Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 5**

***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-13

Гошовська Діана Олегівна

# **Тема роботи:**

Ознайомлення з теорією про файли та їх типи. Дослідження символьних рядків та їх змінні. Робота із файлами.

# **Мета роботи:**

Ознайомитися з навчальною теорією про файли та їх типи, роботу із символьними рядками та принцип їхньої роботи в коді, використавши для цього доступні джерела інформації; закріпити отриманні знання на практичних заняттях; оформити звіт до лабораторної роботи №5.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Ознайомлення з теорією про файли та їх типи.
* Тема №2: Дослідження символьних рядків та їх змінні.
* Тема №3: Робота із файлами та бібліотеками.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Ознайомлення з теорією про файли та їх типи.
  + Джерела Інформації
    - Документ: <https://vns.lpnu.ua/pluginfile.php?file=%2F1116827%2Fmod_resource%2Fcontent%2F1%2Fkonspekt_lekcyi_ukr_1.pdf>
    - Відео:  
      <https://youtu.be/SSNJ7alki-E?si=06q4tyWrHd21cQsd>
  + Що опрацьовано:
    - Опрацьовано джерела, що містять інформацію про створення фалів та методи їх використання у коді.
    - Закріплено отриману інформацію, використавши її для виконання завдань.
* Статус: Ознайомлена
  + Початок опрацювання теми: 03.12.2023
  + Звершення опрацювання теми: 07.12.2023
* Тема №2: Дослідження символьних рядків та їх змінні.
  + Джерела Інформації:
    - Документ:  
      <https://vns.lpnu.ua/pluginfile.php?file=%2F1116827%2Fmod_resource%2Fcontent%2F1%2Fkonspekt_lekcyi_ukr_1.pdf>
  + Що опрацьовано:
    - Опрацьовано інформацію про роботу символьних рядків та їх імплементацію в програмі.
    - Закріплено отриману інформацію, використавши її для виконання завдань.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 28.11.2023
  + Звершення опрацювання теми: 04.12.2023
* Тема №3: Робота із файлами та бібліотеками.
  + Джерела Інформації:
    - Документ - <https://vns.lpnu.ua/pluginfile.php?file=%2F1116827%2Fmod_resource%2Fcontent%2F1%2Fkonspekt_lekcyi_ukr_1.pdf>
    - Стаття:  
      <https://www.w3schools.com/cpp/cpp_files.asp>
  + Що опрацьовано:
    - Опрацьовано інформацію про принцип роботи із бібліотеками під час реалізації коду з файлами.
    - Закріплено отриману інформацію під час написання кодів.
  + Статус: Ознайомлена
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 08.12.2023

# **Виконання роботи:**

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

Завдання №3: написання коду до VNS Lab 6

* Варіант завдання: 9
* Деталі завдання:   
  Надрукувати всі слова-паліндроми, які є в цьому рядку.
* Важливі деталі для врахування в імплементації програми: зрозуміти принцип роботи із створенням функцій та пояснити отримані результати.

Завдання №4: написання коду до VNS Lab 8

* Варіант завдання: 9
* Деталі завдання:

Структура "Пацієнт":

- прізвище, ім'я, по батькові;

- домашня адреса;

- номер медичної карти;

- номер страхового поліса.  
Знищити елемент із заданим номером медичної карти, додати 2 елементи в

початок файлу.

* Важливі деталі для врахування в імплементації програми: виконати завдання, використати знання про структури, роботу з файлами та пояснити отримані результати.

Завдання №5: написання коду до VNS Lab 9

* Варіант завдання: 9
  + Деталі завдання:   
    1) Скопіювати з файлу F1 у файл F2 всі рядки, які містять тільки одне слово.

2) Знайти найдовше слово у файлі F2.

* Важливі деталі для врахування в імплементації програми: дотримуватися умов завдання, пояснити отримані результати.

Завдання №6: написання коду до Algotester Lab 4v2

* Варіант завдання: 2
* Деталі завдання:

Вам дано масив a з N цілих чисел.

Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4].

Після цього оберніть посортовану версію масиву a на K, тобто при K=3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3].

Виведіть результат.

* Важливі деталі для врахування в імплементації програми: дотримуватися умов завдання.

Завдання №7: написання коду до Algotester Lab 6v2

* Деталі завдання: <https://algotester.com/uk/ContestProblem/DisplayWithEditor/134743>
* Важливі деталі для врахування в імплементації програми: дотримуватися умов завдання.

Завдання №8: написання коду Class Practice Task

* Деталі завдання: написати код, використовуючи закріплені раніше знання про створення функцій, файлів та роботу з ними.
* Важливі деталі для врахування в імплементації програми: використати здобуті знання на практиці.

Завдання №9: написання коду Self-Practice Task

* Деталі завдання: написати код, використовуючи закріплені раніше знання про функції, масиви та вказівники.
* Важливі деталі для врахування в імплементації програми: використати здобуті знання на практиці.

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

Програма №3: VNS Lab 6

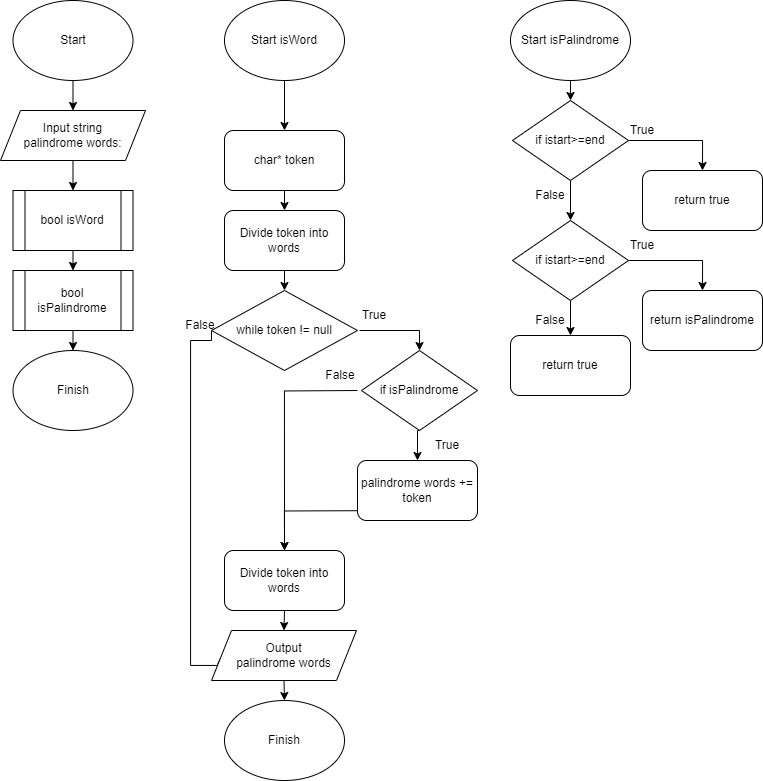
* Блок-схема
* 

Figure Flowchart

* Планований час на реалізацію: 2 години.
* Важливі деталі для врахування в імплементації: використати функції для виконання завдання.

Програма №4: VNS Lab 8

* Планований час на реалізацію: 3 години.
* Важливі деталі для врахування в імплементації: використати знання про файли для роботи зі структурами та пояснити отримані результати.

Програма №5: VNS Lab 9

* Планований час на реалізацію: 3 години.
* Важливі деталі для врахування в імплементації: використати знання про файли, створити відповідні функції та дотримуватися умов завдання.

Програма №6: Algotester Lab 4v2

* Планований час на реалізацію: 2 години.
* Важливі деталі для врахування в імплементації: дотримуватися умов завдання.

Програма №7: Algotester Lab 6v2

* Планований час на реалізацію: 6 годин.
* Важливі деталі для врахування в імплементації: дотримуватися умов завдання.

