Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 5(Algolab 4. Algolab 6.VNS Lab 6.VNS Lab 8. VNS Lab 9. Class Practice Work. Self Practice Work)**

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

***з розділу***: «Епік 5: « Файли. Системи числення. Символи і Рядкові Змінні та Текстові Файли»

***Виконав:***

***Бачик Анатолій Андрійович*** групи ШІ-13

### Тема роботи:

Файли. Системи числення. Символи і Рядкові Змінні та Текстові Файли

### Мета роботи:

Ознайомлення з файлами, cистемами числення, символами і рядковими змінними та текстовими файлами

### Теоретичні відомості:

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Символи і Рядкові Змінні
* Тема №2: Текстові файли
* Тема №3 Робота з файлами

1. Індивідуальний план опрацювання теорії:

**Тема №1 Cимволи і Рядкові Змінні**

* + Джерела Інформації

<https://www.bestprog.net/en/2020/05/14/c-examples-of-working-with-text-files-modification-of-files-sort-data-in-files-convert-file-data-to-list/>

* + Що опрацьовано:

Матеріал з джерел інформації

* + Статус: Ознайомлений
  + Початок опрацювання теми: 29.11.2023
  + Звершення опрацювання теми: 29.11.2023

**Тема №2 Текстові файли**

* + Джерела Інформації:

<https://www.bestprog.net/uk/2019/09/11/examples-of-using-c-tools-for-working-with-files-ua/>

https://www.bestprog.net/uk/2020/05/14/c-examples-of-working-with-text-files-modification-of-files-sort-data-in-files-convert-file-data-to-list-ua/

* + Що опрацьовано:

Матеріал з джерел інформації

* + Статус: Ознайомлений
  + Початок опрацювання теми: 29.11.2023
  + Звершення опрацювання теми: 29.11.2023

**Тема №3 Робота з файлами**

* + Джерела Інформації

<https://purecodecpp.com/archives/2751>

Що опрацьовано:

Матеріал з джерел інформації

* + Статус: Ознайомлений
  + Початок опрацювання теми: 29.11.2023
  + Звершення опрацювання теми: 29.11.2023

