Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 6, 8, 9**

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

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

***Виконала:***

студентка групи ШІ-11

Педченко Юлія Сергіївна

# **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:**

Вивчити теоретичний матеріал та вміти використовувати все вищеперелічене на практиці.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Рядки.
* Тема №2: Робота з файлами.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Рядки.
  + Джерела Інформації
    - <http://cpp.dp.ua/vykorystannya-ryadkiv-typu-string/>
    - <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/73_C++/index.html>
    - <https://youtu.be/1FkTJYm-T34?feature=shared>
    - <https://cppstudy.wordpress.com/2009/03/27/cin-get-and-co/>
  + Що опрацьовано:
    - Вивчено інформацію про рядки, їх синтаксис і використання.
  + Статус: Ознайомлений
  + Початок опрацювання теми: -
  + Звершення опрацювання теми: -
* Тема №2: Робота з файлами.
  + Джерела Інформації:
    - <https://youtu.be/SSNJ7alki-E?feature=shared>
    - <https://youtu.be/FeNqHytI0fA?feature=shared>
    - <https://youtu.be/CBnB2fvfu_I?feature=shared>
    - <https://youtu.be/aUP0eAEIxog?feature=shared>
    - <https://purecodecpp.com/archives/2751>
    - <https://itproger.com/ua/course/cpp/13>
    - <https://ravesli.com/urok-212-bazovyj-fajlovyj-vvod-vyvod/>
    - <https://www.bestprog.net/ru/2019/09/11/examples-of-using-c-tools-for-working-with-files-ru/>
  + Що опрацьовано:
    - Опрацьовано матеріал щодо роботи з файлами через код С++
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: -
  + Звершення опрацювання теми: -

# **Виконання роботи:**

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

Завдання №1 Lab# programming: VNS Lab 6

* Варіант 1
* 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.

* Надрукувати найдовше й найкоротше слово в цьому рядку.

Завдання №2 Lab# programming: VNS Lab 8

* Варіант 1
* Сформувати двійковий файл із елементів, заданої у варіанті структури,

роздрукувати його вмістиме, виконати знищення й додавання елементів у

відповідності зі своїм варіантом, використовуючи для пошуку елементів що

знищуються чи додаються, функцію. Формування, друк, додавання й знищення

елементів оформити у вигляді функцій. Передбачити повідомлення про

помилки при відкритті файлу й виконанні операцій вводу/виводу.

* Структура "Абітурієнт":

- прізвище, ім'я, по батькові;

- рік народження;

- оцінки вступних іспитів (3);

- середній бал атестата.

Знищити елемент із зазначеним номером, додати елемент після елемента із

зазначеним прізвищем.

Завдання №3 Lab# programming: VNS Lab 9

* Варіант 1
* Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього

інформацію

Виконати завдання.

* 1) Скопіювати у файл F2 тільки парні рядки з F1.

2) Підрахувати розмір файлів F1 й F2 (у байтах).

Завдання №4 Lab# programming: Algotester Lab 4.1

* Варіант 2
* Вам дано масив a з N цілих чисел. Спочатку видаліть з масиву a усі елементи що повторюються. Після цього оберніть посортовану версію масиву а на К. Виведіть результат. Використайте засоби STL.

Завдання №5 Lab# programming: Algotester Lab 4.2

* Варіант 2
* Вам дано масив a з N цілих чисел. Спочатку видаліть з масиву a усі елементи що повторюються. Після цього оберніть посортовану версію масиву а на К. Виведіть результат. Використайте свою реалізацію.

Завдання №6 Lab# programming: Algotester Lab 6

* Варіант 2
* У вас є шахова дошка розміром 8х8 та дуже багато фігур. Кожна клітинка має якесь значення. Вам дають позиції фігур на дошці. Далі ідуть q запитів з координатами клітинки. На кожен запит ви маєте вивести стрічку – посортовані за алфавітом букв фігур, які атакують цю клітинку.

