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

**з дисципліни:** «Основи програмування»

до:

ВНС Лабораторної Роботи №6

ВНС Лабораторної Роботи №8

ВНС Лабораторної Роботи №9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних робіт до блоку №5

**Виконав(ла):**

Студентка групи ШІ-11

Ільящук Марта Тарасівна

Львів 2024

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

**Мета роботи:** Навчитись працювати з файлами, символами, рядковими змінними, бібліотеками.

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1. Вступ до Роботи з Файлами
* Тема №2. Символи і Рядкові Змінні
* Тема №3. Текстові Файли
* Тема №4. Бінарні Файли
* Тема №5. Стандартна бібліотека та робота з файлами
* Тема №6. Створення й використання бібліотек

1. Індивідуальний план опрацювання теорії:

* Тема №1. Вступ до Роботи з Файлами

Джерела інформації: [Basics of File Handling in C - GeeksforGeeks](https://www.geeksforgeeks.org/basics-file-handling-c/)

Що опрацьовано: Основні операції з файлами, перевірка стану файлу

* Тема №2. Символи і Рядкові Змінні

Джерела інформації: [C Strings](https://www.w3schools.com/c/c_strings.php), [C String Functions](https://www.w3schools.com/c/c_strings_functions.php)

Що опрацьовано: Робота з char та string

* Тема №3. Текстові Файли

Джерела інформації: [C++ File handling: Create text file and write text](https://www.w3resource.com/cpp-exercises/file-handling/cpp-file-handling-exercise-1.php), [C Files I/O: Opening, Reading, Writing and Closing a file](https://www.programiz.com/c-programming/c-file-input-output)

Що опрацьовано: Особливості читання та запису текстових файлів, обробка рядків

* Тема №4. Бінарні Файли

Джерела інформації: [Reading and writing binary file in C/C++](https://www.tutorialspoint.com/reading-and-writing-binary-file-in-c-cplusplus), [How to store records in binary files in C++](https://www.youtube.com/watch?v=16NXp7c53OQ)

Що опрацьовано: Робота з бінарними файлами, відмінності від текстових

* Тема №5. Стандартна бібліотека та робота з файлами

Джерела інформації: [C++ fstream Library (File Streams) Reference](https://www.w3schools.com/cpp/cpp_ref_fstream.asp)

Що опрацьовано: Cтандартна бібліотека для роботи з файлами, потоки вводу/виводу: ifstream, ofstream, fstream

* Тема №6. Створення й використання бібліотек

Джерела інформації: [How to write your own header file in C? - GeeksforGeeks](https://www.geeksforgeeks.org/write-header-file-c/)

Що опрацьовано: Вступ до створення власних бібліотек у С++

**Виконання роботи:**

1. Опрацювання завдань та вимог до середовища:

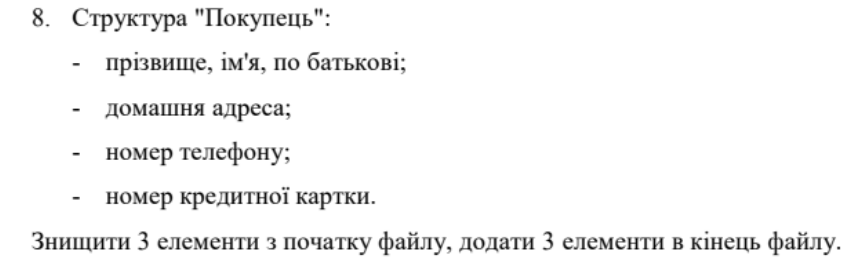
**Завдання №1. VNS Lab 6 Variant 8**

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 слова в ньому стали ідентифікаторами, слова, які складаються тільки із цифр - знищити.

**Завдання №2. VNS Lab 8 Variant 8**

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.



**Завдання №3. VNS Lab 9 Variant 8**

Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію:

1) Скопіювати з файлу F1 у файл F2 всі рядки, які не містять цифри.

