Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

**про виконання лабораторних та практичних робіт блоку № 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

**Виконав:**

Студент групи ШІ-13

Гаргай Юрій Анатолійович

Львів 2023

# 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.

**Задачі:**

1. *VNS Lab 6*
2. *VNS Lab 8*
3. *VNS Lab 9*
4. *Algotester Lab 4*
5. *Algotester Lab 6*
6. *Practice task1*
7. *Practice task2*
8. *Self-practice*

# Мета роботи:

# Ознайомитися з файлами в с++

# Навчитися передавати Символи і Рядкові Змінні та Текстові Файли

# Теоретичні відомості:

* 1. Теоретичні відомості з переліком важливих тем:
* Тема №1: Файли
* Тема №2: Використання бібліотек
* Тема №3: Системи числення.
  1. Індивідуальний план опрацювання теорії:
* Тема №1: Файли
* Джерела Інформації:
  + - https://www.geeksforgeeks.org/file-handling-c-classes/
    - <https://ua.udemy.com/>

Beginning C++ Programming - From Beginner to Beyond

* Що опрацьовано:
* Ознайомився з файлами
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.12.2023
  + Звершення опрацювання теми: 02.12.2023
* Тема №2: Використання бібліотек
* Джерела Інформації:
  + - <https://ua.udemy.com/>

Beginning C++ Programming - From Beginner to Beyond

* + Що опрацьовано**:**
    - Ознайомився з використанням бібліотек для керування файлами
  + Статус: Ознайомлений
  + Початок опрацювання теми:02.12.2023
  + Звершення опрацювання теми: 02.12.2023

* Тема №3: Системи числення.
  + Джерела Інформації:
    - <https://ua.udemy.com/>

Beginning C++ Programming - From Beginner to Beyond

* + Що опрацьовано:
    - Ознайомився з системами числення.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 07.12.2023

# Виконання роботи:

**Завдання №1**: VNS Lab 6

Варіант завдання: **13**

Деталі завдання

Перетворити рядок таким чином, щоб цифри кожного слова в ньому були

відсортовані за спаданням.

**Планована оцінка часу виконання завдань:** 1 день

**Завдання №2** VNS Lab 8

* Варіант завдання: **13**
* 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.

Структура "Спортивна команда":

- назва;

- місто;

- кількість гравців;

- кількість набраних очків.

Знищити всі елементи з кількістю

**Планована оцінка часу виконання завдань:** 1 день

**Завдання №3**: VNS Lab 9

Варіант завдання: **13**

Деталі завдання

1) Скопіювати з файлу F1 у файл F2 всі рядки, що починаються на букву «А» і

розташовані між рядками з номерами N1 й N2.

2) Визначити номер того рядка, у якому найбільше приголосних букв, файлу

F2.

**Планована оцінка часу виконання завдань:** 1 день

**Завдання №4** Algotester Lab 4

* Варіант завдання: **2**
* Деталі завдання:
* Вам дано масив, який складається з N додатнiх цiлих чисел.

Ваше завдання - роздiлити його на три частини, по остачi вiд дiлення на 3, по зростанню

остачi (тобто спочатку йдуть числа, у яких остача 0, далi числа з остачею 1 i тодi нарештi числа з остачею 2). Далi необхiдно тi елементи, остача вiд дiлення на 3 яких парна посортувати по зростанню, а тi, у яких остача 1 - по спаданню. Пiсля цього видалiть усi дублiкати з масиву. Виведiть результуючий масив.

**Планована оцінка часу виконання завдань:** 1 день

**Завдання №5** Algotester Lab 6

* Варіант завдання: **13**
* Деталі завдання:
* Вам дано 2 У Клiнта в черговий раз виключилось свiтло i йому немає чим зайнятися. Так як навiть це не заставить його подивитися збереженi вiдео про програмування на ютубi - вiн вирiшив придумати свою гру на основi судоку.

Гра виглядає так:

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

Завданням гри є написати до кожної координати скiльки чисел туди можна вписати (якщо вона пуста) i якi це числа (обов’язково в посортованi по зростанню!). В клiтинку можна вписати лише тi числа, якi не зустрiчаються в рядку та стовбцi, якi перетинаються у цiй клiтинцi. Пiд час гри поле не мiняється!

