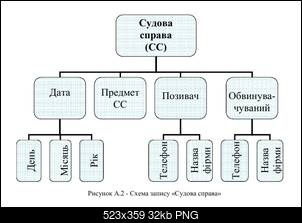
омогите с написанием программы и с поиском ошибки.

Надо создать базу данных, структура изображена на картинке. Первый раз пишу программу такого размера. Большую часть программы я уже написал, но она работает не корректно, а именно - не работает поиск. Помогите найти ошибки и дописать последнюю функцию, в которой надо определить фирму которая чаще всего , за эти годы, выступала в роли истца.

Всем привет!   
Помогите с написанием программы и с поиском ошибки.   
Надо создать базу данных, структура изображена на картинке. Первый раз пишу программу такого размера. Большую часть программы я уже написал, но она работает не корректно, а именно - не работает поиск. Помогите найти ошибки и дописать последнюю функцию, в которой надо определить фирму которая чаще всего , за эти годы, выступала в роли истца.   
  
   
  
Код программы: 

| **C++**[Выделить код](http://www.cyberforum.ru/) | |
| --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171 | #include <iostream>  #include <stdlib.h>  #include <fstream>  #include <cstdio>  #include <string.h>  #include <stdio.h>  #include <cstdlib>  #include <cstring>    using namespace std;    struct DATA{   int day;   int month;   int year;  };    struct vidp {   int telefon;   char nazvFirm[15];   };    struct sprava{   DATA data;   char predmet[20];   vidp v;   vidp obv;  };    sprava\* array\_of\_sprava=new sprava[50];  FILE\* Fs;  int nzap=0;    int i=0,h=0,g=0;      int reading()//чтение;  {  if ((Fs=fopen("my.dat","rb"))==NULL)      {            cout<<("Ne mogy otrit fail**\n**");            return 0; }     do         {           fread(&array\_of\_sprava[nzap],sizeof(sprava),1,Fs);           nzap++;           if (feof(Fs)) break;           } while(1);        nzap--;        cout<<"Schitivanie zaversheno**\n**";      fclose(Fs); // Закрываем файл      system("pause");      return 0;  }    int outp\_to\_file(){ // Вывод в файл  FILE\* Fs;     if ((Fs=fopen("my.dat","wb"))==NULL)      {       cout<<("Fail ne sozdan!**\n**");       return 0;      }     for (int i=0;i<nzap;i++)       fwrite(&array\_of\_sprava[i],sizeof(sprava),1,Fs);     cout<<("Fail zapisan");      fclose(Fs);      system("pause");  }        int input\_in\_array(){//ввод в масив;  int q=1;  while (q!=0)  {  cout<<"**\n**Vvedite den dela: ";  cin>>array\_of\_sprava[nzap].data.day;  cout<<"**\n**Vvedite mesyac dela: ";  cin>>array\_of\_sprava[nzap].data.month;  cout<<"**\n**Vvedite god dela: ";  cin>>array\_of\_sprava[nzap].data.year;  cout<<"**\n**Vvedite predmet dela: ";  cin>>array\_of\_sprava[nzap].predmet;  cout<<"**\n**Vvedite telefon istca: ";  cin>>array\_of\_sprava[nzap].v.telefon;  cout<<"**\n**Vvedite firmy istca: ";  cin>>array\_of\_sprava[nzap].v.nazvFirm;  cout<<"**\n**Vvedite telefon obvinyvachovanogo : ";  cin>>array\_of\_sprava[nzap].obv.telefon;  cout<<"**\n**Vvedite firmy obvinyvachovanogo  : ";  cin>>array\_of\_sprava[nzap].obv.nazvFirm;  nzap++;  cout<<"**\n**Vvedite 0 dlya okonchaniya zapisi ili luboe chislo dlya prodolzheniya: ";  cin>>q;  cout<<"**\n**";    }  }    int VivEkran()// Вывод на экран  {system("cls");      for(int i=0;i<nzap;i++){    cout<<"Zapis #"<<i+1<<"**\n**";  cout<<"\ den dela: "<<array\_of\_sprava[i].data.day<<"**\n**";  cout<<"\ mesyac dela: "<<array\_of\_sprava[i].data.month<<"**\n**";  cout<<"\ god dela: "<<array\_of\_sprava[i].data.year<<"**\n**";  cout<<"\ predmet dela: "<<array\_of\_sprava[i].predmet<<"**\n**";  cout<<"\ telefon istca: "<<array\_of\_sprava[i].v.telefon<<"**\n**";  cout<<"\ firmy istca: "<<array\_of\_sprava[i].v.nazvFirm<<"**\n**";  cout<<"\ telefon obvinyvachovanogo : "<<array\_of\_sprava[i].obv.telefon<<"**\n**";  cout<<"\ firmy obvinyvachovanogo  : "<<array\_of\_sprava[i].obv.nazvFirm<<"**\n**";     system("pause");  return 0;      }}              int poisk() // поиск  { char f[20];  cout<<"Vvedite god dela: ";  cin>>f;  int k=0;  for (int i=0;i<nzap;i++)  if (array\_of\_sprava[i].data.year) {  cout<<"Dannih za etot god ne sushistvyet**\n**";  k++;  }  if(k==0)cout<<"Dannie za etot god ne naideni**\n**";    system("pause");    return 0;  }        int main()// меню  {      int rez=1;      while(rez!=0)  {     system("cls");     cout<<"Gaidyk Artyr RPZ-14 Variant №2**\n**";     cout<<"Menu**\n**";     cout<<"1 - Dobavit dannie;**\n**";     cout<<"2 - Vvod dannih v fail;**\n**";     cout<<"3 - Shitat dannie s faila;**\n**";     cout<<"4 - Vivisti na ekran vsi dannie;**\n**";     cout<<"5 - Poisk po gody;**\n**";     cout<<"6 - Firma kotoraya chahe bila istsom;**\n**";     cout<<"0 - Vihod;**\n**";     cout<<nzap<<":";     cin>>rez;  switch(rez)  {   case 1:input\_in\_array();break;   case 2:outp\_to\_file();break;   case 3:reading();break;   case 4:VivEkran();break;   case 5:poisk();break;   // case 6:Firma kotoraya chahe bila istsom();break;   case 0: outp\_to\_file();exit(0);break;   default :break;  }}  } | |  |