Програма №8&9: Class Practice Task

* Планований час на реалізацію: 2 години.
* Важливі деталі для врахування в імплементації: використати здобуті знання на практиці.

Програма №10: Self-Practice Task

* Планований час на реалізацію: 2 години.
* Важливі деталі для врахування в імплементації: використати здобуті знання на практиці.

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

Програма №3: VNS Lab 6

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1011/commits/ba6f4a5a525e77ed0da922009e7483f99139590a#diff-1d0eb87bcfbee5aa553cedfa95f7c109df0719c6fc2935d5fdd763fe075975bc

#include <iostream>

#include <cstring>

using namespace std;

bool isPalindrome(char token[], int start, int end);

void isWord(char wrd[], string& palindrome\_words);

int main(){

    string palindrome\_words;

    char str[256];

    cout << "Input text: ";

    cin.getline(str, 256);

    isWord(str, palindrome\_words);

    return 0;

}

bool isPalindrome(char token[], int start, int end){

    if (start>=end){

        return true;

    }

    if (token[start]==token[end]){

        return isPalindrome(token, start+1, end-1);

    }

    return false;

}

void isWord(char wrd[], string& palindrome\_words){

    char \*token;

    token = strtok(wrd, " ");

    while (token != NULL){

        if(isPalindrome(token, 0, strlen(token)-1)==true) {

            palindrome\_words+=token;

            palindrome\_words+=" ";

        }

        token = strtok(NULL, " ");

    }

    cout << "Words with palindrome: " << palindrome\_words;

}

Ввід програми VNS Lab 6

Програма №4: VNS Lab 8

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1011/commits/ba6f4a5a525e77ed0da922009e7483f99139590a#diff-46d2bd9468990b04613a9de5314f67aea23211ccd87b4545dff800af9d17cf1e

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

struct PATIENT{ //фактично працює як змінна з підзміннами

    char firstName[100];

    char surName[100];

    char patronymic[100];

    char Address[100];

    int MedCard;

    int StrPolicy;

};

void addPatient(const char\* filename, int number){

    ofstream file(filename, ios::binary | ios::app);

    if (!file.is\_open()){

    cerr << "An error has occured, please try again." << endl; exit(1);}

    for (int i = 0; i < number; i++){

        PATIENT newPatient;

        cout << "\nEnter the details of the " << i+1 << " patient: " << endl;

        cout << "Name: "; cin >> newPatient.firstName;

        cout << "Surname: "; cin >> newPatient.surName;

        cout << "Patronymic: "; cin >> newPatient.patronymic;

        cout << "Address: "; cin >> newPatient.Address;

        cout << "MedCard: "; cin >> newPatient.MedCard;

        cout << "StrPolis: "; cin >> newPatient.StrPolicy;

        file.write((char \*)&newPatient, sizeof(PATIENT));

    }

    file.close();

}

void deleteMedCard(const char\* filename, int num){

    ifstream file(filename, ios::binary);

    ofstream temp("temp.dat", ios::binary);

    if (!file.is\_open()){

    cerr << "An error has occured, please try again." << endl; exit(1);}

    PATIENT newPatient;

    for(int i = 0; file.read((char \*)&newPatient, sizeof(PATIENT)); i++){

        if(newPatient.MedCard != num){

            temp.write((char \*)&newPatient, sizeof(PATIENT));

        }

        else{

            cout << "A patient with a medical card " << newPatient.MedCard << " - is deleting...\n\n";

        }

    }

    file.close();

    temp.close();

    remove(filename);

    rename("temp.dat", filename);

}

void patientNoMedCard(const char\* filename, int num){

    ifstream file(filename, ios::binary);

    if (!file.is\_open()){

    cerr << "An error has occured, please try again." << endl; exit(1);}

    cout << "Patients without medical card: " << endl;