Також необовязково, щоб це було валiдне судоку! Якщо є клiтинка, в яку не можна вписати нiяку цифру - виведiть 0.

Також допускаються рядки та стовпцi, в яких цифра записана кiлька разiв.

**Планована оцінка часу виконання завдань:** 1 день

**Завдання №6** Practice task1

- Деталі завдання:

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

- написати код стійкий до різних варіантів вхідних параметрів

- name – ім’я, може не включати шлях

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

- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.**Планована оцінка часу виконання завдань:** 1 день

**Завдання №7** Practice task2

- Деталі завдання:

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.

**Планована оцінка часу виконання завдань:** 1 день

**Завдання №8** Self-practice

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 eлектричний 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.

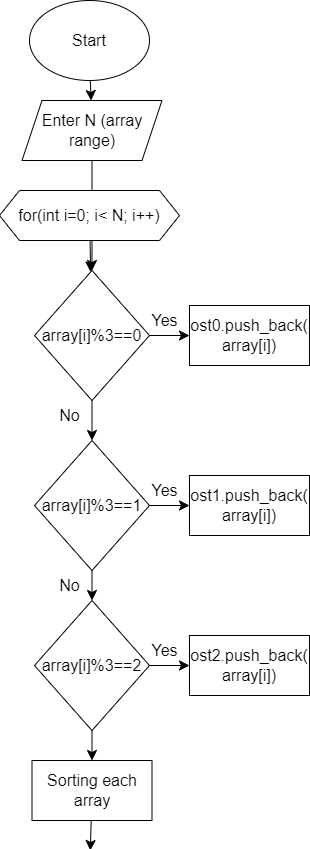
Змінювач не видає сигнал при ненульовому вхідному сигналі та видає сигнал силою 15 при відсутності вхідного сигналу. Формально, нехай сила вхідного сигналу k. Тоді змінювач видає сигнал силою 0, якщо k≥1, і сигнал силою 15, якщо k=0.

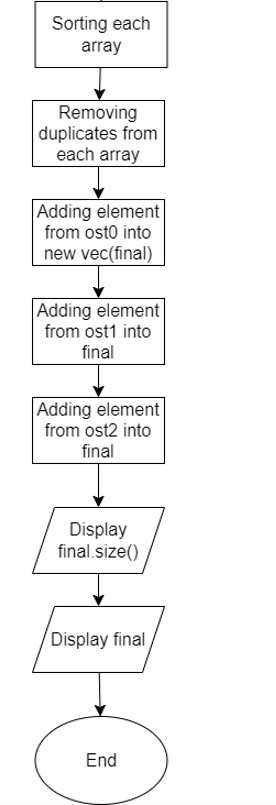
Петрик задає вам електричну лінію, в якій кабелі, повторювачі і змінювачі позначені літерами k, p, z відповідно. На перший елемент лінії подано сигнал силою 15. Допоможіть Петрику знайти вихідний сигнал останнього блоку в лінії.**Планована оцінка часу виконання завдань:** 1 день

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

Завдання №4Algotester lab4

* + Блок-схема





ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss

Рисунок 1 блок схема

Планований час на реалізацію: 1 день sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss

### Код програм з результатом та посиланням на зовнішні ресурси:

Завдання №1 VNS Lab 6

1)Код програми

#include <iostream>

#include <string>

#include <algorithm>

#include <cctype>

using namespace std;

bool isNumber(char );

bool sortDescending(char, char);

int main() {

    string str;

    cout << "Enter row: ";

    getline(cin, str);

    string word;

    for (char c : str) {

        if (isalnum(c)) {

            word += c;

        } else {

            if (!word.empty()) {

                sort(word.begin(), word.end(), sortDescending);

                cout << word;

                word = "";

            }

            cout << c;

        }

    }

    if(!word.empty()) {

        sort(word.begin(), word.end(), sortDescending);

        cout << word;

    }

    return 0;

