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

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

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт №5

***Виконала:***

студентка групи ШІ-13

Мостова Віта Любомирівна

# **Тема роботи:** Файли. Системи числення. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек

# **Мета роботи:** Орієнтуватись в роботі з файлами. Навчитись користуватись вбудованими функціями та методами стандартної бібліотеки.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Файли
* Тема №2: Потоки вводу/виводу
* Тема №3: Стандартна бібліотека

1. Індивідуальний план опрацювання теорії:

* Тема №1: Файли. Бінарні файли. Методи роботи з файлами
  + Джерела Інформації
    - <https://youtu.be/SSNJ7alki-E?si=bL7BSIb8dSsTzkWM>
    - <https://youtu.be/SJMxuXrP_Rk?si=OKq1rDjFfrL-IVcV>
  + Що опрацьовано: Опрацьовано матеріали надані вище. Розібрано як зчитувати інформацію з файлу та записувати інформацію в файл.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 6.12.2023
  + Звершення опрацювання теми: 11.12.2023
* Тема №2: Потоки вводу/виводу
  + Джерела Інформації:
    - [https://acode.com.ua/urok-215-potoky-vvodu-i-vyvodu/#toc-1](https://acode.com.ua/urok-215-potoky-vvodu-i-vyvodu/%23toc-1%20)
  + Що опрацьовано: Опрацьовано матеріали надані вище. Розібрано, що таке потік вводу та виводу.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 6.12.2023
  + Звершення опрацювання теми: 11.12.2023
* Тема №3: Стандартна бібліотека
  + Джерела Інформації:
    - <https://youtu.be/m-WJikuZGuU?si=teQouPBVvCw0rAZ6>
    - [https://youtu.be/L7JGsi4sryc?si=6ee\_n-KxEOb-cgQt](https://youtu.be/L7JGsi4sryc?si=6ee_n-KxEOb-cgQt%20)
  + Що опрацьовано: Опрацьовано матеріали надані вище. Пропрацьовано використання бібліотеки
  + Статус: Ознайомлена
  + Початок опрацювання теми: 6.12.2023
  + Звершення опрацювання теми: 6.12.2023

# **Виконання роботи:**

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

Завдання №1 **«Робота з рядками»**

* Варіант 18
* 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 **«Блоковий ввід-вивід»**

* Варіант 18
* Сформувати двійковий файл із елементів, заданої структури, роздрукувати його вміст. Знищити 3 елементи з початку файлу, додати елемент перед елементом із

зазначеною назвою.

* Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

Структура "Книга":

- назва;

- автор;

- рік видання;

- кількість сторінок.

Завдання №3 **«Ввід-вивід рядків»**

* Варіант 18
* 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 **«Сортування масиву за остачею»**

* Дано масив з N додатніх цілих чисел. Потрібно 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 видалити усі дублікати з масиву. Вивести результуючий масив.

Завдання №5 **«Судоку»**

* 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 посортовані по зростанню!). В клітинку можна вписати лише ті числа, які не зустрічаються в рядку та стовбці, які перетинаються у цій клітинці.

Завдання №6 **«Робота з текстовими файлами»**

* Реалізувати функцію створення файла і запису в нього даних. enum FileOpResult {Success, Failure}; FileOpResult write\_to\_file(const char \*name, const char \*content); FileOpResult copy\_file(const char \*file\_from, const char \*file\_to);
* Потрібно створити файл із заданим ім’ям; якщо файл існує – перезаписати його вміст. Копіювати вміст файла з ім’ям file\_from у файл з ім’ям file\_to;

Завдання №7 **«Новорічна математика»**

* Послідовність новорічних простих чисел нульового рівня — це зростаюча послідовність усіх простих чисел. Тобто це не що інше, як послідовність виду p0=[2,3,5,7,11,…]. Послідовність новорічних простих чисел першого рівня — це зростаюча послідовність усіх простих чисел, які в поcлідовності нульового рівня розміщені на позиціях із простим номером: p1=[3,5,11,17,…].Узагальнюючи ці міркування можна сказати, що послідовність новорічних простих чисел рівня k — це зростаюча послідовність усіх новорічних простих чисел рівня k−1, які розміщені на позиціях із простим номером:

pk=[pk−1[2],pk−1[3],pk−1[5],pk−1[7],…]

* Зауважте, що елементи послідовностей новорічних чисел нумеруються з 1.
* Завдання — визначити n-ий елемент послідовності новорічних простих чисел рівня k