### Виконання роботи:

**Завдання №1**

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

**Завдання №1**

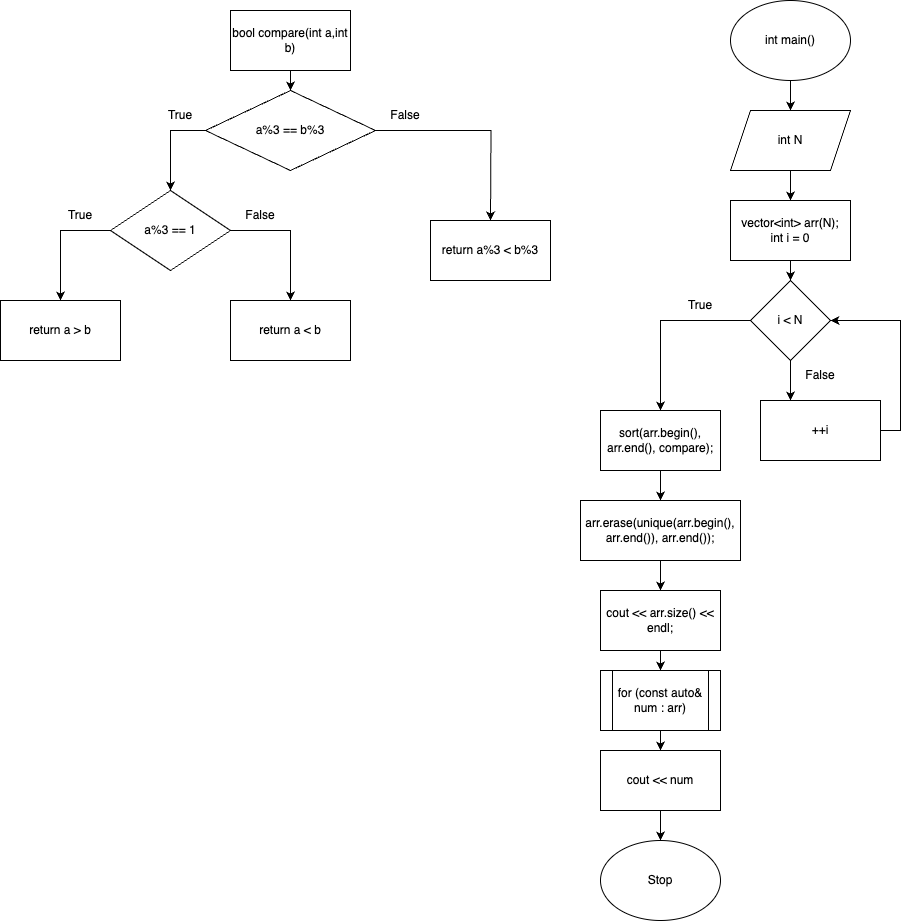
Тут вказуєте

-Варіант 3,Algolab 4

Розділити вхідний масив на три частини за залишком від ділення на 3: спочатку елементи з остачею 0, потім з остачею 1, і, нарешті, елементи з остачею .Сортувати числа з остачею 0 та 2 за зростанням, а числа з остачею 1 - за спаданням. Видалити будь-які дублікати, які можуть зустрітися в масиві після виконання попередніх операцій. Вивести результуючий масив без дублікатів згідно з описаними умовами сортування.

Запланований час для виконання завдання: 1 день

1. *Дизайн та планована оцінка часу виконання завдань:*

**

1. *Код програми*

#include <vector>

#include <algorithm>

#include <iostream>

using namespace std;

bool compare(int a, int b) {

if (a % 3 == b % 3) {

if (a % 3 == 1) {

return a > b;

} else {

return a < b;

}

} else {

return a % 3 < b % 3;

}

}

int main() {

int N;

cin >> N;

vector<int> arr(N);

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

cin >> arr[i];

}

sort(arr.begin(), arr.end(), compare);

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

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

for (const auto& num : arr) {

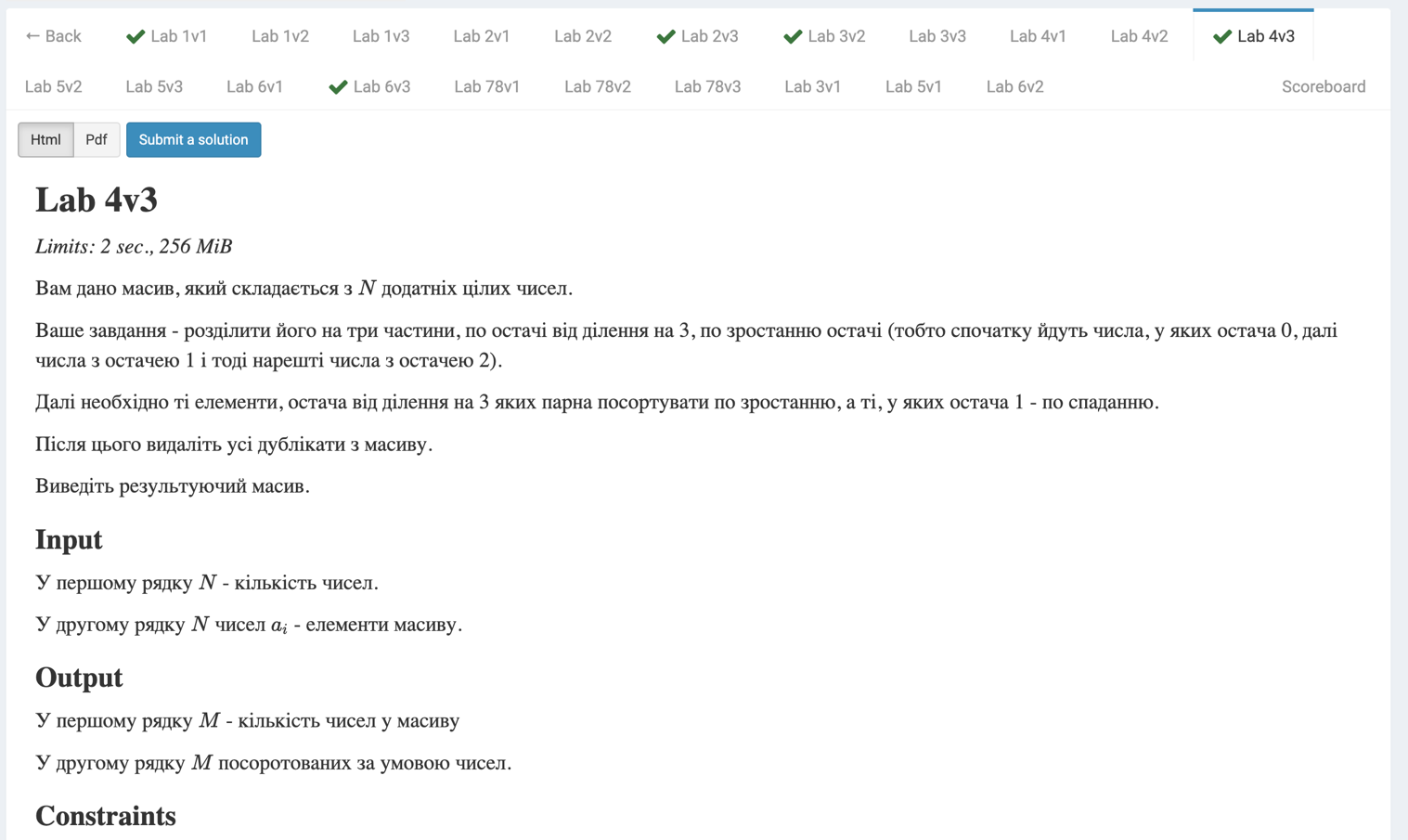
cout << num << ' ' << endl;