}

bool isNumber(char c) {

    return c >= '0' && c <= '9';

}

bool sortDescending(char a, char b) {

    if (isNumber(a) && isNumber(b)) {

        return a > b;

    }

    return false;

}

Код програми

2) Результат



Рисунок 2 результат

3) Посилання на файл програми у пул-запиті GitHub

[Pull-request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/749/files#diff-455a2a00031bf2c16b8564660d23630a76c4d546c7a81f4cb3423417a95361ea)

ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss

Завдання №2 VNS Lab 8

1)Код програми

#include <iostream>

#include <fstream>

#include <vector>

#include <cstring>

using namespace std;

struct Team {

    char name[20];

    char city[20];

    int players;

    int points;

};

void write\_to\_file(const char\* filename, const vector<Team>& teams) {

    ofstream file(filename, ios::binary);

    if (!file) {

        cerr << "Error opening file for writing\n";

        return;

    }

    for (const auto& team : teams) {

        file.write((char\*)&team, sizeof(Team));

    }

    file.close();

}

vector<Team> read\_from\_file(const char\* filename) {

    vector<Team> teams;

    ifstream file(filename, ios::binary);

    if (!file) {

        cerr << "Error opening file for reading\n";

        return teams;

    }

    Team team;

    while (file.read((char\*)&team, sizeof(Team))) {

        teams.push\_back(team);

    }

    file.close();

    return teams;

}

void print\_teams(const vector<Team>& teams) {

    for (const auto& team : teams) {

        cout << "Name: " << team.name << ", City: " << team.city

             << ", Players: " << team.players << ", Points: " << team.points << '\n';

    }

}

vector<Team> remove\_teams(vector<Team>& teams, int min\_points) {

    vector<Team> new\_teams;

    for (const auto& team : teams) {

        if (team.points >= min\_points) {

            new\_teams.push\_back(team);

        }

    }

    return new\_teams;

}

vector<Team> add\_teams(vector<Team>& teams, const vector<Team>& new\_teams) {

    vector<Team> all\_teams = new\_teams;

    all\_teams.insert(all\_teams.end(), teams.begin(), teams.end());

    return all\_teams;

}

int main() {

    // Create some teams

    vector<Team> teams = {

        {"Manchester United", "Manchester", 11, 5},

        {"Real Madrid", "Madrid", 11, 4},

        {"Dynamo Kyiv", "Kyiv", 11, 12}

    };

    // Write teams to file

    write\_to\_file("teams.bin", teams);

    // Read teams from file

    teams = read\_from\_file("teams.bin");

    // Print teams

    cout << "Initial teams:\n";

    print\_teams(teams);

    // Remove teams with points less than 5

    teams = remove\_teams(teams, 5);

    // Print teams after removal

    cout << "\nTeams after removal:\n";

    print\_teams(teams);

    // Add new teams

    vector<Team> new\_teams = {

        {"Chelsea", "London", 11, 7},

        {"Arsenal", "London", 11, 6}

    };

    teams = add\_teams(teams, new\_teams);

    // Print teams after addition

    cout << "\nTeams after addition:\n";

    print\_teams(teams);

    // Write final teams to file

    write\_to\_file("teams.bin", teams);

    return 0;

}

Код програми

2) Результат

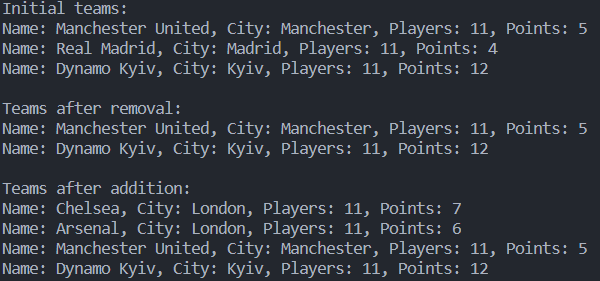


Рисунок 3 результат

3) Посилання на файл програми у пул-запиті GitH ub

Pull-[Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/749/files#diff-ada73fb5725821261f48e5f29fedb6334ecf9d1d1db5fac917c42709aaaefc04)

