**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА КОМП’ЮТЕРНИХ НАУК**

**СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**ПРОЕКТУВАННЯ**

**Пояснювальна записка**

**До курсової роботи**

**З дисципліни**

**«Програмування»**

Виконав: Студент групи IT-91

Дєдовський Д. І.

Варіант: 8

Перевірив: Ващенко. С. М.

**2020**

**ЗМІСТ**

[1 ПОСТАНОВКА ЗАДАЧІ 5](#_Toc40477339)

[2 БЛОК СХЕМИ ФУНКЦІЙ 7](#_Toc40477340)

[2.1 Функція int main() 7](#_Toc40477341)

[2.2 void DeleteInformation() 8](#_Toc40477342)

[2.3 void AmountElement(law ved) 9](#_Toc40477343)

[2.4 void BeginOutPut() 10](#_Toc40477344)

[2.5 void OutPutInfo(law ved, int lich) 11](#_Toc40477345)

[2.6 void ShowInfo() 12](#_Toc40477346)

[2.7 void ShowInfoAccordingVariant() 13](#_Toc40477348)

[2.8 void AddInfo() 14](#_Toc40477349)

[3 ОПИС ПІДПРОГРАМ КОРИСТУВАЧА 17](#_Toc40477350)

[4 ТЕКСТ ПРОГРАМИ 18](#_Toc40477351)

[5 ОПИС ПРОГРАМИ 31](#_Toc40477352)

[6 КОНТРОЛЬНИЙ ВИГЛЯД 33](#_Toc40477353)

**1 ПОСТАНОВКА ЗАДАЧІ**

Взявши за основу свій варіант структури запису, реалізувати роботу з найпростішою базою даних, використовуючи двійкові файли (файли структур).

У програмі реалізувати створення нової бази даних і додавання записів в уже створену. Вхідні дані повинні вводитися з перевіркою на область допустимих значень. Усі дії користувача повинні контролюватися і супроводжуватися обдуманими повідомленнями. Реалізувати у відповідності до свого варіанту (рис.1.1, рис.1.2):

• запит (обчислення окремого значення);

• пошук інформації за ключем.

Передбачити можливість виведення всіх записів бази даних на екран.

Уся обробка бази даних повинна відбуватися шляхом вибору користувачем відповідного пункту з меню.

Обов’язково повинна бути зазначена інформація про виконавця роботи та завдання на курсову роботу.

У програмі повинні бути передбачені відповідні функції, що коректно обробляють вхідні дані, виконують читання-запис файлів бази даних. Інформація повинна запам’ятовуватися у файлі і бути доступною для наступної (багаторазової) обробки.

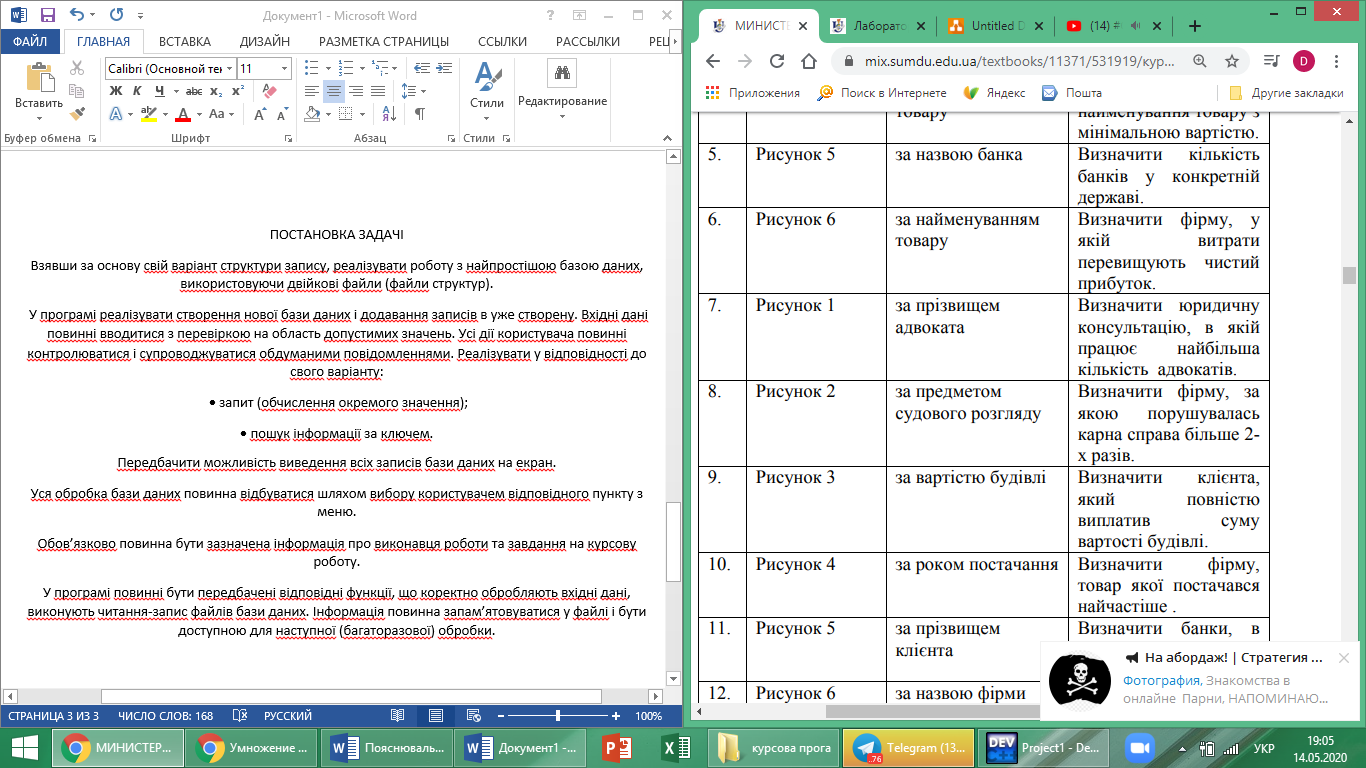


Рисунок 1.1 - Завдання до курсової

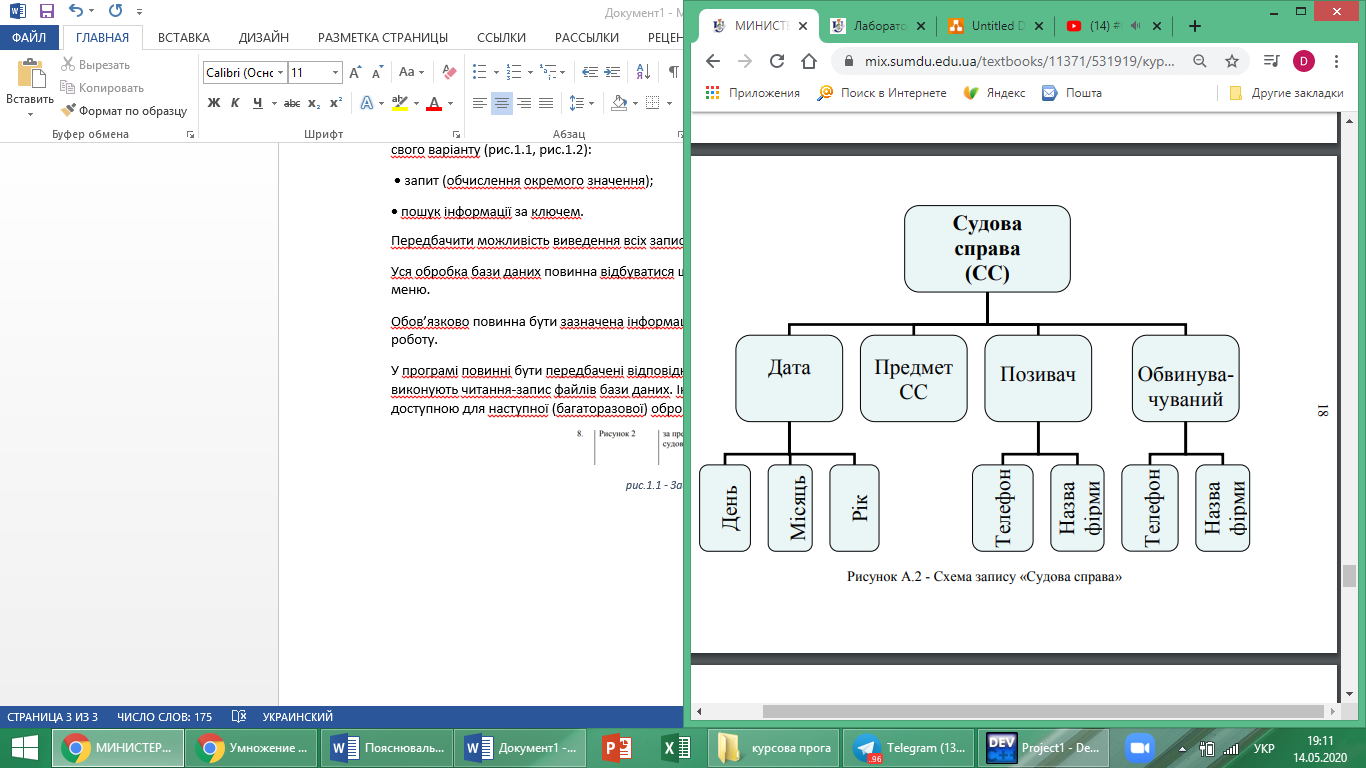
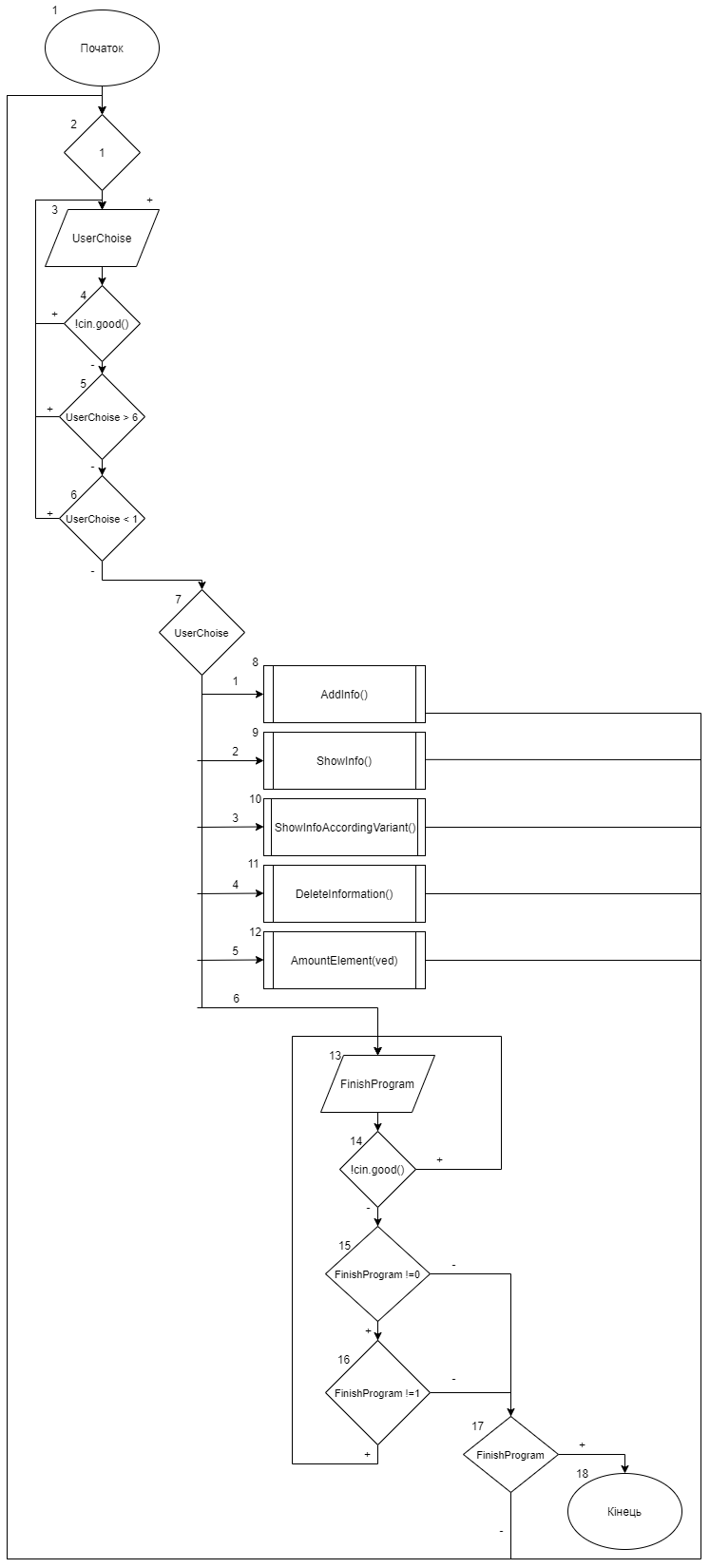


