Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

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

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

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

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт №5

***Виконав:***

студент групи ШІ-12

Яремчук Павло Марекович

# **Тема роботи:**

Виконання лабораторних робіт по темі: «Файли. Системи числення. Бінарні файли. Символи та Рядкові змінні. Текстові файли. Стандартна бібліотека. Створення та використання бібліотек»

# **Мета роботи:**

* Lab# programming: VNS Lab 6
* Lab# programming: VNS Lab 8
* Lab# programming: VNS Lab 9
* programming: Algotester Lab 4
* Lab# programming: Algotester Lab 6
* Practice# programming: Class Practice Task

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: VNS Lab 6
* Тема №2: VNS Lab 8
* Тема №3: VNS Lab 9
* Тема №4: Algotester Lab 2
* Тема №5: Algotester Lab 3
* Тема №6: Class Practice

1. Індивідуальний план опрацювання теорії:

* Тема №1: VNS Lab 6.
  + Джерела Інформації
    - Лекції
    - Практичні
  + Що опрацьовано:
    - Символьні і рядкові змінні і способи їхньої обробки.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 12.12.2023
  + Завершення опрацювання теми: 12.12.2023
* Тема №2: VNS Lab 8.
* Джерела Інформації:
  + - Лекції
    - Практичні
  + Що опрацьовано:
    - Робота із двійковими файлами, організація вводу-виводу структурованої інформації і її зберігання на зовнішніх носіях.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 13.11.2023
  + Завершення опрацювання теми: 13.11.2023
* Тема №3: VNS Lab 9.
  + Джерела Інформації:
    - Лекції
    - Практичні
  + Що опрацьовано:
    - Робота з текстовими файлами, ввід-вивід текстової інформації і її зберігання на зовнішніх носіях.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.11.2023
  + Завершення опрацювання теми: 14.12.2023
* Тема №4 Algotester Lab 4.
  + Джерела Інформації:
    - Лекції
    - Практичні
  + Що опрацьовано:
    - використання засобів STL
  + Статус: Ознайомлений
  + Початок опрацювання теми: 14.12.2023
  + Завершення опрацювання теми: 14.12.2023
* Тема №5 Algotester Lab 6.
  + Джерела Інформації:
    - Лекції
    - Практичні
  + Що опрацьовано:
    - Робота з двовимірними масивами
  + Статус: Ознайомлений
  + Початок опрацювання теми: 12.12.2023
  + Завершення опрацювання теми: .12.2023
* Тема №6 Class Practice.
  + Джерела Інформації:
    - Лекції
    - Практичні
  + Що опрацьовано:
    - Робота з текстовими файлами
  + Статус: Ознайомлений
  + Початок опрацювання теми: 15.12.2023
  + Завершення опрацювання теми: 15.12.2023

# **Виконання роботи:**

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

Завдання № 1 VNS Lab 6 Task 1

* Варіант завдання: 5
* Деталі завдання: 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 таким чином, щоб спочатку в ньому були надруковані тільки букви, а потім тільки цифри, не міняючи порядку проходження символів у рядку.

Завдання № 2 VNS Lab 8 Task 1

* Варіант завдання: 5
* Деталі завдання: Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вмістиме, виконати знищення й додавання елементів у відповідності зі своїм варіантом, використовуючи для пошуку елементів що знищуються чи додаються, функцію. Формування, друк, додавання й знищення елементів оформити у вигляді функцій. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

Структура "Людина": - прізвище, ім'я, по батькові; - рік народження; - ріст; - вага. Знищити усі елементи із зазначеним ростом і вагою, додати елемент після елемента із зазначеним прізвищем.

Завдання № 3 VNS Lab 9 Task 1

* Варіант завдання: 5
* Деталі завдання: 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 K до K+5. 2) Підрахувати кількість голосних букв у файлі F2.

Завдання № 4 Algotester Lab 4

* Варіант завдання: 2
* Деталі завдання: Вам дано масив а з № цілих чисел. Спочатку видаліть масиву в усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4] Після цього оберніть посортовану версію масиву а на К. тобто при К - 3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться m [4,5,6,7,1,2]. Виведіть результат

Завдання № 5 Algotester Lab 6

* Варіант завдання: 2
* Деталі завдання: У вас є шахова дошка розміром 8 х 8 та дуже багато фігур.

Кожна клітинка може мати таке значення:

Пуста клітинка О

Пішак Р

• Typa R

• Кінь N

• Слон В

• Король К

• Королева Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1).