Завдання №3 VNS Lab 9

1)Код програми

#include <fstream>

#include <string>

#include <vector>

#include <cctype>

#include <iostream>

using namespace std;

void createF1();

void copyLines(int N1, int N2);

int findMaxConsonants();

void printF2();

int main() {

    createF1();

    copyLines(0, 9);

    printF2();

    int lineNumber = findMaxConsonants();

    cout << "Most consonants: " << lineNumber << endl;

    return 0;

}

// Створення файлу F1 з 10 рядками інформації

void createF1() {

    ofstream F1("F1.txt");

    for (int i = 0; i < 10; i++) {

        string line;

        cout << "Enter row number " << i + 1 << ": ";

        getline(cin, line);

        F1 << line << endl;

    }

    F1.close();

}

// Копіювання рядків з F1 в F2

void copyLines(int N1, int N2) {

    ifstream F1("F1.txt");

    ofstream F2("F2.txt");

    vector<string> lines;

    string line;

    while (getline(F1, line)) {

        lines.push\_back(line);

    }

    for (int i = N1; i <= N2; i++) {

        if (lines[i][0] == 'A') {

            F2 << lines[i] << endl;

        }

    }

    F1.close();

    F2.close();

}

// Визначення номера рядка з найбільшою кількістю приголосних букв у файлі F2

int findMaxConsonants() {

    ifstream F2("F2.txt");

    string vowels = "aeiouAEIOU";

    vector<string> lines;

    string line;

    int maxConsonants = 0;

    int lineNumber = 0;

    while (getline(F2, line)) {

        lines.push\_back(line);

    }

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

        int consonants = 0;

        for (char c : lines[i]) {

            if (isalpha(c) && vowels.find(c) == string::npos) {

                consonants++;

            }

        }

        if (consonants > maxConsonants) {

            maxConsonants = consonants;

            lineNumber = i+1;

        }

    }

    F2.close();

    return lineNumber;

}

// Друкування вмісту файлу F2

void printF2() {

    ifstream F2("F2.txt");

    string line;

    while (getline(F2, line)) {

        cout << line << endl;

    }

    F2.close();

}

Код програми

2) Результат

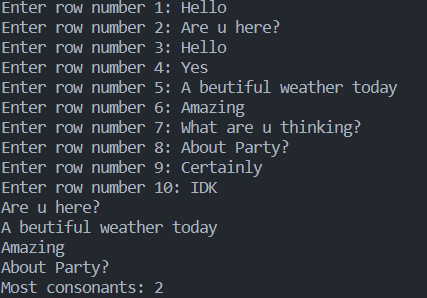


Рисунок 4 результат

3) Посилання на файл програми у пул-запиті GitHub

[Pull-Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/749/files#diff-f21c16f66c1a6b3dcbfedef6d38a467b7e0a2008bbcef44c2956c1bcc6509b2a)

Завдання №4 Algotester Lab 4

1)Код програми

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

vector<int> ost0;

vector<int> ost1;

vector<int> ost2;

vector<int> final;

bool sortDescending(int, int);

int main(){

    int N;

    cin >> N;

    int chusla[N];

    for(int i=0; i<N; i++){

        cin >> chusla[i];

    }

    for(int i=0; i<N; i++){

        if((chusla[i] % 3) == 0){

            ost0.push\_back(chusla[i]);}

        else if((chusla[i] % 3) == 1){

            ost1.push\_back(chusla[i]);}

        else if((chusla[i] % 3) == 2){

            ost2.push\_back(chusla[i]);}

    }

    sort(ost0.begin(), ost0.end());

    ost0.erase(unique(ost0.begin(), ost0.end()), ost0.end());

    sort(ost1.begin(), ost1.end(), sortDescending);

    ost1.erase(unique(ost1.begin(), ost1.end()), ost1.end());

    sort(ost2.begin(), ost2.end());

    ost2.erase(unique(ost2.begin(), ost2.end()), ost2.end());

    for(int i=0; i<ost0.size(); i++)

        final.push\_back(ost0.at(i));

    for(int i=0; i<ost1.size(); i++)

        final.push\_back(ost1.at(i));

    for(int i=0; i<ost2.size(); i++)

        final.push\_back(ost2.at(i));

    cout << final.size() << endl;

    for(int i=0; i<final.size(); i++)

        cout << final.at(i)<< " ";

return 0;

