Министерство образования Республики Беларусь

Учреждение образования

Брестский государственный технический университет

Кафедра ИИТ

Отчет №12

По лабораторной работе

Тема:«Индексные файлы»

Выполнил:

Студ. гр.ИИ-23

Романюк А.П.

Проверил:

Брест 2023

Вариант №8

Цель работы: изучить принципы программирования с использованием индексных файлов.

Текст задания:

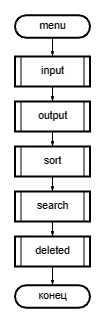
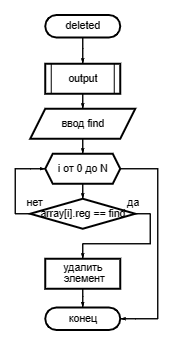
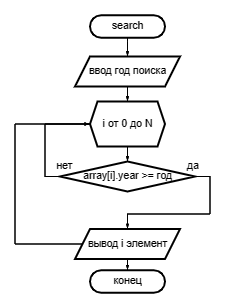
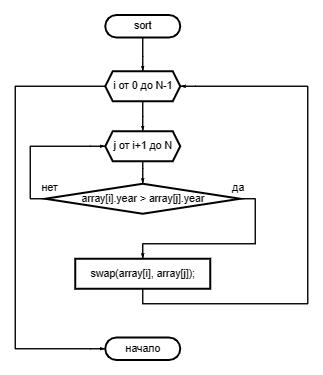
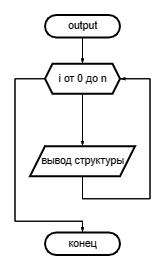
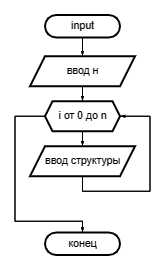
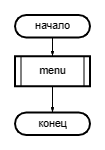
В программу, разработанную в лабораторной работе 11, внести следующие изменения и дополнения:

1. При запуске программы данные читаются в массив структур из файла, при добавлении новой записи в массив структур в файл должна дописываться новая запись, без изменения остальных записей.

2. Все изменения (изменения полей записи, удаление записи) – сохраняются в файле при помощи перезаписи содержимого всего файла.

3. Сортировка должна выполняться по двум полям на выбор при помощи создания индексных файлов. Содержимое индексного файла переписывается в случае изменения значения ключевого поля (поля, по которому выполняется сортировка) или в случае удаления, добавления записей.

Блок-схема:



Текст программы:

#include <iostream>

#include <string>

#include <windows.h>

#include <iomanip>

#include <sstream>

#include <fstream>

using namespace std;

enum Proizv { Belarus = 1, Russia, Lituania, Ukraine };

union Nalichie {

int check;

char access;

};

struct book {

int reg;

char author[80],

name[80],

izdat[80];

int year;

unsigned pages;

Nalichie nal;

Proizv proizv;

};

void index\_file(int N, int\* index);

void writetxt(book\* array, int N);

void writebin(book\* array, int N);

void readtxt(book\*& array, int& N);

void input(book\*& array, int& N)

{

int n;

cout << "Введите количество элементов массива\n";

cin >> n;

N = N + n;

book\* tempArr = new book[N]{};

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

tempArr[i] = array[i];

array = new book[N];

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

array[i] = tempArr[i];

system("cls");

for (int i = N - n; i < N; i++) {

array[i].reg = i + 1;

cout << "1) название книги\n";

cin >> array[i].name;

cout << "2) автор книги\n";

cin >> array[i].author;

cout << "3) название издательства\n";

cin >> array[i].izdat;

cout << "4) год издания книги\n";

int year = 0;

cin >> year;

array[i].year = year;

cout << "5) количество страниц\n";

cin >> array[i].pages;

int choise;

cout << "6) страна поставщик (1 - Беларусь, 2 - Россия, 3 - Латвия, 4 - Украина)\n";

cin >> choise;

array[i].proizv = (Proizv)choise;

cout << "7) есть ли книга в наличии? (1 - да, 2 - нет)\n";

cin >> array[i].nal.check;

if (array[i].nal.check == 1) {

array[i].nal.access = '+';