Задача: разработать на C++ справочно-информационную систему, в базу данных которой (в файле) данные записываются с уникальным идентификационным номером (ID). Вариант с ручным внесением идентификационного номера исключается как неуважительный к пользователю. Да и запутаться легко, внося ID вручную.

Сложность задачи вызвана тем, что данные из базы могут быть удалены, а идентификационные номера сохранённых записей остаются прежними. Поэтому идентификационный номер каждой новой записи должен представлять собой не увеличенное на единицу число элементов массива, а увеличенный на единицу идентификационный номер последней записи данной структуры.

Пусть требуется разработать справочно-информационную систему библиотеки. Создаём структуры с данными о книгах в фонде, книгах в выдаче, читателях и библиотекарях. Данные структуры должны записываться в файл, в каждой структуре есть данные идентификационного номера (ID). Есть два уровня пользователя. Первый - читатель, который может просматривать данные о книгах в фонде. Второй - библиотекарь, он же администратор, который может просматривать все данные, а также вносить новые данные и редактировать уже содержащиеся в базе.

Рассмотрим подробнее, как реализуется запись даннных с уникальным идентификационным номером. Наиболее важны два момента.

1. Происходит открытие файла данных о читателях. Данные считываются из файла. Из вновь созданного массива принимаются данные о номере идентификатора читалеля. Если этот номер превышает количество элементов (записях о читателях), то данные о новом читателе в базу данных записываются с номером, на 1 больше последнего номера из базы данных. Если не превышает, то номер присваивается на единицу больше числа элементов массива (читателей).

Код C++

Reader\* reader\_arr = new Reader[n];

FILE \* ff;

int b = 0;

int c = 0;

fstream fs;

fs.open ("Reader.dat");

if (fs.is\_open())

{

ff = fopen("Reader.dat","rb");

int fsize;

fseek(ff,0,SEEK\_END);

fsize = ftell(ff);

b = fsize / sizeof(Reader);

Reader\* start\_arr2 = new Reader[b]; //создаётся новый массив из данных структуры

fseek(ff,0,SEEK\_SET);

fread(start\_arr2,sizeof(Reader),b,ff);

c = start\_arr2[b - 1].id + 1; //присваиваются данные об ID последней записи масссива

fclose( ff);

fs.close();

}

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

{

if (c >= b)

reader\_arr[i].id = c + i;

else

reader\_arr[i].id = b + i;

cout << "Enter name and surname: "; //далее записываются все

cin.getline(reader\_arr[i].name\_surname, 30); //необходимые данные о читателях

cout << "Enter day of birth: ";

cin >> reader\_arr[i].birth\_date.date;

cout << "Enter month of birth: ";

cin >> reader\_arr[i].birth\_date.month;

cout << "Enter year of birth: ";

cin >> reader\_arr[i].birth\_date.year;

cin.get();

cout << "Enter profession: ";

cin.getline(reader\_arr[i].profession, 20);

cout << "Enter education: ";

cin.getline(reader\_arr[i].education, 16);

cout << "Enter address: ";

cin.getline(reader\_arr[i].address, 50);

cout << "Enter phone: ";

cin >> reader\_arr[i].phone;

cin.get();

cout << endl;

}

ff = fopen("Reader.dat","a");

fwrite(reader\_arr,sizeof(Reader),n,ff); //все внесённые данные записываются

fclose( ff);

2. Точно так же, как данные о читателях, в базу данных записываются данные о библиотекарях. Подводный камень нашей программы кроется в том, что библиотекари имеют доступ к информационной системе с паролем. При выборе библиотекарем пункта меню, извещающего программу о том, что он хочет войти в систему как библиотекарь, открывается файл данных о библиотекарях. И тут... Если библиотекарь в течении сессии будет редактировать данные о библиотекарях (так же, как и о читателях), то файл, открытый при входе в систему, будет продолжать оставаться открытым и данные в базу не запишутся. Для этого предусмотрено закрытие файла данных о библиотекарях после прочтения имени пользователя и пароля вошедшего в систему.