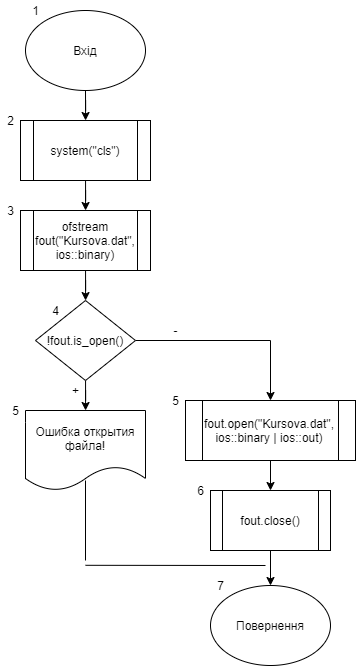
Рисунок 1.2 - Завдання до курсової

# **2 БЛОК СХЕМИ ФУНКЦІЙ**

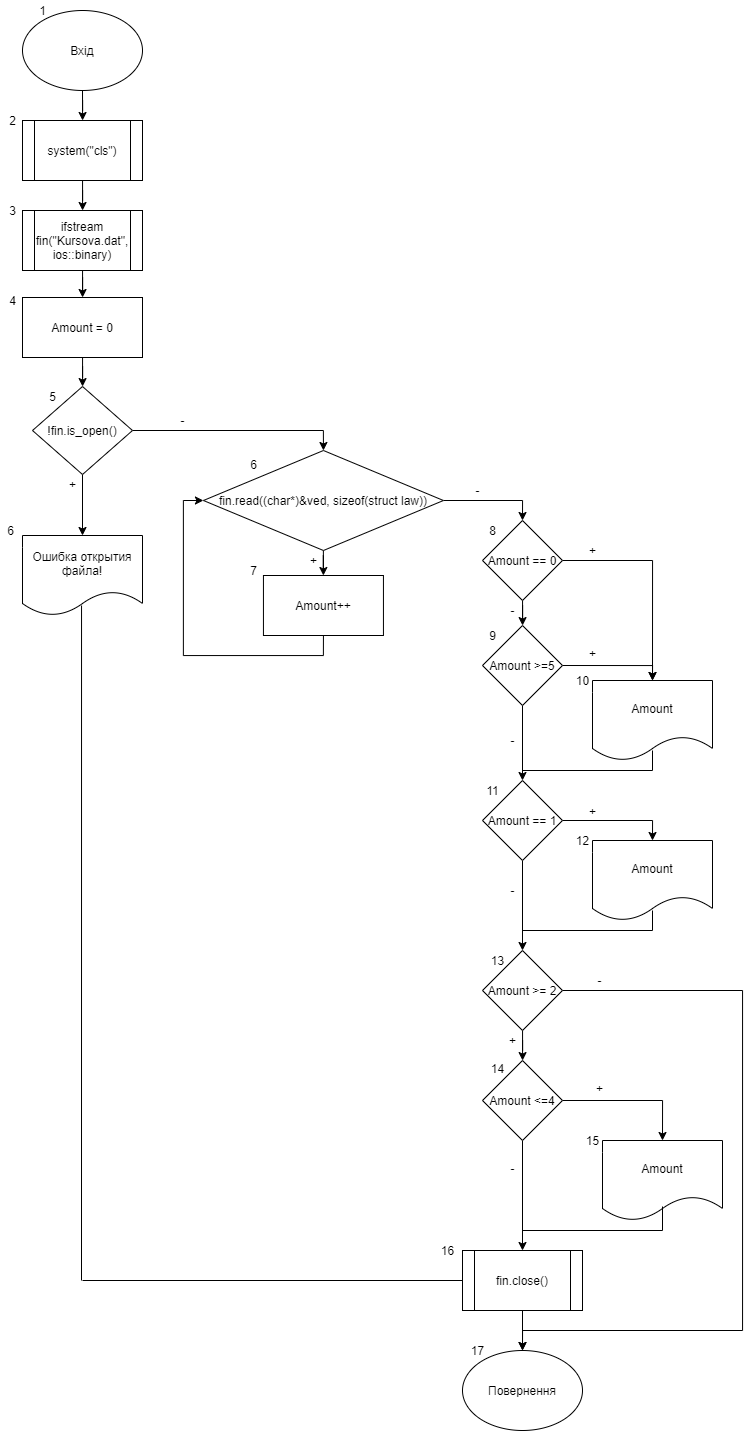
**2.1 Функція int main()**



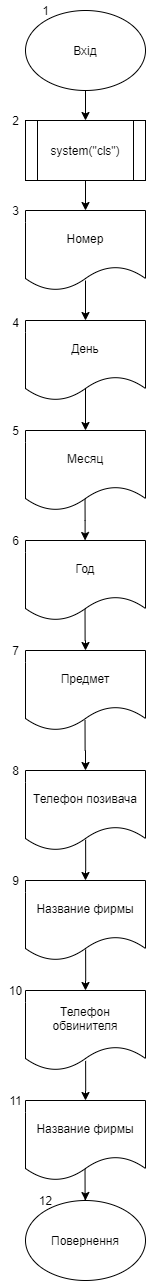
**2.2 void DeleteInformation()**



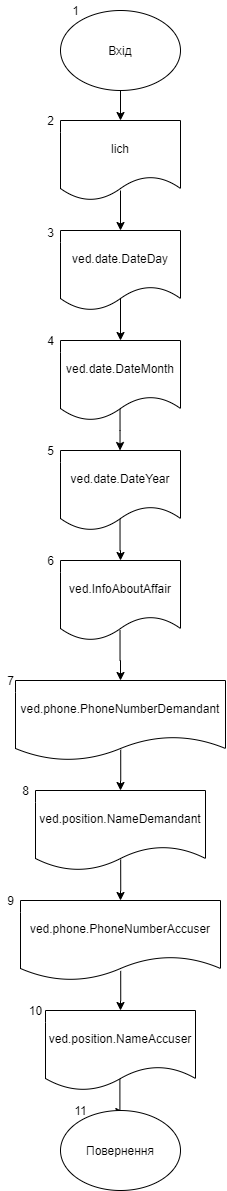
**2.3 void AmountElement(law ved)**



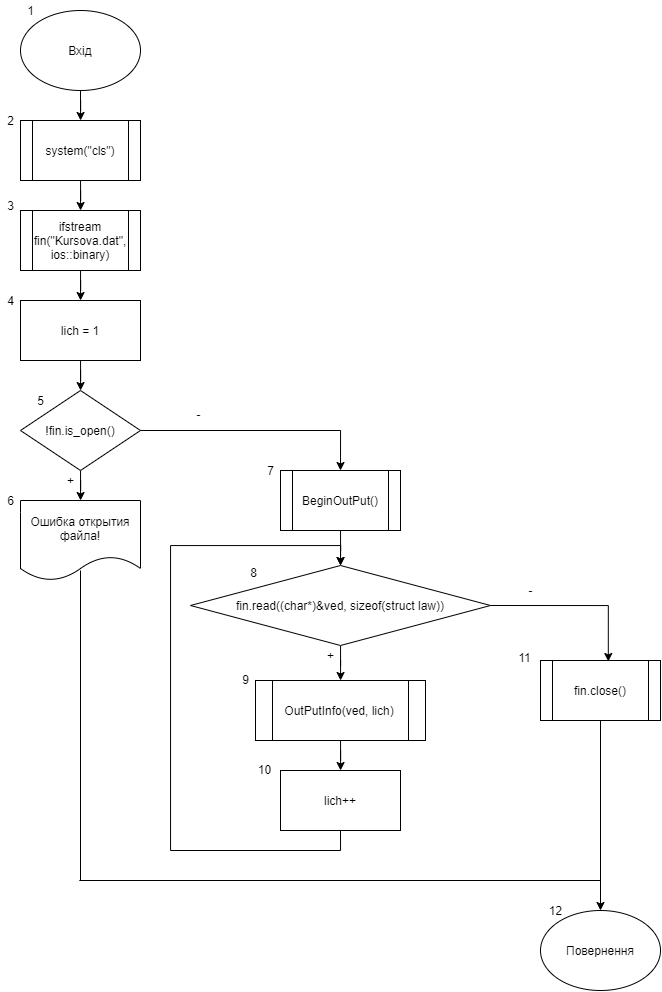
**2.4 void BeginOutPut()**



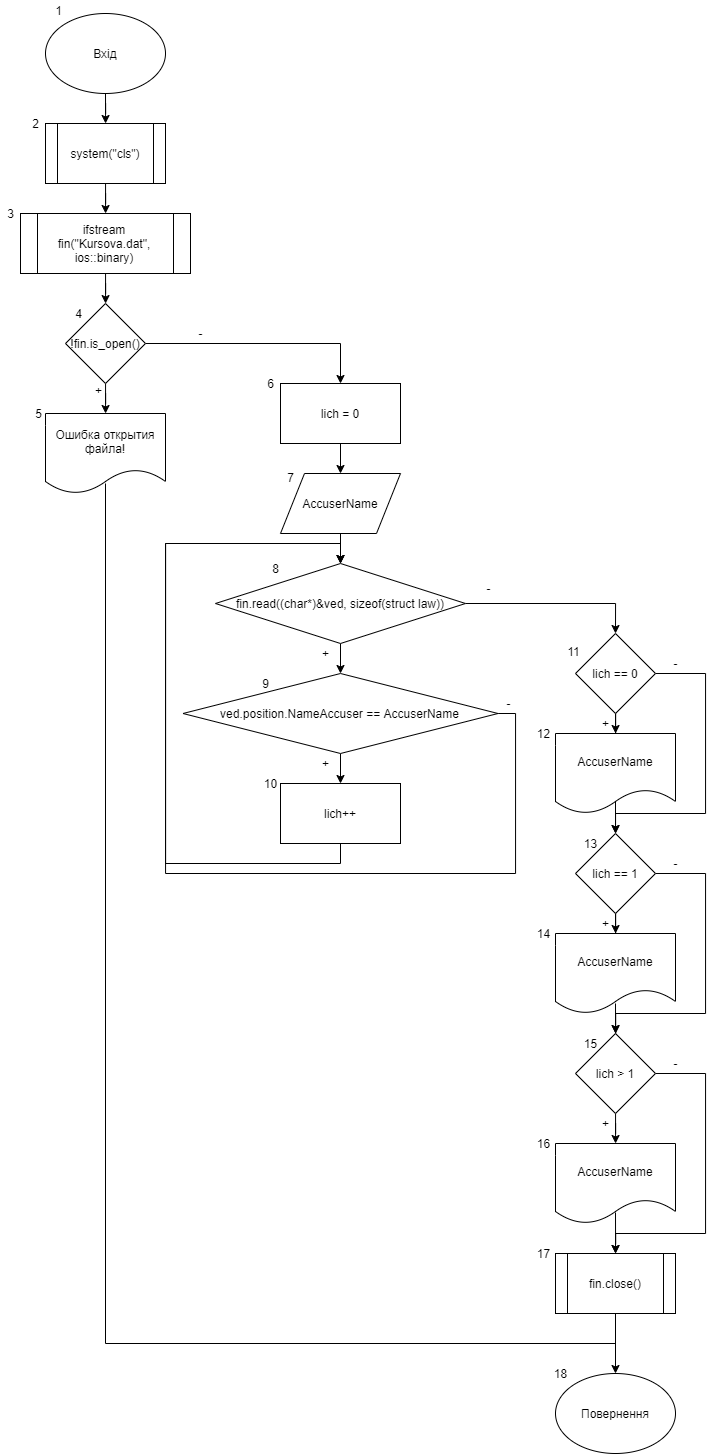
**2.5 void OutPutInfo(law ved, int lich)**



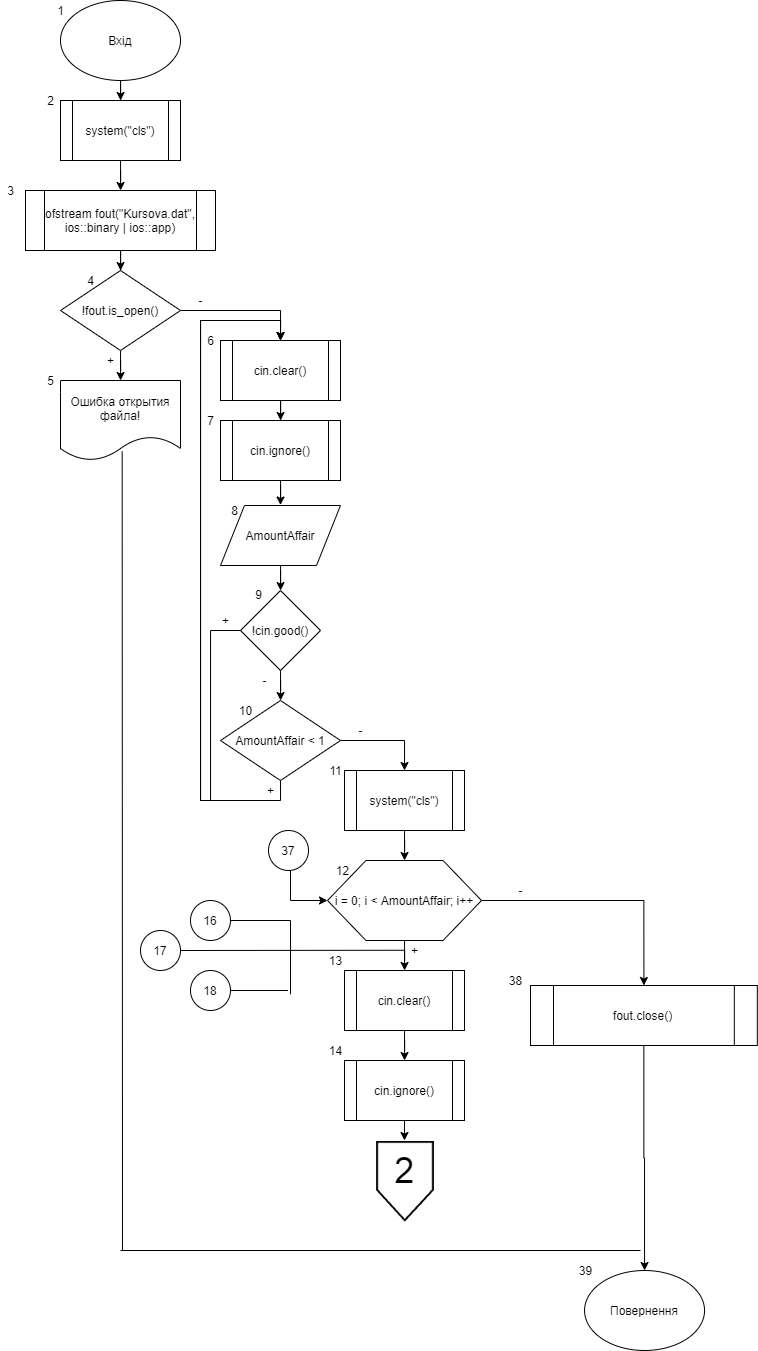
**2.6 void ShowInfo()**

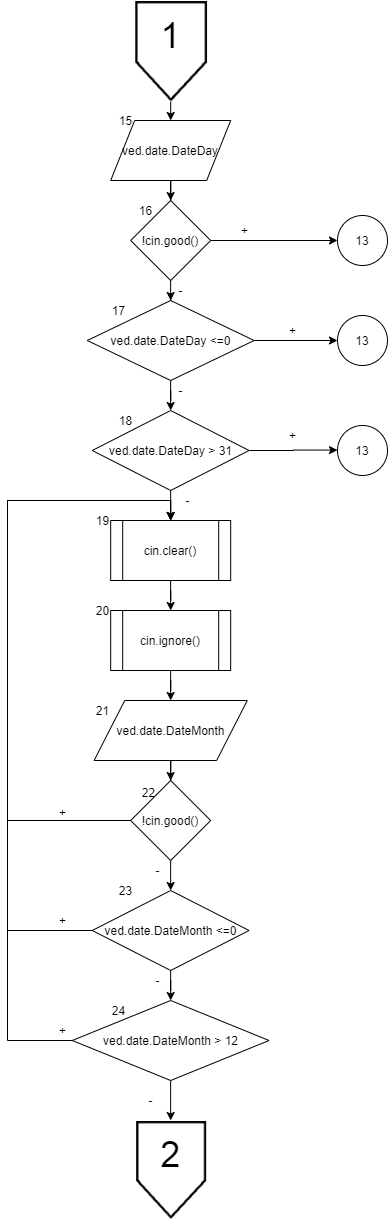


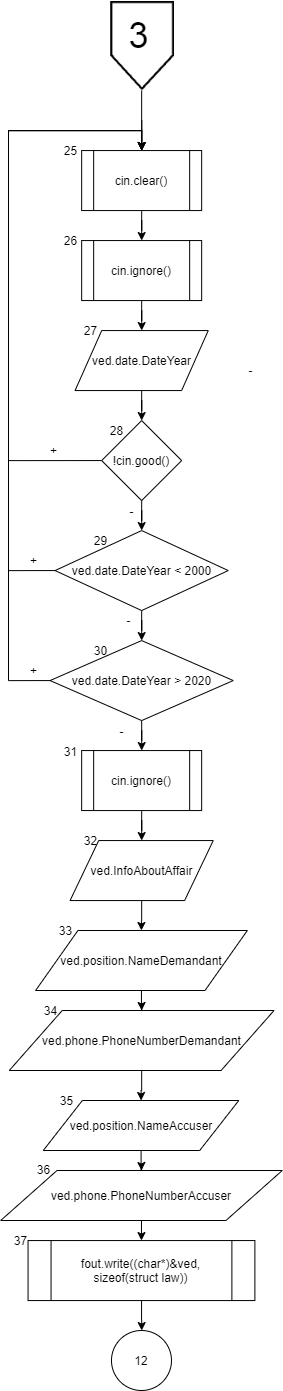
**2.7 void ShowInfoAccordingVariant()**



**2.8 void AddInfo()**







# **3** **ОПИС ПІДПРОГРАМ КОРИСТУВАЧА**

Підпрограма DeleteInformation() реалізує видалення всіх даних з файлу, тобто очищеннЯ бази даних від всієї інформації.

Підпрограма AmountElement() реалізує виведення інформації про кількість елементів, які знаходяться в базі даних.

Підпрограма BeginOutPut() реалізує виведення «шапки»(ключових слів), для кращого сприйняття інформації.

Підпрограма OutPutInfo() реалізує виведення на екран інформації про номер судової справи, дату, місяць, рік, предмет судової справи, номер фірми позивача, назву фірми позивача, номер фірми обвинувача, назву фірми обвинувача.

Підпрограма ShowInfo() реалізує виведення на екран інформації, яка є в файлі, тобто виведення всієї інформації, яка є в базі даних.

Підпрограма ShowInfoAccordingVariant() реалізує виведення на екран інформації про те, скільки разів порушувалась карна справа на конкретну фірму.Назву фірми вводить користувач.