    PATIENT newPatient;

    for(int i = 0; file.read((char \*)&newPatient, sizeof(PATIENT)); i++){

        if(newPatient.MedCard != num){

            cout << "Name: " << newPatient.firstName << endl;

            cout << "Surname: " << newPatient.surName << endl;

            cout << "Patronymic: " << newPatient.patronymic << endl;

            cout << "Address: " << newPatient.Address << endl;

            cout << "MedCard: " << newPatient.MedCard << endl;

            cout << "StrPolis: " << newPatient.StrPolicy << endl << endl;

        }

    }

    file.close();

}

int main(){

    int number, num;

    const char\* filename = "patient.dat";

    cout << "Enter a number of patients: ";

    cin >> number;

    cout << "Enter a number of medical card to delete: ";

    cin >> num;

    addPatient(filename, number);

    deleteMedCard(filename, num);

    patientNoMedCard(filename, num);

    return 0;

}

Ввід програми VNS Lab 8

Програма №5: VNS Lab 9

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1011/commits/ba6f4a5a525e77ed0da922009e7483f99139590a#diff-42d012f8545171620fd9fc0d617d9939b989085273ae44cbbc187773cdffd786

#include <iostream>

#include <fstream>

#include <sstream>

#include <vector>

using namespace std;

void copyLinesWithSingleWord(const string& inputFileName, const string& outputFileName) {

    ifstream inputFile(inputFileName);

    ofstream outputFile(outputFileName);

    if (!inputFile.is\_open()) {

        cerr << "Failed to open file1." << endl;

        return;

    }

    if (!outputFile.is\_open()) {

        cerr << "Failed to open file2." << endl;

        return;

    }

    string line;

    while (getline(inputFile, line)) { //програма зчитує файл (перший рядок в inputFile зберігає в line), допоки не дійде до eof

        istringstream iss(line);

        string word;

        if (!(iss >> word >> ws).eof()) { //якщо рядок з пропусками

            continue; //в рядку більше одного слова, пропустити його

        }

        outputFile << line << endl; //рядок містить тільки одне слово, скопіювати його в інший файл

    }

    cout << "Lines with one word are copied into " << outputFileName << endl;

}

string findLongestWord(const string& fileName) {

    ifstream file(fileName);

    if (!file.is\_open()) {

        cerr << "Failed to open file" << endl;

        return "";

    }

    string longestWord;

    string word;

    while (file >> word) {

        if (word.length() > longestWord.length()) {

            longestWord = word;

        }

    }

    return longestWord;

}

int main() {

    ofstream f1("F1.txt");

    if (f1.is\_open()) {

        f1 << "Word\n";

        f1 << "Another word\n";

        f1 << "Many Words\n";

        f1 << "One\n";

        f1 << "BigWord\n";

        f1 << "So many\n";

        f1 << "Words\n";

        f1 << "ThisIsTheBiggestWord\n";

        f1 << "Wow\n";

        f1 << "Great\n";

        f1.close();

    } else {

        cerr << "Failed to open or create F1." << endl;

        return 1;

    }

    copyLinesWithSingleWord("F1.txt", "F2.txt");

    string longestWord = findLongestWord("F2.txt");

    if (!longestWord.empty()) {

        cout << "The longest word in F2: " << longestWord << endl;

    } else {

        cout << "F2 does not contain long words." << endl;

    }

    return 0;

}

Ввід програми VNS Lab 9

Програма №6: Algotester Lab 4v2

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1011/commits/ba6f4a5a525e77ed0da922009e7483f99139590a#diff-708e351ce84bfc922d26a671042f710d2d1ae13ba37fa56b4f5744749b9ece85

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

void del(vector <int>& a, int& n){

    int j = 0;

    vector <int> b;

    for(int i = 0; i < n; i++){

        if(a[i] != a[i+1]){

            b.push\_back(a[i]);

            j++;

        }

    }

    n=j;

    a.clear();

    for(int i=0; i<n; i++){

        a.push\_back(b[i]);

    }

}

int main(){

    int n, k;

    cin >> n >> k;

    vector <int> arr(n);

    for(int i = 0; i < n; i++){

        cin >> arr[i];