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

Програма №1 **«Робота з рядками»**

* Планований час на реалізацію: до 3 годин

Програма №2 **«Блоковий ввід-вивід»**

* Планований час на реалізацію: до 5 годин

Програма №3 **«Ввід-вивід рядків»**

* Планований час на реалізацію: до 4 годин

Програма №4 **«Сортування масиву за остачею»**

* Планований час на реалізацію: до 3 годин

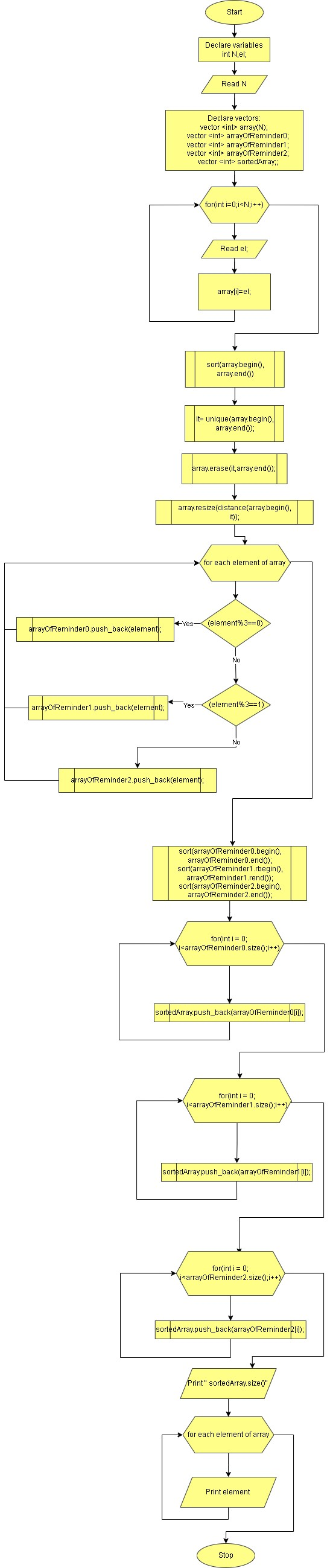


Рисунок 1: Блок-схема до завдання №4

Програма №5 **«Судоку»**

* Планований час на реалізацію: до 3 годин

Програма №6 **«Робота з текстовими файлами»**

* Планований час на реалізацію: до 3 годин

Програма №7 **«Новорічна математика»**

* Планований час на реалізацію: до 3 годин

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

Завдання №1 **«Робота з рядками»**

[Pull Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/813/files%23diff-297d9dd88e71c5d6a71939635c3a1994a1ab879711cf5b3c9c66572ce94e28d1)

#include <iostream>

#include <sstream>

#include <algorithm>

#include <vector>

#include <cstring>

#define n 255

using namespace std;

bool isDigit(char c);

int main() {

    char text[n];

    cout << "Enter text:\n";

    cin.getline(text, n);

    istringstream iss(text);

    string word;

    vector<string> wordsWithNumber;

        while (iss >> word) {

        if (isDigit(word[0])) {

            wordsWithNumber.push\_back(word);

        }

    }

    if (wordsWithNumber.empty()) {

        cout << "There are no words that start with a digit.\n";

    } else {

    sort(wordsWithNumber.rbegin(), wordsWithNumber.rend());

    cout << "Words that start with a digit (in descending order):\n";

    for (string &element : wordsWithNumber) {

        cout << element << endl;

    }}

    return 0;

}

bool isDigit(char c) {

    if (c >= '0' && c <= '9'){

    return true;}

    else{ return false;}

}

## *Програмний код до завдання №1*

Програма №2 **«Блоковий ввід-вивід»**

[Pull Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/813/files%23diff-a7b3e561094fca030a725a1bc3b5cf80351940a94d32cc5863fac5e68647768b)

#include <iostream>

#include <vector>

#include <fstream>

#include <sstream>

using namespace std;

struct Book {

    string title;

    string author;

    int year;

    int pages;

};

void write\_to\_file(vector<Book> books, string file\_name);

void remove\_first\_three\_lines(vector<Book> &books, const string& file\_name);

void print\_file(const string& file\_name);

int main()

{

        Book book1={"HarryPotter","Joanne Rowling",1997,319};

        Book book2={"Emma","Jane Austen",1813,432};

        Book book3= {"Hobbit","Tolkien",1937,310};

        Book book4={"1984","George Orwell",1949,328};

        Book book5={"TheGreatGatsby","Scott Fitzgerald",1997,319};

        vector<Book> books;

        books.push\_back(book1);

        books.push\_back(book2);

        books.push\_back(book3);

        books.push\_back(book4);

        books.push\_back(book5);

    write\_to\_file(books, "file.bin");

    cout<<"Now you have this books:"<<endl;

    print\_file("file.bin");

    remove\_first\_three\_lines(books, "file.bin");

    cout<<"----------------------------------------------"<<endl;

    cout<<"After deleting first three lines:"<<endl;

    print\_file("file.bin");

    cout<<"----------------------------------------------"<<endl;

    Book yourBook;

    cout << "Enter title of the book you want to add: ";

    getline(cin, yourBook.title);

    cout << "Enter name of author: ";

    getline(cin, yourBook.author);

    cout << "Enter year of publication: ";

    cin >> yourBook.year;

    cout << "Enter number of pages: ";

    cin >> yourBook.pages;

    cin.ignore();

    string bookBefore;

    cout << "Enter name of the book before which  you want to insert a new one: ";

    getline(cin, bookBefore);

    ifstream fileToRead("file.bin");

    if (!fileToRead)

    {

        cout << "This file can not be read:(" << endl;

        return 1;

    }

    ofstream fileToWrite("temp.bin");

    if (!fileToWrite)

    {

        cout << "This file can not be opened:(" << endl;

        return 1;

    }

    string text;

    bool IsBookPresent = false;

    while (getline(fileToRead, text))

    {

        istringstream iss(text); //Це створює "потік", який можна використовувати для введення, але не з файлу чи консолі, а з рядка.

        string word;

        iss >> word;

        if (word == bookBefore)

        {

            fileToWrite << yourBook.title << " " << yourBook.author << " " << yourBook.year << " " << yourBook.pages << endl;

            IsBookPresent = true;

        }

        fileToWrite << text << endl;

    }

    if (!IsBookPresent)

    {

        cout << "Oops! Book not found:(" << endl;

        return 1;

    }

    fileToRead.close();

    fileToWrite.close();

    remove("file.bin");

    rename("temp.bin", "file.bin");

     cout<<"----------------------------------------------"<<endl;

    cout<<"After inserting:"<<endl;

    print\_file("file.bin");

    return 0;

}

void write\_to\_file(vector<Book> books, string file\_name)

{

    ofstream creatingFile(file\_name);

    if (!creatingFile)

    {

        cout << "This file can not be opened:(" << endl;

        return;

    }

    for (auto it = books.begin(); it != books.end(); ++it)