}

return 0;

}

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



Потрачено часу: 1 день

**Завдання №2**

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

-Варіант 3,Algolab 6

- У грі є поле розміром N × N , де деякі клітинки мають числові значення, а інші - пусті. Гравець має вказати, які числа можна вписати в кожну пусту клітинку. Завдання полягає в тому, щоб для кожної порожньої клітинки визначити, які числа можна в неї записати. При цьому потрібно враховувати, що в клітинку можна записати тільки ті числа, які не зустрічаються в тому ж рядку та стовпці, де знаходиться ця клітинка. Якщо ж для певної порожньої клітинки неможливо записати жодне число, виводиться 0.

1. *Код програми*

#include <iostream>

#include <vector>

using namespace std;

vector<int> PossibleValues(const vector<vector<int>>& matrix, int x, int y) { // Функція повертає можливі значення для клітинки матриці

int size = matrix.size();

vector<bool> Values\_present(size + 1, false);// Створення вектору для перевірки наявності значень

for (int i = 0; i < size; ++i) { // Створення вектору для перевірки наявності значень в клітинці

if (matrix[x - 1][i] != 0) {

Values\_present[matrix[x - 1][i]] = true;

}

if (matrix[i][y - 1] != 0) {

Values\_present[matrix[i][y - 1]] = true;

}

}

vector<int> values; // Створення вектору для можливих значень

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

if (!Values\_present[i]) {

values.push\_back(i);

}

}

return values;

}

int main() {

int size;

cin >> size;

if (size < 1 || size > 9) {

return 0;

}

vector<vector<int>> matrix(size, vector<int>(size, 0)); // Створення та заповнення матриці введеними значеннями

string matrixRow;

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

cin >> matrixRow;

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

matrix[i][j] = matrixRow[j] - '0';

if (matrix[i][j] < 0 || matrix[i][j] > 9) {

return 0;

}

}

}

int Q;

cin >> Q;

if (Q < 1 || Q > 1000) {

return 0;

}

vector<pair<int, int>> coordinates(Q);

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

int x, y;

cin >> x >> y;

if (x < 1 || x > size || y < 1 || y > size) {

return 0;

}

coordinates[q] = make\_pair(x, y);

}

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

int x = coordinates[q].first;

int y = coordinates[q].second;

if (matrix[x - 1][y - 1] != 0) {

cout << 1 << endl;

cout << matrix[x - 1][y - 1] << endl << endl;

continue;

}

vector<int> possibleValues = PossibleValues(matrix, x, y); // Отримання можливих значень для порожньої клітинки та виводимо їх

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

for (const auto& value : possibleValues) {

cout << value << endl;