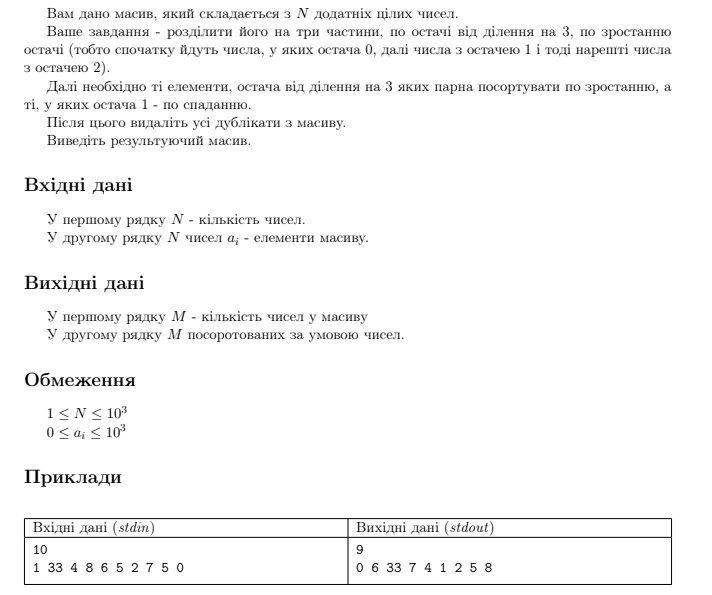
2) Підрахувати кількість рядків, які починаються на букву «А» у файлі F2.

**Завдання №4. Algotester Lab 4.1 Variant 3**

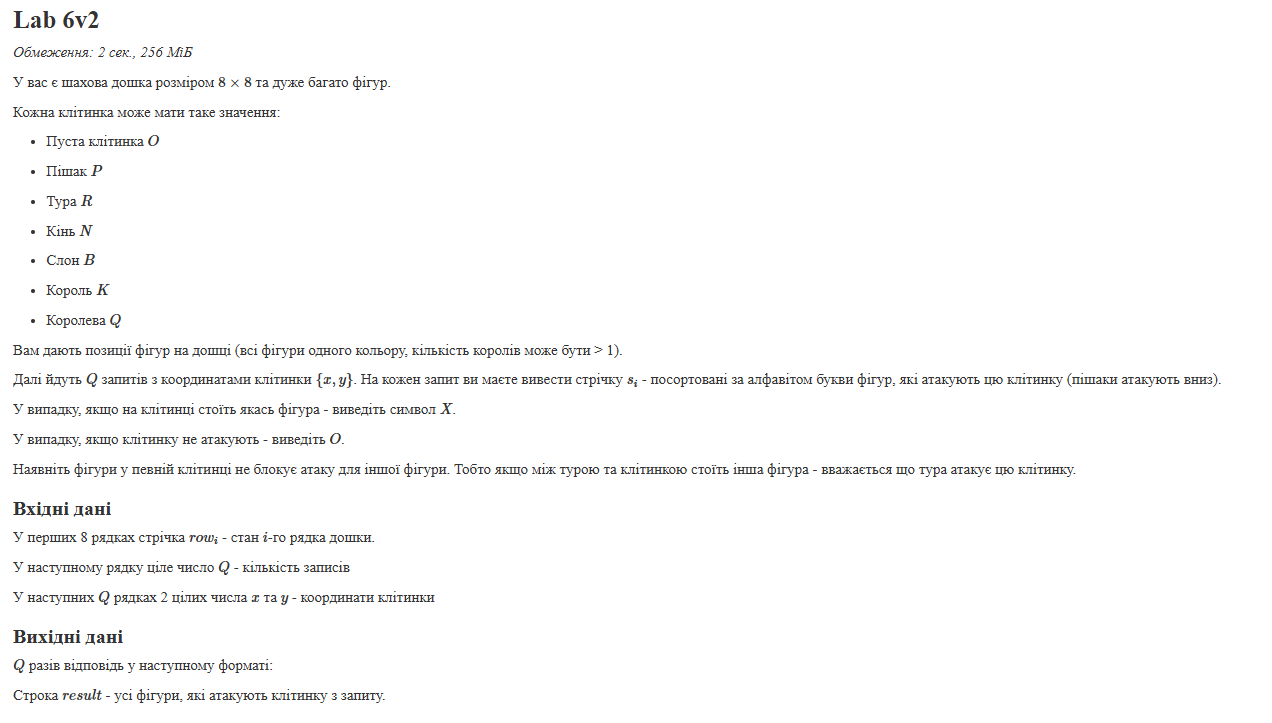
(std::partition + std::sort + std::unique)