Далі йдуть запитів з координатами клітинки {х, у). На кожен запит ви маєте вивести стрічку в посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз).

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. Тобто якщо між турою та клітинкою стоїть інша фігура вважається що тура атакує цю клітинку.

Завдання № 6 Class Practice

* Деталі завдання: Реалізувати функцію створення файла і запису в нього даних.

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

Програма № 1 VNS Lab 6 Task 1

* Планований час на реалізацію: 60 хв

Програма № 2 VNS Lab 8 Task 1

* Планований час на реалізацію: 150 хв

Програма №\_3 VNS Lab 9 Task 1

* Планований час на реалізацію: 90 хв

Програма №\_4 Algotester Lab 4

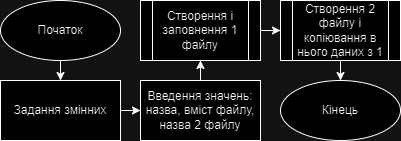
* Планований час на реалізацію: 60 хв

Програма №\_5 Algotester Lab 6

* Планований час на реалізацію: 180 хв

Програма №\_6 Class Practice

* Блок-схема:



* Планований час на реалізацію: 120 хв

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

Додаткова конфігурація не потрібна.

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

Завдання №1 Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

#include <iostream>

#include <cstring>

#include <cctype>

using namespace std;

int main()

{

const int max = 256;

char text[max];

char letters[max] = "";

char numbers[max] = "";

char editedText[max] = "";

//gets(text);

cin.getline(text, max);

for (int i = 0; i < strlen(text); ++i)

{

if(isalpha(text[i]) || text[i] == ' ' && isalpha(text[i - 1]))

{

char l[2] = {text[i]};

strcat(letters, l);

}

else

{

char n[2] = {text[i]};

strcat(numbers, n);

}

}

strcat(editedText, letters);

strcat(editedText, numbers);

puts(editedText);

return 0;

}

Підпис та № до блоку з кодом програми

Завдання №2

#include <iostream>

#include <fstream>

#include <stdio.h>

#include <stdlib.h>

#include <vector>

#include <algorithm>

using namespace std;

struct Person

{

char surname[20];

char name[20];

char fathername[20];

int birthYear;

int height;

int weight;

};

void addtoFile(const char\* filename, vector<Person>& people)

{

ofstream fileP(filename, ios::binary);

if (!fileP.is\_open())

{

cout << "Помилка запису в файл" << endl;

return;

}

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

{

fileP.write(reinterpret\_cast<const char\*>(&people[i]), sizeof(Person));

}

fileP.close();

}

void readFile(int countPeople, vector<Person>& person)

{

for (int i = 0; i < countPeople; ++i)

{

cout << person[i].surname << "\t\t";

}

cout << endl;

for (int i = 0; i < countPeople; ++i)

{

cout << person[i].name << "\t\t";

}

cout << endl;

for (int i = 0; i < countPeople; ++i)

{

cout << person[i].fathername << "\t\t";

}

cout << endl;

for (int i = 0; i < countPeople; ++i)

{

cout << person[i].birthYear << "\t\t";

}

cout << endl;

for (int i = 0; i < countPeople; ++i)

{

cout << person[i].height << "\t\t";

}

cout << endl;

for (int i = 0; i < countPeople; ++i)

{

cout << person[i].weight << "\t\t";

}

cout << endl;

}

void DeletebyHW(int& countPeople, vector<Person>& person, int banedH, int banedW, const char\* filename)

{

person.erase(remove\_if(person.begin(), person.end(),

[banedW, banedH](const Person& person)

{

return person.weight == banedW && person.height == banedH;

}),

person.end());

--countPeople;

addtoFile(filename, person);

}

void PersonToAddF(Person& PersonToAdd)

{

cin >> PersonToAdd.surname;

cin >> PersonToAdd.name;

cin >> PersonToAdd.fathername;

cin >> PersonToAdd.birthYear;

cin >> PersonToAdd.height;

cin >> PersonToAdd.weight;

return;

}

void AddAfterSurname(vector<Person>& person, const char\* surnameToAddAfter, int& countPeople, const char\* filename)

{

for (int i = 0; i < countPeople; ++i)

{

if (person[i].surname == surnameToAddAfter);

{

Person PersonToAdd;

PersonToAddF(PersonToAdd);

auto position = person.begin() + (i + 1);

person.insert(position, PersonToAdd);

++countPeople;

addtoFile(filename, person);

}

}

}

void AddPerson(vector<Person>& person, int countPeople)

{

for (int i = 0; i < countPeople; ++i)