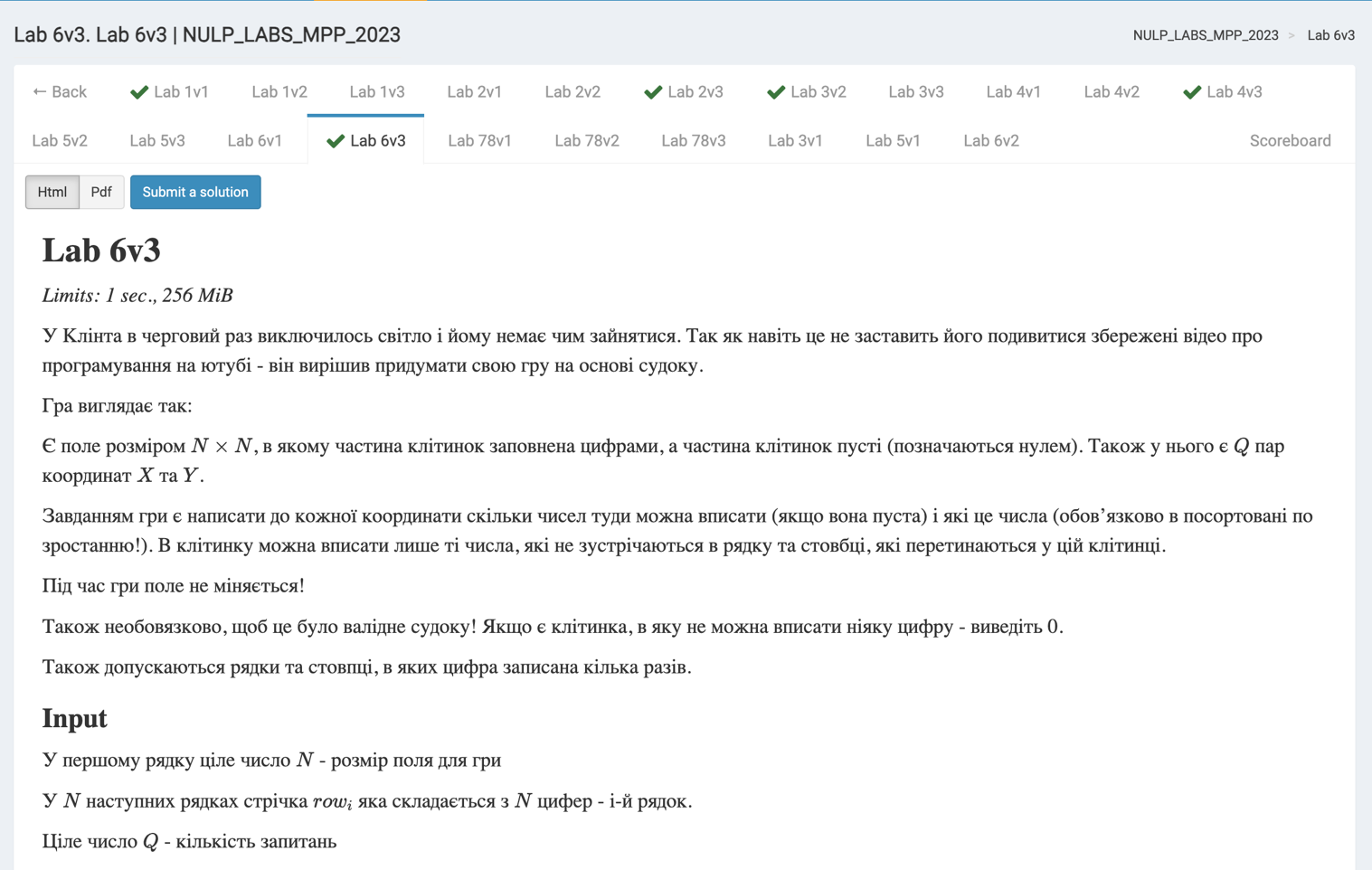
}

}

return 0;

}

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

**

Потрачено часу: 1 день

**Завдання №3**

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

-Варіант 4,VNS Lab 6

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 всі слова, які співпадають з її першим словом.

1. *Код програми*

#include <iostream>

#include <cstring>

using namespace std;

void printMatchingWords(const char \*input) {

char inputCopy[256]; // Створення копії введеного рядка, оскільки функція strtok модифікує вхідний рядок

strcpy(inputCopy, input);

char \*firstWord = strtok(inputCopy, " "); // Отримання першого слова

cout << "Слова, які співпадають з першим словом '" << firstWord << "':" << endl;

char \*word;

while ((word = strtok(NULL, " ")) != NULL) { // Отримання інших слів та їх порівняння з першим словом

if (strcmp(word, firstWord) == 0) {

cout << word << endl;

}

}

}

int main() {

cout << "Введіть рядок тексту (не більше 255 символів) з крапкою в кінці:" << endl;

char input[256];