{

    const Book& book = \*it;

    creatingFile << book.title << " " << book.author << " " << book.year << " " << book.pages << endl;

}

    creatingFile.close();

}

void remove\_first\_three\_lines(vector<Book> &books, const string& file\_name)

{

    if (books.size() >= 3)

    {

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

    }

    ofstream container("temp.bin");

    if (!container)

    {

        cout << "Sorry, something went wrong while removing first three books :(" << endl;

        return;

    }

    for (auto it = books.begin(); it != books.end(); ++it)

{

    const Book& book = \*it;

    container << book.title << " " << book.author << " " << book.year << " " << book.pages << endl;

}

    container.close();

    remove(file\_name.c\_str());

    rename("temp.bin", file\_name.c\_str());

}

## *Програмний код до завдання №2*

Завдання №3 **«Ввід-вивід рядків»**

[Pull Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/813/files%23diff-ecd55a87ff37390591592030372824101f24f407cec922b68a2ddb03671d1edb)

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

#include <unordered\_set>

using namespace std;

enum FileOpResult { Success, Failure };

FileOpResult write\_to\_file(const char \*name);

FileOpResult copy\_to\_file(const char \*fileName1, const char \*fileName2);

int count\_vowels(const string& line);

int main() {

    string fileName1;

    string fileName2;

    cout << "Enter name of the file you want to create and write to:\n";

    getline(cin, fileName1);

    cout << "Enter name of the file you want to copy lines without duplicates:\n";

    getline(cin, fileName2);

    FileOpResult resultOfWriting = write\_to\_file(fileName1.c\_str());

   if (resultOfWriting == Success) {

        FileOpResult resultOfCopying = copy\_to\_file(fileName1.c\_str(), fileName2.c\_str());

        if (resultOfCopying == Success) {

            cout << "Success\n";

            ifstream FileNumber2(fileName2.c\_str());

            string firstLine;

            if (getline(FileNumber2, firstLine, '.')) {

                int vowels = count\_vowels(firstLine);

                cout << "Number of vowels in the first line in the second file: " << vowels << endl;

            }

            FileNumber2.close();

        } else {

            cout << "Failure\n";

        }

    } else {

        cout << "Failure\n";

    }

    return 0;

}

FileOpResult write\_to\_file(const char \*name) {

    ofstream myFile(name);

    if (!myFile.is\_open()) {

        return Failure;

    }

    for (int i = 0; i < 10; i++) {

        string text;

        cout << "Enter text for line " << i + 1 << ": ";

        getline(cin, text);

        myFile << text << endl;

    }

    myFile.close();

    return Success;

}

FileOpResult copy\_to\_file(const char \*fileName1, const char \*fileName2) {

    ifstream copyFromThisFile(fileName1);

    ofstream copyToThisFile(fileName2);

    if (!copyFromThisFile.is\_open() || !copyToThisFile.is\_open()) {

        return Failure;

    }

    string line;

    while (getline(copyFromThisFile, line)) {

        istringstream iss(line);

        unordered\_set<string> wordsInLine;

        bool duplicate = false;

        string word;

        while (iss >> word) {

  if (!wordsInLine.insert(word).second) {

// вставка в сет провалилась, отже слово вже присутнє

                duplicate = true;

                break;

            }

        }

        if (!duplicate) {

            copyToThisFile << line << endl;

        }

    }

    copyFromThisFile.close();

    copyToThisFile.close();

    return Success;

}

int count\_vowels(const string& line) {

    int vowelCount = 0;

    for (char c : line) {

        if (tolower(c) == 'a' || tolower(c) == 'e' || tolower(c) == 'i' || tolower(c) == 'o' || tolower(c) == 'u'|| tolower(c)=='y') {

            vowelCount++;

        }

    }

    return vowelCount;

}

## *Програмний код до завдання №3*

Завдання №4 **«Сортування масиву за остачею»**

[Pull Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/813/files%23diff-3e81b80e31f9493f22067fdd8bdf42959f81778306183680c6a39a47696f46e4)

#include <iostream>

#include<vector>

#include <algorithm>

using namespace std;

int main(){

    int N,el;

    cin>>N;

    vector <int> array(N);

    vector <int> arrayOfReminder0;

    vector <int> arrayOfReminder1;

    vector <int> arrayOfReminder2;

    vector <int> sortedArray;

for(int i=0;i<N;i++){

    cin>>el;

    array[i]=el;}

sort(array.begin(), array.end());

auto it = unique(array.begin(), array.end());

array.erase(it,array.end());

array.resize(distance(array.begin(), it));

for(int element:array){

    if(element%3==0){

        arrayOfReminder0.push\_back(element);

    }

    else if(element%3==1){

        arrayOfReminder1.push\_back(element);

    }

    else if(element%3==2){

        arrayOfReminder2.push\_back(element);

    }

}

sort(arrayOfReminder0.begin(), arrayOfReminder0.end());

sort(arrayOfReminder1.rbegin(), arrayOfReminder1.rend());

sort(arrayOfReminder2.begin(), arrayOfReminder2.end());

    for(int i = 0;i<arrayOfReminder0.size();i++)

    {

        sortedArray.push\_back(arrayOfReminder0[i]);

    }

    for(int i = 0;i<arrayOfReminder1.size();i++)

    {

        sortedArray.push\_back(arrayOfReminder1[i]);

    }

    for(int i = 0;i<arrayOfReminder2.size();i++)

    {

        sortedArray.push\_back(arrayOfReminder2[i]);

    }

cout<<sortedArray.size()<<"\n";

for(int j:sortedArray){

    cout<<j<<" ";}

    return 0;

}

## *Програмний код до завдання №4*

Завдання №5 **«Судоку»**

[Pull Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/813/files%23diff-70e1e73aff40b914be80f7228234ded77769742a01723ef568021a34783a21c2)