У випадку, якщо на клітинці стоїть якась фігура – виведіть символ «Х». У випадку, якщо клітинку не атакують – виведіть «О». Наявність фігури у певній клітинці не блокує атаку для іншої фігури.

Завдання №7 Practice# programming: Class Practice Task 1

* - 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.

Завдання №8 Practice# programming: Class Practice Task 2

* -       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.

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

Програма №1 Lab# programming: VNS Lab 6

* Планований час на реалізацію: 30 хвилин

Програма №2 Lab# programming: VNS Lab 8

* Планований час на реалізацію: 3 години

Програма №3 Lab# programming: VNS Lab 9

* Планований час на реалізацію: 30 хвилин

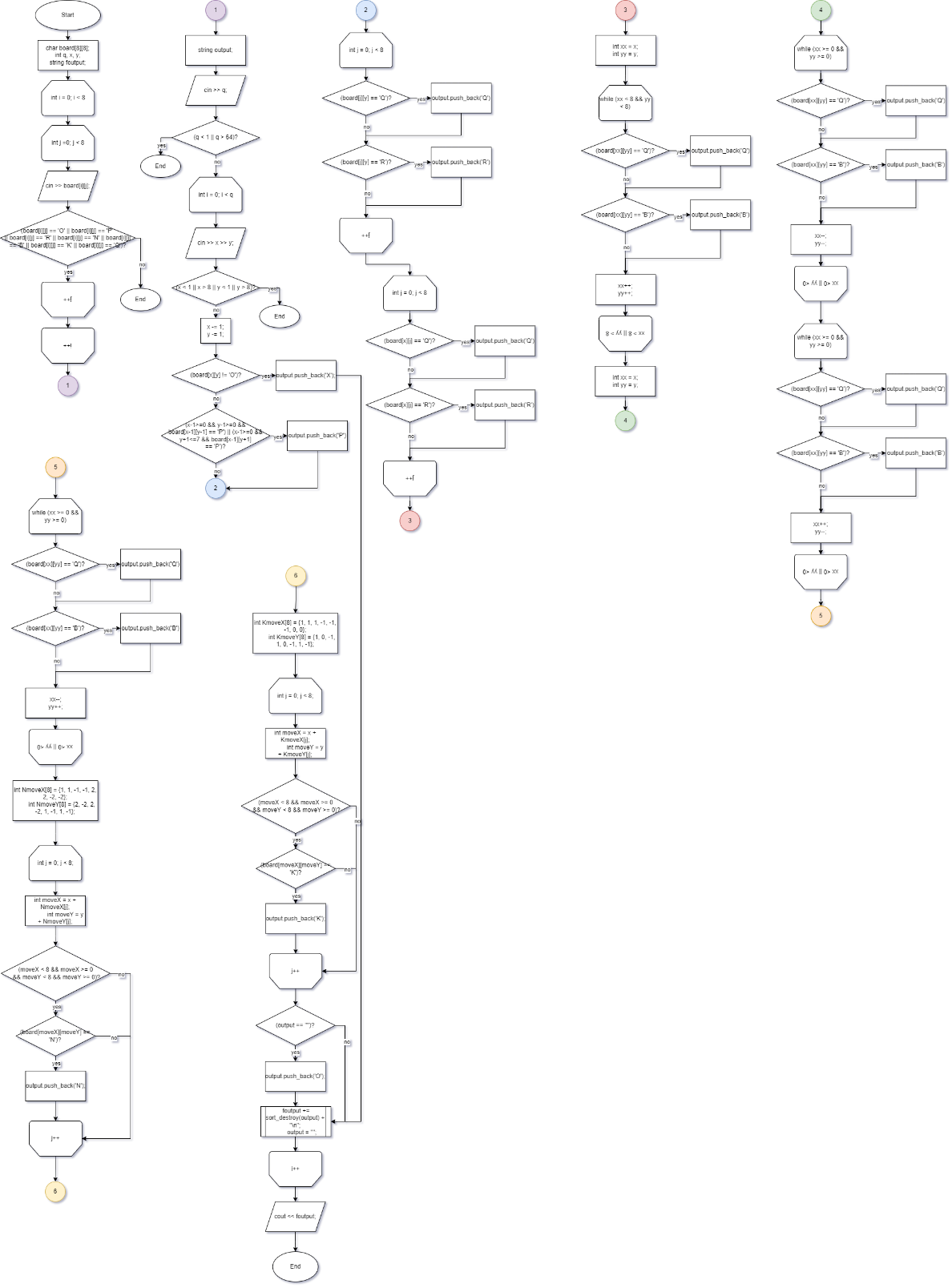
Програма №4 Lab# programming: Algotester Lab 4.1