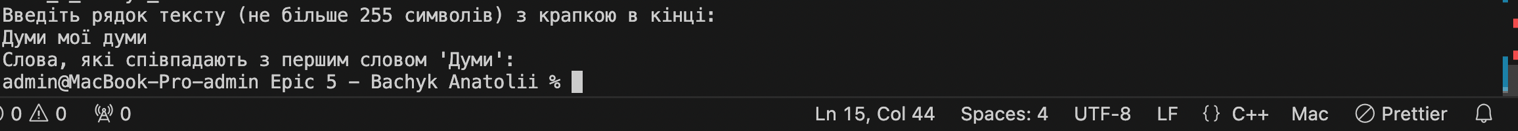
cin.getline(input, sizeof(input)); // Введення рядка тексту

printMatchingWords(input); // Виклик функції для обробки та виведення слів

return 0;

}

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

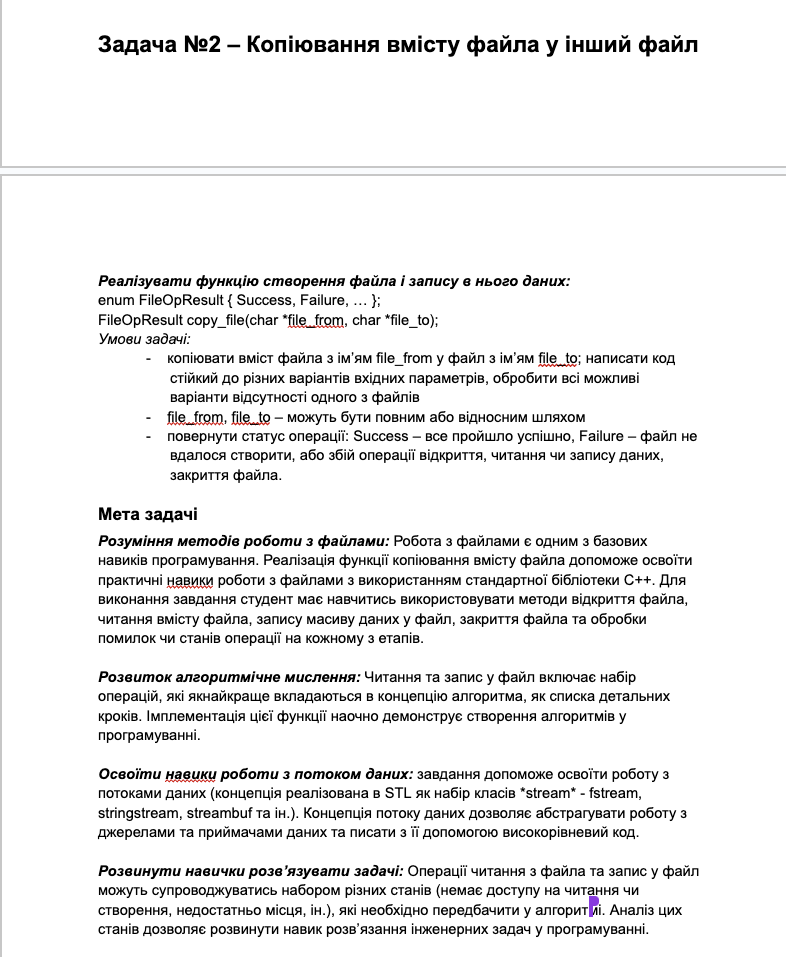
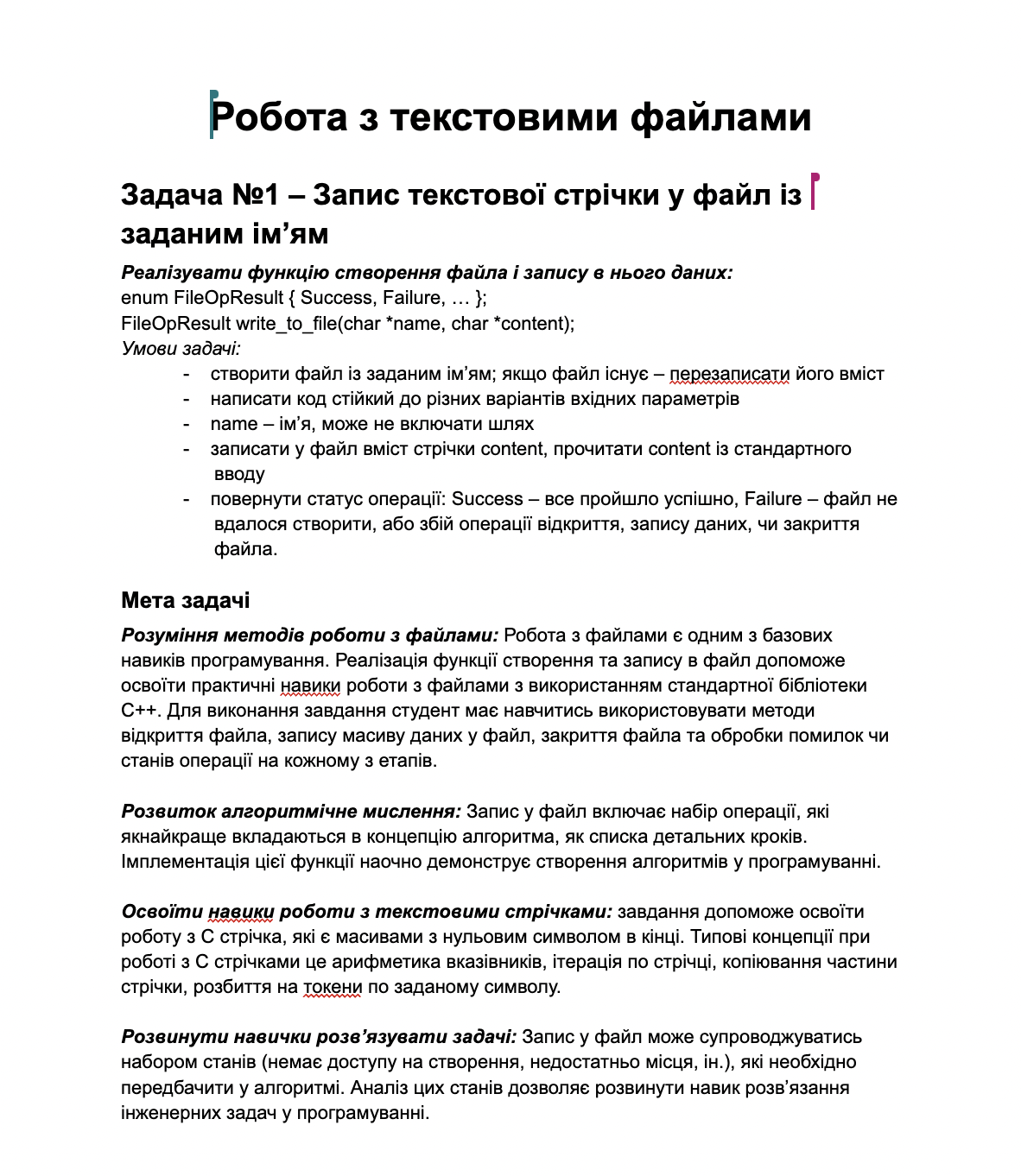