Підпрограма AddInfo() реалізує введення інформації користувачем про судову справу.Кількість судових справ, які користувач хоче ввести задає самостійно.Користувач вводить дату, місяць, рік, предмет судової справи, номер фірми позивача, назву фірми позивача, номер фірми обвинувача, назву фірми обвинувача.Коректність введення даних передбачена.

# **4** **ТЕКСТ ПРОГРАМИ**

**Таблиця 4.1 – Опис змінних в програмі**

|  |  |
| --- | --- |
| ***Позначення в програмі*** | ***Пояснення*** |
| DateDay | Поле структури, що описує дату судової справи |
| DateMonth | Поле структури, що описує місяць судової справи |
| DateYear | Поле структури, що описує рік судової справи |
| NameDemandant | Поле структури, що описує назву фірми позивача |
| NameAccuser | Поле структури, що описує назву фірми обвинувача |
| PhoneNumberDemandant | Поле структури, що описує номер фірми позивача |
| PhoneNumberAccuser | Поле структури, що описує номер телефону фірми обвинувача |
| InfoAboutAffair | Поле структури, що описує предмет судової справи |
| Amount | Змінна, що відображає поточну кількість елементів в базі даних |
| lich | Лічильник |
| AccuserName | Змінна, яка зберігає назву фірми, по якій хочуть дізнатися скільки судових справ з нею порушено |
| LichVariant | Змінна, яка відображає кількість судових справ, які порушені на конкретну фірму |
| AmountAffair | Змінна, яка відображає кількість судових справ, які користувач хоче додати в базу даних |
| i | Лічильник |
| UserChoise | Змінна, яка відображає, який пункт меню користувач хоче обрати |
| FinishProgram | Змінна, яка відображає чи дійсно користувач хоче завершити роботу програми |

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

struct date{

unsigned int DateDay;

unsigned int DateMonth;

unsigned int DateYear;

};

struct position{

char NameDemandant[15];

char NameAccuser[15];

};

struct phone{

char PhoneNumberDemandant[15];

char PhoneNumberAccuser[15];

};

struct law{

date date;

char InfoAboutAffair[15];

position position;

phone phone;

};

law ved;

void DeleteInformation(){

system("cls");

ofstream fout("Kursova.dat", ios::binary);

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

cout << "Ошибка открытия файла!" << endl;

}

else{

fout.open("Kursova.dat", ios::binary | ios::out);

}

fout.close();

system("pause");

}

void AmountElement(law ved){

system("cls");

unsigned short int Amount = 0;

ifstream fin("Kursova.dat", ios::binary);

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

cout << "Ошибка открытия файла!" << endl;

}

else{

while(fin.read((char\*)&ved, sizeof(struct law))){

Amount++;

}

if(Amount == 0 || Amount >=5){

cout << "Сейчас в базе данных имеется " << Amount << " елементов." << endl;

}

if(Amount == 1){

cout << "Сейчас в базе данных имеется " << Amount << " елемент." << endl;

}

if(Amount >= 2 && Amount <=4){

cout << "Сейчас в базе данных имеется " << Amount << " елемента." << endl;

}

}

fin.close();

system("pause");

}

void BeginOutPut(){

system("cls");

cout << "--------------------------------------------------------------------------------------------------------------------------------" << endl;

cout << "|" << setw(7) << "Номер" << "|" << setw(6) << "День" << "|" << setw(7) << "Месяц" << " | " << setw(7) << "Год" << "|" << setw(13) << "Предмет" << "|" << setw(20) << "Телефон позивача"

<< "|" << setw(18) << "Название фирмы" << "|" << setw(20) << "Телефон обвинителя" << "|" << setw(18) << "Название фирмы" << "|" << endl;

cout << "--------------------------------------------------------------------------------------------------------------------------------" << endl;

}

void OutPutInfo(law ved, int lich){

cout << "|" << setw(7) << lich << "|" << setw(6) << ved.date.DateDay << "|" << setw(8) << ved.date.DateMonth << "|" << setw(8) << ved.date.DateYear << "|" << setw(13) << ved.InfoAboutAffair << "|" << setw(20) << ved.phone.PhoneNumberDemandant

<< "|" << setw(18) << ved.position.NameDemandant << "|" << setw(20) << ved.phone.PhoneNumberAccuser << "|" << setw(18) << ved.position.NameAccuser << "|" << endl;

cout << "--------------------------------------------------------------------------------------------------------------------------------" << endl;

}

void ShowInfo(){

system("cls");

ifstream fin("Kursova.dat", ios::binary);

unsigned int lich = 1;

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

cout << "Ошибка открытия файла!" << endl;

}

else{

BeginOutPut();

while(fin.read((char\*)&ved, sizeof(struct law))){

OutPutInfo(ved, lich);

lich++;

}

}

fin.close();

system("pause");

}

void ShowInfoAccordingVariant(){

system("cls");

ifstream fin("Kursova.dat", ios::binary);

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

cout << "Ошибка открытия файла!" << endl;

}

else{

int LichVariant = 0;

string AccuserName;

cout << "Введите название фирмы по которой хотите получить информацию было ли возбуждено на эту фирму уголовное дело: ";

cin >> AccuserName;

while(fin.read((char\*)&ved, sizeof(struct law))){

if(ved.position.NameAccuser == AccuserName){

LichVariant++;

}

}

if(LichVariant == 0){

cout << "С фирмой " << AccuserName << " не возбуждалось уголовное дело.Возбуждено " << LichVariant << " уголовных дел." << endl;

}

if(LichVariant == 1){

cout << "С фирмой " << AccuserName << " возбуждалось уголовное дело.Возбуждено " << LichVariant << " уголовное дело." << endl;

}

if(LichVariant > 1){

cout << "С фирмой " << AccuserName << " возбуждалось уголовное дело.Возбуждено " << LichVariant << " уголовных дела." << endl;

}

}

fin.close();

system("pause");

}

void AddInfo(){

system("cls");

ofstream fout("Kursova.dat", ios::binary | ios::app);

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

cout << "Ошибка открытия файла!" << endl;

}

else{

short int AmountAffair;

do{

cout << "Введите количество судебных дел которые хотите добавить.Будьте внимательные нужно вводить число и адекватные значения, ввод:";

cin.clear();

cin.ignore();

cin >> AmountAffair;

}while(!cin.good() || AmountAffair < 1);

system("cls");

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

do{

cout << "Введите день подачи заявления.Будьте внимательные нужно вводить число и адекватные значения, ввод:";

cin.clear();

cin.ignore();

cin >> ved.date.DateDay;

}while(!cin.good() || ved.date.DateDay <=0 || ved.date.DateDay > 31);

cout << "Введите месяц подачи заявления:" << endl;

do{

cout << "Будьте внимательные нужно вводить число и адекватные значения, ввод:";

cin.clear();

cin.ignore();

cin >> ved.date.DateMonth;

}while(!cin.good() || ved.date.DateMonth <=0 || ved.date.DateMonth > 12);

cout << "Введите год подачи заявления:" << endl;

do{

cout << "Будьте внимательные нужно вводить число и адекватные значения, ввод:";

cin.clear();

cin.ignore();

cin >> ved.date.DateYear;

}while(!cin.good() || ved.date.DateYear < 2000 || ved.date.DateYear > 2020);

cout << "Введите предмет подачи заявления:";

cin.ignore();

cin.getline(ved.InfoAboutAffair, 15);

cout << "Введите название фирмы позивача:";

cin.getline(ved.position.NameDemandant, 15);

cout << "Введите телефон фирмы позивача:";

cin.getline(ved.phone.PhoneNumberDemandant, 15);

cout << "Введите название фирмы обвинителя:";

cin.getline(ved.position.NameAccuser, 15);

cout << "Введите телефон фирмы обвинителя:";

cin.getline(ved.phone.PhoneNumberAccuser, 15);

fout.write((char\*)&ved, sizeof(struct law));

}

}

fout.close();

system("pause");

}

int main(){

setlocale(LC\_ALL, "Russian");

int UserChoise;

while(1){

system("cls");

cout << "Выберите пункт" << endl;

cout << "1)Добавить информацию в базу данных" << endl;

cout << "2)Посмотреть информацию, которая есть в базе данных" << endl;

cout << "3)Посмотреть список фирм, на которых было заведено больше 2-ох дел" << endl;

cout << "4)Удалить всю базу данных" << endl;

cout << "5)Узнать сколько елемент находится в базе данных" << endl;

cout << "6)Завершить программу" << endl;

do{

cout << "Введите число от 1 до 6(число - это пункт который хотите выполнить).Будьте внимательные нужно вводить число и адекватные значения, ввод:";

cin.clear();

cin.ignore();

cin >> UserChoise;

}while(!cin.good() || UserChoise > 6 || UserChoise < 1);

switch(UserChoise){

case 1:

AddInfo();

break;

case 2:

ShowInfo();

break;

case 3:

ShowInfoAccordingVariant();

break;

case 4:

DeleteInformation();

break;

case 5:

AmountElement(ved);

break;

case 6:

cout << "Вы уверены, что хотите завершить програму(1 - да, 0 - нет)" << endl;

int FinishProgram;

do{

cout << "Будьте внимательные нужно вводить число и адекватные значения, ввод:";

cin.clear();

cin.ignore();

cin >> FinishProgram;

}while(!cin.good() || ( FinishProgram !=0 && FinishProgram !=1));

if(FinishProgram){

system("pause");

return 0;

}

else{

break;

}

}

}

system("pause");

return 0;

}

# **5** **ОПИС ПРОГРАМИ**

Після запуску програми користувач може побачити контекстне меню.На його вибір наведено 6 пунктів меню.Для подальшої роботи програми користувач повинен ввести цифру від 1 до 6(включно).Кожна цифра – це пункт меню, тобто що програма буде виконувати, якщо натиснути ту чи іншу клавішу.

Цифра 1 запускає роботу функції AddInfo(), яка відповідає за введення інформації в базу даних(файл).Якщо користувач натисне цифру 1 перед ним з’явиться наступне повідомлення: «Введіть будь-ласка кількість судових справ, які хочете додати до бази даних.».Цифра, яку введе користувач і буде тією кількістю судових справ, які йому потрібно ввести.Після закінчення введення інформації, користувач буде перенаправлений в контекстне меню(6 пунктів на вибір).

Цифра 2 запускає роботу функції ShowInfo(), яка відповідає за виведення інформації, яка знаходиться в базі данних.Якщо користувач натисне цифру 2 перед ним з’явиться повна інформацію про всі судові справи, які на даний момент знаходяться в базі даних(файлі). Після закінчення виведення інформації, користувач буде перенаправлений в контекстне меню(6 пунктів на вибір).

Цифра 3 запускає роботу функції ShowInfoAccordingVariant(), яка відповідає за виведення інформації про фірму, з якою порушувалася карна справа.Тобто, користувач повинен ввести назву фірми обвинувача і програма надасть інформацію про те скільки разів порушувалася карна справа з цією фірмою. Після закінчення виведення інформації, користувач буде перенаправлений в контекстне меню(6 пунктів на вибір).