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

struct Coordinates{

    int x;

    int y;

};

vector<int> Variantu(const vector <vector<int>>& array, const Coordinates& coordinates);

int main (){

int N;

cin>>N;

if(N<1 || N>9){

    return 0;

}

vector<vector<int>> pole(N,vector<int>(N));

string row;

for(int i=0;i<N;i++){

    cin>>row;

    for(int j=0;j<N;j++){

        pole[i][j] = row[j] - '0';

    }

}

int Q;

cin>>Q;

vector <Coordinates> coordinates(Q);

for (int k = 0; k < Q; k++) {

    cin >> coordinates[k].x >> coordinates[k].y;}

    for (int k = 0; k < Q; k++){

    if (pole[coordinates[k].x-1][coordinates[k].y-1] == 0) {

        vector<int> yourOptions = Variantu(pole, coordinates[k]);

        sort(yourOptions.begin(),yourOptions.end());

        cout << yourOptions.size() << endl;

        for (int &element : yourOptions) {

            cout << element <<" ";

        }cout<<endl<<endl;

    }else {

        cout<<"1"<<endl;

         cout << pole[coordinates[k].x-1][coordinates[k].y-1] << endl;

    }

}

return 0;}

vector<int> Variantu(const vector <vector<int>>& array, const Coordinates& coordinates) {

    vector<int> PossibleNumbers;

    for (int i = 1; i <= array.size(); i++) {

        PossibleNumbers.push\_back(i);

    }

       for (int j = 0; j < array.size(); j++) {

        if (array[j][coordinates.y-1] != 0) {

            auto it = find(PossibleNumbers.begin(), PossibleNumbers.end(), array[j][coordinates.y-1]);

            if (it != PossibleNumbers.end()) {

                PossibleNumbers.erase(it);

            }

        }

    }

    for (int j = 0; j < array.size(); j++) {

        if (array[coordinates.x - 1][j] != 0) {

            auto it = find(PossibleNumbers.begin(), PossibleNumbers.end(), array[coordinates.x - 1][j]);

            if (it != PossibleNumbers.end()) {

                PossibleNumbers.erase(it);

            }

        }

    }

    return PossibleNumbers;

}

## *Програмний код до завдання №5*

Завдання №6 **«Робота з текстовими файлами»**

[Pull Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/813/files%23diff-70eb3ec2a7bae9365ad0e37c4606080a7671dcc41f1cf912b30d2e732600f5c1)

#include <iostream>

#include <fstream>

#include <string>

enum FileOpResult { Success, Failure};

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

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

using namespace std;

int main(){

string fileName;

cout<<"Enter name of the file you want to write to\n";

getline(cin,fileName);

string text;

cout<<"Enter text you want to write into the file\n";

getline(cin,text);

FileOpResult resultOfWriting=write\_to\_file((fileName.c\_str()),(text.c\_str()));

if(resultOfWriting==Success){

    cout<<"Success\n";

}else{

    cout<<"Failure\n";

}

string fileNameForCopy;

cout<<"Enter name of the file you want to copy from\n";

getline(cin,fileNameForCopy);

string fileNamtoCopy;

cout<<"Enter name of the file you want to copy this into\n";

getline(cin,fileNamtoCopy);

FileOpResult resultOfCopying=copy\_file((fileNameForCopy.c\_str()),(fileNamtoCopy.c\_str()));

if(resultOfCopying==Success){

    cout<<"Success\n";

}else{

    cout<<"Failure\n";

}

    return 0;

}

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

ofstream myFile(name);

if (!myFile.is\_open()) {

        return Failure;

    }

myFile<<content;

    if(myFile.fail()){

    myFile.close();

    return Failure;

    }

else{

    myFile.close();

    return Success;

}

}

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

ifstream myReadFile(file\_from);

ofstream myCopiedFile(file\_to);

string myText;

if(!myReadFile.is\_open() || !myCopiedFile.is\_open()){

    return Failure;

}

while (getline(myReadFile, myText)) {

   myCopiedFile << myText << "\n";

}

myReadFile.close();

myCopiedFile.close();

return Success;

}

## *Програмний код до завдання №6*

Завдання №7 **«Новорічна математика»**

[Pull Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/813/files%23diff-84f12c600d69965a035349a788f7f2f48afabe288ad3e69c271c4dda8636fa63)

#include <iostream>

#include <vector>

using namespace std;

vector<int> get\_primes(int n);

int solve(int n,int k);

int main(){

int n,k;

cin>>n>>k;

cout<<solve(n,k);

    return 0;

}

vector<int> get\_primes(int n){

vector<bool> vykresleni(n+1,true);

vykresleni[0]=vykresleni[1]=false;

vector<int> primes;

for(int i=2;i<=n;i++){

    if(vykresleni[i]){

        primes.push\_back(i);

        for(int j=2\*i;j<=n;j+=i){

            vykresleni[j]=false;}

        }

    }

return primes;

}

int solve(int n,int k){

    int max=2000000;

    vector<vector<int>>np(k+1);

    vector<int> primes=get\_primes(max);

    np[0]=primes;

    for(int i=1;i<=k;++i){

        for(int j=0;j<primes.size();++j){

            if(primes[j]-1>=np[i-1].size())

            break;

            np[i].push\_back(np[i-1][primes[j]-1]);

        }

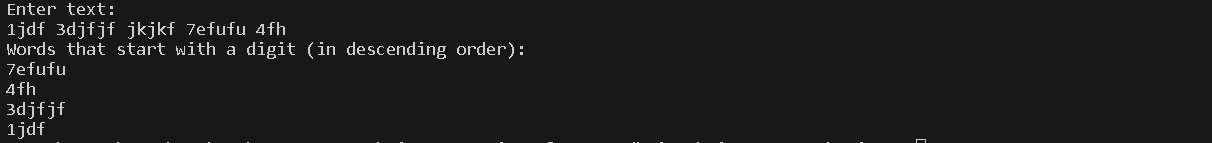
    }

    return np[k][n-1];