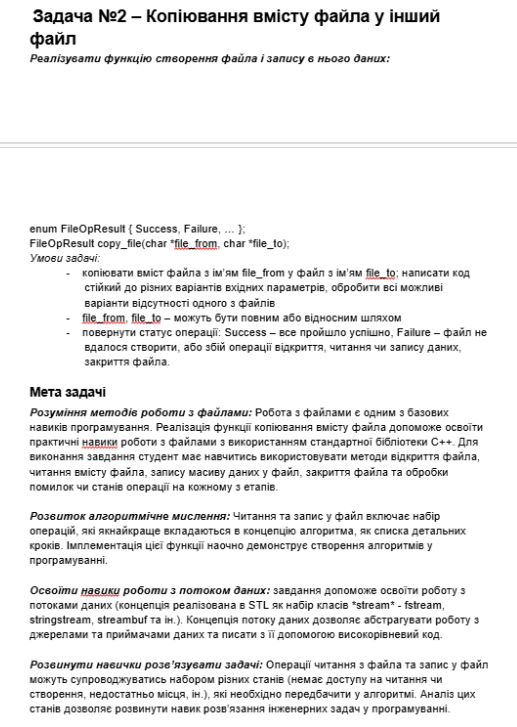
**Завдання №5. Algotester Lab 4.2 Variant 3**

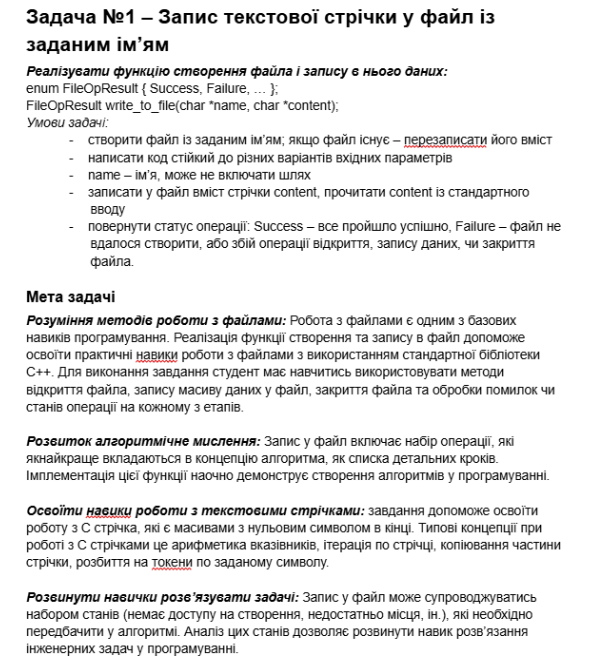
(Зi своєю реалiзацiєю)

****

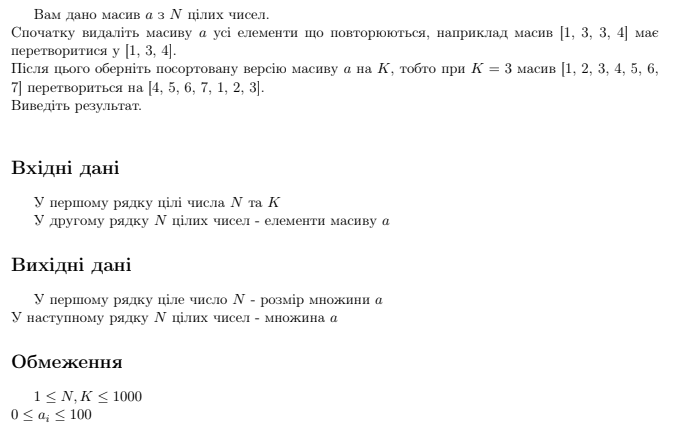
**Завдання №6. Algotester Lab 6 Variant 2**