* Планований час на реалізацію: 15 хвилин

Програма №5 Lab# programming: Algotester Lab 4.2

* Планований час на реалізацію: 25 хвилин

Програма №6 Lab# programming: Algotester Lab 6



* Планований час на реалізацію: 2 години

Програма №7 Practice# programming: Class Practice Task 1

* Планований час на реалізацію: 15 хвилин

Програма №8 Practice# programming: Class Practice Task 2

* Планований час на реалізацію: 15 хвилин

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

-

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

Pull-request: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1116>

Програма №1 Lab# programming: VNS Lab 6

#include <iostream>

#include <string>

#include <sstream>

using *namespace* std;

*int* main()

{

    string longestWord, shortestWord, input;

*bool* first = true;

    cout << "Введіть рядок англійською, відокремлюючи слова пробілом і поставивши в кінці крапку" << endl;

    getline(cin, input);

    if (input[input.size() - 1] == '.') { // - прибираємо крапку в кінці останнього слова

        input = input.substr(0, input.size() - 1);

    }

    stringstream stream(input);

    while (stream >> input) {

        if (first) {  // - першим значення найдовшого і найкоротшого слова задається перше

                      // слово вводу, щоб було з чим порівнювати

            longestWord = input;

            shortestWord = input;

            first = false;

        }

        if (input.size() > longestWord.size()) {

            longestWord = input;

        }

        if (input.size() < shortestWord.size()) {

            shortestWord = input;

        }

    }

    cout << "The longest word in sentence is: " << longestWord << endl;

    cout << "The shortest word in sentence is: " << shortestWord;

    return 0;

}

Код завдання №1

Програма №2 Lab# programming: VNS Lab 8

#include <iostream>

#include <fstream>

#include <vector>

using *namespace* std;

///////////////////////////////////////////////////////////////////////

*struct* Abiturients {

    string sec\_name;

    string name;

    string po\_bati;

    string birth\_year;

    string mark1;

    string mark2;

    string mark3;

    string average;      };

const string file\_path = "C:/REPOSITORY/ai\_programming\_playground/ai\_11/yuliia\_pedchenko/textfiles/Abiturients.txt";

///////////////////////////////////////////////////////////////////////

*void* data\_line(string *path*, Abiturients *info*) {

    ofstream output\_info;

    output\_info.open(*path*, ios::app);

    output\_info << *info*.sec\_name << ", "

                << *info*.name << ", "

                << *info*.po\_bati << ", "

                << *info*.birth\_year << ", "

                << *info*.mark1 << ", "

                << *info*.mark2 << ", "

                << *info*.mark3 << ", "

                << *info*.average << ";" << endl;

    output\_info.close();

}

///////////////////////////////////////////////////////////////////////

string read\_file(string *path*) {

    string info;

    string line;

    ifstream read\_file;

    read\_file.open(*path*, ios::app);

    while (getline(read\_file, line)) {

        cout << line << endl;

        info += line;

    }

    return info;

}

///////////////////////////////////////////////////////////////////////

vector<Abiturients> info\_as\_mass(string *info*) {

    vector<string> s\_vector;

    string temp;

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

        if (*info*[i] == ',' || *info*[i] == ';' ) {

            s\_vector.push\_back(temp);

            temp = "";

        }

        else if (*info*[i] == ' ') {}

        else {

            temp += *info*[i];