}

bool sortDescending(int a, int b) {

    return a > b;}

Код програми

2) Результат



Рисунок 5 результат в алготестері

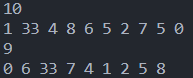


Рисунок 6 результат

3) Посилання на файл програми у пул-запиті GitHub

Pull-[Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/749/files#diff-0ac17233f4afe7b8bd02b23e8bec11ebcdd9f5aef6785f51df484496247eacb3)

Завдання № 5. Algotester Lab 6

1)Код програми

#include <iostream>

#include <vector>

#include <algorithm>

#include <sstream>

using namespace std;

vector<int> ToCheck {};

vector<int> Sorted {};

int main() {

    int N;

    cin >> N;

    //Ініціалізую числа, які я маю перевірити

    for(int i=1; i<=N; i++){

        ToCheck.push\_back(i);

    }

    int field[N][N];

    for (int i = 0; i < N; i++) {

        string row;

        cin >> row;

        for (int j = 0; j < N; j++) {

            field[i][j] = row[j] - '0';

        }

    }

    int times;

    cin >> times;

    int M = times\*2;

    int positions[M];       // координати

    for(int i = 0; i < M; i++){

        cin >> positions[i];

    }

    // for(auto pos: positions){

    //     cout << pos << " ";}

    for (int i = 0; i < times; i++) {   // Початок обчислень

        int row = positions[i\*2]-1;

        int col = positions[(i\*2)+1]-1;

        vector<int> numbers; // Вектор для додавання всіх чисел навхрест

        // прохожусь по рядку

        for (int j = 0; j < N; j++) {

            if (field[row][j] != 0)

                numbers.push\_back(field[row][j]);

        }

        // прохожусь по колонці

        for (int j = 0; j < N; j++) {

            if (field[j][col] != 0)

                numbers.push\_back(field[j][col]);

        }

        // Сортую числа

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

        // знаходжу числа

         for (int targetNumber : ToCheck) {

            auto fin = find(numbers.begin(), numbers.end(), targetNumber);

        //знову сортую

            if (fin == numbers.end()) {

                Sorted.push\_back(targetNumber);

            }

         }

         cout << endl;

         if (field[row][col]!=0){

            cout << "1" << endl;

            cout << field[row][col] << endl;}

        else if(field[row][col]==0){

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

            for (int num : Sorted) {

                cout << num << " ";

            }

        cout << endl;

        }

        Sorted.clear();

    }

    return 0;

}

Код програми

1. Результат



Рисунок 7 результат в алготестері



Рисунок 8 результат

3) Посилання на файл програми у пул-запиті GitHub

Pull-[Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/749/files#diff-87f4fb065cf6fbe4c1a3e218f304a8ea8bf667492c2add86d0cdd215cd0980f8)

Завдання № 6. Practice task1

1)Код програми

#include <fstream>

#include <string>

#include <iostream>

using namespace std;

enum FileOpResult { Success, Failure };

FileOpResult write\_to\_file(string, string);

int main() {

       cout << "Enter name: ";

        string name;

    getline(cin, name);

    cout << "Enter content: ";

    string content;

    getline(cin, content);

    FileOpResult result = write\_to\_file(name, content);

    if (result == Success) {

        cout << "File was written successfully.\n";

    } else {

        cout << "Failed to write to file.\n";

    }

    return 0;

}

FileOpResult write\_to\_file(string name, string content) {

    ofstream file(name);

    if (!file) {

        return Failure;

    }

    file << content;

    if (!file) {

        return Failure;

    }

    file.close();

    if (!file) {

        return Failure;

    }

    return Success;

