МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



# Кафедра: САП

Звіт до виконаної лабораторної роботи №8 з

дисципліни “Алгоритмізація та програмування”

на тему: "Структури та об’єднання в мові програмування С++ "

*Виконав:*

*студент групи ПП-16*

*Якіб’юк*

*Ігор Прийняла/в*

*: Гілета І.В.*

*Львів – 2023*

Лабораторна робота № 8

**Варіант 15**

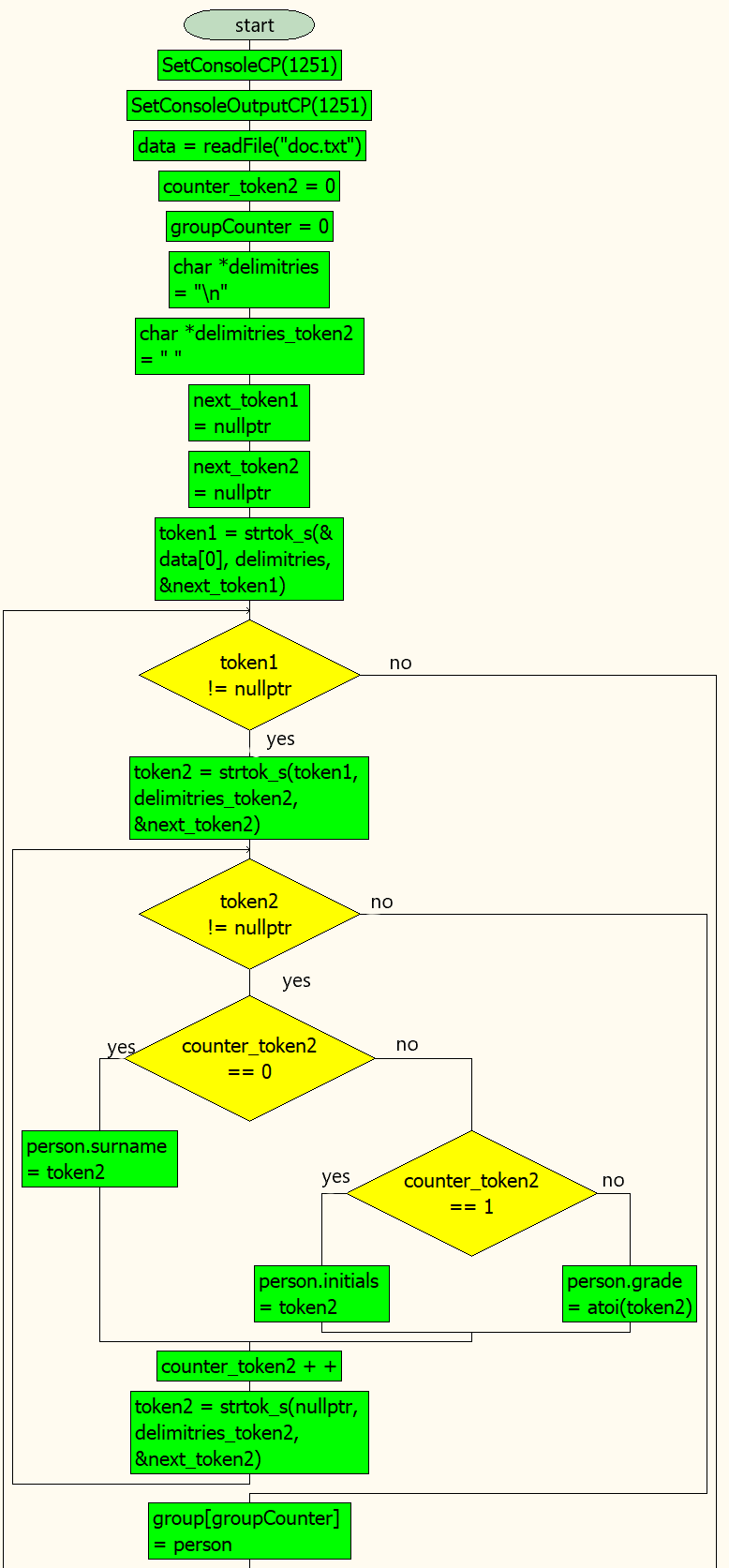
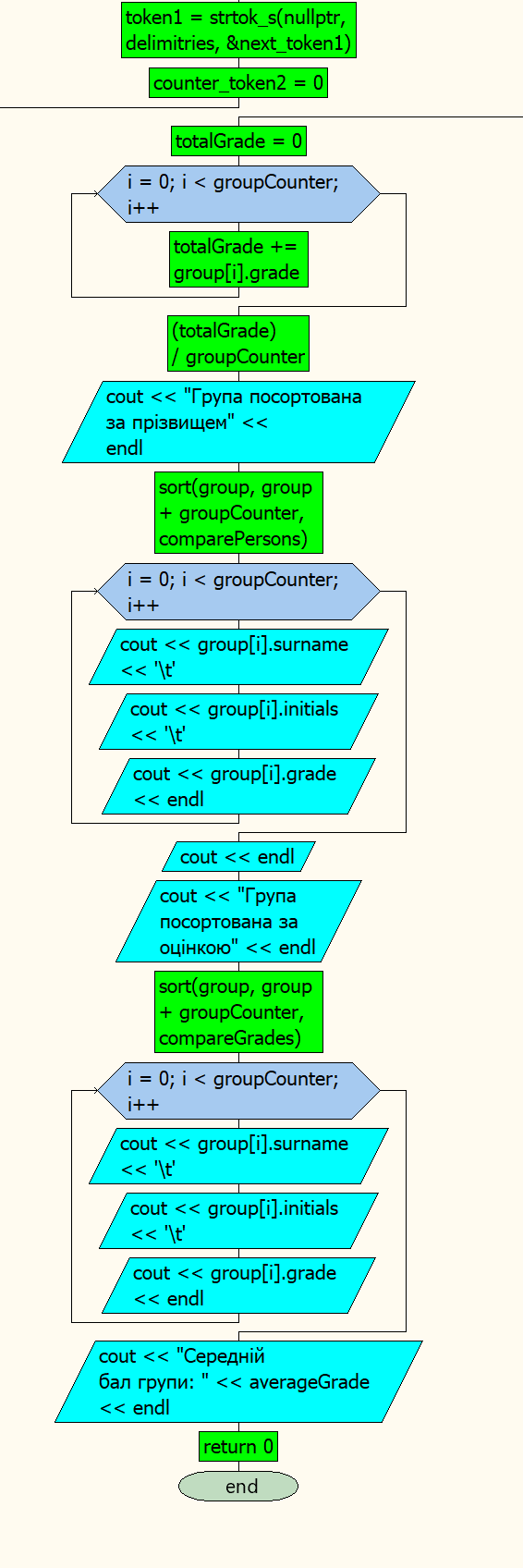
**Мета:** Навчитися використовувати структури та об’єднання у мові С++.

