

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Кафедра систем штучного інтелекту



## **Звіт**

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

На тему: «Програмування: алгоритм, програма, код. Системи числення.  
Двійкова система числення. Розробка та середовище розробки програми.»

**з дисципліни:** «Основи програмування»

до:

Практичних Робіт до блоку № 5

**Виконав:**  
Студент групи ШІ-11  
Силіч Богдан

Львів 2024

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

**Мета:** Ознайомитися з основними поняттями роботи з файлами, включаючи бінарні та текстові файли, символи й рядкові змінні, а також засвоїти методи роботи з файлами за допомогою стандартної бібліотеки, створення та використання власних бібліотек.

## Теоретичні відомості:

1. Вступ до Роботи з Файлами:
  - Основні операції з файлами: відкриття, читання, запис, закриття
  - Робота з файловими дескрипторами
  - C-style читання з файлу та запис до файлу
  - Перевірка стану файлу: перевірка помилок, кінець файлу
  - Базові приклади читання та запису в файл
2. Символи і Рядкові Змінні:
  - Робота з char та string: основні операції і методи
  - Стрічкові літерали та екранування символів
  - Конкатенація, порівняння та пошук у рядках
3. Текстові Файли:
  - Особливості читання та запису текстових файлів
  - Обробка рядків з файлу: getline, ignore, peek
  - Форматування тексту при записі: setw, setfill, setprecision
  - Парсинг текстових файлів: розділення на слова, аналіз структури
  - Обробка помилок при роботі з файлами
4. Бінарні Файли:
  - Вступ до бінарних файлів: відмінності від текстових, приклади (великі дані, ігрові ресурси, зображення)
  - Читання та запис бінарних даних
  - Робота з позиціонуванням у файлі: seekg, seekp
  - Серіалізація об'єктів у бінарний формат
5. Стандартна бібліотека та робота з файлами:

- Огляд стандартної бібліотеки для роботи з файлами
  - Потоки вводу/виводу: ifstream, ofstream, fstream
  - Обробка помилок при роботі з файлами
6. Створення й використання бібліотек:
- Вступ до створення власних бібліотек у C++
  - Правила розбиття коду на header-и(.h) та source(.cpp) файли
  - Статичні проти динамічних бібліотек: переваги та використання
  - Інтерфейси бібліотек: створення, документування, версіонування
  - Використання сторонніх бібліотек у проектах

## **Індивідуальний план опрацювання теорії:**

Вступ до Роботи з Файлами

Символи і Рядкові Змінні

Текстові Файли

Бінарні Файли

Стандартна бібліотека та робота з файлами

Створення й використання бібліотек

### **Джерела:**

- Chat gpt
- Список відтворення на YouTube (  
<https://youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&si=sXvmPdnGkwwJLXUi> )
- Лекції та практичні

## **Виконання роботи:**

### **VNS Lab 6 - Task 6:**

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 так, щоб всі букви в ньому були відсортовані за зростанням.

## **VNS Lab 8 - Task 6:**

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 "Школяр":

- прізвище, ім'я, по батькові;
- клас;
- номер телефону;
- оцінки по предметах (математика, фізика, українська мова, література).

Знищити всі елементи, у яких є 2 хоча б з одного предмету, додати елемент у початок файлу.

## **VNS Lab 9 - Task 6:**

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 рядки, починаючи з N до K.
- 2) Підрахувати кількість приголосних букв у файлі F2.

## **Algotester Lab 4v1:**

Вам дано 2 цілих чисел масиви, розміром NN та MM.

Ваше завдання вивести:

1. Різницю N-M
2. Різницю M-N

3. Їх перетин
4. Їх об'єднання
5. Їх симетричну різницю

### **Algotester Lab 4v3:**

Вам дано масив, який складається з  $NN$  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.

### **Algotester Lab 6v3:**

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  $N \times NN \times N$ , в якому частина клітинок заповнена цифрами, а частина клітинок пуста (позначаються нулем). Також у нього є  $QQ$  пар координат  $XX$  та  $YY$ .

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 необов'язково, 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.