Потрачено часу: 1 день

**Завдання №4**

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

- Class Practice Work



1. *Код програми*

#include <fstream>

#include <iostream>

using namespace std;

enum FileOpResult { Success, Failure };

FileOpResult write\_to\_file(char \*name, char \*content) { // Функція для запису в файл

ofstream file(name);

if (!file.is\_open()) {

return Failure;

}

file << content;// Запис вмісту в файл

if (file.fail()) {

return Failure;

}

file.close();

if (file.fail()) {

return Failure;

}

return Success;

}

int main() {

char name[100];

char content[1000];

cout << "Enter file name: ";

cin.getline(name, 100);

cout << "Enter content: ";

cin.getline(content, 1000);

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

if (result == Success) {

cout << "File written successfully." << endl;

} else {

cout << "Failed to write to file." << endl;

}

return 0;

}

#include <fstream>

#include <string>

using namespace std;

enum FileOpResult { Success, Failure };

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

ifstream src(file\_from, ios::binary); // Відкривання вихідного файлу

ofstream dst(file\_to, ios::binary); // Відкривання цільового файлу

if (!src.is\_open() || !dst.is\_open()) {

return Failure;

}

dst << src.rdbuf(); // Копіювання вмісту вихідного файлу у цільовий файл

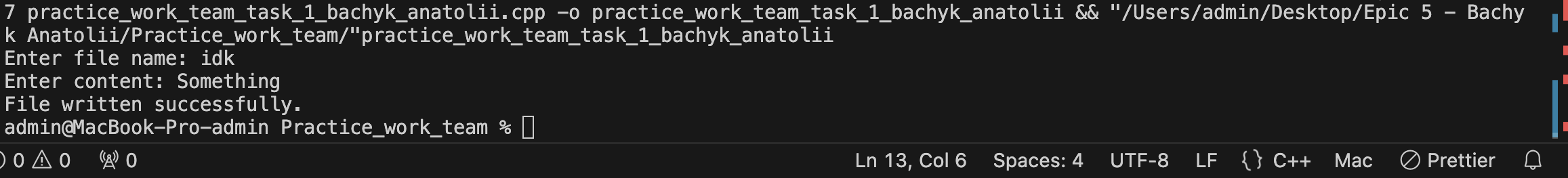
if (src.bad() || dst.bad()) {

return Failure;