**Умова завдання(1):** Завдання 1

Скласти програму що дає змогу з використанням структур та об'єднань реалізувати розв’язок поставленої задачі. Всі вхідні дані беруться з текстового файлу (створити не менше десяти відповідних записів у файлі). Ввід-вивід даних та виконання інших окремих логічних дій необхідно реалізувати в окремих функціях. У головній функції необхідно виконувати лише їх виклик. Використання глобальних змінних не допускається. Інформація повинна передаватися у функції лише за допомогою параметрів.

15. У вхідному файлі записана відомість, де входить прізвище, ініціали, кількість набраних балів. Вивести на друк розсортовану відомість по прізвіщу і по набраних балах та середній бал групи.

* **Блок-схема до завдання 1:**



* **Код програми завдання(1):**

#include <iostream>

#include <fstream>

#include <windows.h>

#include <string>

#include <algorithm>

using namespace std;

string readFile(string nameFile);

struct Person {

char\* surname;

char\* initials;

int grade;

};

bool comparePersons(const Person& a, const Person& b);

bool compareGrades(const Person& a, const Person& b);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Person group[10];

string data = readFile("doc.txt");

int counter\_token2 = 0;

int groupCounter = 0;

char\* profileData[256];

const char\* delimitries = "\n";

const char\* delimitries\_token2 = " ";

char\* next\_token1 = nullptr;

char\* next\_token2 = nullptr;

char\* token1 = strtok\_s(&data[0], delimitries, &next\_token1);

while (token1 != nullptr) {

Person person;

char\* token2 = strtok\_s(token1, delimitries\_token2, &next\_token2);

while (token2 != nullptr) {

if (counter\_token2 == 0) {

person.surname = token2;

}

else if (counter\_token2 == 1) {

person.initials = token2;

}

else

{

person.grade = atoi(token2); // atoi для перетворення рядка на число

}

counter\_token2++;

token2 = strtok\_s(nullptr, delimitries\_token2, &next\_token2);