### **Class Practice Work:**

Реалізувати функцію створення файлу і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult write_to_file(char *name, char *content);
```

Реалізувати функцію створення файлу і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult copy_file(char *file_from, char *file_to);
```

## **Self Practice Task:**

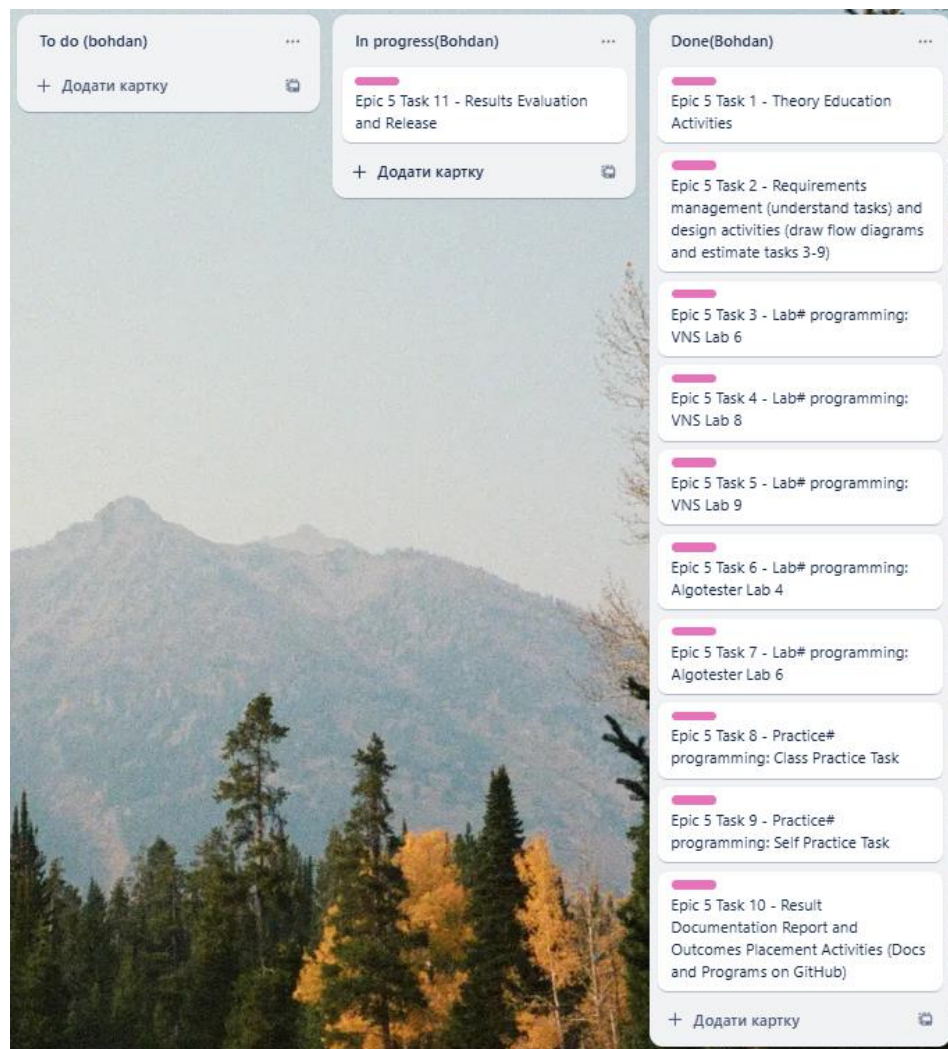
Програма що дозволяє користувачеві працювати з текстовими файлами.

Вона:

1. Запитує назву файлу.
2. Перевіряє, чи файл уже існує:
  - Якщо існує, додає текст до кінця файлу.
  - Якщо не існує, створює новий файл і записує текст.
3. Відображає вміст файлу після кожного запису.
4. Повторює процес для іншого файлу, якщо користувач бажає.
5. Завершує роботу, коли користувач обирає вихід.

## **1. Requirements management and design activities**

Team Trello dashboard for task control



Team meeting in zoom



## Time planning for tasks & tasks design

### VNS Lab 6

```
#include <iostream>
#include <algorithm>
#include <string>

using namespace std;

int main() {
    string input;
    cout << "Введіть рядок: ";
    getline(cin, input);

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

    cout << "Відсортований рядок: " << input << endl;

    return 0;
}
```

```
Введіть рядок: 1 23 543 1
Відсортований рядок: 1123345
```