}

Код програми

2) Результати



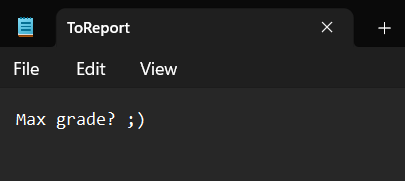
Рисунок 9 результат

Рисунок 10 результат

3) Посилання на файл програми у пул-запиті GitHub

Pull-[Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/749/files#diff-e63bc04d38f9ad273fd02512699d5604437b7536297cd351d6f3faabed2af07a)

Завдання № 7. Practice task2

1)Код програми

#include <fstream>

#include <string>

#include <iostream>

using namespace std;

enum FileOpResult { Success, Failure };

FileOpResult write\_to\_file(string, string);

FileOpResult copy\_file(string, string);

int main() {

    cout << "Enter content: ";

    string content;

    getline(cin, content);

    FileOpResult result = write\_to\_file("file\_from", content);

    if (result == Success) {

        cout << "File 'file\_from' was written successfully.\n";

    } else {

        cout << "Failed to write to file 'file\_from'.\n";

    }

    result = copy\_file("file\_from", "file\_to");

    if (result == Success) {

        cout << "File 'file\_to' was written successfully.\n";

    } else {

        cout << "Failed to write to file 'file\_to'.\n";

    }

    return 0;

}

FileOpResult write\_to\_file(string name, string content) {

    ofstream file(name);

    if (!file) {

        return Failure;

    }

    file << content;

    if (!file) {

        return Failure;

    }

    file.close();

    if (!file) {

        return Failure;

    }

    return Success;

}

FileOpResult copy\_file(string from, string to) {

    ifstream src(from, ios::binary);

    ofstream dst(to, ios::binary);

    dst << src.rdbuf();

    return src && dst ? Success : Failure;

}

Код програми

2) Результати



Рисунок 11 результат

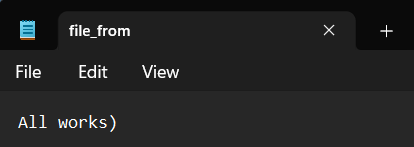
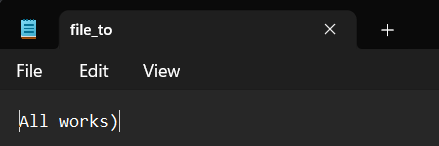


Рисунок 13 результат

Рисунок 12 результат

3) Посилання на файл програми у пул-запиті GitHub

[Pull-Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/749/files#diff-fdc03aba91e2f8ff5f1aab0cbd907db219adae70b491e2f02f4e811f79348861)

Завдання 8. Self-practice

1)Код програми

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int kabel(int&);

int povtoruvach (int&);

int zminuvach (int&);

string a;

vector<char> blocks;

int k {15};

size\_t rozmir {};

int main(){

    cin >> rozmir;

    cin >> a;

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

            blocks.push\_back(a.at(i));

        }

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

         if (blocks.at(i) == 'k'){

            kabel(k);

        }

        else if (blocks.at(i) == 'p'){

            povtoruvach(k);

        }

        else if (blocks.at(i) == 'z'){

            zminuvach(k);

        }

    }

    cout << k;

}

int kabel(int &k){

    k = (k-1);

    if (k<0)

        k=0;

return k;

}

int povtoruvach (int &k){

    if (k>0){

        k=15;

    }

    else if(k<=0){

        k=0;

    }

return k;

}

int zminuvach (int &k){

    if (k<=0){

        k=15;

    }

    else if(k>0){

        k=0;

    }

return k;

}

Код програми

2) Результати



Рисунок 14 результат

3) Посилання на файл програми у пул-запиті GitHub

[Pull-Request](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/749/files#diff-d075f978de11ff36d65a026ef919c16c5d25a88d959d2a75b731ea2116b9d78d)