}

group[groupCounter] = person;

groupCounter++;

token1 = strtok\_s(nullptr, delimitries, &next\_token1);

counter\_token2 = 0;

}

int totalGrade = 0;

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

totalGrade += group[i].grade;

}

double averageGrade = static\_cast<double>(totalGrade) / groupCounter;

cout << "Група посортована за прізвищем" << endl;

sort(group, group + groupCounter, comparePersons);

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

cout << group[i].surname << '\t';

cout << group[i].initials << '\t';

cout << group[i].grade << endl;

}

cout << endl;

cout << "Група посортована за оцінкою" << endl;

sort(group, group + groupCounter, compareGrades);

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

cout << group[i].surname << '\t';

cout << group[i].initials << '\t';

cout << group[i].grade << endl;

}

cout << "Середній бал групи: " << averageGrade << endl;

return 0;

}

string readFile(string nameFile) {

ifstream file(nameFile);

string data;

if (file.is\_open()) {

getline(file, data, '\0');

file.close();

}

return data;

}

bool comparePersons(const Person& a, const Person& b) {

int surnameComparison = strcmp(a.surname, b.surname);

return surnameComparison < 0;

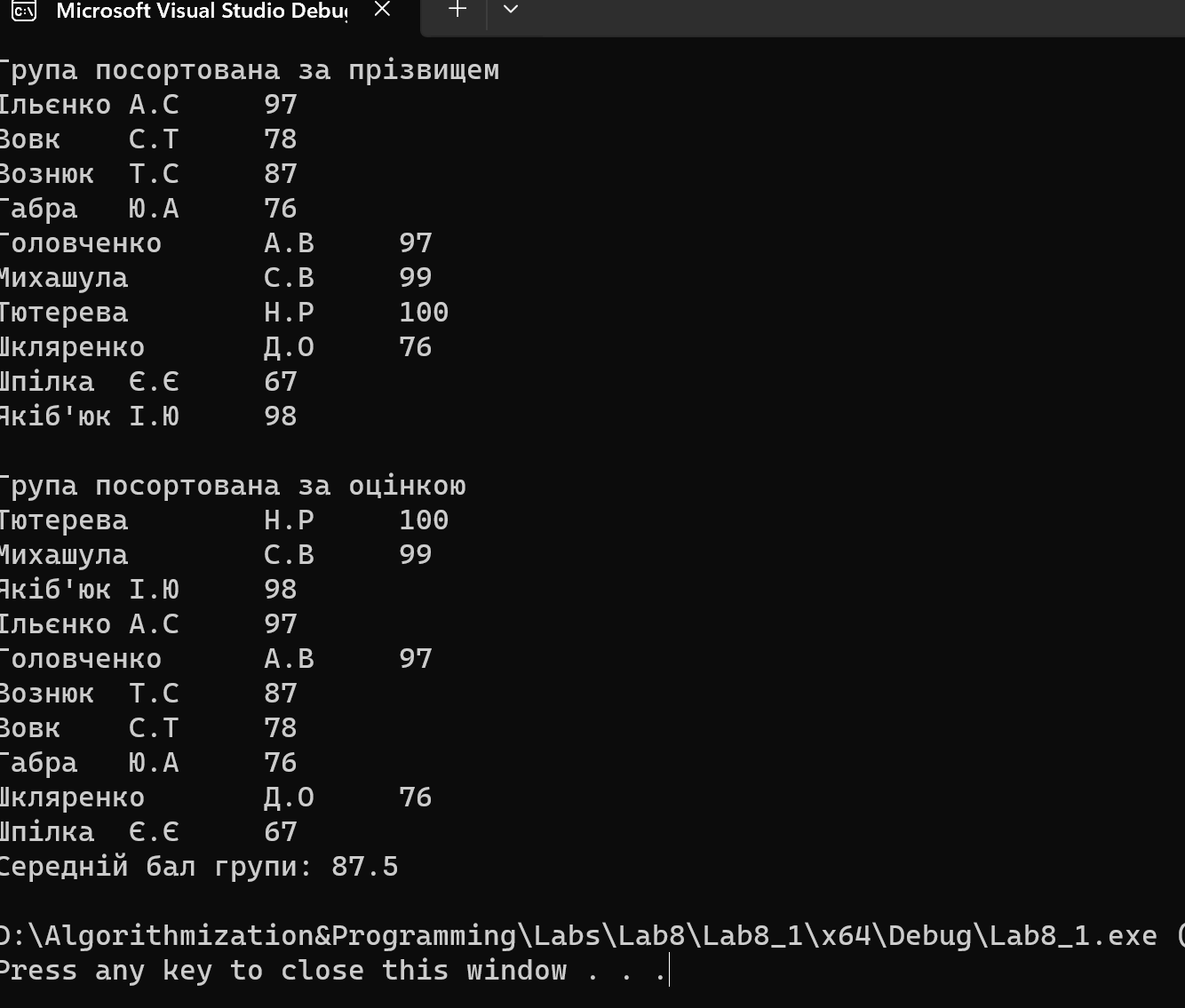
}

bool compareGrades(const Person& a, const Person& b) {

return a.grade > b.grade;