}

## *Програмний код до завдання №7*

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

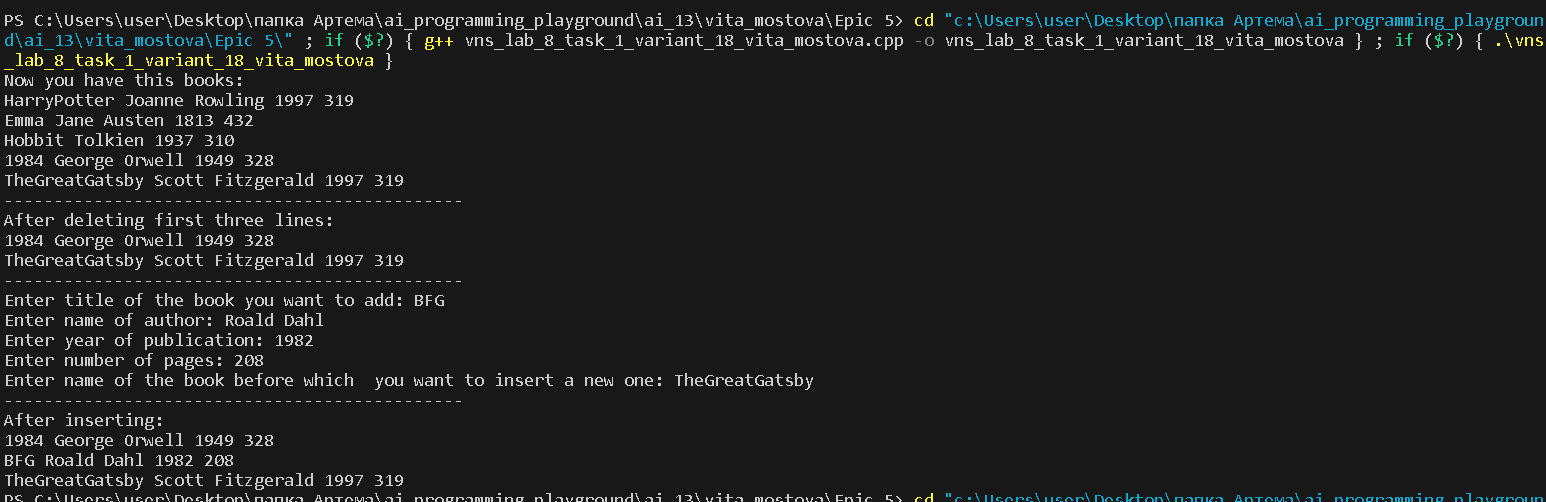
Завдання №1 **«Робота з рядками»***Рисунок 2.1: Приклад виконання завдання №1*



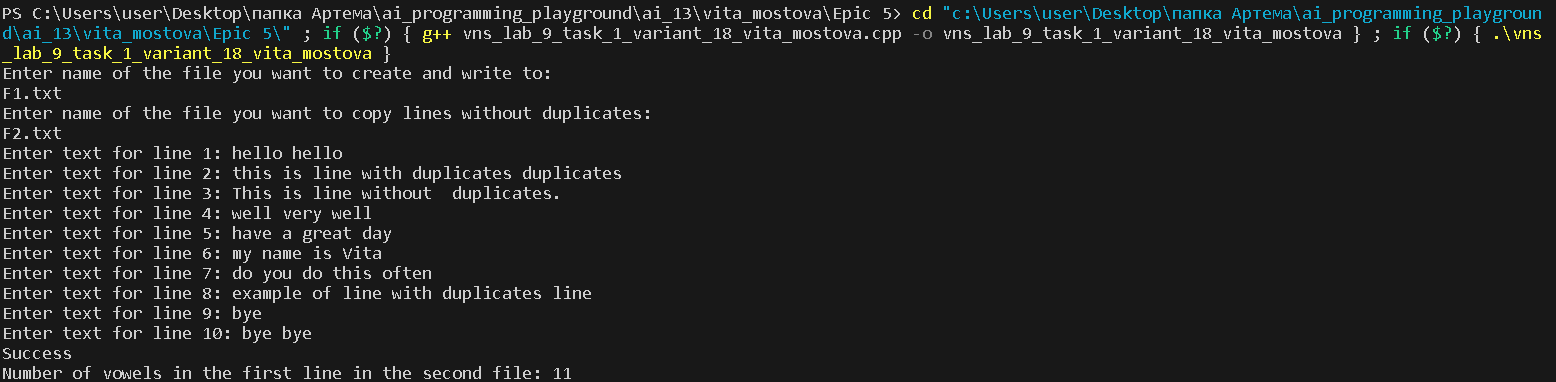
*Рисунок 2.2: Приклад виконання завдання №1*

