сигналу. Формально, нехай сила вхідного сигналу k. Тоді змінювач видає сигнал силою 0, якщо k≥1, і сигнал силою 15, якщо k=0.

Петрик задає вам електричну лінію, в якій кабелі, повторювачі і змінювачі позначені літерами k, p, z відповідно. На перший елемент лінії подано сигнал силою 15. Допоможіть Петрику знайти вихідний сигнал останнього блоку в лінії.

Завдання №9 **Self-Practice Work 2**

* Варіант завдання **Algotester Lab 4v2**
* Деталі завдання: 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.

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

Програма №1 **VNS Lab 6 - Task 1-N**

* Блок-схема

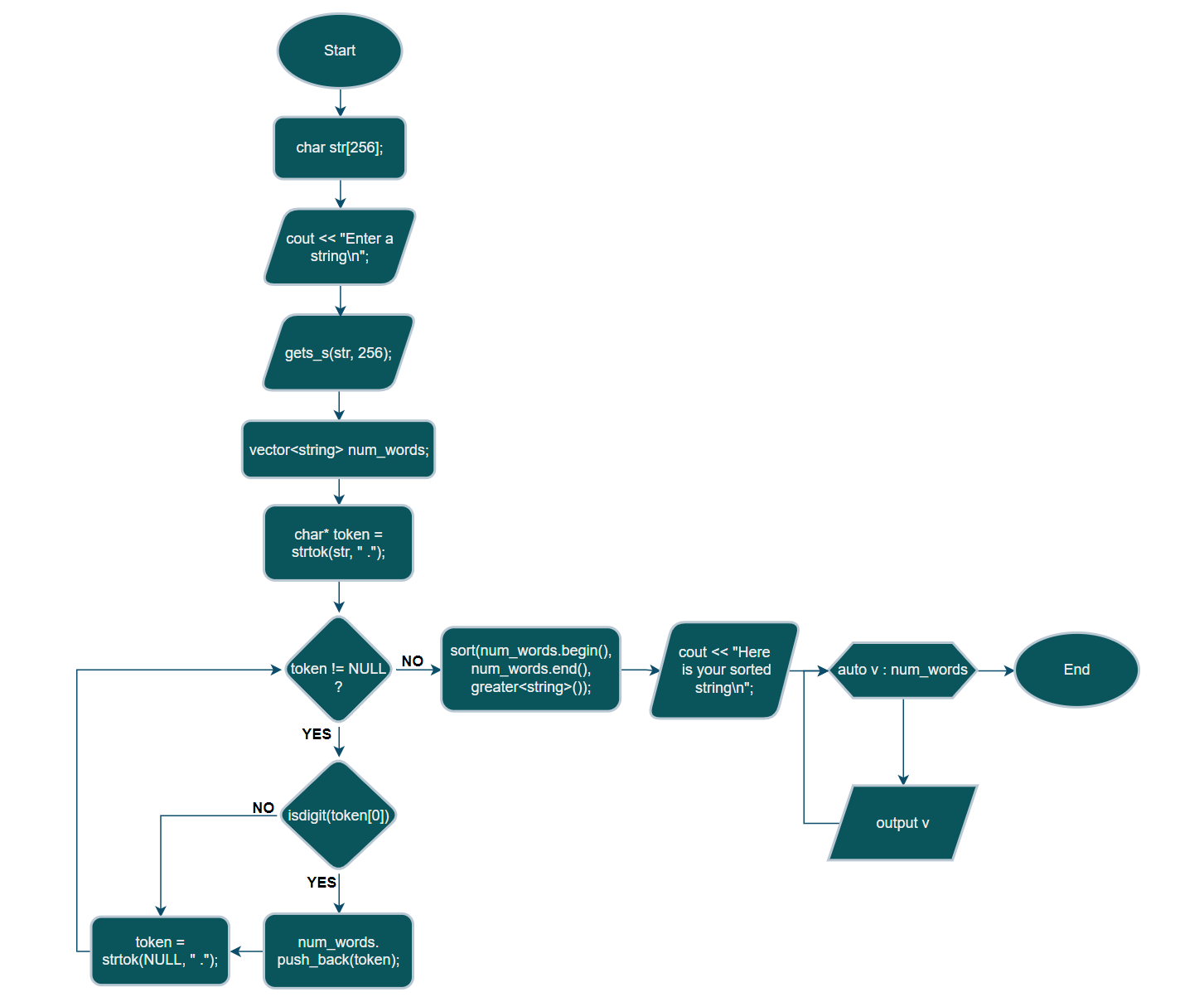


Рисунок 1: блок-схема до програми 1

* Планований час на реалізацію – 2 години

Програма №2 **VNS Lab 8 - Task 1-N**

* Планований час на реалізацію – 3 години

Програма №3 **VNS Lab 9 - Task 1-N**

* Планований час на реалізацію – 3 години

Програма №4 **Algotester Lab 4v3 (варіант 1)**

* Планований час на реалізацію – 1 година

Програма №5 **Algotester Lab 4v3 (варіант 2)**

* Планований час на реалізацію – 2 години

Програма №6 **Algotester Lab 6v3**

* Планований час на реалізацію – 1 година

Програма №7 **Class Practice Work**

* Планований час на реалізацію 1 година

Програма №8 **Self-Practice Work 1**

* Планований час на реалізацію: 15 хвилин

Програма №9 **Self-Practice Work 2**

* Планований час на реалізацію: 30 хвилин

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

Використав конфігурацію з Епіку 1.

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

Завдання №1 **VNS Lab 6- Task 1-N**

#include <iostream>

#include <vector>

#include <algorithm>

#include <stdio.h>

#include <string.h>

using namespace std;

int main()

{

    char str[256];

    cout << "Enter your string:\n";

    gets\_s(str, 256);

    vector<string> num\_words;

    char\* token = strtok(str, " .");

    while (token != NULL)

    {

        if (isdigit(token[0]))

        {

            num\_words.push\_back(token);

        }

        token = strtok(NULL, " .");

    }

    sort(num\_words.begin(), num\_words.end(), greater<string>());

    cout << "Here are the sorted words that start with numbers:\n";

    for (auto v : num\_words)

    {

        cout << v << " ";

    }

}

Завдання №2 **VNS Lab 8 - Task 1-N**

#include <iostream>

#include <vector>

#include <fstream>

#include <sstream>

using namespace std;

struct Book {

    char name[30];

    char author[30];

    int year;

    int pages;

};

void FillFile(vector<Book> books, string FileName);

void DeleteThreeBooks(vector<Book> &books, const char\* FileName);

bool AddBook(const char\* FileName);