}

else array[i].nal.access = '-';

system("cls");

}

}

void output(book\* array, int N)

{

ifstream index\_file;

index\_file.open("Indexes.txt");

cout << setw(3) << "№";

cout << setw(20) << "name:";

cout << setw(20) << "author:";

cout << setw(20) << "publisher:";

cout << setw(20) << "year:";

cout << setw(20) << "pages:";

cout << setw(20) << "made in";

cout << setw(20) << "are in stock" << endl;

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

int k;

if (index\_file.is\_open() && !index\_file.eof()) {

index\_file >> k;

index\_file >> k;

}

else k = i;

cout << setw(10) << array[i].reg;

cout << setw(20) << array[i].name;

cout << setw(20) << array[i].author;

cout << setw(20) << array[i].izdat;

cout << setw(20) << array[i].year;

cout << setw(20) << array[i].pages;

switch (array[i].proizv) {

case Belarus:

cout << setw(20) << "Беларусь";

break;

case Russia:

cout << setw(20) << "Россия";

break;

case Lituania:

cout << setw(20) << "Литва";

break;

case Ukraine:

cout << setw(20) << "Украина";

break;

}

cout << setw(20) << array[i].nal.access;

cout << endl;

}

}

void sort(book\*& array, int N, int key) //sort by year

{

cout << "выберите пункт:\n";

cout << "1 - сортировка по названию книги\n";

cout << "2 - сортировка по году выпуска\n";

int choise;

cin >> choise;

int\* index = new int[N];

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

index[i] = i;

}

for (int i = 0; i < N - 1; i++) {

for (int j = i + 1; j < N; j++) {

if (choise == 2) {

if (array[index[i]].year > array[index[j]].year)

swap(index[i], index[j]);

}

if (choise == 1) {

if (strcmp(array[index[j]].name, array[index[i]].name) < 0)

swap(index[i], index[j]);

}

}

}

index\_file(N, index);

if (key == 1) writebin(array, N);

else writetxt(array, N);

readtxt(array, N);

}

void search(book\* array, int N) //search books after X year

{

int fond;

cout << "Введите книги с какого года вам показать?\n";

cin >> fond;

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

if (array[i].year >= fond) {

cout << "№" << array[i].reg << "; ";

cout << "Название: " << array[i].name << "; ";

cout << "Автор: " << array[i].author << "; ";

cout << "Издательство: " << array[i].izdat << "; ";

cout << "Год выпуска: " << array[i].year << "; ";

cout << "Кол-во страниц: " << array[i].pages << "; ";

switch (array[i].proizv) {

case Belarus:

cout << "Поставщик: " << "Беларусь" << "; ";

break;

case Russia:

cout << "Поставщик: " << "Россия" << "; ";

break;

case Lituania:

cout << "Поставщик: " << "Литва" << "; ";

break;

case Ukraine:

cout << "Поставщик: " << "Украина" << "; ";

break;

}

cout << "Есть в наличии: " << array[i].nal.access << "; ";

cout << endl;

}

}

}

void deleted(book\*& array, int& N, int key) {

int find;

output(array, N);

cout << "Введите номер книги которую нужно удалить: ";

cin >> find;

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

if (array[i].reg == find) {

N--;

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

array[j] = array[j + 1];

i--;

}

}

if (key == 1) writebin(array, N);

else writetxt(array, N);

int\* tempArray = new int[N];

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

tempArray[i] = i;

}

index\_file(N, tempArray);

delete[] tempArray;

writetxt(array, N);

output(array, N);

}

void writetxt(book\* array, int N) {

ofstream f1;

ifstream index\_file;

index\_file.open("Indexes.txt");

f1.open("file.txt");

f1 << setw(10) << "№";

f1 << setw(25) << "name:";

f1 << setw(25) << "author:";

f1 << setw(25) << "publisher:";

f1 << setw(25) << "year:";

f1 << setw(25) << "pages:";

f1 << setw(25) << "made in";

f1 << setw(25) << "are in stock" << endl;