{

cout << "Прізвище: ";

cin >> person[i].surname;

cout << "Ім'я: ";

cin >> person[i].name;

cout << "По батькові: ";

cin >> person[i].fathername;

cout << "Рік народження: ";

cin >> person[i].birthYear;

cout << "Ріст: ";

cin >> person[i].height;

cout << "Bara: ";

cin >> person[i].weight;

}

}

int main()

{

char filename[20];

cout << "Введіть назву файлу: ";

cin >> filename;

int countPeople;

cout << "Введіть кількість людей: ";

cin >> countPeople;

int banedH, banedW;

cout << "Введіть зріст та вагу для видалення: ";

cin >> banedH >> banedW;

vector<Person> person(countPeople);

addtoFile(filename, person);

AddPerson(person, countPeople);

addtoFile(filename, person);

readFile(countPeople, person);

DeletebyHW(countPeople, person, banedH, banedW, filename);

cout << "Після видалення:" << endl;

readFile(countPeople, person);

char surnameToAddAfter[20];

cout << "Введіть прізвище людини, після якої треба вставити нову: ";

cin >> surnameToAddAfter;

AddAfterSurname(person, surnameToAddAfter, countPeople, filename);

cout << endl;

readFile(countPeople, person);

return 0;

}

Завдання №3

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main()

{

ofstream file1("F1");

if (!file1.is\_open())

{

cout << "Помилка запису в файл" << endl;

return 0;

}

for (int i = 0; i < 10; ++i)

{

string textLine;

cout << "Введіть " << i + 1 << "-ий рядок: ";

cin >> textLine;

file1 << textLine << endl;

}

file1.close();

ifstream file1Read("F1");

ofstream file2("F2");

string lineToAdd;

int K;

cout << "Введіть з якого рядка почати копіювання: ";

cin >> K;

int N = K;

int M = 0;

while (getline(file1Read, lineToAdd))

{

++M;

if (M >= N)

{

if (K < N + 6)

{

file2 << lineToAdd << endl;

++K;

}

}

}

file1Read.close();

file2.close();

ifstream file2Read("F2");

string line3;

char symbol;

int count = 0;

while (getline(file2Read, line3))

{

for (int i = 0; i < line3.size(); ++i)

{

symbol = line3[i];

if (symbol == 'A' || symbol == 'a' || symbol == 'E' || symbol == 'e' || symbol == 'I' || symbol == 'i' || symbol == 'O' || symbol == 'o' || symbol == 'U' || symbol == 'u' || symbol == 'Y' || symbol == 'y')

{

++count;

}

}

}

file2Read.close();

cout << "Кількість голосних букв y файлі F2: " << count;

return 0;

}

Завдання №4

#include <iostream>

#include <algorithm>

#include <vector>

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];

}

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

auto last = unique(a.begin(), a.end());

a.erase(last, a.end());

N = a.size();

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

cout << N << endl;

for (int i = 0; i < N; ++i)

{

cout << a[i] << " ";

}

return 0;

}

Завдання №5

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int main()

{

vector<vector<char>> chessboard(8, vector<char>(8));

for (int i = 0; i < 8; ++i)

{

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

{

cin >> chessboard[i][j];

}

}

int Q, x, y;

cin >> Q;

vector<vector<char>> attackingF(Q, vector<char>());

vector<char> attRow;

vector<int> size;

for(int q = 0; q < Q; ++q)

{

cin >> x >> y;

x -= 1;

y -= 1;

attRow.clear();

for (int i = 0; i < 8; ++i)

{

char o = 'O';

if (chessboard[x][y] != o)

{

attRow.push\_back('X');

break;

}

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

{

char chessman = chessboard[i][j];

switch (chessman)

{

case 'P':

if (((x - i == 1) && (abs(y - j) == 1)))

{

attRow.push\_back(chessman);

}

break;

case 'R':

if ((i == x) || (j == y))

{

attRow.push\_back(chessman);

}

break;

case 'N':

if ((abs(x - i) == 2) && (abs(y - j) == 1) || (abs(x - i) == 1) && (abs(y - j) == 2))

{

attRow.push\_back(chessman);

}

break;

case 'B':

if ((abs(x - i) == abs(y - j)))

{

attRow.push\_back(chessman);

}

break;

case 'K':

if (((abs(x - i) == 1) && y == j) || ((abs(y - j) == 1) && x == i) || ((abs(x - i) == 1) && (abs(y - j) == 1)))

{

attRow.push\_back(chessman);

}

break;

case 'Q':

if ((i == x) || (j == y) || (abs(x - i) == abs(y - j)))