Цифра 4 запускає роботу функції DeleteInformation(), яка відповідає за видалення всієї інформації з бази даних(файлу).Ніяких дій від користувача не треба програма сама видалить всі дані. Після закінчення видалення інформації, користувач буде перенаправлений в контекстне меню(6 пунктів на вибір).

Цифра 5 запускає роботу функції AmountElement(), яка відповідає за виведення інформації про кількість елементів, які знаходяться на даний момент в базі данних(файлі).Ніяких дій від користувача не потрібно, програма сама виведе кількість елементів, які знаходяться в базі данних(файлі). Після закінчення виведення інформації, користувач буде перенаправлений в контекстне меню(6 пунктів на вибір).

Цифра 6 відповідає за закінчення роботи програми.Якщо користувач натисне цифру 6, то перед ним з’явиться повідомлення чи дійсно він хоче закінчити роботу з програмою.Якщо далі користувач вводить цифру 1 – програма завершує свою роботу, але якщо користувач вводить цифру 0 – то він буде перенаправлений в контекстне меню(6 пунктів на вибір).

Також при роботу функції ShowInfo(), запускаються дві функції BeginOutPut(), яка відповідає за виведення на екран «шапки»(ключових слів) і функція OutPutInfo(), яка відповідає за виведення на екран елементів судової справи із бази даних(файлу).