}

return Success;

}

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

Потрачено часу: 1 день

**Завдання №5**

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

-Self Practice Work

- Дана прямокутна матриця розміром n × m . Є два типи запитів:

1. Заміна місцями двох підматриць: Обміняти місцями дві підматриці з h рядками та w стовпцями. Координати верхніх лівих кутів першої та другої підматриць задаються як ( x 1 , y 1 ) та ( x 2 , y 2 ) відповідно. Підматриці не перетинаються.

2. Виведення значення: Вивести значення елементу матриці за координатами ( x , y ) .

Необхідно обробити q запитів, кожен з яких може бути одного з описаних типів вище.

1. *Код програми*

#include <iostream>

#include <vector>

using namespace std;

int main() {

int n, m;

cin >> n >> m;

vector<vector<int>> matrix(n, vector<int>(m));

// Зчитуємо матрицю

for (int i = 0; i < n; ++i) {

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

cin >> matrix[i][j];

}

}

int q;

cin >> q;

while (q--) { // виконується поки q не стане 0

int t;

cin >> t;

if (t == 1) {

int x1, y1, x2, y2, h, w;

cin >> x1 >> y1 >> x2 >> y2 >> h >> w;

// Поміняти місцями підматриці

vector<vector<int>> temp(h, vector<int>(w));// Створення тимчасової матриці

for (int i = 0; i < h; ++i) {

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

temp[i][j] = matrix[x1 + i - 1][y1 + j - 1]; // Копіюємо значення з першої підматриці в тимчасову

matrix[x1 + i - 1][y1 + j - 1] = matrix[x2 + i - 1][y2 + j - 1];// Копіюємо значення з другої підматриці в першу

matrix[x2 + i - 1][y2 + j - 1] = temp[i][j]; // Копіюємо значення з тимчасової матриці в другу підматрицю

}

}

} else if (t == 2) {

int x, y;

cin >> x >> y;

cout << matrix[x - 1][y - 1] << endl;

}

}

return 0;

}

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

Потрачено часу: *2 години*

**Завдання №5**

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

-Варіант 4,VNS Lab 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.

Структура "Людина":

- прізвище, ім'я, по батькові;

- домашня адреса;

- номер телефону;

- вік.

Знищити усі елементи із заданим віком, додати елемент після елемента із

заданим номером.

1. *Код програми*

#include <fstream>

#include <iostream>

#include <vector>

#include <string>

using namespace std;

struct Human {

string surname;

string name;

string patronymic;

string address;

string phone\_number;

int age;

};

void writeData(const string& filename, const vector<Human>& data) {

ofstream file(filename, ios::binary);

if (!file) {

cerr << "Помилка при відкритті файлу для запису.\n";

return;

}

for (const auto& item : data) {

file.write(reinterpret\_cast<const char\*>(&item), sizeof(Human));

}

}

vector<Human> readData(const string& filename) {

ifstream file(filename, ios::binary);

if (!file) {

cerr << "Помилка при відкритті файлу для читання.\n";

return {};

}

vector<Human> data;

Human item;

while (file.read(reinterpret\_cast<char\*>(&item), sizeof(Human))) {

data.push\_back(item);

}

return data;

}

void printData(const vector<Human>& data) {

for (const auto& item : data) {

cout << item.surname << ' ' << item.name << ' ' << item.patronymic << '\n';

cout << item.address << '\n';

cout << item.phone\_number << '\n';

cout << item.age << '\n';

}

}

vector<Human> removeByAge(vector<Human>& data, int age) {

vector<Human> newData;

for (const auto& item : data) {

if (item.age != age) {

newData.push\_back(item);

}

}

return newData;

}

void insertAfterNumber(vector<Human>& data, const string& number, const Human& newHuman) {

for (auto it = data.begin(); it != data.end(); ++it) {

if (it->phone\_number == number) {

data.insert(it + 1, newHuman);

break;

}

}

}

int main() {

vector<Human> data = {

{"Бачик", "Анатолій", "Андрійович", "Львів, вул. Шевченка, 1", "+38066234567", 17},