{

attRow.push\_back(chessman);

}

break;

}

}

}

if (attRow.empty())

{

attRow.push\_back('O');

}

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

auto last = unique(attRow.begin(), attRow.end());

attRow.erase(last, attRow.end());

size.push\_back(attRow.size());

for (int n = 0; n < attRow.size(); ++n)

{

attackingF[q].push\_back(attRow[n]);

}

}

for (int i = 0; i < Q; ++i)

{

for (int j = 0; j < size[i]; ++j)

{

cout << attackingF[i][j];

}

cout << endl;

}

return 0;

}

Завдання №6

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

enum FileOpResult {Success, Failure};

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

{

ofstream file(name);

if (!file.is\_open())

{

cout << "Помилка відкриття файлу" << endl;

return Failure;

}

file << content;

file.close();

cout << "Файл успішно створений i заповнений" << endl;

return Success;

}

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

{

ifstream file1(file\_from);

ofstream file2(file\_to);

if (!file1.is\_open())

{

cout << "Помилка відкриття файлу" << endl;

return Failure;

}

if (!file2.is\_open())

{

cout << "Помилка відкриття файлу" << endl;

return Failure;

}

string text;

while (getline(file1, text))

{

file2 << text;

}

file1.close();

file2.close();

cout << "Вміст файлу 1 успішно копійовано y файл 2" << endl;

return Failure;

}

int main()

{

string name, content, file\_to, file\_from;

cout << "Введіть назву файлу: ";

cin >> name;

cout << "Введіть вміст файлу: ";

cin >> content;

cout << "Введіть назву файлу, в який потрібно копіювати: ";

cin >> file\_to;

file\_from = name;

FileOpResult wrtfile = write\_to\_file(name.c\_str(), content.c\_str());

FileOpResult cpfile = copy\_file(file\_from.c\_str(), file\_to.c\_str());

return 0;

}

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

Завдання №1 Деталі по виконанню і тестуванню програми

aaa bbb 10 abab 1 0

aaa bbb abab 10 1 0

Час затрачений на виконання завдання: 60 хв

Завдання №2

Введіть назву файлу: people.bin

Введіть кількість людей: 2

Введіть зріст та вагу для видалення: 179 70

Прізвище: qqq

Ім'я: www

По батькові: eee

Рік народження: 2006

Ріст: 179

Bara: 70

Прізвище: rrr

Ім'я: ttt

По батькові: yyy

Рік народження: 2007

Ріст: 165

Bara: 50

qqq rrr

www ttt

eee yyy

2006 2007

179 165

70 50

Після видалення:

rrr

ttt

yyy

2007

165

50

Введіть прізвище людини, після якої треба вставити нову: rrr

aaa

sss

ddd

2008

171

64

rrr aaa

ttt sss

yyy ddd

2007 2008

165 171

50 64

Час затрачений на виконання завдання: 150 хв

Завдання №3

Введіть 1-ий рядок: qqq

Введіть 2-ий рядок: www

Введіть 3-ий рядок: eee

Введіть 4-ий рядок: rrr

Введіть 5-ий рядок: ttt

Введіть 6-ий рядок: yyy

Введіть 7-ий рядок: uuu

Введіть 8-ий рядок: iii

Введіть 9-ий рядок: ooo

Введіть 10-ий рядок: ppp

Введіть з якого рядка почати копіювання: 3

Кількість голосних букв y файлі F2: 12

Час затрачений на виконання завдання: 90 хв

Завдання №4

10 3

1 2 2 3 3 3 4 5 6 7

7

4 5 6 7 1 2 3

Час затрачений на виконання завдання: 60 хв

Завдання №5

OOOOOOOO

OROOOOOO

OONOOOOO

OOOOPOOO

OOOOOOOO

OOOOOOOO

KOQOOOOO

OOOOOOOR

7

8 1

1 2

5 4

5 1

6 2

8 4

6 7

KR

NR

NP

Q

KQR

QR

O

Час затрачений на виконання завдання: 180 хв

# **Висновки:**

Під час ознайомлення з цим розділом я вивчив, як працювати з символьними та рядковими змінними, а також освоїв основні алгоритми для обробки їхніх даних. Я засвоїв принципи роботи з текстовими та бінарними файлами, навчився організовувати введення та виведення структурованої та текстової інформації. Крім того, я успішно реалізував зберігання цієї інформації на зовнішніх носіях. Мною були створені декілька програм, де використані отримані знання та функціонал стандартної бібліотеки.