# 

# **6** **КОНТРОЛЬНИЙ ВИГЛЯД**

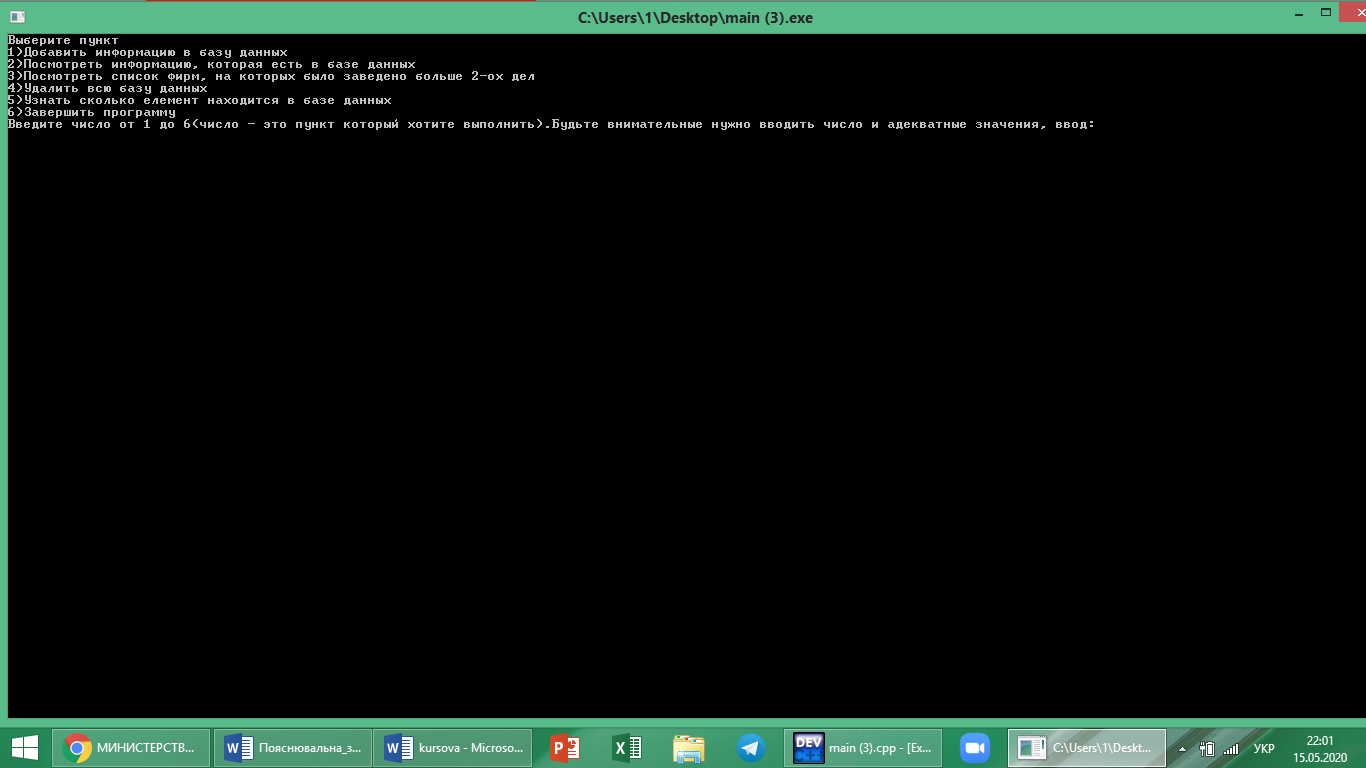


Рисунок 6.1 - Контекстне меню програми

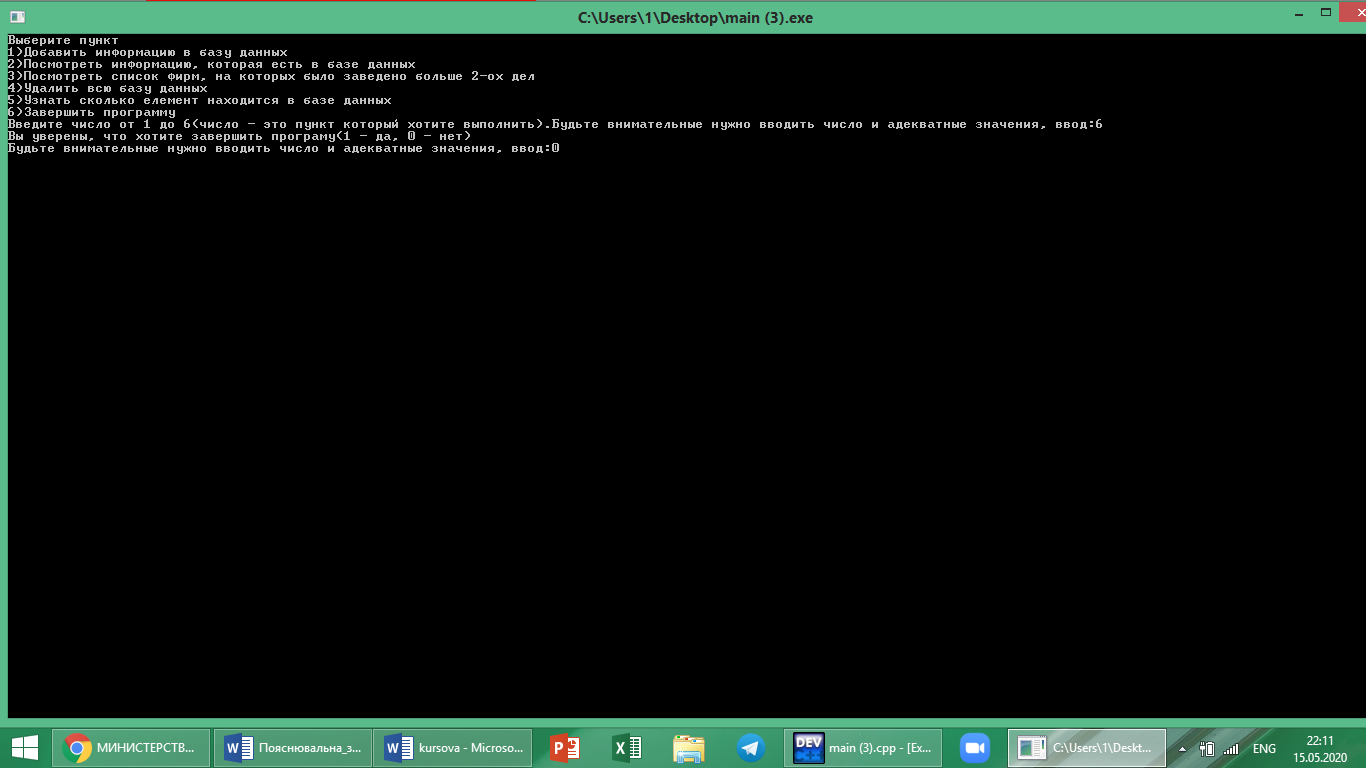


Рисунок 6.2 - Введення пункту

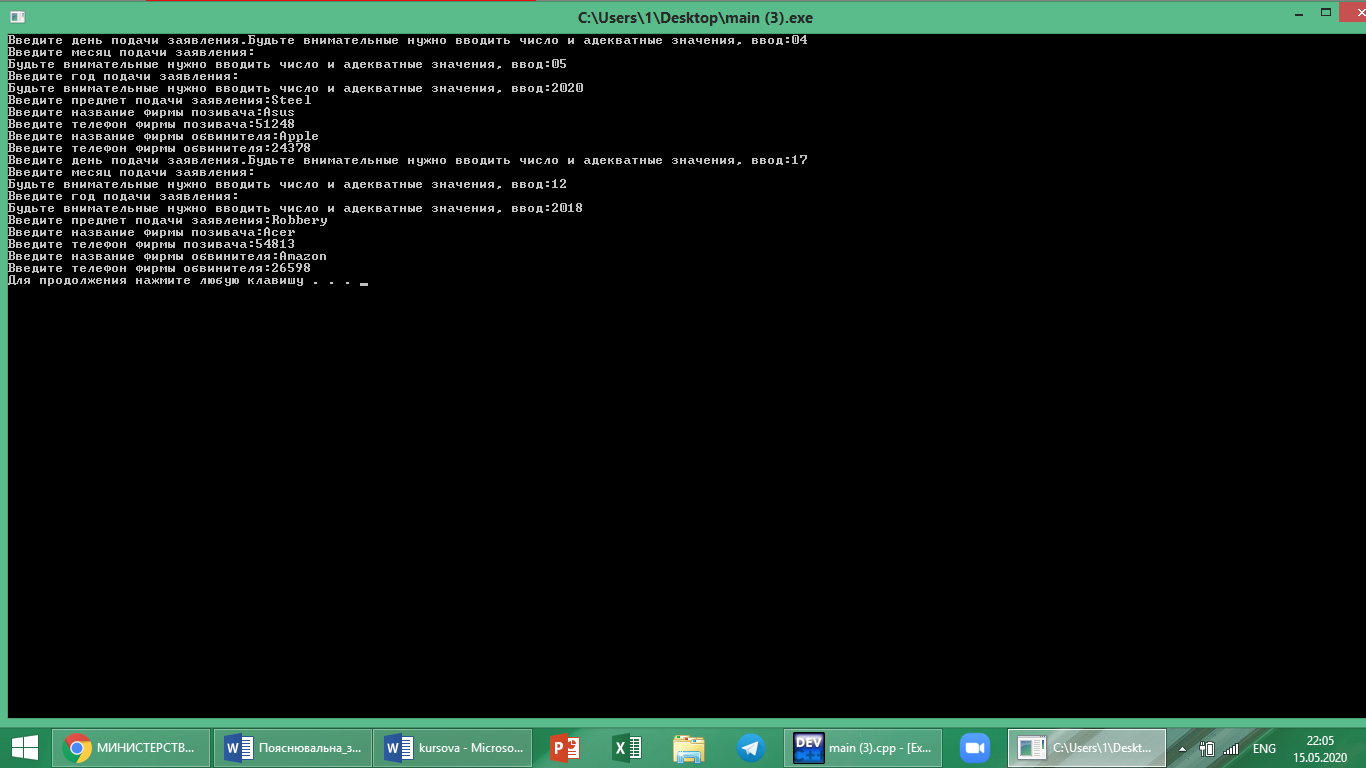


Рисунок 6.3 - Додавання інформації до бази даних