        }

    }

    vector <Abiturients> inf\_as\_mass;

    for (*int* i = 0; i < s\_vector.size()-1; i += 8) {

        Abiturients mtemp = {s\_vector[i], s\_vector[i+1], s\_vector[i+2], s\_vector[i+3], s\_vector[i+4], s\_vector[i+5], s\_vector[i+6], s\_vector[i+7]};

        inf\_as\_mass.push\_back(mtemp);

    }

    return inf\_as\_mass;

}

///////////////////////////////////////////////////////////////////////

vector<Abiturients> erase\_element(vector<Abiturients> *mass*, *int* *number*) {

    vector<Abiturients> new1mass;

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

        if (i != (*number*-1)) {

            new1mass.push\_back(*mass*[i]);

        }

    }

    return new1mass;

}

///////////////////////////////////////////////////////////////////////

*int* main() {

    ifstream check\_file;

    check\_file.open(file\_path);

     if (!check\_file) {

        cout << "File with this path does not exist" << endl;

        return 0;

    }

    check\_file.close();

    ofstream delete\_data(file\_path, ios::trunc);

    delete\_data.close();

    data\_line(file\_path, {"Pedchenko", "Yuliia", "Serghiivna", "2006", "200", "186", "181", "9.9" });

    data\_line(file\_path, {"Bartysheva", "Valentyna", "Serghiivna", "2007", "143", "180", "164", "8.5" });

    data\_line(file\_path, {"Piven", "Tymofiy", "Konstantynovych", "2006", "175", "186", "170", "10.3" });

    data\_line(file\_path, {"Shchurlin", "Prokhor", "Oleksandrovych", "2005", "200", "200", "200", "10.9" });

    data\_line(file\_path, {"Koriakov", "Pavlo", "Olegovych", "2005", "172", "200", "193", "11" });

    cout << "Current info in file:" << endl;

    string cinfo = read\_file(file\_path);

    vector<Abiturients> element\_mass = info\_as\_mass(cinfo);

*int* number;

    string nazwisko;

    cout << "Which line do you want to delete? (Enter number)" << endl;

    cin >> number;

    if (number < 1 || number > 5) {

        cout << "Please try number between 1 and 5";

        return 0;

    }

    vector <Abiturients> new\_element\_mass = erase\_element(element\_mass, number);

    cout << "List now is builded with these elements:" << endl;

    for (*int* i = 0; i < new\_element\_mass.size(); i++) {

        cout    << new\_element\_mass[i].sec\_name << ", "

                << new\_element\_mass[i].name << ", "

                << new\_element\_mass[i].po\_bati << ", "

                << new\_element\_mass[i].birth\_year << ", "

                << new\_element\_mass[i].mark1 << ", "

                << new\_element\_mass[i].mark2 << ", "

                << new\_element\_mass[i].mark3 << ", "

                << new\_element\_mass[i].average << "; " << endl;

    }

    cout << endl << "After which last name you want to add new line?" << endl;

    cin >> nazwisko;

    ofstream delete\_data2(file\_path, ios::trunc);

    delete\_data2.close();

    for (*int* i = 0; i < new\_element\_mass.size(); i++) {

        if (new\_element\_mass[i].sec\_name == nazwisko) {

        data\_line(file\_path,    {new\_element\_mass[i].sec\_name, new\_element\_mass[i].name, new\_element\_mass[i].po\_bati,

                                 new\_element\_mass[i].birth\_year, new\_element\_mass[i].mark1, new\_element\_mass[i].mark2,

                                 new\_element\_mass[i].mark3, new\_element\_mass[i].average});

        data\_line(file\_path, {"Burunduk", "Roman", "Viktorovych", "2004", "100", "109", "132", "3.3" });

        }

        else data\_line(file\_path,    {new\_element\_mass[i].sec\_name, new\_element\_mass[i].name, new\_element\_mass[i].po\_bati,

                                 new\_element\_mass[i].birth\_year, new\_element\_mass[i].mark1, new\_element\_mass[i].mark2,