***Міт з командою***

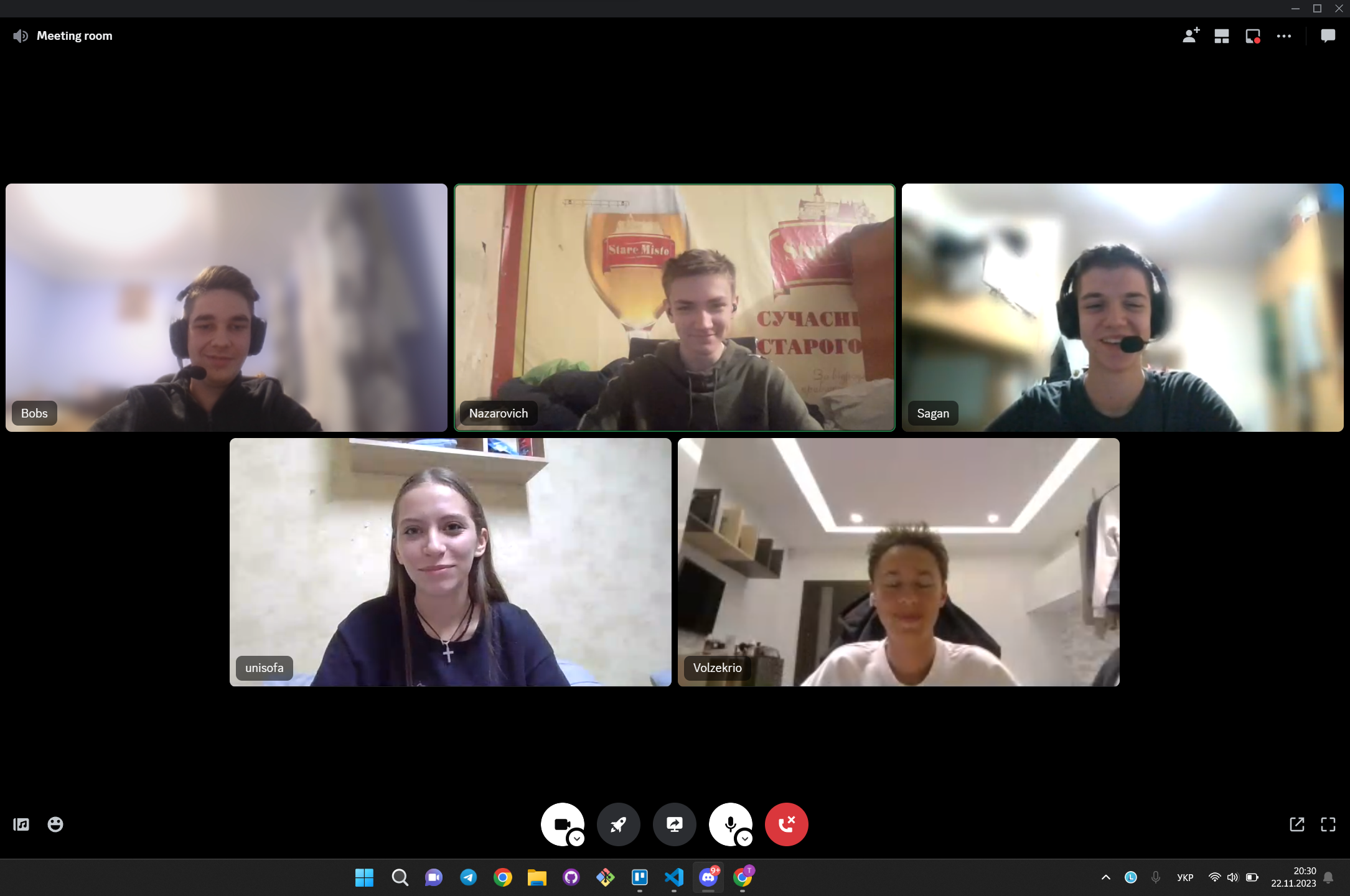


Рисунок 15 Міт з командою в Discord

# Висновок:

У процесі цього епіку я успішно оволодів основними конструкціями, файли, системи числення. бінарні файли, символи і рядкові змінні та текстові файли. Виконуючи різноманітні завдання, такі як Algotester Lab 6 мені вдалося застосувати отримані знання на практиці. Це мій ще один крок у свiт програмування і впевнений, що ці набутi навички будуть корисними для подальшого навчання та програмування.