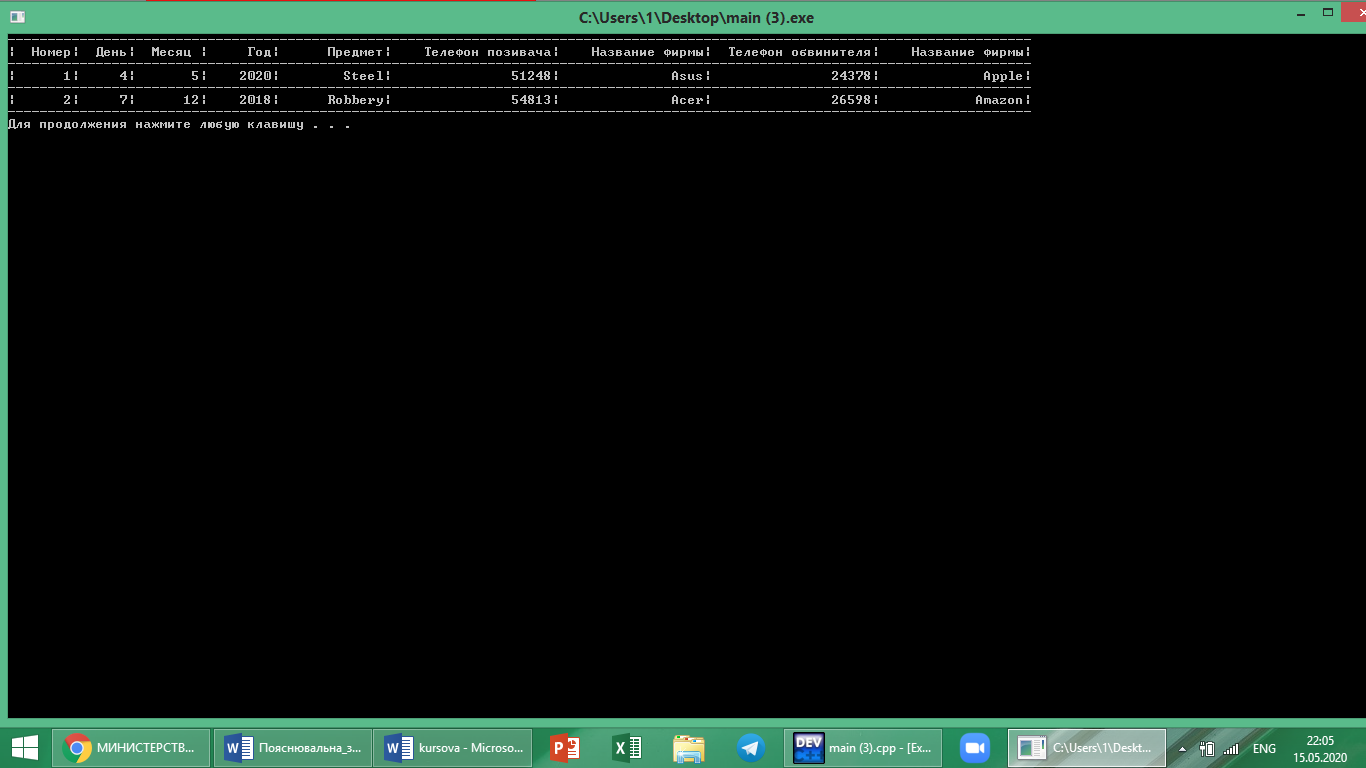


Рисунок 6.4 - Виведення інформації

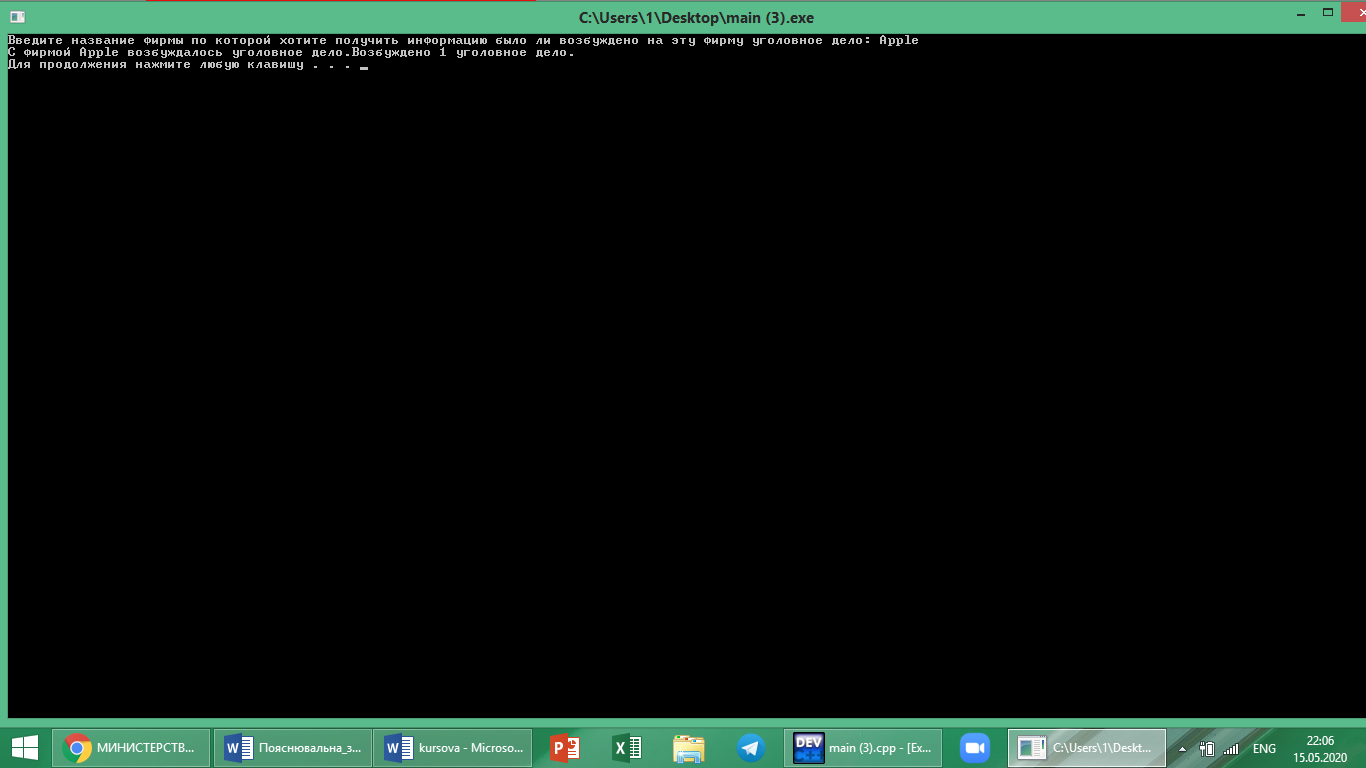


Рисунок 6.5 - Фірма, з якою порушувалася судова справа

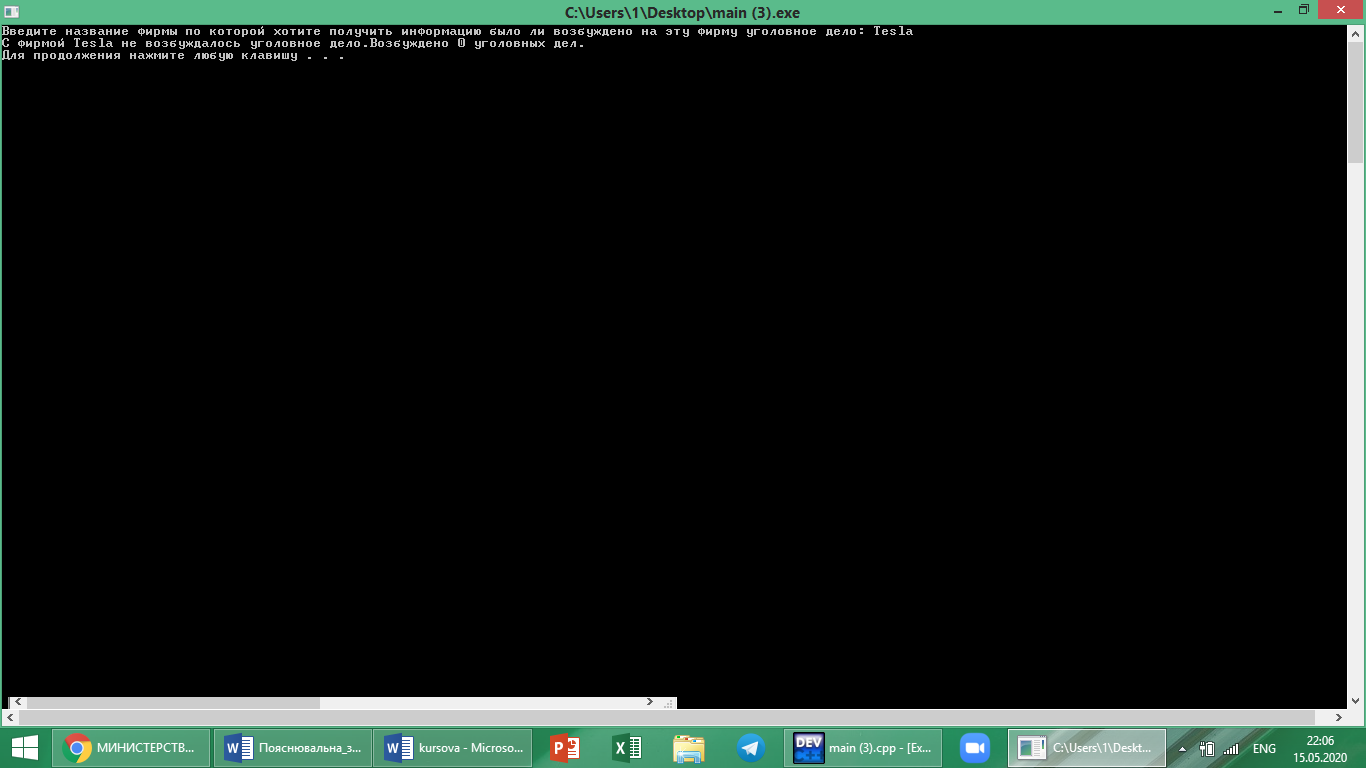


Рисунок 6.6 - Фірма, з якою не порушувалася судова справа