void PrintFile(const char\* FileName);

int main()

{

   vector<Book> books = {

        {"1984\t", "Orwell\t", 1948, 312},

        {"Meditations", "Marcus Aurelius", 175, 180},

        {"Faust\t", "Goethe\t", 1808, 416},

        {"Hamlet\t", "Shakespear", 1600, 238},

        {"Metamorphosis", "Kafka\t", 1915, 160}};

    FillFile(books, "file1.bin");

    PrintFile("file1.bin");

    DeleteThreeBooks(books, "file1.bin");

    if (!AddBook("file1.bin")) return 1;

    PrintFile("file1.bin");

    return 0;

}

void FillFile(vector<Book> books, string FileName)

{

    ofstream fout(FileName);

    if (!fout)

    {

        cerr << "An error occurred while opening/creating the file" << endl;

        return;

    }

    for (const Book b : books)

    {

        fout << b.name << "\t" << b.author << "\t" << b.year << "\t" << b.pages << endl;

    }

    fout.close();

}

void DeleteThreeBooks(vector<Book> &books, const char\* FileName)

{

    if (books.size() >= 3)

    {

        books.erase(books.begin(), books.begin() + 3);

    }

    ofstream fout("temp.bin");

    if (!fout)

    {

        cerr << "An error occurred while deleting first 3 books" << endl;

        return;

    }

    for (const Book b : books)

    {

        fout << b.name << "\t" << b.author << "\t" << b.year << "\t" << b.pages << endl;

    }

    fout.close();

    remove(FileName);

    rename("temp.bin", FileName);

}

bool AddBook(const char\* FileName)

{

    Book newbook;

    cout << "Enter the information about a book you want to add:\n";

    cout << "Title: ";

    cin >> newbook.name;

    cout << "Author: ";

    cin >> newbook.author;

    cout << "Year: ";

    cin >> newbook.year;

    cout << "Pages: ";

    cin >> newbook.pages;

    string beforeBook;

    cout << "Before which book do you want to insert a new one?\n";

    cin >> beforeBook;

    ifstream fin1(FileName);

    if (!fin1)

    {

        cerr << "An error occurred while reading data from the file" << endl;

        return false;

    }

    ofstream fout2("temp.bin");

    if (!fout2)

    {

        cerr << "An error occurred while writing data into the file" << endl;

        return false;

    }

    string line;

    bool found = false;

    while (getline(fin1, line))

    {

        istringstream iss(line);

        string w;

        iss >> w;

        if (w == beforeBook)

        {

            fout2 << newbook.name << "\t" << newbook.author << "\t" << newbook.year << "\t" << newbook.pages << endl;

            found = true;

        }

        fout2 << line << endl;

    }

    if (!found)

    {

        cerr << "This book doesn't exist." << endl;

        return false;