****

**Завдання №7. Class Practice Task**

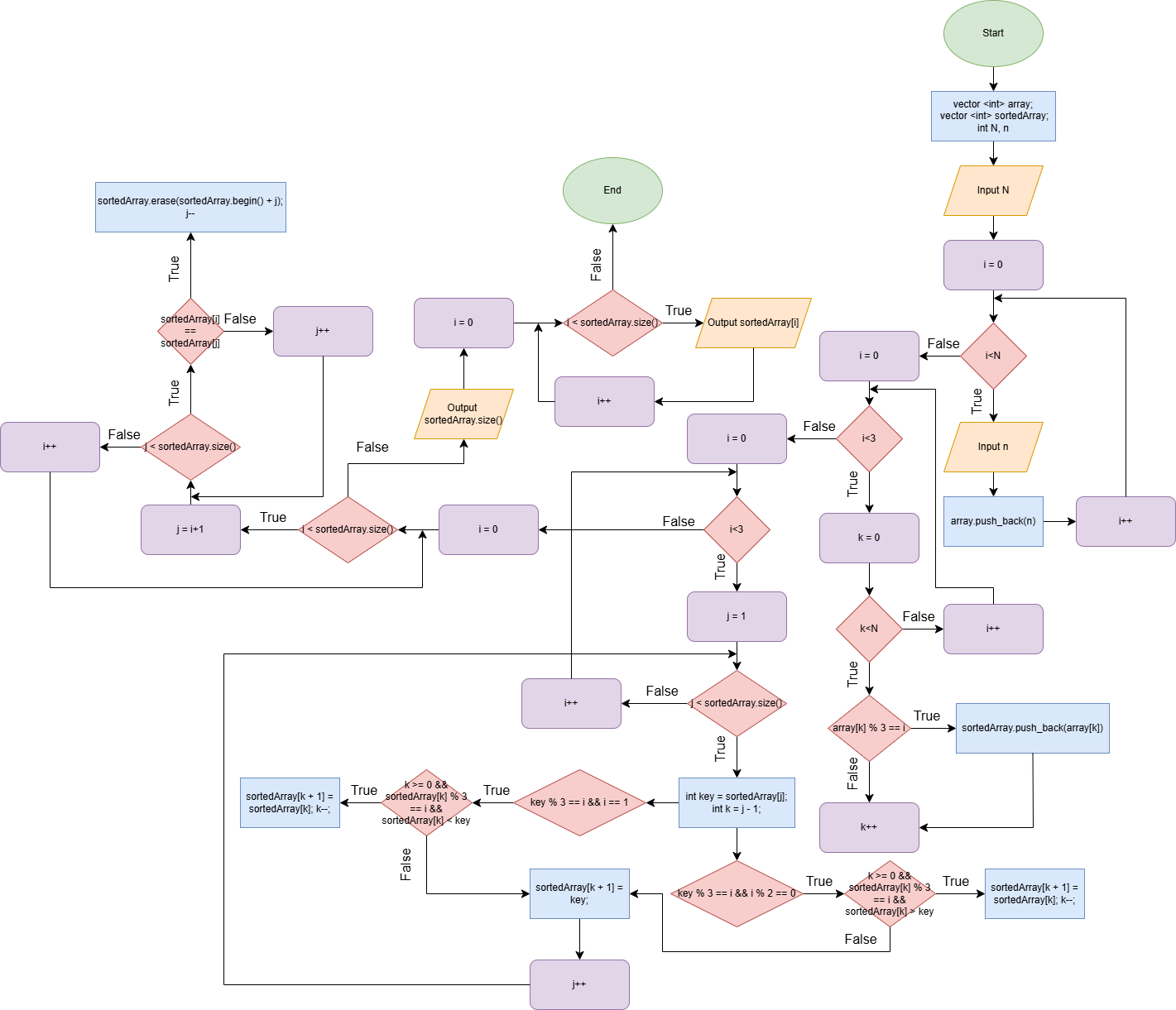


**Завданя №8. Self Practice Task Algotester Lab 4.1 Variant 2**

****

1. Дизайн виконання завдань

**Завдання №5. Algotester Lab 4.2 Variant 3**



1. Код програм з посиланням на зовнішні ресурси

[Epic 5 - Marta Iliashchuk by martailiashchuk · Pull Request #593 · artificial-intelligence-department/ai\_programming\_playground\_2024 · GitHub](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/593)

**Завдання №1. VNS Lab 6 Variant 8**

#include <iostream>

#include <string.h>

using namespace std;

int main(){

    string result;

    char str[256];

    cout<<"Enter a string: ";

    gets(str);

    char\* token = strtok(str, " ");

    while (token != nullptr){

        bool isNumeric = true;

        for (int i = 0; token[i] != '\0'; i++){

            if (!isdigit(token[i])){

                isNumeric = false;

                result+=token;

                result+="\_";

                break;