                                 new\_element\_mass[i].mark3, new\_element\_mass[i].average});

    }

    cout << "List now is builded with these elements:" << endl << endl;

    read\_file(file\_path);

    return 0;

}

Код завдання №2

Програма №3 Lab# programming: VNS Lab 9

#include <iostream>

#include <fstream>

#include <vector>

using *namespace* std;

const string f1\_path = "C:/REPOSITORY/ai\_programming\_playground/ai\_11/yuliia\_pedchenko/textfiles/F1.txt";

const string f2\_path = "C:/REPOSITORY/ai\_programming\_playground/ai\_11/yuliia\_pedchenko/textfiles/F2.txt";

////////////////////////////////////////////////////////////////////////////////////////////////////////

vector<string> read\_file(string *path*) {

    vector<string> info;

    string line;

    ifstream read\_file;

    read\_file.open(*path*, ios::app);

    while (getline(read\_file, line)) {

        cout << line << endl;

        info.push\_back(line);

    }

    return info;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////

*void* copy\_info(vector<string> *info*, string *path*) {

    ofstream put\_info;

    put\_info.open(*path*, ios::app);

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

        if (i%2==0) {

            put\_info << *info*[i] << endl;

        }

    }

}

////////////////////////////////////////////////////////////////////////////////////////////////////////

*int* main() {

    ofstream delete\_data(f2\_path, ios::trunc);

    delete\_data.close();

    ifstream check\_file;

    check\_file.open(f1\_path);

     if (!check\_file) {

        cout << "File 'F1' does not exist" << endl;

        return 0;

    }

    check\_file.close();

    ifstream check\_file2;

    check\_file2.open(f2\_path);

     if (!check\_file2) {

        cout << "File 'F2' does not exist" << endl;

        return 0;

    }

    check\_file2.close();

    cout << endl <<"Data of the F1 file is: " << endl;

    vector <string> F1\_info = read\_file(f1\_path);

    copy\_info(F1\_info, f2\_path);

    cout << endl <<"Data of the F2 file is: " << endl;

    read\_file(f2\_path);

    ifstream in\_file1(f1\_path, ios::binary);

    in\_file1.seekg(0, ios::end);

*int* sizef1 = in\_file1.tellg();

     ifstream in\_file2(f2\_path, ios::binary);

    in\_file2.seekg(0, ios::end);

*int* sizef2 = in\_file2.tellg();

    cout    << endl << "Size of F1 file is " << sizef1 << " bytes" << endl

            << "Size of F2 file is " << sizef2 << " bytes" << endl;

    return 0;

}

Код завдання №3

Програма №4 Lab# programming: Algotester Lab 4.1

#include <iostream>

#include <vector>

#include <algorithm>

using *namespace* std;

*int* main() {

    vector<*int*> a;

*int* N, K;

    cin >> N >> K;

    if (N < 1 || K < 1 || N > 1000 || K > 1000) return 0;

    for (*int* i = 0; i < N; i++) {

*int* ai;

        cin >> ai;

        if (ai < 0 || ai > 100) return 0;

        a.push\_back(ai);

    }

    sort(a.begin(), a.end());

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

    if ( K > a.size()) {

        while (K > a.size()) {

            K -= a.size();

        }

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

    }

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

    cout << a.size() << endl;

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

        cout << a[i] << " ";

    }

    return 0;

}

Код завдання №4

Програма №5 Lab# programming: Algotester Lab 4.2

#include <iostream>

#include <vector>

using *namespace* std;

//////////////////////////////////////////////////////////////////

vector<*int*> bubblesort(vector<*int*> *massive*) {

    for (*int* i = 0; i < *massive*.size()-1; i++) {

        for (*int* j = 0; j < *massive*.size()-i-1; j++) {

            if (*massive*[j] > *massive*[j+1]) {

               swap(*massive*[j], *massive*[j+1]);

            }

        }

    }

    return *massive*;