    }

    fin1.close();

    fout2.close();

    remove(FileName);

    rename("temp.bin", FileName);

    return true;

}

void PrintFile(const char\* FileName)

{

    ifstream fin(FileName);

    if (!fin)

    {

        cerr << "An error occurred while printing the file's content" << endl;

        return;

    }

    string str;

    while (getline(fin, str))

    {

        cout << str << endl;

    }

    fin.close();

}

Завдання №3 **VNS Lab 9 - Task 1-N**

#include <iostream>

#include <vector>

#include <fstream>

#include <sstream>

#include <cctype>

using namespace std;

string CleanWord(string &word)

{

    string cleaned;

    for (char c : word)

    {

        if (isalnum(c))

        {

            cleaned += tolower(c);

        }

    }

    return cleaned;

}

int main()

{

    string text[] =

    {"Binary files store data in a compact format, using sequences of bytes.\n",

    "Text files, in contrast, contain readable text encoded in ASCII or Unicode.\n",

    "Reading and writing binary files require dealing with raw data at a byte level.\n",

    "Text files, being human-readable, are commonly used for documents and configuration files.\n",

    "Binary files are suitable for preserving complex structures and non-textual data.\n",

    "Handling text files involves operations on strings and characters.\n",

    "When working with binary files, attention to data representation and endianness is crucial.\n",

    "Textual content in files facilitates easier comprehension and editing by humans.\n",

    "The <fstream> library in C++ offers classes for file handling tasks.\n",

    "Understanding the differences between these file types aids in selecting the appropriate format for specific tasks.\n"};

    ofstream fout1("F1.txt");

    if (!fout1)

    {

        cerr << "An error occurred" << endl;

        return 1;

    }

    for (const string str : text)

    {

        fout1 << str;

    }

    string line, word;

    fout1.close();

    ifstream fin1("F1.txt");

    ofstream fout2("F2.txt");

    if (!fin1 || !fout2)

    {

        cerr << "An error occurred" << endl;

        return 1;

    }

    while(getline(fin1, line))

    {

        vector<string> line\_words;

        istringstream iss(line);

        bool unique = true;

        while (iss >> word)

        {

            line\_words.push\_back(CleanWord(word));

        }

        for (int i = 0; i < line\_words.size() && unique; i++)

        {

            for (int j = i + 1; j < line\_words.size(); j++)

            {

                if (line\_words[i] == line\_words[j])

                {

                    unique = false;

                    break;

                }

            }

        }

        if (unique)

        {

            fout2 << line << endl;

        }

    }

    fout2.close(), fin1.close();

    ifstream fin2("F2.txt");

    if (!fin2)

    {

        cerr << "An error occurred" << endl;

        return 1;

    }

    string first\_line;

    int vowels = 0;

    getline(fin2, first\_line);

    for (char c : first\_line)

    {

        if (c = tolower(c) && (c == 'a' || c == 'o' || c == 'e' || c == 'i' || c == 'u' || c == 'y')) vowels++;

    }

    fin2.close();

    cout << "There are " << vowels << " vowels in the first line\n";

    return 0;

}

Завдання №4 **Algotester Lab 4v3 (варіант 1)**

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int main()

{

    vector<int> v1;

    int n;

    cin >> n;

    for (int i = 0; i < n; i++)

    {

        int el;

        cin >> el;

        v1.push\_back(el);

    }

    auto firstPartition = partition(v1.begin(), v1.end(), [](int x)

    {

        return x % 3 == 0;

    });

    auto secondPartition = partition(firstPartition, v1.end(), [](int x)

    {

        return x % 3 == 1;

    });

    sort(v1.begin(), firstPartition);

    sort(firstPartition, secondPartition, greater<int>());

    sort(secondPartition, v1.end());

    auto rep = unique(v1.begin(), v1.end());

    v1.erase(rep, v1.end());

    cout << v1.size() << endl;

    for (auto a: v1)

    {

        cout << a << ' ';

    }

}

Завдання №5 **Algotester Lab 4v3 (варіант 2)**

#include <iostream>

#include <vector>

using namespace std;

void QuickSort(vector<int>& arr, int low, int high)

{

    if (low < high)

    {

        int pivot = arr[high];

        int i = low - 1;

        for (int j = low; j <= high - 1; j++) {

            if (arr[j] < pivot)

            {

                i++;

                int temp = arr[i];

                arr[i] = arr[j];

                arr[j] = temp;

            }

        }

        int temp = arr[i + 1];

        arr[i + 1] = arr[high];

        arr[high] = temp;

        int pivotIndex = i + 1;

        QuickSort(arr, low, pivotIndex - 1);

        QuickSort(arr, pivotIndex + 1, high);

    }

}

void SortDown(vector<int>& arr)