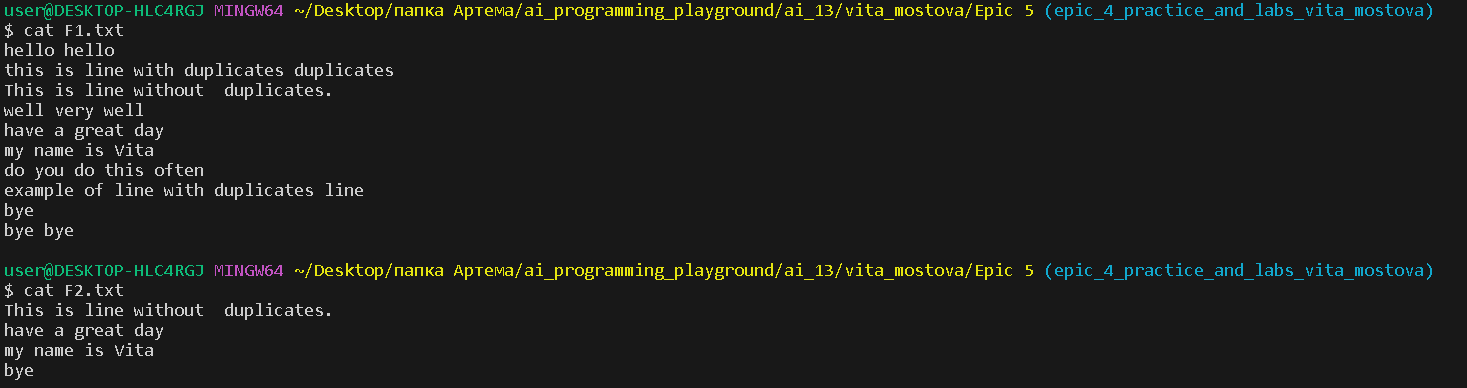
Час затрачений на виконання завдання: близько 2 годин

Програма №2 **«Блоковий ввід-вивід»**

*Рисунок 3: Приклад виконання завдання №2*

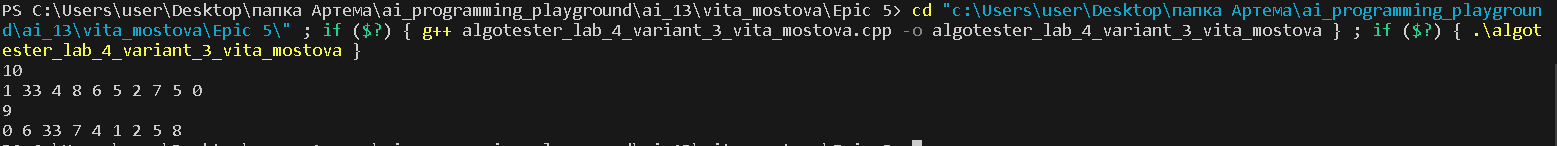
Час затрачений на виконання завдання: близько 4 годин

Завдання №3 **«Ввід-вивід рядків»**

*Рисунок 4: Приклад виконання завдання №3*

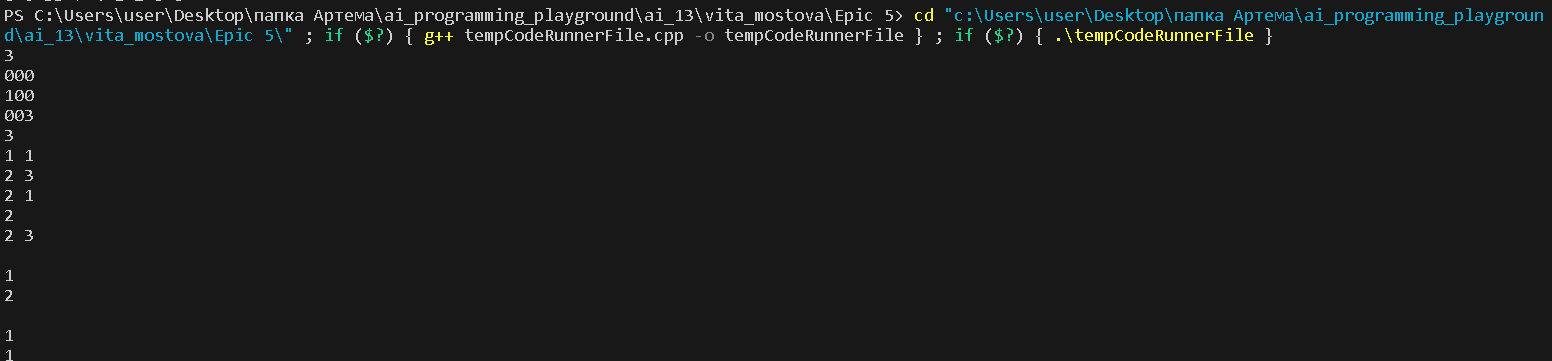
Час затрачений на виконання завдання: близько 3 годин

Завдання №4 **«Сортування масиву за остачею»**

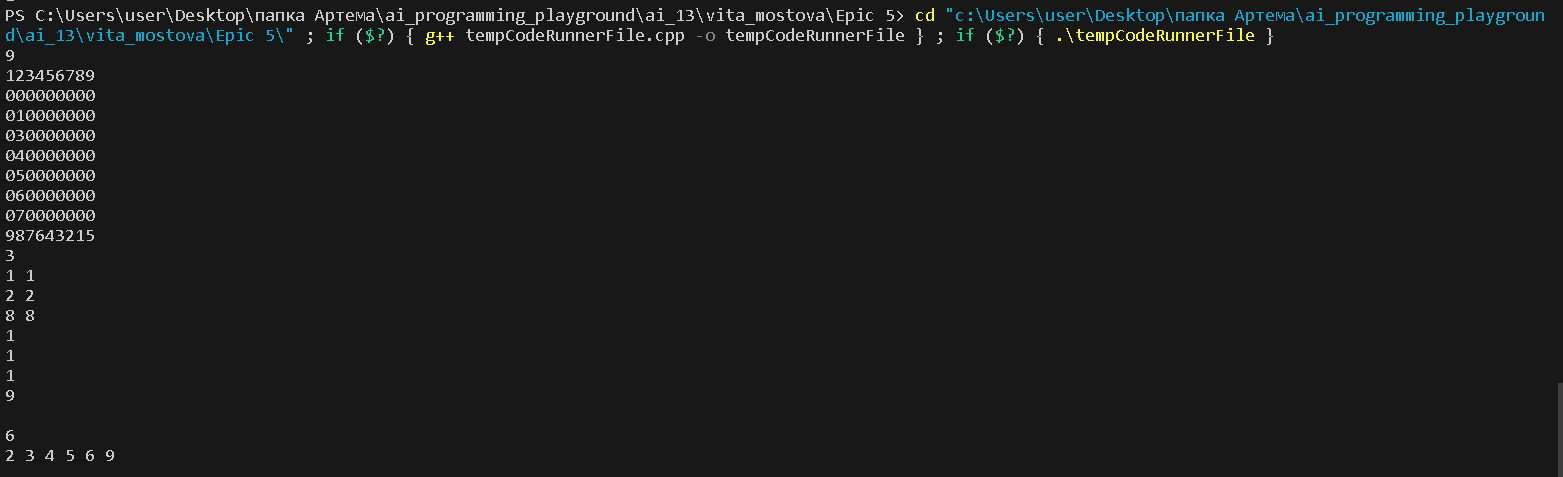
*Рисунок 5: Приклад виконання завдання №4*