            }

        }

        token = strtok(nullptr, " ");

    }

    result.pop\_back();

    cout<<"Changed string: "<<result<<endl;

    return 0;

}

**Завдання №2. VNS Lab 8 Variant 8**

#include <iostream>

#include <vector>

using namespace std;

struct Buyer{

    char fullName[66];

    char homeAddress[66];

    int mobilePhoneNumber;

    int creditCardNumber;

};

void createFile(const char \*filename){

    FILE\* f;

    f = fopen(filename, "wb");

    if(f==NULL){

        cout<<"Error opening the file";

        exit(1);

    }

    int number;

    cout<<"Enter number of buyers: ";

    cin>>number;

    Buyer b;

    for (int i=0; i<number; i++){

        cout<<"Enter fullname: ";

        cin>>b.fullName;

        cout<<"Enter address: ";

        cin>>b.homeAddress;

        cout<<"Enter phone number: ";

        cin>>b.mobilePhoneNumber;

        cout<<"Enter credit card number: ";

        cin>>b.creditCardNumber;

        fwrite(&b, sizeof(Buyer), 1, f);

        if(ferror(f)){

            cout<<"Error writing data to the file";

            exit(2);

        }

    }

    fclose(f);

}

void printFile(const char \*filename){

    FILE\* f;

    f = fopen(filename, "rb");

    if(f==NULL){

        cout<<"Error opening the file";

        exit(1);

    }

    Buyer b;

    cout<<endl<< "File: "<<endl;

    while (fread(&b, sizeof(Buyer), 1, f)) {

        cout<<"Fullname: "<<b.fullName<<endl;

        cout<< "Home address: "<<b.homeAddress<<endl;

        cout<< "Phone number: " <<b.mobilePhoneNumber<<endl;

        cout<<"Credit card number: "<<b.creditCardNumber<<endl;

        }

    fclose(f);

}

void deleteBuyerData(const char \*filename){

    FILE\* f;

    f = fopen(filename, "rb");

    if(f==NULL){

        cout<<"Error opening the file";

        exit(1);

    }

    vector<Buyer> buyers;

    Buyer b;

    int count = 0;

    while (fread(&b, sizeof(Buyer), 1, f)){

        if(count>=3){

            buyers.push\_back(b);

        }

        count++;

    }

    fclose(f);

    f = fopen(filename, "wb");

    if(f==NULL){

        cout<<"Error opening the file";

        exit(1);

    }

    for (const auto& buyer : buyers) {

        fwrite(&buyer, sizeof(Buyer), 1, f);

    }

    if(ferror(f)){

        cout<<"Error writing data to the file";

        exit(2);

    }

    fclose(f);

}

void addBuyerData(const char \*filename){

    FILE\* f;

    f = fopen(filename, "rb");

    if(f==NULL){

        cout<<"Error opening the file";

        exit(1);

    }

    vector<Buyer> buyers;

    Buyer b;

    int count = 0;

    while (fread(&b, sizeof(Buyer), 1, f)){

        buyers.push\_back(b);

    }

    for (int i=0; i<3; i++){

        cout<<"Enter fullname: ";

        cin>>b.fullName;

        cout<<"Enter address: ";

        cin>>b.homeAddress;

        cout<<"Enter phone number: ";

        cin>>b.mobilePhoneNumber;

        cout<<"Enter credit card number: ";

        cin>>b.creditCardNumber;

        buyers.push\_back(b);

    }

    fclose(f);

    f = fopen(filename, "wb");

    if(f==NULL){

        cout<<"Error opening the file";

        exit(1);

    }

    for (const auto& buyer : buyers) {

        fwrite(&buyer, sizeof(Buyer), 1, f);

    }

    if(ferror(f)){

        cout<<"Error writing data to the file";

        exit(2);

    }

    fclose(f);

}

int main(){

    const char\* fileName = "buyerdata.dat";

    createFile(fileName);

    printFile(fileName);

    deleteBuyerData(fileName);

    printFile(fileName);

    addBuyerData(fileName);

    printFile(fileName);

    return 0;

}

**Завдання №3. VNS Lab 9 Variant 8**

#include <iostream>

#include <vector>

#include <cctype>

using namespace std;

int main() {

    string result;

    string input;

    vector<string> lines;

    string A = "A";

    int count = 0;