{

    int n = arr.size() - 1;

    QuickSort(arr, 0, n);

    for (int i = 0; i < n; i++, n--)

    {

        int temp = arr[i];

        arr[i] = arr[n];

        arr[n] = temp;

    }

}

void removeDuplicates(vector<int>& arr)

{

    int n = arr.size();

    if (n <= 1)

        return;

    int j = 0;

    for (int i = 0; i < n - 1; i++)

    {

        if (arr[i] != arr[i + 1])

            arr[j++] = arr[i];

    }

    arr[j++] = arr[n - 1];

    arr.erase(arr.begin() + j, arr.end());

}

int main()

{

    vector<int> v1;

    int n;

    cin >> n;

    for (int i = 0; i < n; i++)

    {

        int el;

        cin >> el;

        v1.push\_back(el);

    }

    vector<int> remainderZero, remainderOne, remainderTwo;

    for (int i = 0; i < n; i++)

    {

        if (v1[i] % 3 == 0)

            remainderZero.push\_back(v1[i]);

        else if (v1[i] % 3 == 1)

            remainderOne.push\_back(v1[i]);

        else

            remainderTwo.push\_back(v1[i]);

    }

    QuickSort(remainderZero, 0, remainderZero.size() - 1);

    SortDown(remainderOne);

    QuickSort(remainderTwo, 0, remainderTwo.size() - 1);

    removeDuplicates(remainderZero);

    removeDuplicates(remainderTwo);

    removeDuplicates(remainderOne);

    cout << remainderZero.size() + remainderOne.size() + remainderTwo.size() << endl;

    for (auto value : remainderZero)

        cout << value << " ";

    for (auto value : remainderOne)

        cout << value << " ";

    for (auto value : remainderTwo)

        cout << value << " ";

    return 0;

}

Завдання №6 **Algotester Lab 6v3**

#include <iostream>

#include <string>

using namespace std;

int main()

{

    int N, Q;

    cin >> N;

    string field[N];

    for (int i = 0; i < N; i++)

    {

        cin >> field[i];

    }

    cin >> Q;

    int xy[Q][2];

    for (int i = 0; i < Q; i++)

    {

        cin >> xy[i][0] >> xy[i][1];

        xy[i][0]--, xy[i][1]--;

    }

    for (int i = 0; i < Q; i++)

    {

        int M = 0;

        string answers = "";

        if (field[xy[i][0]][xy[i][1]] != '0')

        {

            cout << '1' << endl;

            cout << field[xy[i][0]][xy[i][1]] << endl << endl;

            continue;

        }

        for (int num = 1; num <= N; num++)

        {

            bool meets = false;

            for (int p = 0; p < N; p++)

            {

                if (field[xy[i][0]][p] - '0' == num || field[p][xy[i][1]] - '0' == num)

                {

                    meets = true;

                    break;

                }

            }

            if (!meets)

            {

                M++;

                answers += '0' + num;

            }

        }

        cout << M << endl;

        if (M > 0)

        {

        for (int j = 0; j < answers.length(); j++)

        {

            cout << answers[j] << ' ';

        }

        cout << endl;

        }

        cout << endl;

    }

}

Завдання №7 **Class Practice Work**

#include <iostream>

#include <fstream>

using namespace std;

enum FileOpResult {Success, Failure};

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

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

int main()

{

    // Task 1

    char content[100], name[50];

    cout << "What do you want to write into the file?\n";

    cin.getline(content, sizeof(content));

    cout << "In which file to you want to write it?\n";

    cin.getline(name, sizeof(name));

    if (write\_to\_file(name, content) == Success)

    {

        cout << "The text was successfully written\n";

    }

    else cout << "Something went wrong\n";

    // Task 2

    char file\_from[50], file\_to[50];

    cout << "From what file do you want to copy?\n";

    cin.getline(file\_from, sizeof(file\_from));

    cout << "To what file do you want to copy?\n";

    cin.getline(file\_to, sizeof(file\_to));

    copy\_file(file\_from, file\_to);

    if (copy\_file(file\_from, file\_to) == Success)

    cout << "Text copied successfully\n";

}

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

{

    ofstream fout(name);

    if (!fout) return Failure;

    fout << content;

    fout.close();

    return Success;

}

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

{

    ifstream fin(file\_from);

    ofstream fout(file\_to);

    if (!fin)

    {

        cerr << "File " << file\_from << " does not exist" << endl;

        fin.close();

        return Failure;

    }

    if (!fout)

    {

        cerr << "An error occured while writing into the file " << file\_to << endl;

        fout.close();

        return Failure;

    }

    string str;

    while (getline(fin, str))

    {

        fout << str << endl;

    }

    fin.close(), fout.close();

    return Success;