}

//////////////////////////////////////////////////////////////////

vector<*int*> destroy(vector<*int*> *massive*) {

    for (*int* i = 1; i < *massive*.size()-1; i++) {

        if (*massive*[i] == *massive*[i-1] || *massive*[i] == *massive*[i+1]) {

*massive*.erase(*massive*.begin()+i);

        i--;

        }

    }

    return *massive*;

}

//////////////////////////////////////////////////////////////////

vector<*int*> rotat(vector<*int*> *massive*, *int* *K*) {

    for (*int* i = 0; i < *K*; i++) {

        for (*int* j = 0; j < *massive*.size()-1; j++) {

            swap(*massive*[j], *massive*[j+1]);

        }

    }

    return *massive*;

}

//////////////////////////////////////////////////////////////////

*int* main() {

    vector<*int*> a;

*int* N, K;

    cin >> N >> K;

    if (N < 1 || K < 1 || N > 1000 || K > 1000) return 0;

    for (*int* i = 0; i < N; i++) {

*int* ai;

        cin >> ai;

        if (ai < 0 || ai > 100) return 0;

        a.push\_back(ai);

    }

    a = bubblesort(a);

    a = destroy(a);

    a = rotat(a, K);

    cout << a.size() << endl;

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

        cout << a[i] << " ";

    }

    return 0;

}

Код завдання №5

Програма №6 Lab# programming: Algotester Lab 6

#include <iostream>

#include <string>

#include <algorithm>

using *namespace* std;

/////////////////////////////////////////////////////////////////////////////

string sort\_destroy(string *answer*) {

    sort(*answer*.begin(), *answer*.end());

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

        if (*answer*[i]==*answer*[i+1]) {

*answer*.erase(*answer*.begin()+i);

            i--;

        }

    }

    return *answer*;

}

/////////////////////////////////////////////////////////////////////////////

*int* main() {

*char* board[8][8];

*int* q, x, y;

    string foutput;

    for (*int* i = 0; i < 8; i++) {

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

        cin >> board[i][j];

        if (board[i][j] == 'O' || board[i][j] == 'P' || board[i][j] == 'R' ||

            board[i][j] == 'N' || board[i][j] == 'B' || board[i][j] == 'K' || board[i][j] == 'Q');

        else return 0;

        }

    }

    string output;

    cin >> q;

    if (q < 1 || q > 64) return 0;

    for (*int* i = 0; i < q; i++) {

        cin >> x >> y;

        if (x < 1 || x > 8 || y < 1 || y > 8) return 0;

        x -= 1;

        y -= 1;

/////////////////////////////////////////////////////////////////////////////

        if (board[x][y] != 'O') {

            output.push\_back('X');

        }

        else {

/////////////////////////////////////////////////////////////////////////////

        if((x-1>=0 && y-1>=0 && board[x-1][y-1] == 'P') || (x-1>=0 && y+1<=7 && board[x-1][y+1] == 'P')) {

            output.push\_back('P');

        }

/////////////////////////////////////////////////////////////////////////////

        for (*int* j = 0; j < 8; j++) {   //перевірка на королеву і туру

            if(board[j][y] == 'Q') {

                output.push\_back('Q');

            }

            if(board[j][y] == 'R') {

                output.push\_back('R');

            }

        }

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

            if(board[x][j] == 'Q') {

                output.push\_back('Q');

            }

            if(board[x][j] == 'R') {

                output.push\_back('R');

            }

        }

/////////////////////////////////////////////////////////////////////////////

*int* xx = x;

*int* yy = y;

        while (xx < 8 && yy < 8) {

            if (board[xx][yy] == 'Q') {

                output.push\_back('Q');

            }

            if (board[xx][yy] == 'B') {

                output.push\_back('B');

            }

            xx++; yy++;

        }

        xx = x;

        yy = y;

        while (xx >= 0 && yy >= 0) {

            if (board[xx][yy] == 'Q') {

                output.push\_back('Q');

            }

            if (board[xx][yy] == 'B') {

                output.push\_back('B');

            }

            xx--; yy--;