    FILE\* first = fopen("F1.txt", "w");

    FILE\* second = fopen("F2.txt", "w");

    if (first == nullptr || second == nullptr) {

        cerr << "Error opening file" << endl;

        exit(1);

    }

    cout << "Enter 10 strings:" << endl;

    for (int i = 0; i < 10; i++) {

        cout << i + 1 << ") ";

        getline(cin, input);

        fputs(input.c\_str(), first);

        fputs("\n", first);

        string noDigits;

        for (char wordInLine : input) {

            if (!isdigit(wordInLine)) {

                noDigits += wordInLine;

            }

        }

        lines.push\_back(noDigits);

    }

    fclose(first);

    for (const string& line : lines) {

        fputs(line.c\_str(), second);

        fputs("\n", second);

    }

    fclose(second);

    for (string line : lines){

        if(line[0] == A[0]){

            count++;

        }

    }

    cout << "Done" << endl;

    cout<<count;

    return 0;

}

**Завдання №4 - 5. Algotester Lab 4 Variant 3**

#include <iostream>

#include <vector>

#include <set>

#include <algorithm>

using namespace std;

int main(){

    int N;

    cin >> N;

    vector<int> a(N);

    for (int i = 0; i < N; ++i) cin >> a[i];

    set<int> unique\_elements(a.begin(), a.end());

    vector<int> unique\_array(unique\_elements.begin(), unique\_elements.end());

    vector<int> group0, group1, group2;

    for (int x : unique\_array) {

        if (x % 3 == 0)

            group0.push\_back(x);

        else if (x % 3 == 1)

            group1.push\_back(x);

        else

            group2.push\_back(x);

    }

    sort(group0.begin(), group0.end());

    sort(group1.rbegin(), group1.rend());

    sort(group2.begin(), group2.end());

    vector<int> result;

    result.insert(result.end(), group0.begin(), group0.end());

    result.insert(result.end(), group1.begin(), group1.end());

    result.insert(result.end(), group2.begin(), group2.end());

    cout << result.size() << endl;

    for (int x : result) cout << x << " ";

    cout << endl;

    return 0;

}

#include <iostream>

#include <vector>

using namespace std;

int main() {

    vector<int> array;

    vector<int> sortedArray;

    int N, n;

    cin >> N;

    for (int i = 0; i < N; i++) {

        cin >> n;

        array.push\_back(n);

    }

    for (int i = 0; i < 3; i++) {

        for (int k = 0; k < N; k++) {

            if (array[k] % 3 == i) {

                sortedArray.push\_back(array[k]);

            }

        }

    }

    // Сортування вставками

    for (int i = 0; i < 3; i++) {

        for (int j = 1; j < sortedArray.size(); j++) {

            int key = sortedArray[j];

            int k = j - 1;

            if (key % 3 == i && i % 2 == 0) {

                while (k >= 0 && sortedArray[k] % 3 == i && sortedArray[k] > key) {

                    sortedArray[k + 1] = sortedArray[k];

                    k--;

                }

            }

            else if (key % 3 == i && i == 1) {

                while (k >= 0 && sortedArray[k] % 3 == i && sortedArray[k] < key) {

                    sortedArray[k + 1] = sortedArray[k];

                    k--;

                }

            }

            sortedArray[k + 1] = key;

        }

    }

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

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

            if (sortedArray[i] == sortedArray[j]) {

                sortedArray.erase(sortedArray.begin() + j);

                j--;

            }

        }

    }

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

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

        cout << sortedArray[i] << " ";

    }

    return 0;

}

**Завдання №6. Algotester Lab 6 Variant 2**

#include <iostream>

#include <string>

#include <vector>

#include <set>

#include <algorithm>

using namespace std;

bool movePiece(char element, int elementX, int elementY, int x, int y) {

    switch (element) {

        case 'R':

            return elementX == x || elementY == y;

        case 'B':

            return abs(elementX - x) == abs(elementY - y);

        case 'N':

            return (abs(elementX - x) == 2 && abs(elementY - y) == 1) ||

                   (abs(elementX - x) == 1 && abs(elementY - y) == 2);

        case 'P':

            return elementX + 1 == x && abs(elementY - y) == 1;

        case 'K':

            return abs(elementX - x) <= 1 && abs(elementY - y) <= 1;

        case 'Q':

            return (elementX == x || elementY == y) ||

                   (abs(elementX - x) == abs(elementY - y));

        default:

            return false;