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

int k;

if (index\_file.is\_open() && !index\_file.eof()) {

index\_file >> k;

index\_file >> k;

}

else k = i;

f1 << setw(10) << array[k].reg;

f1 << setw(25) << array[k].name;

f1 << setw(25) << array[k].author;

f1 << setw(25) << array[k].izdat;

f1 << setw(25) << array[k].year;

f1 << setw(25) << array[k].pages;

switch (array[k].proizv) {

case Belarus:

f1 << setw(25) << "Беларусь";

break;

case Russia:

f1 << setw(25) << "Россия";

break;

case Lituania:

f1 << setw(25) << "Литва";

break;

case Ukraine:

f1 << setw(25) << "Украина";

break;

}

f1 << setw(25) << array[k].nal.access;

f1 << endl;

}

f1.close();

}

void readtxt(book\*& array, int& N) {

N = 0;

ifstream f1;

string s;

char temp[1024];

f1.open("file.txt");

if (!f1.is\_open()) {

input(array, N);

writetxt(array, N);

}

else {

f1.getline(temp, 1024, '\n');

for (int i = 0; !f1.eof(); i++) {

N++;

book\* buf = new book[N - 1];

for (int j = 0; j < N - 1; j++) {

buf[j] = array[j];

}

array = new book[N];

for (int j = 0; j < N - 1; j++) {

array[j] = buf[j];

}

f1 >> array[i].reg >> array[i].name >> array[i].author >> array[i].izdat >> array[i].year >> array[i].pages;

char buf1[20];

f1 >> buf1;

if (strcmp(buf1, "Беларусь") == NULL) {

array[i].proizv = Belarus;

}

else if (strcmp(buf1, "Россия") == NULL) {

array[i].proizv = Russia;

}

else if (strcmp(buf1, "Литва") == NULL) {

array[i].proizv = Lituania;

}

else if (strcmp(buf1, "Украина") == NULL) {

array[i].proizv = Ukraine;

}

f1 >> array[i].nal.access;

}

N--;

f1.close();

}

}

void writebin(book\* array, int N) {

ofstream f1;

f1.open("file.bin", ios\_base::binary);

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

f1.write((char\*)&array[i], sizeof(array[i]));

}

f1.close();

}

void readbin(book\*& array, int& N) {

ifstream f1;

f1.open("file.bin", ios\_base::binary);

N = 0;

if (!f1.is\_open()) {

input(array, N);

writebin(array, N);

}

else {

for (int i = 0; !f1.eof() && (f1.peek() != std::ifstream::traits\_type::eof()); i++) {

N++;

book\* tempArr = new book[N - 1];

for (int j = 0; j < N - 1; j++) {

tempArr[j] = array[j];

}

array = new book[N];

for (int j = 0; j < N - 1; j++) {

array[j] = tempArr[j];

}

f1.read((char\*)&array[i], sizeof(array[i]));

}

cout << "Данные считаны из файла.\n";

f1.close();

}

}

void index\_file(int N, int\* index) {

ofstream fout("Indexes.txt");

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

fout << i << " " << index[i];

if (i < N - 1) {

fout << endl;

}

}

fout.close();

}

int menu()

{

int key = 2;

/\*while (key != 1 && key != 2) {

cout << "выберите тип файла (1.bin 2.txt)";

cin >> key;

}\*/

int choise = 0, N = 0;

book\* array = new book[N];

do {

cout << "Выберите действие:\n";

cout << "1) Ввод массива структур\n";

cout << "2) Вывод массива структур\n";

cout << "3) Сортировка книг \n";

cout << "4) Вывести список книг с фамилиями авторов, изданных после заданного года.\n";

cout << "5) Удаление заданной структуры\n";

cout << "6) Добавление записи\n";

cout << "7) Выход\n";

cin >> choise;

system("cls");

switch (choise) {

case 1:

if (key == 1) readbin(array, N);

else readtxt(array, N);

break;

case 2:

output(array, N);

break;

case 3:

sort(array, N, key);

system("cls");

output(array, N);

break;

case 4:

search(array, N);

break;

case 5:

deleted(array, N, key);

break;

case 6:

input(array, N);

if (key == 1) writebin(array, N);

else writetxt(array, N);

break;

case 7:

return 0;

break;