}

Завдання №8 **Self-Practice Work 1**

#include <iostream>

using namespace std;

int main()

{

    int n;

    string sequence;

    cin >> n;

    cin.ignore();

    getline(cin, sequence);

    int signal = 15;

    for (int i = 0; i < n; i++)

    {

        switch (sequence[i])

        {

            case 'k':

            {

                signal = max(signal - 1, 0);

                break;

            }

            case 'p':

            {

                if (signal >= 1) signal = 15;

                else signal = 0;

                break;

            }

            case 'z':

            {

                if (signal >= 1) signal = 0;

                else signal = 15;

                break;

            }

        }

    }

    cout << signal;

}

Завдання №9 **Self-Practice Work 2**

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int main()

{

    int N, K;

    cin >> N >> K;

    vector<int> a;

    for (int i = 0; i < N; i++)

    {

        int el;

        cin >> el;

        a.push\_back(el);

    }

    sort(a.begin(), a.end());

    auto last = unique(a.begin(), a.end());

    a.erase(last, a.end());

    K %= a.size();

    rotate(a.begin(), a.begin() + K, a.end());

    cout << a.size() << endl;

    for (auto element : a)

    {

        cout << element << ' ';

    }

}

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

Завдання №1 **VNS Lab 6 - Task 1-N**

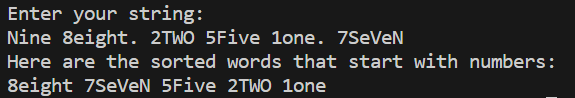


Рисунок 2: результат виконання програми 1

Час затрачений на виконання завдання: 1 година 30 хвилин

Завдання №2 **VNS Lab 8 - Task 1-N**

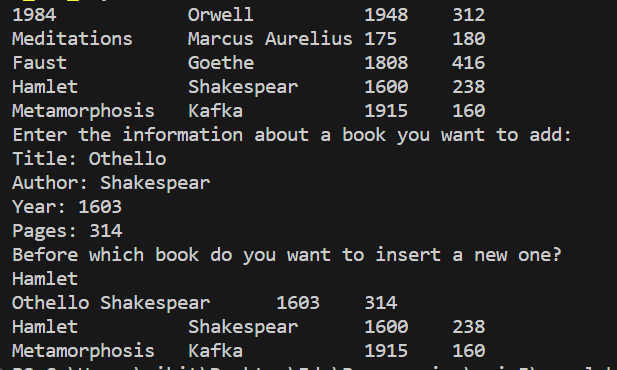


Рисунок 3: результат виконання програми 2

Час затрачений на виконання завдання: 2 години

Завдання №3 **VNS Lab 9 - Task 1-N**



Рисунок 4: результат виконання програми 3(консоль)

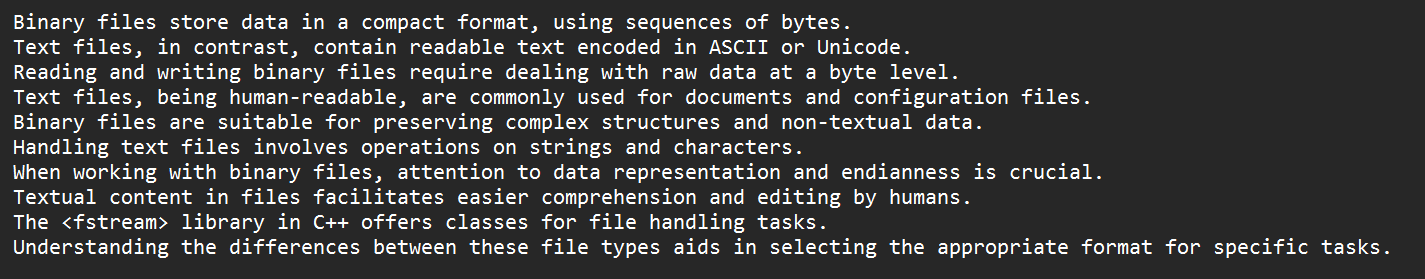


Рисунок 5: результат виконання програми 3(F1.txt)

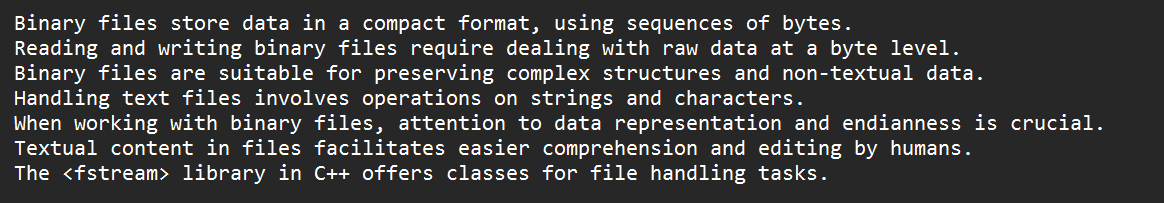


Рисунок 6: результат виконання програми 3(F2.txt)