{"Сидоров", "Олег", "Васильович", "Київ, вул. Хрещатик, 47", "+380501777568", 50},

{"Мельник", "Сергій", "Олексійович", "Одеса, вул. Дерибасівська, 3", "+38413234569", 63}

};

writeData("data.bin", data);

vector<Human> read\_data = readData("data.bin");

printData(read\_data);

read\_data = removeByAge(read\_data, 40);

printData(read\_data);

Human newHuman = {"Щербовських", "Андрій", "Миколайович", "Харків, вул. Сумська, 4", "+380511234570", 41};

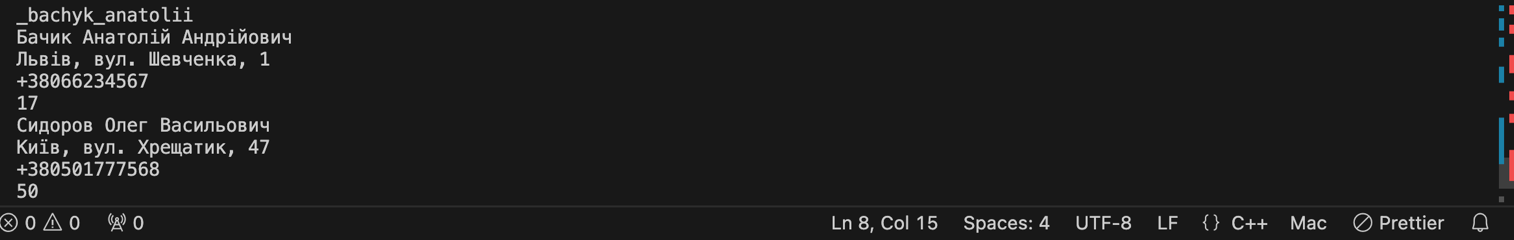
insertAfterNumber(read\_data, "+380501774567", newHuman);

printData(read\_data);

return 0;

}

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



Потрачено часу: *2 дня*

**Завдання №6**

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

-Варіант 4,VNS Lab 9

- 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 4.

2) Підрахувати кількість символів в останньому слові F2.

1. *Код програми*

#include <fstream>

#include <string>

#include <sstream>

#include <iostream>

using namespace std;

int main() {

ofstream file("F1.txt");// Створення об'єкта для запису в файл F1.txt

if (file.is\_open()) {

for (int i = 1; i <= 10; i++) {

file << "Це рядок номер " << i << endl;

}

file.close();

}

// Копіювання рядків, починаючи з 4-го, з файлу F1.txt до файлу F2.txt

ifstream file1("F1.txt");// Створення об'єкта для читання з файлу F1.txt

ofstream file2("F2.txt"); // Створення об'єкта для запису в файл F2.txt

string line;

int line\_number = 0;

if (file1.is\_open() && file2.is\_open()) {

while (getline(file1, line)) {// Читання рядків з файлу F1.txt

line\_number++;

if (line\_number >= 4) {

file2 << line << endl;// Запис рядка в файл F2.txt

}

}

file1.close();

file2.close();

}

// Підрахунок кількості символів в останньому слові F2

ifstream file2\_read("F2.txt");// Створення об'єкта для читання з файлу F2.txt

string last\_word;

if (file2\_read.is\_open()) {

while (file2\_read >> last\_word) {}

file2\_read.close();

}

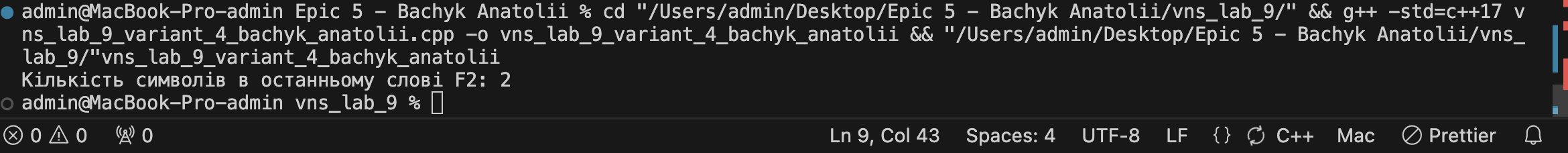
int num\_chars = last\_word.length();

cout << "Кількість символів в останньому слові F2: " << num\_chars << endl;

return 0;

}

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



Потрачено часу: *1 день*

### Висновки:

Ознайомлений з файлами, cистемами числення, символами і рядковими змінними та текстовими файлами