+ - 15 хв  
VNS Lab 8



```

#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

struct Schoolboy {
    string surname;
    string name;
    string patronymic;
    int classNumber;
    string phoneNumber;
    int grades[4]; // 0 - математика, 1 - фізика, 2 - українська мова, 3 - література
};

void createBinaryFile(const string& filename, const vector<Schoolboy>& schoolboys) {
    ofstream outFile(filename, ios::binary);
    if (!outFile) {
        cerr << "Error: Failed to open the file for writing!" << endl;
        return;
    }

    for (const auto& schoolboy : schoolboys) {
        outFile.write(reinterpret_cast<const char*>(&schoolboy), sizeof(Schoolboy));
    }

    outFile.close();
}

void printBinaryFile(const string& filename) {
    ifstream inFile(filename, ios::binary);
    if (!inFile) {
        cerr << "Error: Failed to open the file for reading!" << endl;
        return;
    }

    Schoolboy schoolboy;
    while (inFile.read(reinterpret_cast<char*>(&schoolboy), sizeof(Schoolboy))) {
        cout << "Прізвище: " << schoolboy.surname << ", Ім'я: " << schoolboy.name
              << ", По батькові: " << schoolboy.patronymic << ", Клас: " << schoolboy.classNumber
              << ", Телефон: " << schoolboy.phoneNumber << ", Оцінки: "
              << schoolboy.grades[0] << ", " << schoolboy.grades[1] << ", "
              << schoolboy.grades[2] << ", " << schoolboy.grades[3] << endl;
    }

    inFile.close();
}

```

```

void addSchoolboy(const string& filename, const Schoolboy& newSchoolboy) {
    ofstream outFile(filename, ios::binary | ios::app);
    if (!outFile) {
        cerr << "Error: Failed to open the file for appending!" << endl;
        return;
    }

    outFile.write(reinterpret_cast<const char*>(&newSchoolboy), sizeof(Schoolboy));
    outFile.close();
}

void deleteSchoolboy(const string& filename, const string& surname) {
    ifstream inFile(filename, ios::binary);
    if (!inFile) {
        cerr << "Error: Failed to open the file for reading!" << endl;
        return;
    }

    vector<Schoolboy> schoolboys;
    Schoolboy schoolboy;

    while (inFile.read(reinterpret_cast<char*>(&schoolboy), sizeof(Schoolboy))) {
        schoolboys.push_back(schoolboy);
    }

    inFile.close();

    ofstream outFile(filename, ios::binary);
    if (!outFile) {
        cerr << "Error: Failed to open the file for writing!" << endl;
        return;
    }

    for (const auto& sb : schoolboys) {
        if (sb.surname != surname) {
            outFile.write(reinterpret_cast<const char*>(&sb), sizeof(Schoolboy));
        }
    }

    outFile.close();
}

```

```

int main() {
    string filename = "schoolboys.dat";

    vector<Schoolboy> schoolboys = {
        {"Іваненко", "Іван", "Іванович", 10, "123456789", {5, 4, 5, 5}},
        {"Петренко", "Петро", "Петрович", 9, "987654321", {4, 4, 3, 5}},
        {"Сидоренко", "Сидір", "Сидорович", 11, "456789123", {5, 5, 5, 5}}
    };

    createBinaryFile(filename, schoolboys);

    cout << "Вміст файлу:" << endl;
    printBinaryFile(filename);

    Schoolboy newSchoolboy = {"Коваленко", "Олег", "Олегович", 8, "111222333", {3, 4, 4, 3}};
    addSchoolboy(filename, newSchoolboy);

    cout << "Вміст файлу після додавання нового школяра:" << endl;
    printBinaryFile(filename);

    deleteSchoolboy(filename, "Петренко");

    cout << "Вміст файлу після видалення школяра Петренка:" << endl;
    printBinaryFile(filename);

    return 0;
}