    }

}

int main() {

    vector<string> rows(8);

    for (int i = 0; i < 8; i++) {

        cin >> rows[i];

    }

    int Q;

    cin >> Q;

    vector<string> result(Q);

    for (int k = 0; k < Q; k++) {

        int x, y;

        cin >> x >> y;

        x--;

        y--;

        if (rows[x][y] != 'O') {

            result[k] = "X";

            continue;

        }

        set<char> pieces;

        for (int i = 0; i < 8; i++) {

            for (int j = 0; j < 8; j++) {

                char element = rows[i][j];

                if (element != 'O' && movePiece(element, i, j, x, y)) {

                    pieces.insert(element);

                }

            }

        }

        if (pieces.empty()) {

            result[k] = "O";

        } else {

            string attackPiece(pieces.begin(), pieces.end());

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

            result[k] = attackPiece;

        }

    }

    for (const string& r : result) {

        cout << r << endl;

    }

    return 0;

}

**Завдання №7. Class Practice Task**

#include <iostream>

#include <fstream>

using namespace std;

enum FileOpResult {Success, FailureCreate, FailureOpen, FailureWrite};

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

    if (name == nullptr){

        return FailureCreate;

    }

    ofstream file(name);

    if (!file.is\_open()) {

        return FailureOpen;

    }

    file<<content;

    if(file.fail()){

        return FailureWrite;

    }

    file.close();

    return Success;

}

int main(){

    char fileName[124];

    char content[777];

    cout<<"Enter filename: ";

    cin.getline(fileName, 100);

    cout<<"Enter content: ";

    cin.getline(content, 1000);

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

    if (result == Success) {

        cout << "Data successfully written to the file";

    }

    else if (result == FailureCreate) {

        cout << "Failed to create the file: missing filename";

    }

    else if (result == FailureOpen) {

        cout << "Error opening the file";

    }

    else if (result == FailureWrite) {

        cout << "Error writing data to the file";

    }

    return 0;

}

**Завданя №8. Self Practice Task Algotester Lab 4.1 Variant 2**

#include <iostream>

#include <vector>

#include <algorithm>

#include <set>

using namespace std;

int main() {

    int N, K;

    cin>>N>>K;

    vector<int> a(N);

    for (int i = 0; i < N; i++) {

        cin>>a[i];

    }

    set<int> uniqueElements(a.begin(), a.end());

    vector<int> uniqueArray(uniqueElements.begin(), uniqueElements.end());

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

    rotate(uniqueArray.begin(), uniqueArray.begin() + K % uniqueArray.size(), uniqueArray.end());

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

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

        cout<<uniqueArray[i]<<" ";

    }

    return 0;

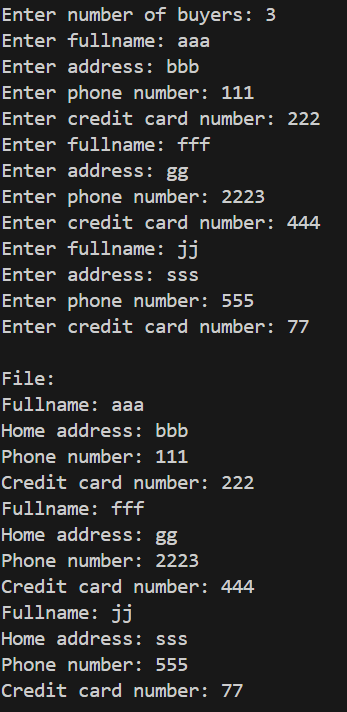
}

1. Результати виконання завдань

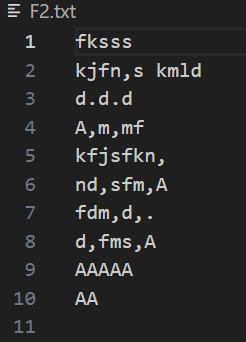
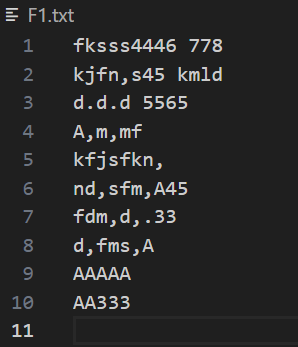
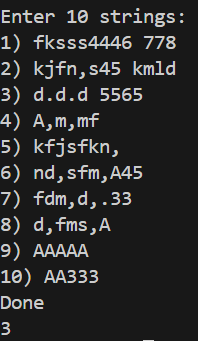
**Завдання №1. VNS Lab 6 Variant 8**