        }

        xx = x;

        yy = y;

        while (xx < 8 && yy >= 0) {

            if (board[xx][yy] == 'Q') {

                output.push\_back('Q');

            }

            if (board[xx][yy] == 'B') {

                output.push\_back('B');

            }

            xx++; yy--;

        }

        xx = x;

        yy = y;

        while (xx >= 0 && yy < 8) {

            if (board[xx][yy] == 'Q') {

                output.push\_back('Q');

            }

            if (board[xx][yy] == 'B') {

                output.push\_back('B');

            }

            xx--; yy++;

        }

/////////////////////////////////////////////////////////////////////////////

*int* NmoveX[8] = {1, 1, -1, -1, 2, 2, -2, -2};

*int* NmoveY[8] = {2, -2, 2, -2, 1, -1, 1, -1};

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

*int* moveX = x + NmoveX[j];

*int* moveY = y + NmoveY[j];

            if (moveX < 8 && moveX >= 0 && moveY < 8 && moveY >= 0) {

                if (board[moveX][moveY] == 'N') {

                    output.push\_back('N');

                }

            }

        }

/////////////////////////////////////////////////////////////////////////////

*int* KmoveX[8] = {1, 1, 1, -1, -1, -1, 0, 0};

*int* KmoveY[8] = {1, 0, -1, 1, 0, -1, 1, -1};

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

*int* moveX = x + KmoveX[j];

*int* moveY = y + KmoveY[j];

            if (moveX < 8 && moveX >= 0 && moveY < 8 && moveY >= 0) {

                if (board[moveX][moveY] == 'K') {

                    output.push\_back('K');

                }

            }

        }

/////////////////////////////////////////////////////////////////////////////

    }

        if (output == "") {

            output.push\_back('O');

        }

/////////////////////////////////////////////////////////////////////////////

        foutput += sort\_destroy(output) + "\n";

        output = "";

    }

    cout << foutput;

    return 0;

}

Код завдання №6

Програма №7 Practice# programming: Class Practice Task 1

#include <iostream>

#include <fstream>

using *namespace* std;

///////////////////////////////////////////////////////////

*void* data\_line(string *path*, string *info*) {

    ofstream output\_info;

    output\_info.open(*path*, ios::app);

    output\_info << *info*;

    output\_info.close();

}

///////////////////////////////////////////////////////////

string read\_file(string *path*) {

    string info;

    string line;

    ifstream read\_file;

    read\_file.open(*path*, ios::app);

    while (getline(read\_file, line)) {

        info += line;

    }

    return info;

}

///////////////////////////////////////////////////////////

*int* main() {

    const string file\_path = "C:/REPOSITORY/ai\_programming\_playground/ai\_11/yuliia\_pedchenko/textfiles/myfile.txt";

    string input;

    cin >> input;

    data\_line(file\_path, input);

    if (input == read\_file(file\_path)) cout << "Success";

    else cout << "Failure";

    return 0;

}

Код завдання №7

Програма №8 Practice# programming: Class Practice Task 2

#include <iostream>

#include <fstream>

using *namespace* std;

////////////////////////////////////////////////////////////

string read\_file(string *path*) {

    string info;

    string line;

    ifstream read\_file;

    read\_file.open(*path*, ios::app);

    while (getline(read\_file, line)) {

        info += line;

    }

    return info;

}

////////////////////////////////////////////////////////////

*void* data\_line(string *path*, string *info*) {

    ofstream output\_info;

    output\_info.open(*path*, ios::app);

    output\_info << *info*;

    output\_info.close();

}

////////////////////////////////////////////////////////////

*int* main() {

    const string fromfile\_path = "C:/REPOSITORY/ai\_programming\_playground/ai\_11/yuliia\_pedchenko/textfiles/file\_from.txt";

    const string tofile\_path = "C:/REPOSITORY/ai\_programming\_playground/ai\_11/yuliia\_pedchenko/textfiles/file\_to.txt";

     ifstream check\_file1;

    check\_file1.open(fromfile\_path);

     if (!check\_file1) {

        cout << "Failure" << endl;

        return 0;