```

Вміст файлу:

```

Прізвище: Іваненко, Ім'я: Іван, По батькові: Іванович, Клас: 10, Телефон: 123456789, Оцінки: 5, 4, 5, 5
Прізвище: Петренко, Ім'я: Петро, По батькові: Петрович, Клас: 9, Телефон: 987654321, Оцінки: 4, 4, 3, 5
Прізвище: Сидоренко, Ім'я: Сидір, По батькові: Сидорович, Клас: 11, Телефон: 456789123, Оцінки: 5, 5, 5, 5
Вміст файлу після додавання нового школяра:
Прізвище: Іваненко, Ім'я: Іван, По батькові: Іванович, Клас: 10, Телефон: 123456789, Оцінки: 5, 4, 5, 5
Прізвище: Петренко, Ім'я: Петро, По батькові: Петрович, Клас: 9, Телефон: 987654321, Оцінки: 4, 4, 3, 5
Прізвище: Сидоренко, Ім'я: Сидір, По батькові: Сидорович, Клас: 11, Телефон: 456789123, Оцінки: 5, 5, 5, 5
Прізвище: Коваленко, Ім'я: Олег, По батькові: Олегович, Клас: 8, Телефон: 111222333, Оцінки: 3, 4, 4, 3
Вміст файлу після видалення школяра Петренка:
Прізвище: Іваненко, Ім'я: Іван, По батькові: Іванович, Клас: 10, Телефон: 123456789, Оцінки: 5, 4, 5, 5
Прізвище: Сидоренко, Ім'я: Сидір, По батькові: Сидорович, Клас: 11, Телефон: 456789123, Оцінки: 5, 5, 5, 5
Прізвище: Коваленко, Ім'я: Олег, По батькові: Олегович, Клас: 8, Телефон: 111222333, Оцінки: 3, 4, 4, 3

```

+ - 2 години

VNS Lab 9

```

#include <iostream>
#include <fstream>
#include <string>
#include <cctype>
using namespace std;

int countConsonants(const string& line) {
    int count = 0;
    for (char c : line) {
        char lower = tolower(c);
        if (isalpha(lower) && !(lower == 'a' || lower == 'e' || lower == 'i' || lower == 'o' || lower == 'u' || lower == 'y')) {
            count++;
        }
    }
    return count;
}

int main() {

    ofstream fileF1("F1.txt");
    if (!fileF1.is_open()) {
        cerr << "Помилка при створенні файлу F1.txt" << endl;
        return 1;
    }

    fileF1 << "This is the first line.\n";
    fileF1 << "The second line.\n";
    fileF1 << "The third line.\n";
    fileF1 << "The fourth line.\n";
    fileF1 << "This is the fifth line.\n";
    fileF1 << "The sixth line.\n";
    fileF1 << "The seventh line.\n";
    fileF1 << "The eighth line.\n";
    fileF1 << "The ninth line.\n";
    fileF1 << "Tenth line.\n";
    fileF1.close();

    ifstream fileF1Read("F1.txt");
    ofstream fileF2("F2.txt");
    if (!fileF1Read.is_open() || !fileF2.is_open()) {
        cerr << "Помилка при відкритті файлів." << endl;
        return 1;
    }

    int N = 3, K = 6;
    string line;
    int lineNumber = 0;

    while (getline(fileF1Read, line)) {
        lineNumber++;
        if (lineNumber >= N && lineNumber <= K) {
            fileF2 << line << '\n';
        }
    }
}

```

```

fileF1Read.close();
fileF2.close();

ifstream fileF2Read("F2.txt");
if (!fileF2Read.is_open()) {
    cerr << "Помилка при відкритті файлу F2.txt" << endl;
    return 1;
}

int consonantCount = 0;
while (getline(fileF2Read, line)) {
    consonantCount += countConsonants(line);
}
fileF2Read.close();

cout << "Кількість приголосних у файлі F2: " << consonantCount << endl;

return 0;
}

```

Кількість приголосних у файлі F2: 36

+ - 40 хв

Algotester Lab 4v1

```

int main() {
    int N, M;
    cin >> N;
    vector<int> arr1(N);
    for (int i = 0; i < N; ++i) {
        cin >> arr1[i];
    }

    cin >> M;
    vector<int> arr2(M);
    for (int i = 0; i < M; ++i) {
        cin >> arr2[i];
    }

    sort(arr1.begin(), arr1.end());
    sort(arr2.begin(), arr2.end());

    vector<int> res;

    res = difference(arr1, arr2);
    cout << res.size() << endl;
    for (int num : res) cout << num << " ";
    cout << endl;

    res = difference(arr2, arr1);
    cout << res.size() << endl;
    for (int num : res) cout << num << " ";
    cout << endl;

    res = intersection(arr1, arr2);
    cout << res.size() << endl;
    for (int num : res) cout << num << " ";
    cout << endl;

    res = unionSet(arr1, arr2);
    cout << res.size() << endl;
    for (int num : res) cout << num << " ";
    cout << endl;

    res = symmetricDifference(arr1, arr2);
    cout << res.size() << endl;
    for (int num : res) cout << num << " ";
    cout << endl;

    return 0;
}