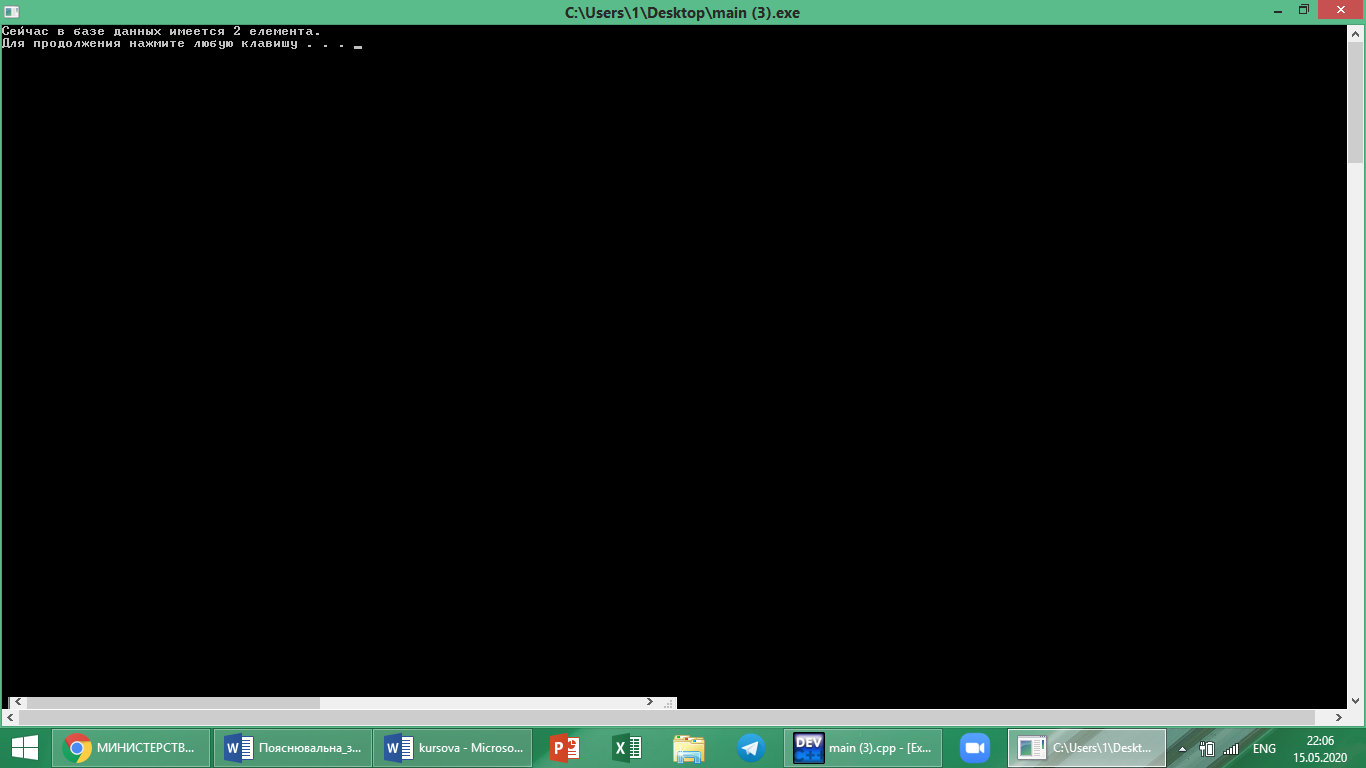


Рисунок 6.7 - Кількість елементів у базі даних

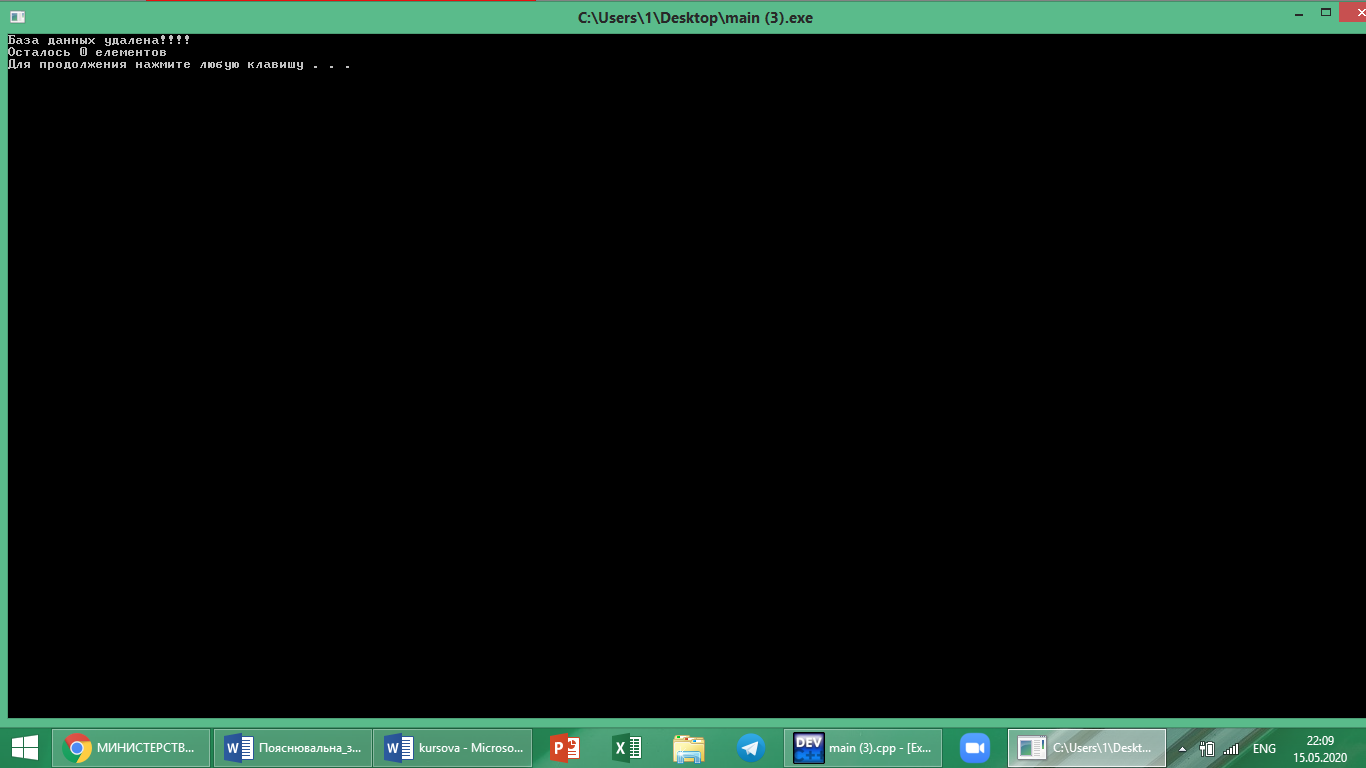


Рисунок 6.8 - Видалення всієї інформації з файлу

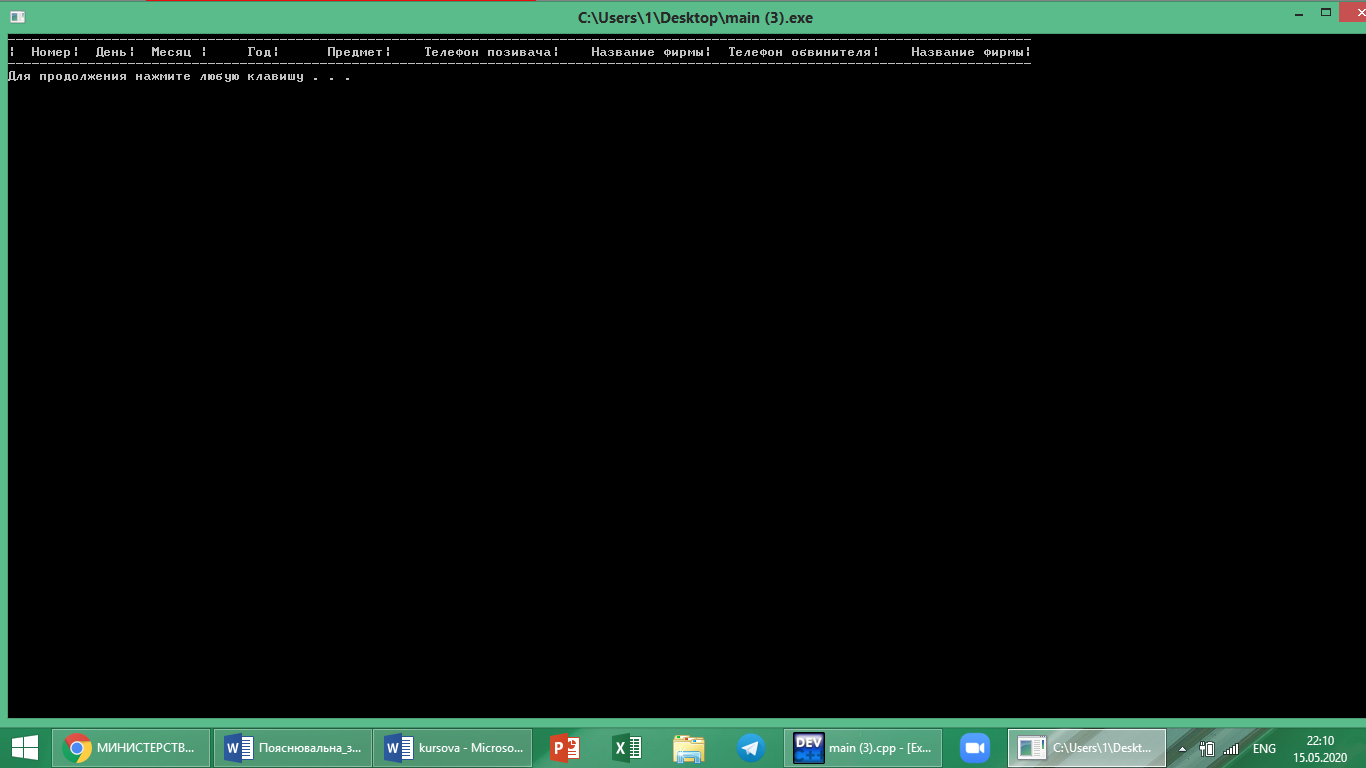


Рисунок 6.9 - Виведення шапки(ключових слів)