    }

    sort(arr.begin(), arr.end());

    del(arr, n);

    cout << n << endl;

    if(k>arr.size()){

        k=k % arr.size();

    }

    rotate(arr.begin(), arr.begin()+k, arr.end());

    for(int i = 0; i < n; i++){

        cout << arr[i] << " ";

    }

    return 0;

}

Ввід програми Algotester Lab 4v2

Програма №7: Algotester Lab 6v2

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1011/commits/ba6f4a5a525e77ed0da922009e7483f99139590a#diff-976220b558d3753f244c6ecc44721e74b7f01efe44ebd7fbe2cca79d08bfd83f

#include <iostream>

#include <algorithm>

#include <string>

#include <vector>

using namespace std;

int main(){

    string row[18]; //дошка

    int Qn;

    for (int i = 0; i < 8; i++){

        cin >> row[i]; //стан рядків

    }

    cin >> Qn;

    for (int i = 0; i < Qn; i++){

        string result;

        int x=0, y=0, n = 0;

        cin >> x >> y;

        x-=1; y-=1;

        result.resize(18);

        if (row[x][y] != 'O'){

            cout << 'X';

        }

        else{

            bool

            P = false,

            R = false,

            N = false,

            B = false,

            K = false,

            Q = false;

            //pawn

            if((((x-1>=0 && y-1>=0) && x > 0 && y > 0) && row[x-1][y-1] == 'P')

            || ((x-1>=0 && y+1 < 8) && x > 0 && (row[x-1][y+1]) == 'P')){

                P = true;

            }

            //rook

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

                if (row[x][j] == 'R' || row[j][y] == 'R' ){

                    R = true;

                }

            }

            //horse

            if ((x+2 < 8 && y-1>=0) && y > 0 && (row[x+2][y-1] == 'N')

            || ((x+1 < 8 && y-2>=0) && y > 1 && row[x+1][y-2] == 'N')

            || ((x-2 >= 0 && y+1 < 8) && x > 1 && row[x-2][y+1] == 'N')

            || ((x-1 >=0 && y+2<8) && x > 0 && row[x-1][y+2] == 'N')

            || ((x+2 < 8 && y+1 < 8) && row[x+2][y+1] == 'N')

            || ((x+1 < 8 && y+2 < 8) && row[x+1][y+2] == 'N')

            || ((x-2 >=0 && y-1>=0) && (x > 1 && y > 0) && row[x-2][y-1] == 'N')

            || ((x-1 >=0 && y-2>=0) && (x > 0 && y > 1) && row[x-1][y-2] == 'N') ){

                N = true;

            }

            //officer

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

                if ((x+j<8 && y+j<8 && row[x + j][y + j] == 'B') ||

                (x-j>=0 && y+j<8 && row[x - j][y + j] == 'B') ||

                (x+j<8 && y-j>=0 && row[x + j][y - j] == 'B') ||

                (x-j>=0 && y-j>=0 && row[x - j][y - j] == 'B')) {

                    B = true;

                }

            }

            //king

            if ((y+1 < 8) && (row[x][y+1] == 'K')

            || ((x+1 < 8) && row[x+1][y] == 'K')

            || ((x+1 < 8 && y+1 < 8) && row[x+1][y+1] == 'K')

            || ((x-1 >=0 && y+1 < 8) && (x>0) && row[x-1][y+1] == 'K')

            || ((x-1 >=0) && (x>0) && row[x-1][y] == 'K')

            || ((x+1 < 8 && y-1 >= 0) && (y>0) && row[x+1][y-1] == 'K')

            || ((y-1 >= 0) && (y>0) && row[x][y-1] == 'K')

            || ((x-1 >=0 && y-1 >=0) && (x>0 && y>0) && row[x-1][y-1] == 'K') ){

                K = true;