```

```

#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
using namespace std;

int main() {
    int N, M;
    cin >> N;
    vector<int> arr1(N);
    for (int i = 0; i < N; ++i) {
        cin >> arr1[i];
    }

    cin >> M;
    vector<int> arr2(M);
    for (int i = 0; i < M; ++i) {
        cin >> arr2[i];
    }

    sort(arr1.begin(), arr1.end());
    sort(arr2.begin(), arr2.end());

    vector<int> result;

    set_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(result));
    cout << result.size() << endl;
    for (int num : result) cout << num << " ";
    cout << endl;
    result.clear();

    set_difference(arr2.begin(), arr2.end(), arr1.begin(), arr1.end(), back_inserter(result));
    cout << result.size() << endl;
    for (int num : result) cout << num << " ";
    cout << endl;
    result.clear();

    set_intersection(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(result));
    cout << result.size() << endl;
    for (int num : result) cout << num << " ";
    cout << endl;
    result.clear();

    set_union(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(result));
    cout << result.size() << endl;
    for (int num : result) cout << num << " ";
    cout << endl;
    result.clear();

    set_symmetric_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(result));
    cout << result.size() << endl;
    for (int num : result) cout << num << " ";
    cout << endl;

    return 0;
}

```

```

5
1 2 3 4 5
5
4 5 6 7 8
3
1 2 3
3
1 2 3
3
6 7 8
2
4 5
8
1 2 3 4 5 6 7 8
6
1 2 3 6 7 8

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (MB)	Дії
денішка секунд тому	C++ 23	Зараховано	0.003	1.266	<a href="#">Перегляд</a>
хвилину тому	C++ 23	Неправильна відповідь: 2	0.002	0.926	<a href="#">Перегляд</a>
2 дні тому	C++ 23	Неправильна відповідь: 2	0.002	0.922	<a href="#">Перегляд</a>
2 дні тому	C++ 23	Помилка компілювання	-	-	<a href="#">Перегляд</a>
2 дні тому	C++ 23	Неправильна відповідь: 2	0.002	0.941	<a href="#">Перегляд</a>
2 дні тому	C++ 23	Неправильна відповідь: 2	0.002	0.930	<a href="#">Перегляд</a>
2 дні тому	C++ 23	Зараховано	0.003	1.230	<a href="#">Перегляд</a>
8 днів тому	C++ 23	Неправильна відповідь: 2	0.002	0.965	<a href="#">Перегляд</a>
8 днів тому	C++ 23	Зараховано	0.003	1.371	<a href="#">Перегляд</a>
8 днів тому	C++ 23	Неправильна відповідь: 2	0.002	0.934	<a href="#">Перегляд</a>

+ - 1 година

## Algotester Lab 4v3

```

#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int N;
    cin >> N;
    vector<int> arr(N);
    for (int i = 0; i < N; ++i) {
        cin >> arr[i];
    }
    sort(arr.begin(), arr.end(), [](int a, int b) {
        int modA = a % 3, modB = b % 3;
        if (modA != modB) return modA < modB;
        if (modA == 1) return a > b;
        return a < b;
    });
    arr.erase(unique(arr.begin(), arr.end()), arr.end());
    cout << arr.size() << endl;
    for (int num : arr) {
        cout << num << " ";
    }
    return 0;
}

```