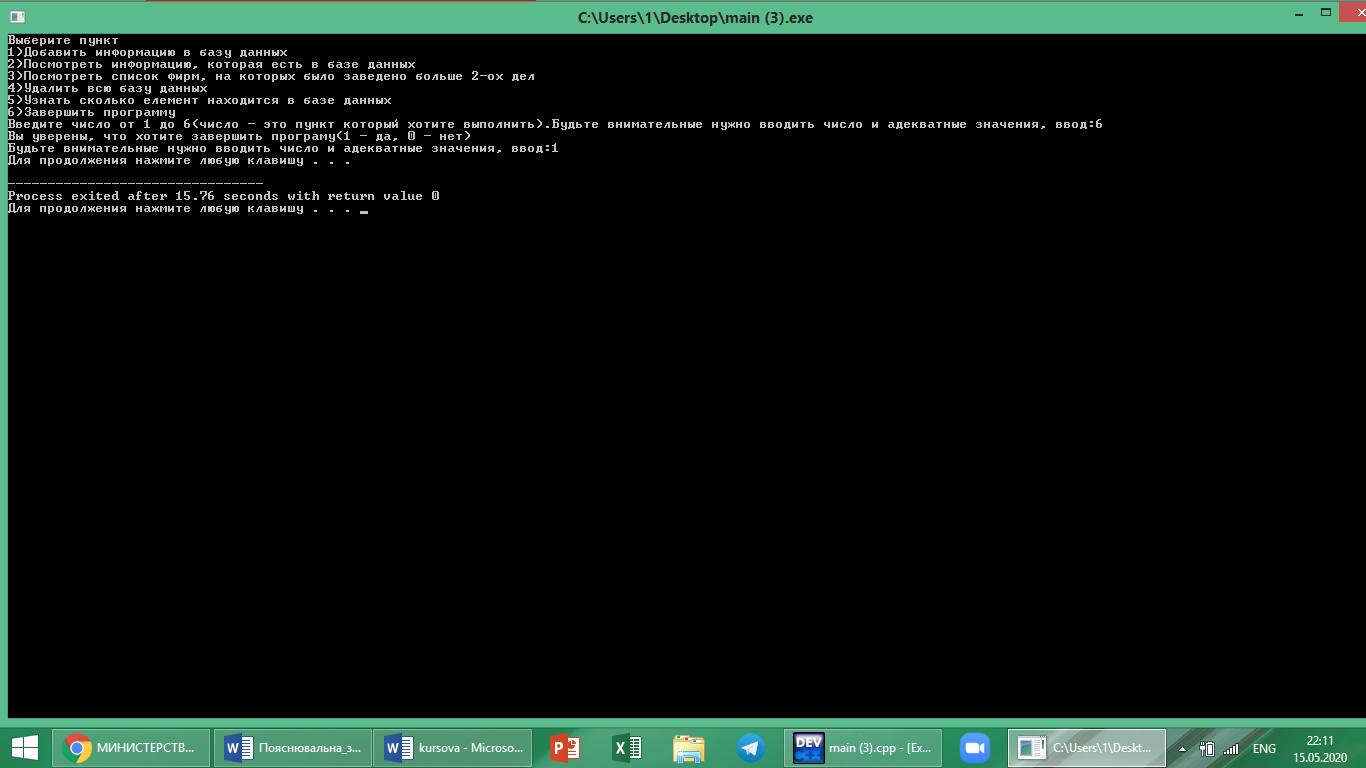


Рисунок 6.10 – Закриття програми

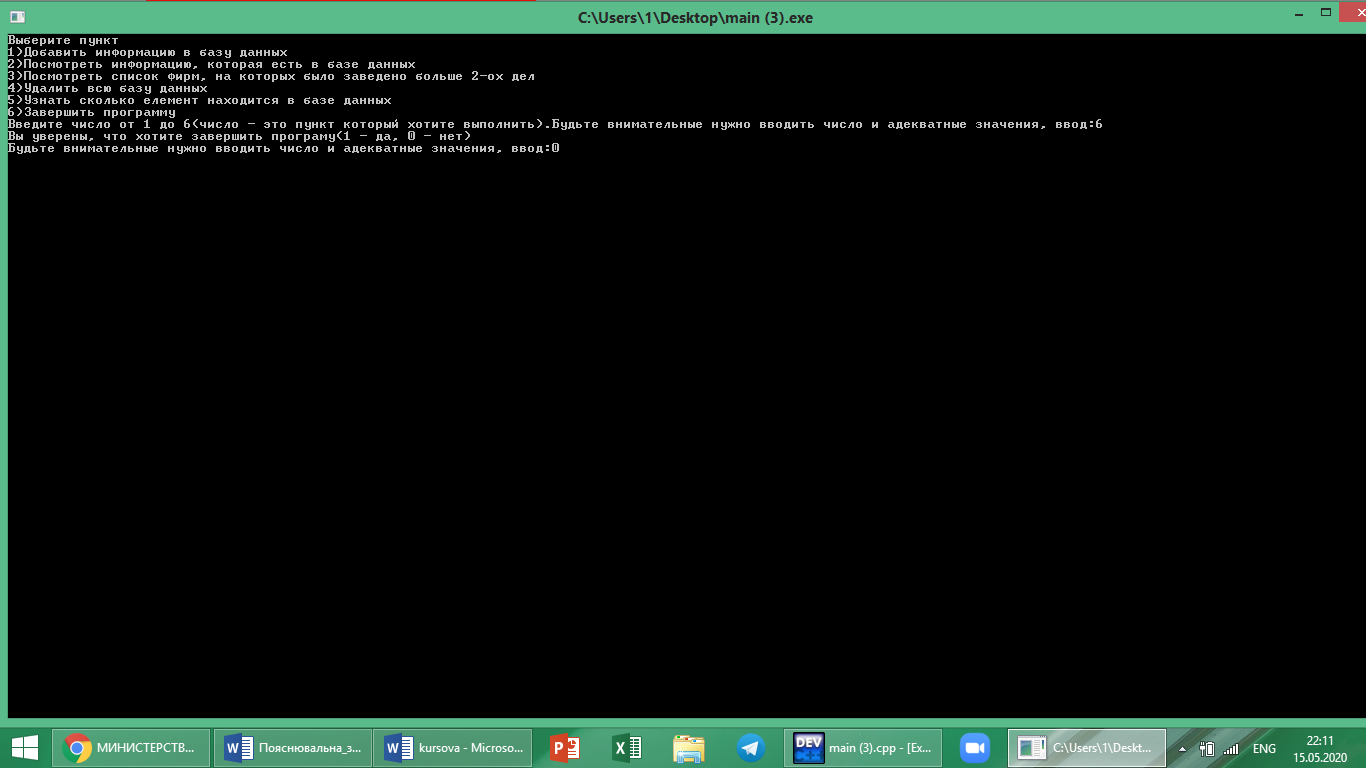


Рисунок 6.11 – Перевірка на закриття програми

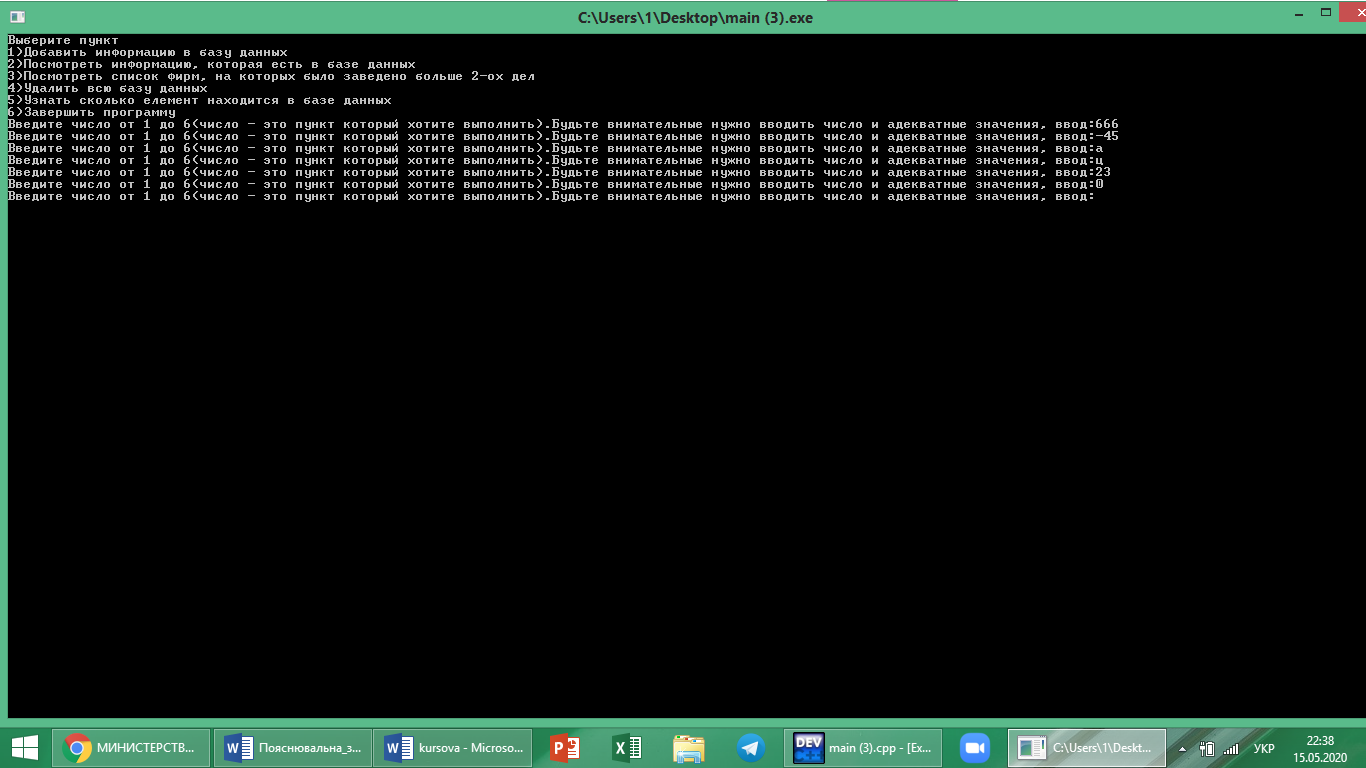


Рисунок 6.12 – Перевірка на некоректне введення пункта меню