    }

    check\_file1.close();

    ifstream check\_file2;

    check\_file2.open(tofile\_path);

     if (!check\_file2) {

        cout << "Failure" << endl;

        return 0;

    }

    check\_file2.close();

    data\_line(tofile\_path, read\_file(fromfile\_path));

    if (read\_file(fromfile\_path) == read\_file(tofile\_path)) cout << "Success";

    else cout << "Failure";

    return 0;

}

Код завдання №8

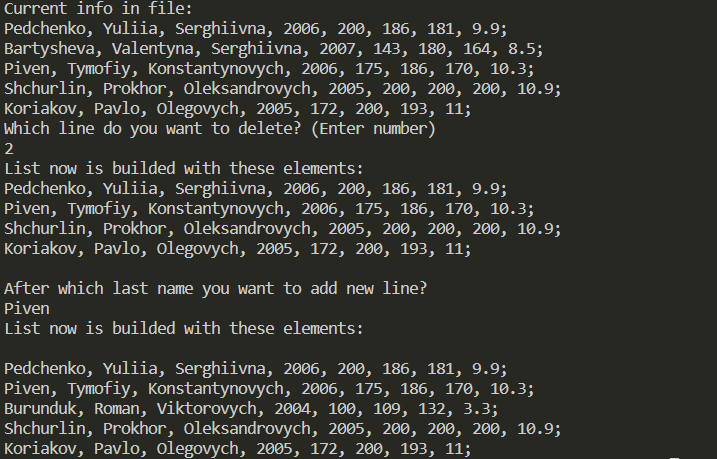
## **5. Результати виконання завдань, тестування та фактично затрачений час:**

Програма №1 Lab# programming: VNS Lab 6



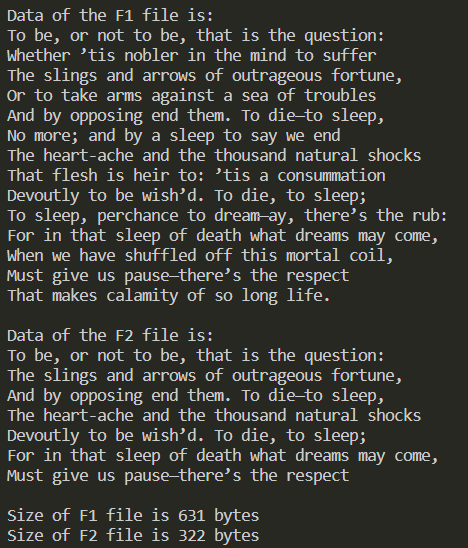
Час затрачений на виконання завдання: 40 хвилин.

Програма №2 Lab# programming: VNS Lab 8



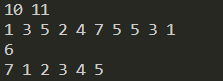
Час затрачений на виконання завдання: 3 дні

Програма №3 Lab# programming: VNS Lab 9



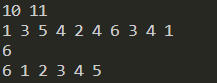
Час затрачений на виконання завдання: 50 хвилин

Програма №4 Lab# programming: Algotester Lab 4.1



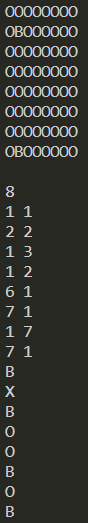
Час затрачений на виконання завдання: 10 хвилин

Програма №5 Lab# programming: Algotester Lab 4.2



Час затрачений на виконання завдання: 30 хвилин

Програма №6 Lab# programming: Algotester Lab 6



Час затрачений на виконання завдання: 1,5 днів

Програма №7 Practice# programming: Class Practice Task 1



Час затрачений на виконання завдання: 15 хвилин

Програма №8 Practice# programming: Class Practice Task 2



Час затрачений на виконання завдання: 15 хвилин

# **Висновки:**

Я навчилася працювати з рядками та файлами у мові С++ та успішно використала ці знання у коді.