Час затрачений на виконання завдання: близько 2 годин

Завдання №5 **«Судоку»**



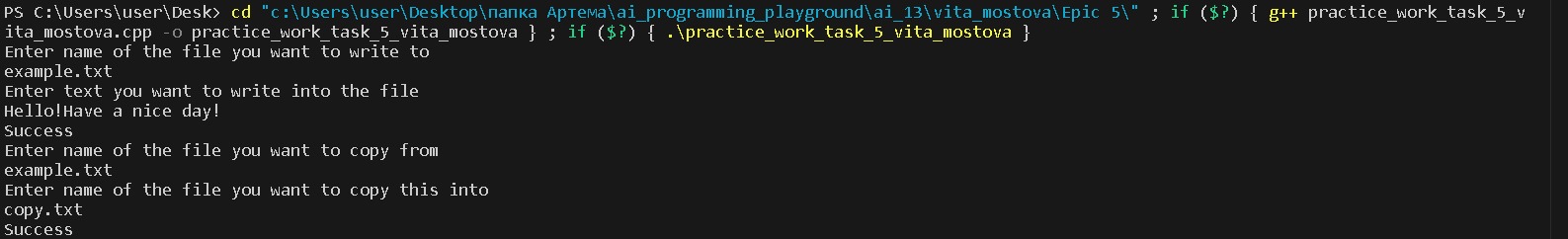
*Рисунок 6.1: Приклад виконання завдання №5*

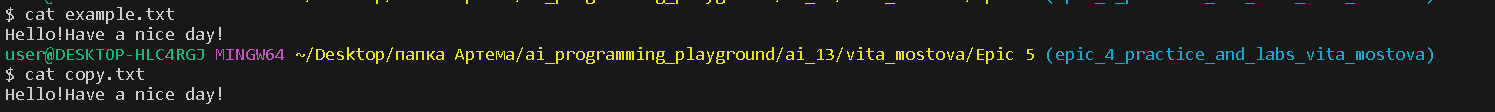


*Рисунок 6.2: Приклад виконання завдання №5*

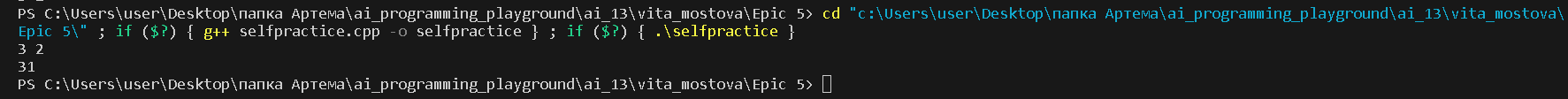
Час затрачений на виконання завдання: близько 3 годин

Завдання №6 **«Робота з текстовими файлами»**



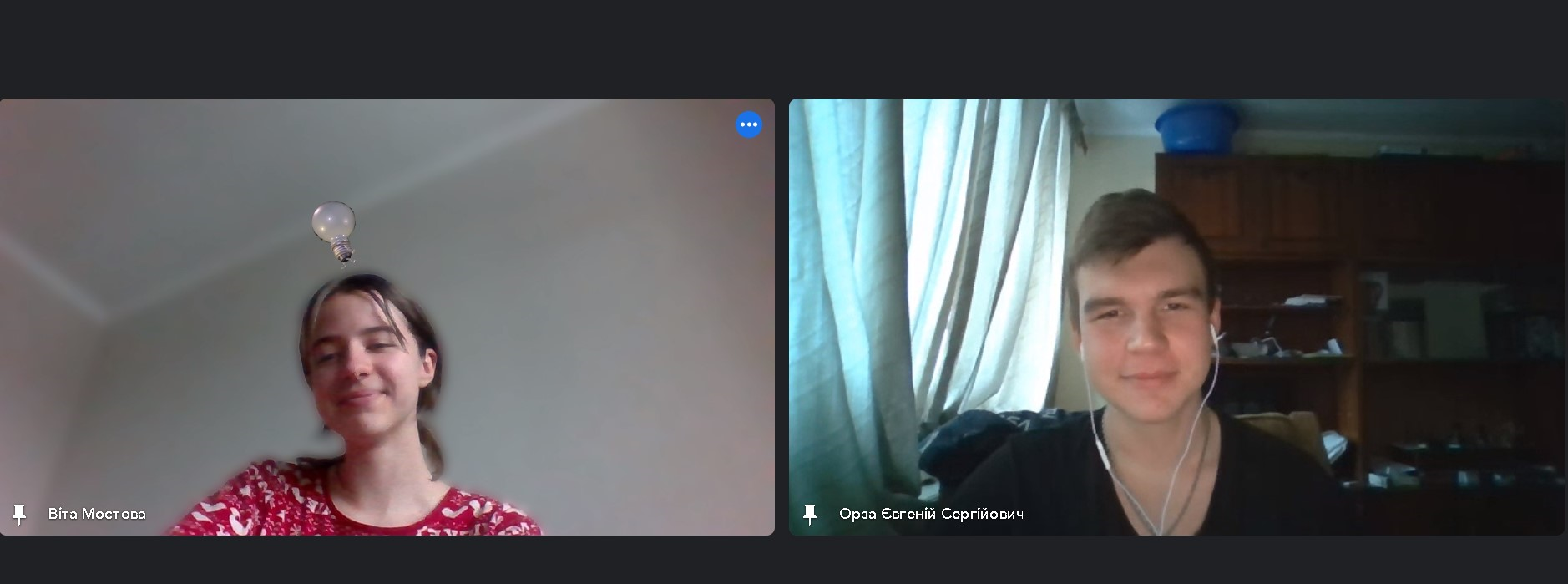
*Рисунок 7: Приклад виконання завдання №6*