* }

**Результат виконання програми(1):**



**Умова завдання(2):**

Завдання 2

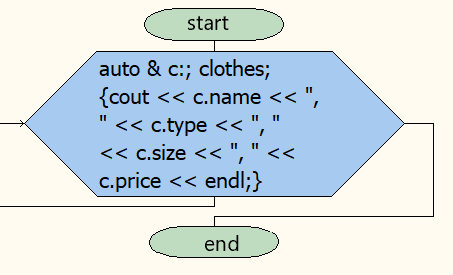
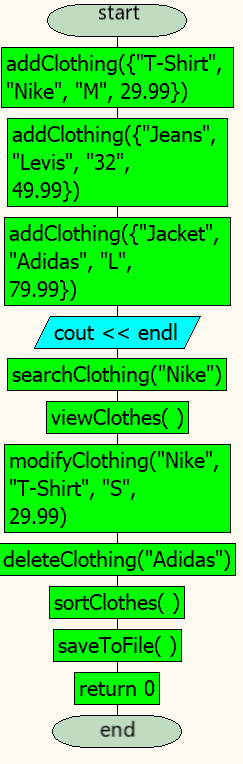
Розробити програму яку забезпечує опрацювання структур даних. Необхідно забезпечити опрацювання 3-5 полів елементів з використанням різних простих типів даних (стрічки, символи, числа). Забезпечити виконання таких операцій:

* 1. додавання нового елементу;
  2. пошук елементу за значенням полів;
  3. послідовний перегляд елементів;
  4. модифікація значень полів елемену;
  5. видалення елементу;
  6. сортування за значеннями полів.

Результати всіх операцій повинні зберігатись у файлі (створити не менше десяти відповідних записів у файлі). Елемент

1. Одяг (тип, назва, розмір, ціна).

**Блок-схема до завдання 2:**



**Код програми завдання (2):**

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

#include <string>

using namespace std;

struct Clothing {

string type;

string name;

string size;

double price;

};

vector<Clothing> clothes;

void addClothing(Clothing c) {

clothes.push\_back(c);

}

void searchClothing(string name) {

for (const auto& c : clothes) {

if (c.name == name) {

cout << "Found: " << c.name << ", " << c.type << ", " << c.size << ", " << c.price << endl;

}

}

}

void viewClothes() {

for (const auto& c : clothes) {

cout << c.name << ", " << c.type << ", " << c.size << ", " << c.price << endl;

}

}

void modifyClothing(string name, string newType, string newSize, double newPrice) {

for (auto& c : clothes) {

if (c.name == name) {

c.type = newType;

c.size = newSize;

c.price = newPrice;

}

}

}

void deleteClothing(string name) {

clothes.erase(remove\_if(clothes.begin(), clothes.end(), [name](const Clothing& c) { return c.name == name; }), clothes.end());

}

bool compareByName(const Clothing& a, const Clothing& b) {

return a.name < b.name;

}

void sortClothes() {

sort(clothes.begin(), clothes.end(), compareByName);

}

void saveToFile() {

ofstream file("clothes.txt");

if (file.is\_open()) {

for (const auto& c : clothes) {

file << c.name << ", " << c.type << ", " << c.size << ", " << c.price << endl;

}

}

file.close();

}

int main() {

// Adding new items

cout << "Adding new items" << endl;

addClothing({ "T-Shirt", "Nike", "M", 29.99 });

addClothing({ "Jeans", "Levis", "32", 49.99 });

addClothing({ "Jacket", "Adidas", "L", 79.99 });

saveToFile();

viewClothes();

cout << "Searching for an item by name" << endl;

// Searching for an item by name

searchClothing("Nike");

cout << endl;

// Viewing all items

cout << "Viewing all items" << endl;

viewClothes();

cout << endl;

// Modifying item details

cout << "Modifying item details" << endl;

modifyClothing("Nike", "T-Shirt", "S", 29.99);

saveToFile();

viewClothes();