            }

            // queen

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

            if (row[x][j] == 'Q' ||

            row[j][y] == 'Q' ||

            (x + j < 8 && y + j < 8 && row[x + j][y + j] == 'Q') ||

            (x - j >= 0 && y + j < 8 && row[x - j][y + j] == 'Q') ||

            (x + j < 8 && y - j >= 0 && row[x + j][y - j] == 'Q') ||

            (x - j >= 0 && y - j >= 0 && row[x - j][y - j] == 'Q')) {

            Q = true;

        }

        }

            if(P){

                result[n] = 'P';

                n++;

            }

            if(R){

                result[n] = 'R';

                n++;

            }

            if(N){

                result[n] = 'N';

                n++;

            }

            if(B){

                result[n] = 'B';

                n++;

            }

            if(K){

                result[n] = 'K';

                n++;

            }

            if(Q){

                result[n] = 'Q';

                n++;

            }

        }

        if(n == 0 && row[x][y] == 'O'){

            cout << 'O';

        }

        else{

            result.resize(n);

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

            cout << result;

        }

        cout << endl;

    }

}

Ввід програми Algotester Lab 6v2

Програма №8: Class Practice Task 1

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1011/commits/ba6f4a5a525e77ed0da922009e7483f99139590a#diff-257a24d2884bbe4d5b287574cdfca697ca544af1347798057843a2fb00ecf1c7

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

enum FileOpResult { Success, Failure }; //розібрати детальніше про енам

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

    ofstream file(Name, ios::binary); //відкриваємо для запису у файл

    if (!file.is\_open()){ //файл не відкрився

        cerr << "An error has occured, please try again." << endl;

        return Failure;}

    file << content;

    if(file.fail()){ //файл не записався

        cerr << "An error has occured, please try again." << endl;

        file.close();

        return Failure;

    }

    file.close();

    return Success;

}

int main(){

    string file = "my\_file", content;

    cout << "Enter data to write in file: ";

    getline(cin, content);

    FileOpResult res = write\_to\_file(file.c\_str(), content.c\_str()); //розвідати, шо таке c\_str

    if (res == Success){

        cout << "\nAn operation is proceed successfully";

    }

    else{

        cout << "\nSomething went wrong, please try again.";

    }

    return 0;

}

Ввід програми Class Practice Task 1

Програма №9: Class Practice Task 2

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1011/commits/ba6f4a5a525e77ed0da922009e7483f99139590a#diff-f878230721acb34f9ee59479168ebf705f711c659ee1a955afe4a2ff90310fe2

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

enum FileOpResult { Success, Failure };

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

    string text = "This is an example text.";

    ofstream file1(file\_from, ios::binary);

    if (!file1.is\_open()){

        cerr << "An error has occurred, please try again." << endl;

        return Failure;

    }

    file1 << text;

    if(file1.fail()){

        cerr << "An error while creating a text for file1 has occurred, please try again." << endl;

        file1.close();

        return Failure;

    }

    file1.close();

    ifstream input\_file(file\_from, ios::binary);

    if (!input\_file.is\_open()){

        cerr << "An error has occurred, please try again." << endl;

        return Failure;

    }

    ofstream file2(file\_to, ios::binary);

    if (!file2.is\_open()){

        cerr << "An error has occurred, please try again." << endl;

        return Failure;

    }

    file2 << input\_file.rdbuf();

    if (file2.fail()) {

        cerr << "An error while writing has occurred, please try again." << endl;

        return Failure;

    }

    input\_file.close();

    file2.close();

    return Success;

}

int main(){

    const char\* file\_from = "from";

    const char\* file\_to = "to";

    FileOpResult res = copy\_file(file\_from, file\_to);

    if (res == Success){

        cout << "Operation proceeded successfully!";

    }

    else{

        cout << "Something went wrong, please try again.";

    }

    return 0;

}

Ввід програми Class Practice Task 2

Завдання №10: Self-Practice Task

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1011/commits/ba6f4a5a525e77ed0da922009e7483f99139590a#diff-56a70f54c3ed149b82e26fe853e6f471173514b65e501f813c1f06e68314b367

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

void ccout(vector <int> &a){

    cout << a.size() << endl;