Час затрачений на виконання завдання: близько 3 годин

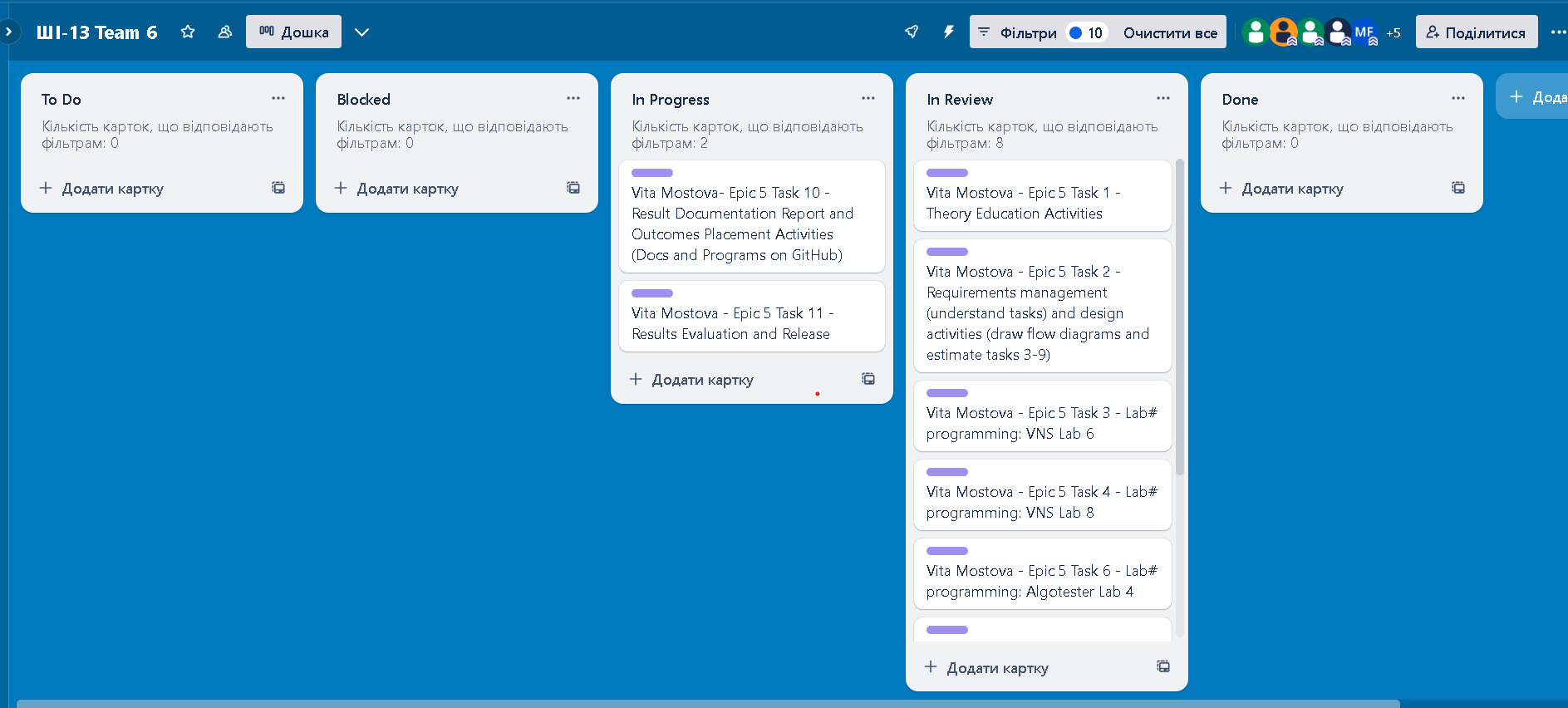
Завдання №7 **«Новорічна математика»***Рисунок 8: Приклад виконання завдання №7*

Час затрачений на виконання завдання: близько 2 годин

## **6. Кооперація з командою:**



*Рисунок 9: Скрін зустрічі з командою)*



*Рисунок 10: Скрін з Трелло*

# **Висновки:**

Під час виконання лабораторних та практичних завдань з цього епіку я дізналась багато нового, а саме: як працювати з файлами, як записувати в них інформацію та зчитувати її. Попрактикувалась і в використанні вбудованих функцій стандартної бібліотеки. Намагалась розібратись і в сортуванні, для виконання завдання №4 було використано sort, uniquе. Також було розібрано, як працювати з символами та рядковими змінними. Загалом, вдалось отримати базові навички роботи з файлами.