Час затрачений на виконання завдання: 2 години

Завдання №4 **Algotester Lab4v3 (варіант 1)**

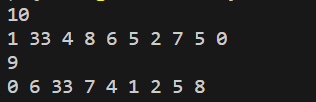


Рисунок 7: результат виконання програми 4

Час затрачений на виконання завдання: 1,5 години

Завдання №5 **Algotester Lab4v3 (варіант 2)**

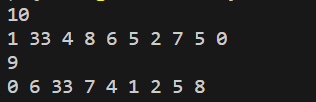


Рисунок 8: результат виконання програми 5

Час затрачений на виконання завдання: 3 години

Завдання №6 **Algotester Lab6v3**

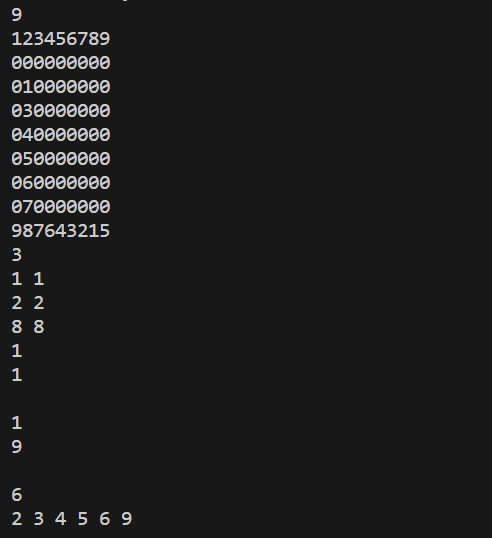


Рисунок 9: результат виконання програми 6

Час затрачений на виконання завдання: 1 година