```

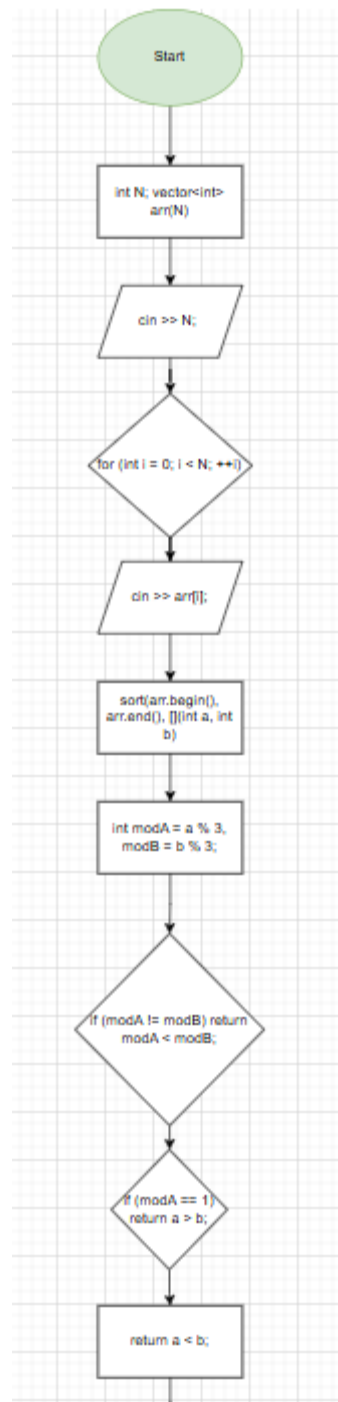
10
1 33 4 8 6 5 2 7 5 0
9
0 6 33 7 4 1 2 5 8

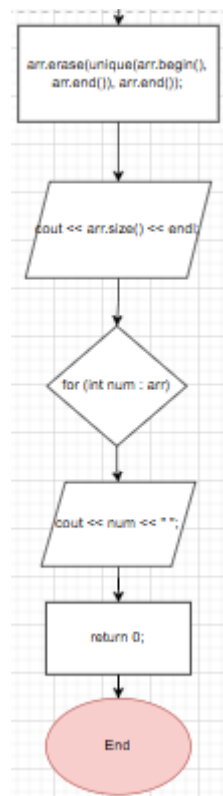
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (MB)	Дії
2 дні тому	C++ 23	Зараховано	0.003	1.313	<a href="#">Перегляд</a>
2 дні тому	C++ 23	Зараховано	0.003	1.203	<a href="#">Перегляд</a>
8 днів тому	C++ 23	Зараховано	0.003	1.258	<a href="#">Перегляд</a>

+ - 20 хв







Algotester Lab 6

```

#include <iostream>
#include <vector>
#include <set>
#include <string>

using namespace std;

vector<int> findNumbers(const vector<string> &sudoku, int x, int y) {
    vector<int> possibleNumbers;
    int N = sudoku.size();
    if (sudoku[x][y] != '0') {
        possibleNumbers.push_back(sudoku[x][y] - '0');
        return possibleNumbers;
    }
    set<int> usedNumbers;
    for (int i = 0; i < N; i++) {
        if (sudoku[x][i] != '0') usedNumbers.insert(sudoku[x][i] - '0');
        if (sudoku[i][y] != '0') usedNumbers.insert(sudoku[i][y] - '0');
    }
    for (int num = 1; num <= N; num++) {
        if (usedNumbers.find(num) == usedNumbers.end()) {
            possibleNumbers.push_back(num);
        }
    }
    return possibleNumbers;
}

int main() {
    int N;
    cin >> N;
    vector<string> sudoku(N);
    for (int i = 0; i < N; i++) {
        cin >> sudoku[i];
    }
    int Q;
    cin >> Q;
    vector<pair<int, int>> queries(Q);

    for (int i = 0; i < Q; i++) {
        int x, y;
        cin >> x >> y;
        queries[i] = {x - 1, y - 1};
    }

    for (const auto &query : queries) {
        int x = query.first;
        int y = query.second;
        vector<int> result = findNumbers(sudoku, x, y);
        cout << result.size() << endl;
        for (int num : result) {
            cout << num << " ";
        }
        cout << endl;
    }
    return 0;
}

```

```

3
000
100
003
3
1 1
3
1 1
2 3
2 1
2
2 3
1
2
1
1
1

```

Створено	Компілятор	Результат	Час (сек.)	Память (MB)	Дії
2 дні тому	C++ 23	Зараховано	0.003	1.230	<a href="#">Перегляд</a>
2 дні тому	C++ 23	Неправильна відповідь 1	0.002	0.914	<a href="#">Перегляд</a>
8 днів тому	C++ 23	Неправильна відповідь 1	0.002	0.914	<a href="#">Перегляд</a>