****

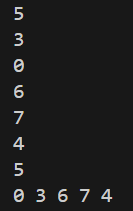
**Завдання №2. VNS Lab 8 Variant 8**

****

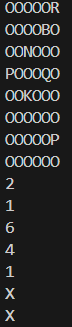
**Завдання №3. VNS Lab 9 Variant 8**

****

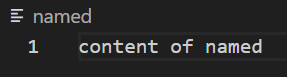
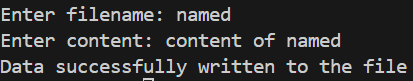
**Завдання №4 - 5. Algotester Lab 4 Variant 3**

****

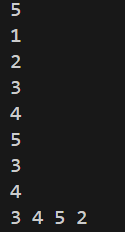
**Завдання №6. Algotester Lab 6 Variant 2**

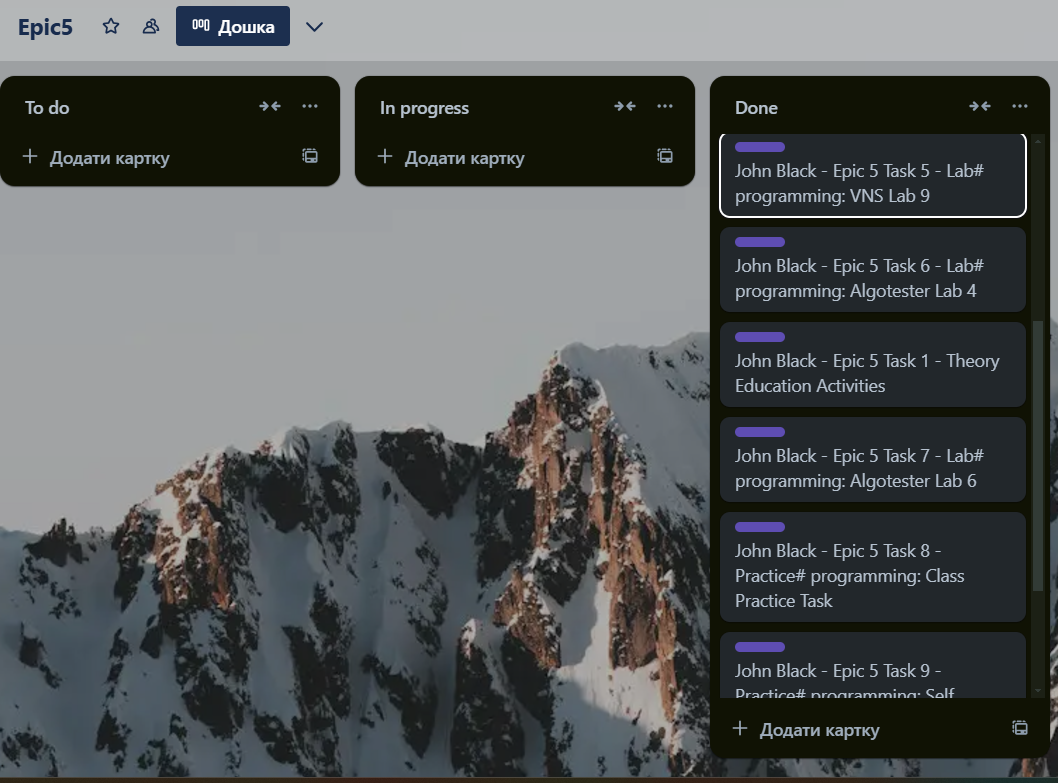
****

**Завдання №7. Class Practice Task**

****

**Завданя №8. Self Practice Task Algotester Lab 4.1 Variant 2**

****

1. ****Кооперація з командою

**Висновок:** Під час виконання роботи я теоретично ознайомилась та на практиці закріпила знання про роботу з текстовими, бінарними файлами у мові С/С++.