// Deleting an item

cout << endl << "Deleting an item" << endl;

deleteClothing("Adidas");

saveToFile();

viewClothes();

// Sorting items by name

cout << endl << "Sorting items by name" << endl;

sortClothes();

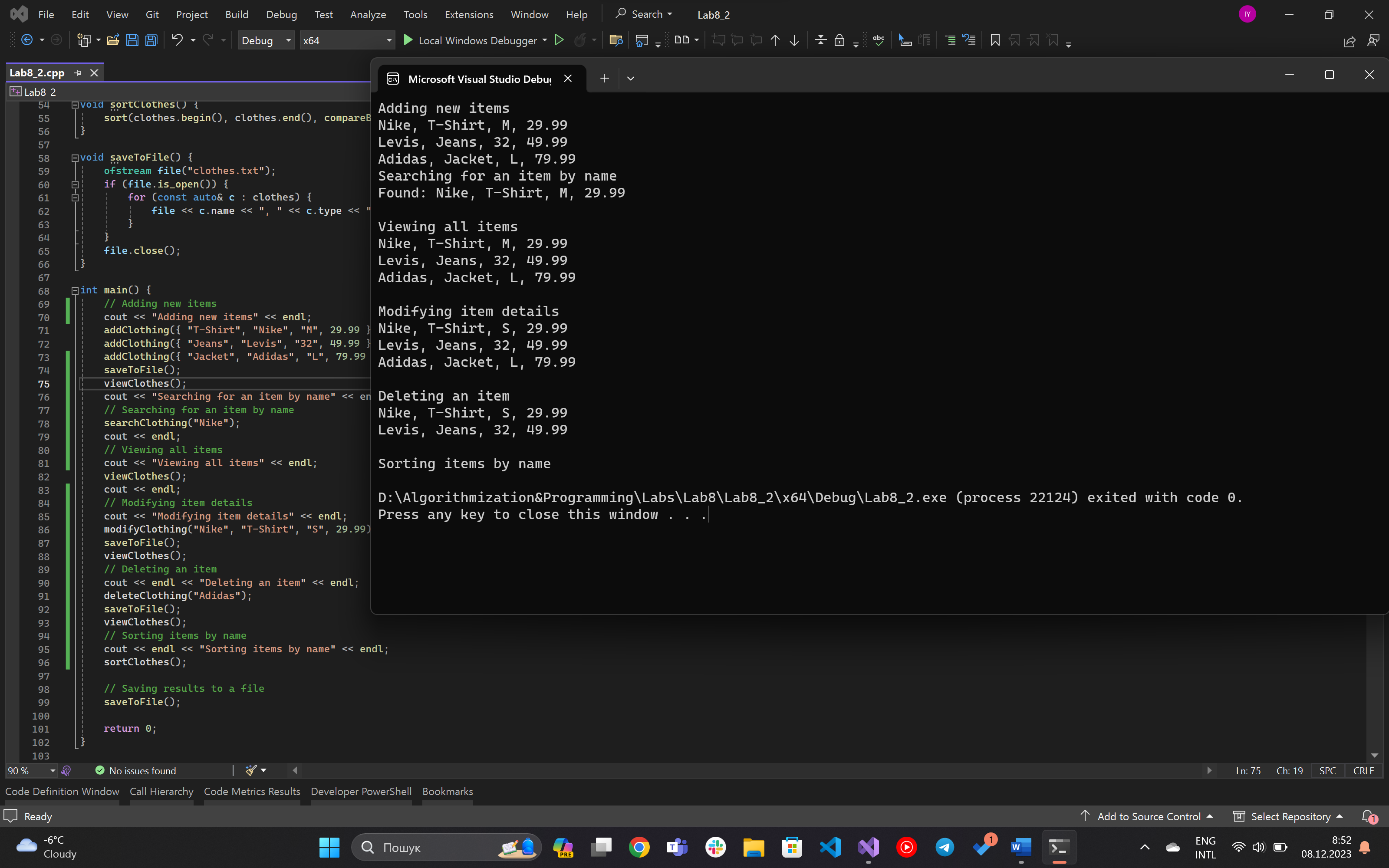
// Saving results to a file

saveToFile();

return 0;

}

**Результат виконання завдання(2):**



**Висновок:** На даній лабораторній роботі я навчився використовувати структури у коді, задля оптимізації мікролокальних баз даних, при написанні програм на мові С++. В обидвох завданнях використовував структури. При виконанні роботи, труднощів не виникло. Блоксхеми і код продемонстровані вище.