+ - 1 година

## Class Practice Work

```

#include <iostream>
#include <fstream>
#include <cstring>

using namespace std;

enum FileOpResult { Success, Failure };

FileOpResult write_to_file(const char *name, const char *content) {
    if (name == nullptr || content == nullptr) {
        return Failure;
    }

    ofstream file(name);
    if (!file.is_open()) {
        return Failure;
    }

    file << content;

    if (file.fail()) {
        file.close();
        return Failure;
    }

    file.close();
    return Success;
}

int main() {
    char filename[256];
    char content[1024];

    cout << "Enter filename: ";
    cin.getline(filename, sizeof(filename));

    cout << "Enter content: ";
    cin.getline(content, sizeof(content));

    FileOpResult result = write_to_file(filename, content);
    if (result == Success) {
        cout << "File written successfully." << endl;
    } else {
        cout << "Failed to write to file." << endl;
    }

    return 0;
}

```

```

#include <iostream>
#include <fstream>

using namespace std;

enum FileOpResult { Success, Failure };

FileOpResult copy_file(const char *file_from, const char *file_to) {
    if (file_from == nullptr || file_to == nullptr) {
        return Failure;
    }

    ifstream source(file_from, ios::binary);
    if (!source.is_open()) {
        return Failure;
    }

    ofstream destination(file_to, ios::binary);
    if (!destination.is_open()) {
        source.close();
        return Failure;
    }

    destination << source.rdbuf();

    if (source.fail() || destination.fail()) {
        source.close();
        destination.close();
        return Failure;
    }

    source.close();
    destination.close();
    return Success;
}

int main() {
    char source_filename[256];
    char destination_filename[256];

    cout << "Enter source filename: ";
    cin.getline(source_filename, sizeof(source_filename));

    cout << "Enter destination filename: ";
    cin.getline(destination_filename, sizeof(destination_filename));

    FileOpResult result = copy_file(source_filename, destination_filename);
    if (result == Success) {
        cout << "File copied successfully." << endl;
    } else {
        cout << "Failed to copy file." << endl;
    }

    return 0;
}

```

```
Enter filename: f3
Enter content: 123123
File written successfully.
```

```
Enter source filename: f3
Enter destination filename: f2
File copied successfully.
```

+ - 50 XB

Self Practice Work

```

#include <iostream>
#include <fstream>
#include <string>
using namespace std;

void displayFileContent(const string& filename) {
    ifstream file(filename);
    if (file.is_open()) {
        cout << "Вміст файлу \" << filename << "\":\n";
        string line;
        while (getline(file, line)) {
            cout << line << "\n";
        }
        file.close();
    } else {
        cout << "Не вдалося відкрити файл \" << filename << "\" для читання.\n";
    }
}

int main() {
    string filename, text;
    char choice;

    do {
        cout << "Введіть назву файлу (з розширенням, наприклад, file.txt): ";
        cin >> filename;

        cout << "Файл \" << filename << "\" вже існує? (y/n): ";
        cin >> choice;

        ofstream file;
        if (choice == 'y' || choice == 'Y') {
            file.open(filename, ios::app);
            if (!file.is_open()) {
                cout << "Помилка відкриття файлу \" << filename << "\".\n";
                continue;
            }
            cout << "Введіть текст для додавання до файлу: ";
        } else {
            file.open(filename);
            if (!file.is_open()) {
                cout << "Помилка створення файлу \" << filename << "\".\n";
                continue;
            }
            cout << "Введіть текст для запису нового файлу: ";
        }

        cin.ignore();
        getline(cin, text);

        file << text << "\n";
        file.close();
        cout << "Текст успішно записано до файлу \" << filename << "\".\n";

        displayFileContent(filename);

        cout << "Бажаєте працювати з іншим файлом? (y/n): ";
        cin >> choice;
    } while (choice == 'y' || choice == 'Y');

    cout << "Програма завершена.\n";
    return 0;
}

```

```
Введіть назву файлу (з розширенням, наприклад, file.txt): F2.txt
Файл "F2.txt" вже існує? (y/n): y
Введіть текст для додавання у файл: wsdfsdffs
Текст успішно записано у файл "F2.txt".
Вміст файлу "F2.txt":
The third line.
The fourth line.
This is the fifth line.
The sixth line.
wsdfsdffs
Бажаєте працювати з іншим файлом? (y/n): n
Програма завершена.
```

+/- 30 хв

## **Висновки:**

Я навчився працювати з файлами різних типів, включаючи бінарні та текстові, використовуючи символи й рядкові змінні. Завдяки стандартній бібліотеці та створенню власних бібліотек, я отримав практичні навички ефективного збереження, обробки та взаємодії з даними.