}

} while (true);

}

int main()

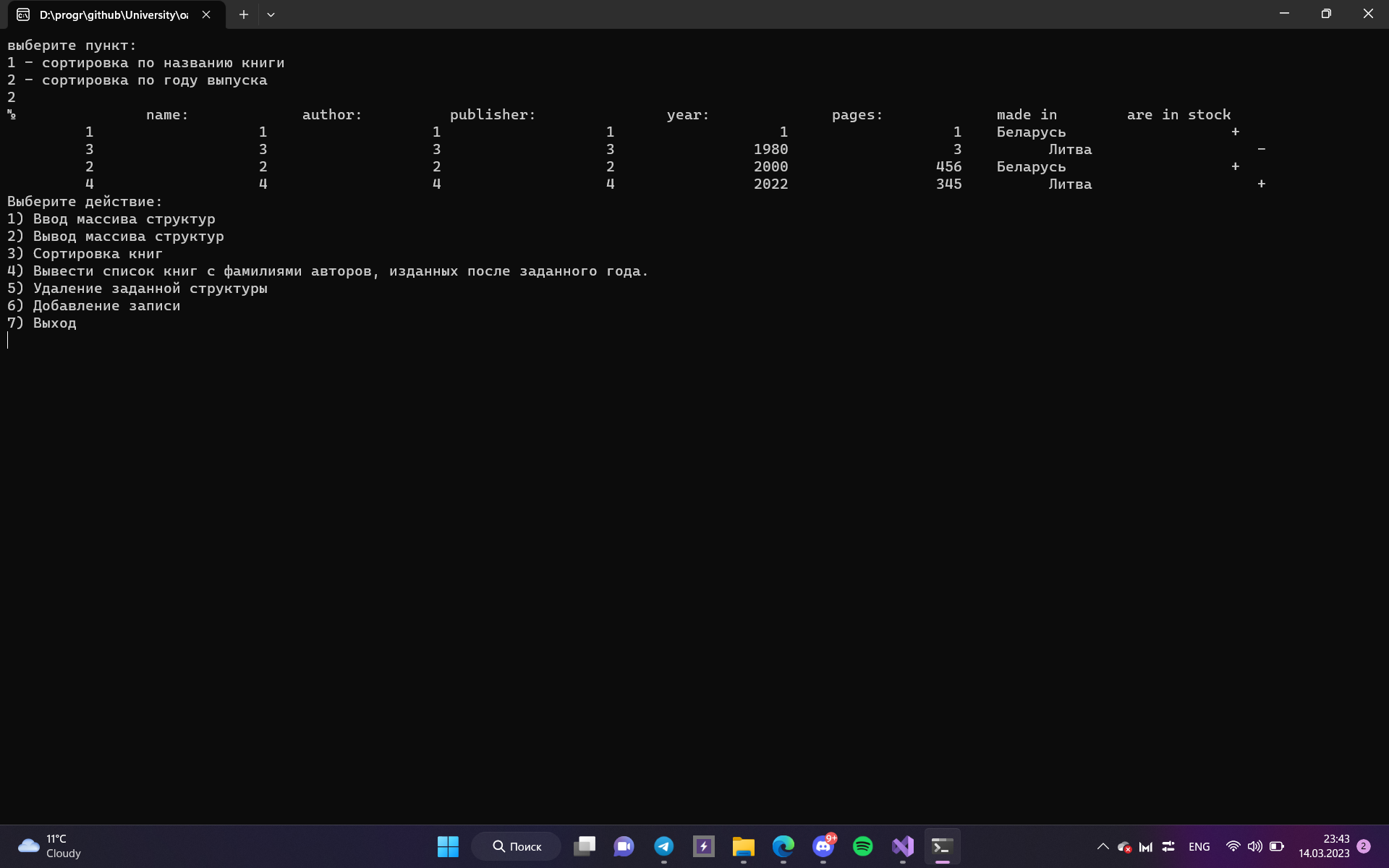
{

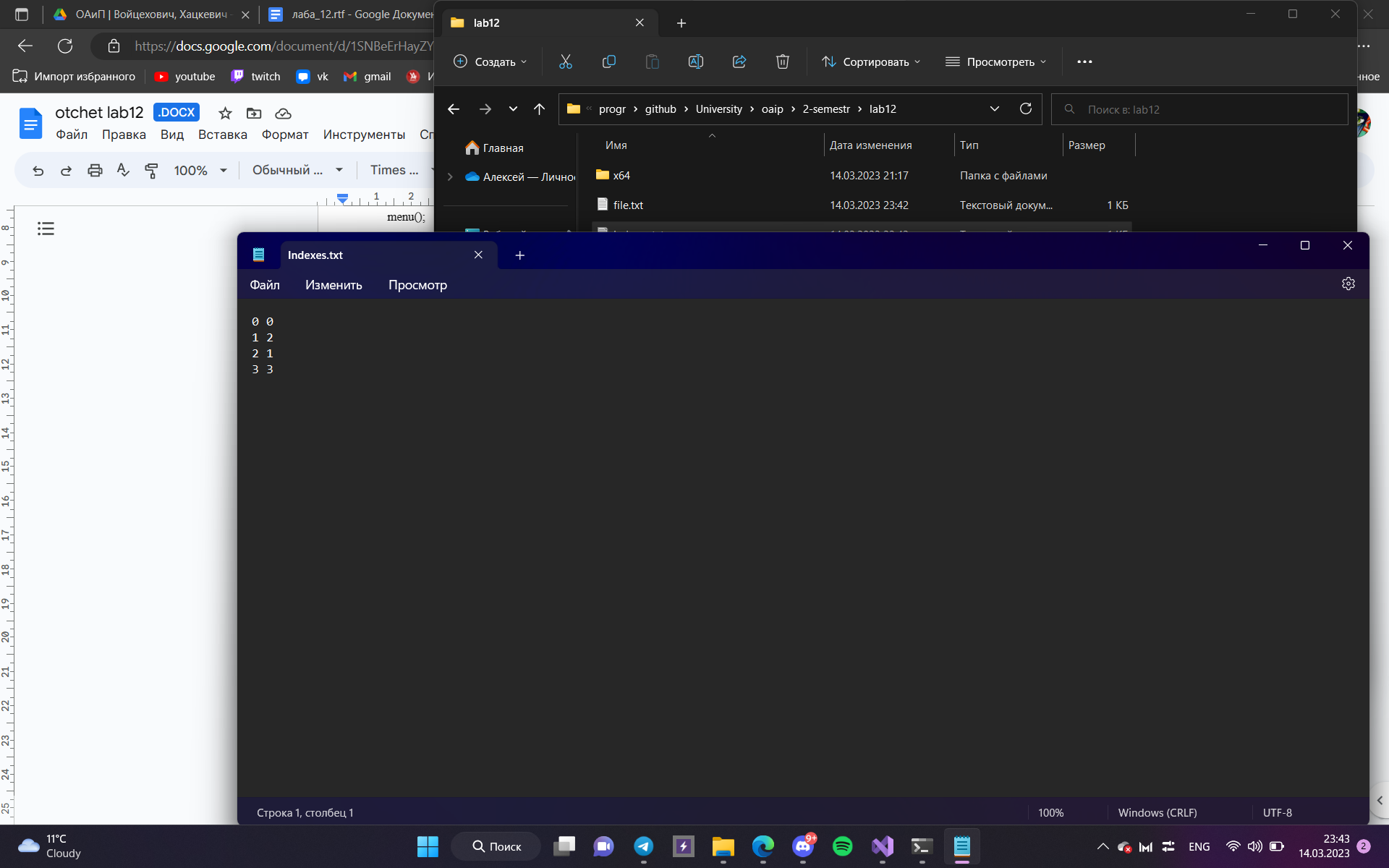
menu();

return 0;

}

Результаты выполнения программы:





Вывод: изучил принципы программирования с использованием индексных файлов.