Завдання №7 **Class Practice Work**

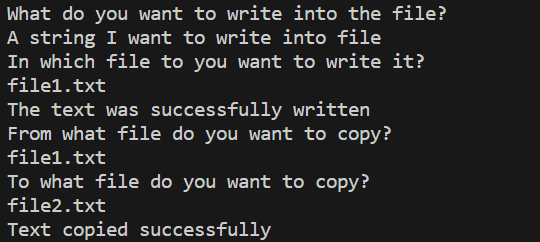


Рисунок 10: результат виконання програми 7

Час затрачений на виконання завдання: 1 годинa 30 хвилин

Завдання №8 **Self-Practice Work 1 (The Algo Battles 2023 - Етап 7 - А)**

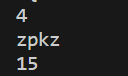


Рисунок 11: результат виконання програми 8

Час затрачений на виконання завдання: 10 хвилин

Завдання №9 **Self-Practice Work 2 (Algotester Lab 4v2)**

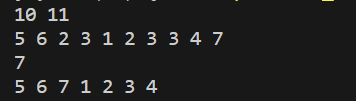


Рисунок 12: результат виконання програми №9

Час затрачений на виконання завдання: 40 хвилин

## **6. Кооперація з командою:**

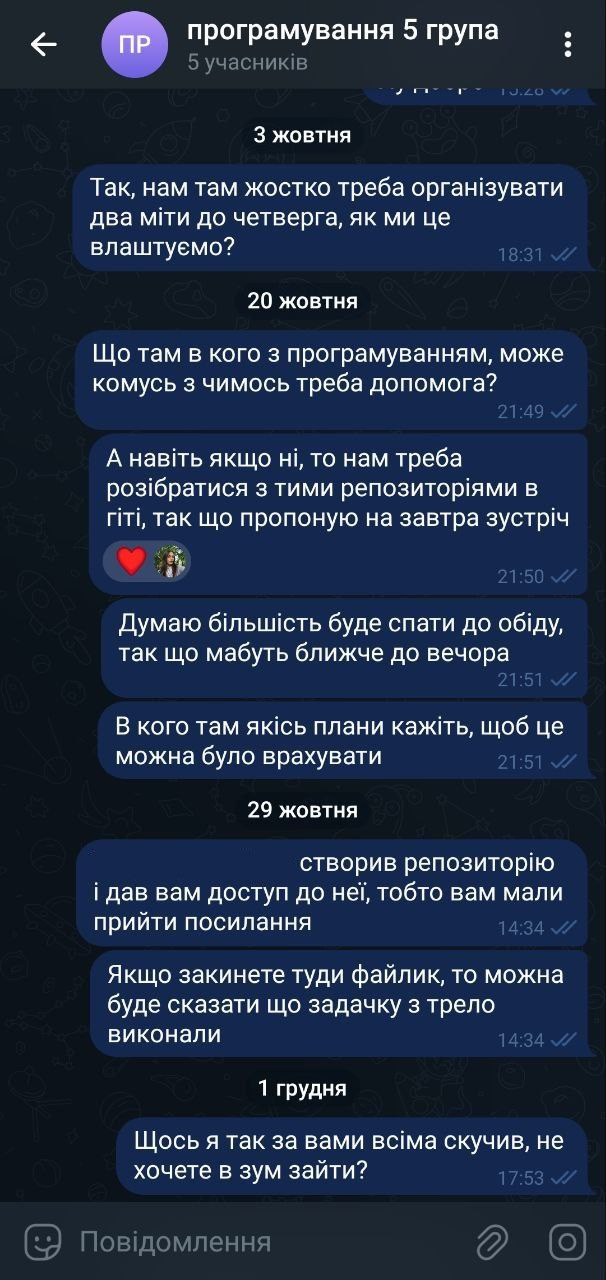


Рисунок 13: скрін спроб кооперації з командою

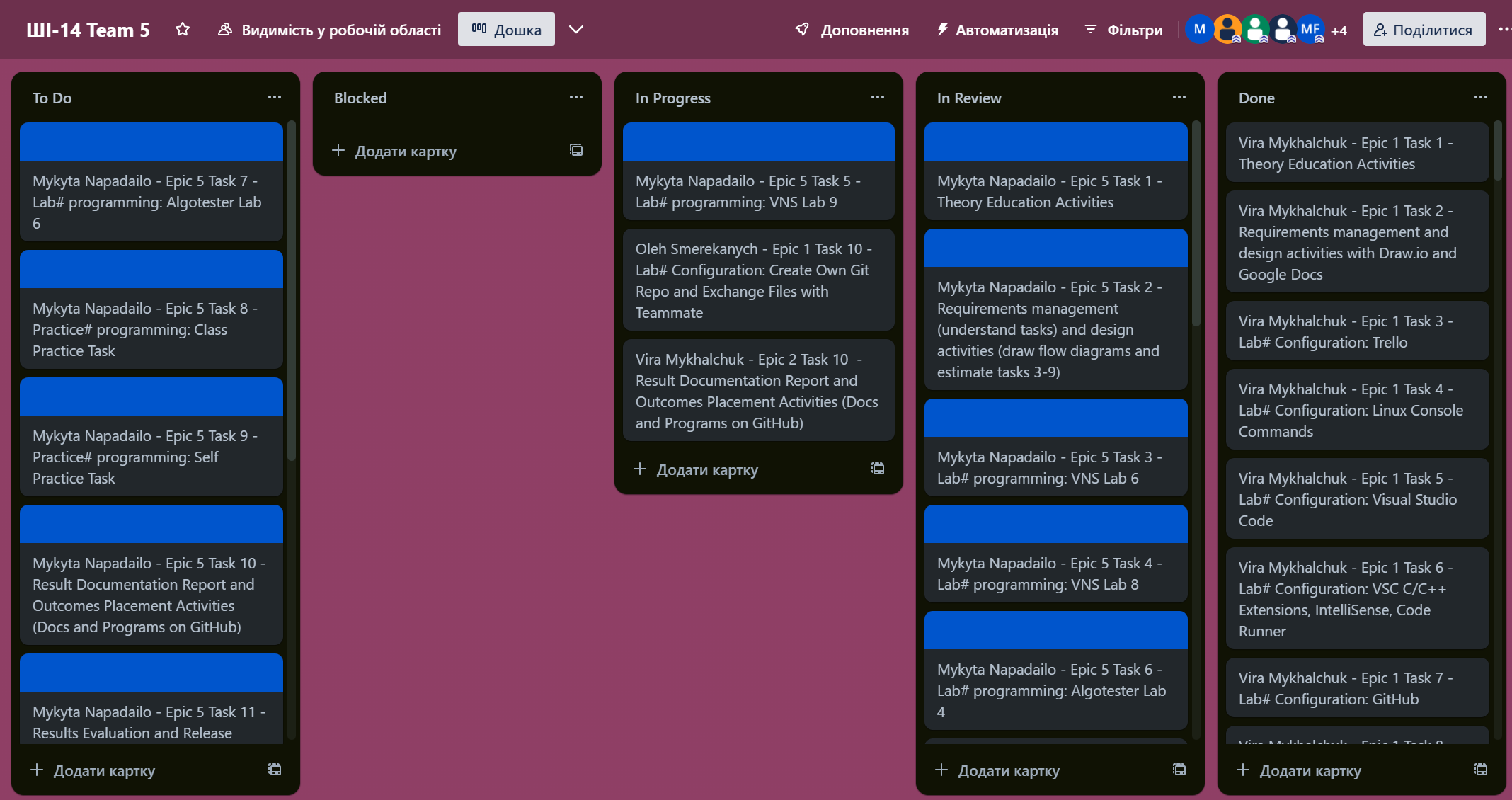
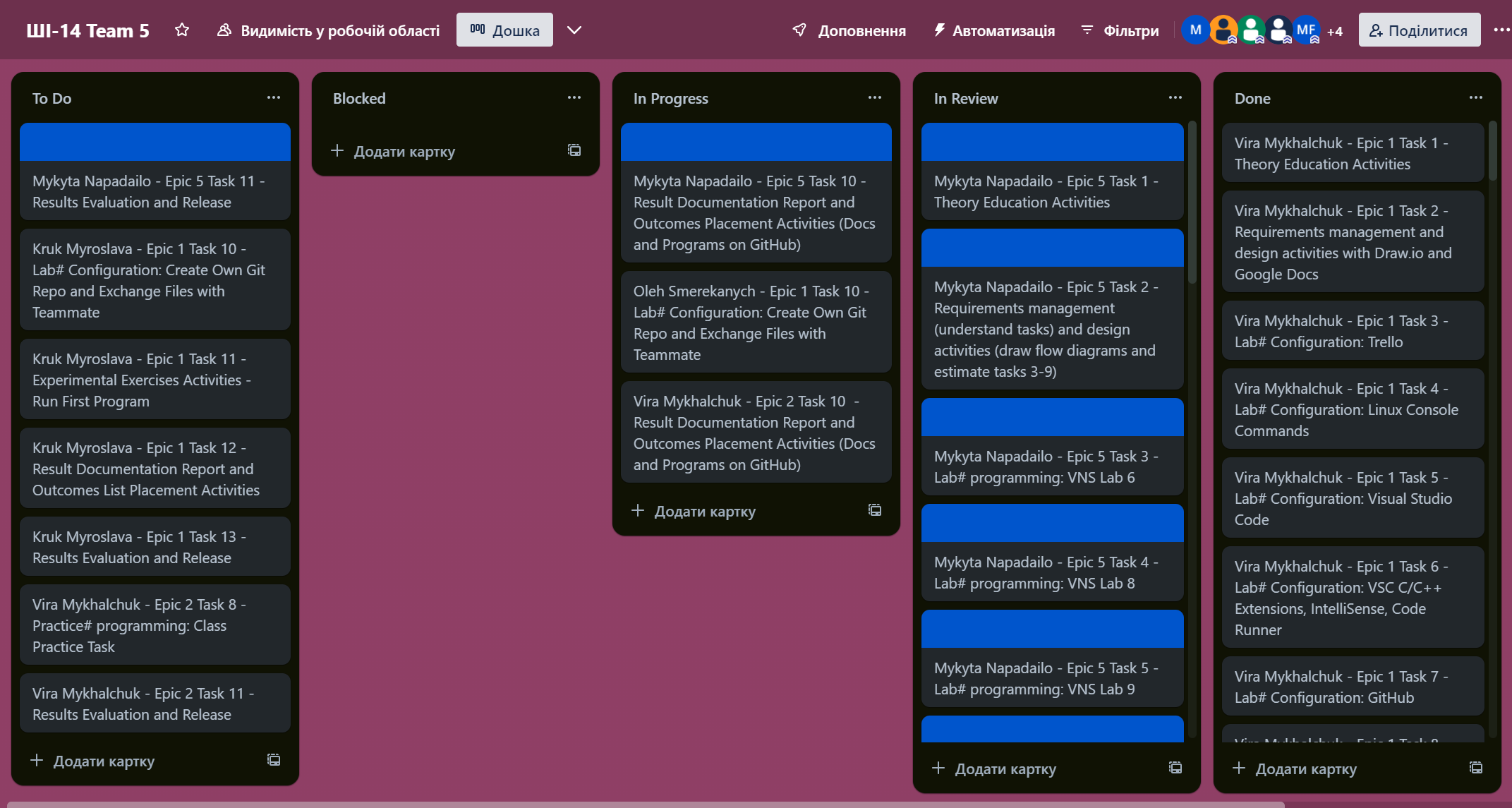


Рисунок 14: скрін прогресу Trello ранок 09.12.23

Рисунок 15: скрін прогресу Trello ніч 10.12.23

# **Висновки:**

В ході виконання лабораторних та практичних робіт в цьому епіку, я попрактикувався в використанні string-ів, вивчив кілька корисних функцій пов’язаних з ними; навчився працювати з текстовими та бінарними файлами, а саме зчитувати та записувати інформацію з/в них.