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

        cout << a[i] << " ";

    }

    cout << endl << endl;

}

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];

    }

    cout << endl;

    sort(arr1.begin(), arr1.end());

    sort(arr2.begin(), arr2.end());

    //різниця N-M

    vector <int> a;

    set\_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back\_inserter(a));

    ccout(a);

    //різниця M-N

    a.clear();

    set\_difference(arr2.begin(), arr2.end(), arr1.begin(), arr1.end(), back\_inserter(a));

    ccout(a);

    //їх перетин

    a.clear();

    set\_intersection(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back\_inserter(a));

    ccout(a);

    //їх об'єднання

    a.clear();

    set\_union(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back\_inserter(a));

    ccout(a);

    //їх симетричну різницю

    a.clear();

    set\_symmetric\_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back\_inserter(a));

    ccout(a);

    return 0;

}

Ввід програми Self-Practice Task

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

Завдання №3: VNS Lab 6

Input text: hfkedg lkkl tooot njrkdsf eljle

Words with palindrome: lkkl tooot eljle

Вивід програми VNS Lab 6

Час затрачений на виконання завдання: 2 години.

Завдання №4: VNS Lab 8

Enter a number of patients: 2

Enter a number of medical card to delete: 14

Enter the details of the 1 patient:

Name: Vasya

Surname: Bilyy

Patronymic: Petrovych

Address: Narodna28

MedCard: 12

StrPolis: 18

Enter the details of the 2 patient:

Name: Ahmed

Surname: Ezz

Patronymic: Ali

Address: Konovaltsya32

MedCard: 14

StrPolis: 18

A patient with a medical card 14 - is deleting...

Patients without medical card:

Name: Vasya

Surname: Bilyy

Patronymic: Petrovych

Address: Narodna28

MedCard: 12

StrPolis: 18

Вивід програми VNS Lab 8

Час затрачений на виконання завдання: 4 години.

Завдання №5: VNS Lab 9

Lines with one word are copied into F2.txt

The longest word in F2: ThisIsTheBiggestWord

Вивід програми VNS Lab 9

Час затрачений на виконання завдання: 3 години.

Завдання №6: Algotester Lab 4v2

10 3

1 2 2 3 3 3 4 5 6 7

7

4 5 6 7 1 2 3

Вивід програми Algotester Lab 4v2

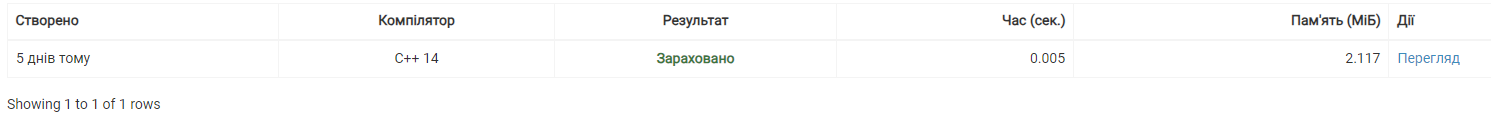


Figure : Результат тестів з Algotester Lab 4v2

Час затрачений на виконання завдання: 1,5 години.

Завдання №7: Algotester Lab 6v2

KOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

OOOOOOOO

5

1 1

X

1 2

K

2 1

K

2 2

K

3 1

O

Вивід програми Algotester Lab 6v2

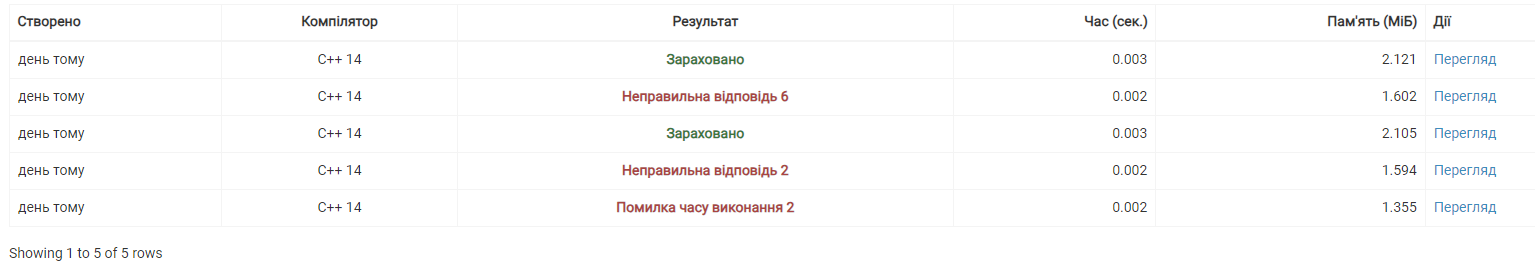


Figure : Результат тестів з Algotester Lab 3v3

Час затрачений на виконання завдання: 6 годин.

Завдання №8: Class Practice Task 1

Enter data to write in file: let

An operation is proceed successfully!

Вивід програми Class Practice Task 1

Час затрачений на виконання завдання: 2 години.

Завдання №9: Class Practice Task 2

Operation proceeded successfully!

Вивід програми Class Practice Task 2

Час затрачений на виконання завдання: 2 години.

Завдання №10: Self-Practice Task (Algotester 4v1)

5

3 2 5 4 1

5

8 5 7 4 6

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

Вивід програми Self-Practice Task (Algotester 4v1)

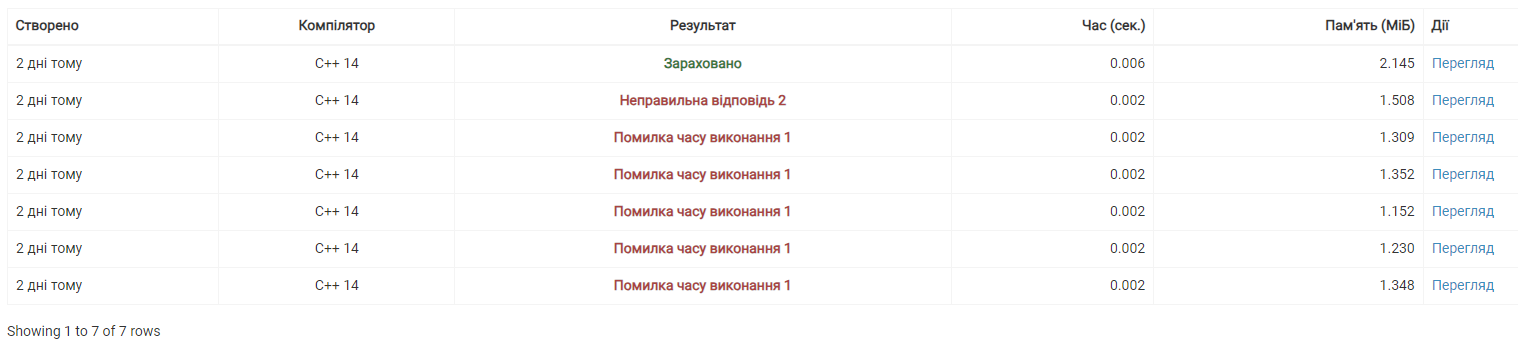


Figure : Результат тестів з Algotester 4v1

Час затрачений на виконання завдання: 3 години.

# **Висновки:**

В ході виконання лабораторної роботи №5 я ознайомилася з навчальною теорією про файли та їх різновид, а також дослідила властивості символьних рядків; це дало мені можливість покращити знання програмування на мові C++. Виконала три лабораторних завдання із обчисленням значень виразів із застосуванням знань про файли та символьні рядки. Написала три коди у середовищі Algotester (один з них є Self-Practice), а також виконала практичне завдання, використовуючи у коді базові операції з файлами. Оформила звіт за зразком. Ця лабораторна робота розвинула мої навички розуміння роботи комп’ютера з кодом, принцип побудови блок-схем та написання програм на мові C++.