Код C++

Reader\* reader\_arr = new Reader[n];

FILE \* ff;

int b = 0;

int c = 0;

fstream fs;

fs.open ("Reader.dat");

if (fs.is\_open())

{

ff = fopen("Reader.dat","rb");

int fsize;

fseek(ff,0,SEEK\_END);

fsize = ftell(ff);

b = fsize / sizeof(Reader);

Reader\* start\_arr2 = new Reader[b]; //создаётся новый массив из данных структуры

fseek(ff,0,SEEK\_SET);

fread(start\_arr2,sizeof(Reader),b,ff);

c = start\_arr2[b - 1].id + 1; //присваиваются данные об ID последней записи масссива

fclose( ff);

fs.close();

}

for (int i = 0; i < n; i++)

{

if (c >= b)

reader\_arr[i].id = c + i;

else

reader\_arr[i].id = b + i;

cout << "Enter name and surname: "; //далее записываются все

cin.getline(reader\_arr[i].name\_surname, 30); //необходимые данные о читателях

cout << "Enter day of birth: ";

cin >> reader\_arr[i].birth\_date.date;

cout << "Enter month of birth: ";

cin >> reader\_arr[i].birth\_date.month;

cout << "Enter year of birth: ";

cin >> reader\_arr[i].birth\_date.year;

cin.get();

cout << "Enter profession: ";

cin.getline(reader\_arr[i].profession, 20);

cout << "Enter education: ";

cin.getline(reader\_arr[i].education, 16);

cout << "Enter address: ";

cin.getline(reader\_arr[i].address, 50);

cout << "Enter phone: ";

cin >> reader\_arr[i].phone;

cin.get();

cout << endl;

}

ff = fopen("Reader.dat","a");

fwrite(reader\_arr,sizeof(Reader),n,ff); //все внесённые данные записываются

fclose( ff);

Точно так же, как данные о читателях, в базу данных записываются данные о библиотекарях. Подводный камень нашей программы кроется в том, что библиотекари имеют доступ к информационной системе с паролем. При выборе библиотекарем пункта меню, извещающего программу о том, что он хочет войти в систему как библиотекарь, открывается файл данных о библиотекарях. И тут... Если библиотекарь в течении сессии будет редактировать данные о библиотекарях (так же, как и о читателях), то файл, открытый при входе в систему, будет продолжать оставаться открытым и данные в базу не запишутся. Для этого предусмотрено закрытие файла данных о библиотекарях после прочтения имени пользователя и пароля вошедшего в систему.

**Код C++**

if (t == 2)

{

int b;

int count1 = 0;

int count2 = 0;

char Slogin[16];

char Spassw[16];

cout << "Enter login: ";

cin >> Slogin;

cin.get();

FILE \* fn;

fn = fopen("Lib.dat","rb");

int fsize;

fseek(fn,0,SEEK\_END);

fsize = ftell(fn);

b = fsize / sizeof(Librarian);

Librarian\* librarian\_arr2 = new Librarian[b];

fseek(fn,0,SEEK\_SET);

fread(librarian\_arr2,sizeof(Librarian),b,fn);

fclose(fn);

for (int i = 0; i < b; i++)

{

while (strstr(librarian\_arr2[i].login, Slogin))

{

cout << "Enter password: ";

cin >> Spassw;

while (strstr(librarian\_arr2[i].password, Spassw))

{

menu();

count2++;

break;

}

count1++;

break;

}

}

Внимание! Закрытие файла с данными о библиотекарях должно произойти сразу же после считывания массива данных о библиотекарях. Если задать закрыте ниже, например, после кода, отвечающего за ввод и приём данных логина и пароля, то файл уже при внесении или редактировании данных о библиотекарях будет открытым и новые данные в него не запишутся.

if (t == 2)

{

int b;

int count1 = 0;

int count2 = 0;

char Slogin[16];

char Spassw[16];

cout << "Enter login: ";

cin >> Slogin;

cin.get();

FILE \* fn;

fn = fopen("Lib.dat","rb");

int fsize;

fseek(fn,0,SEEK\_END);

fsize = ftell(fn);

b = fsize / sizeof(Librarian);

Librarian\* librarian\_arr2 = new Librarian[b];

fseek(fn,0,SEEK\_SET);

fread(librarian\_arr2,sizeof(Librarian),b,fn);

fclose(fn);

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

{

while (strstr(librarian\_arr2[i].login, Slogin))

{

cout << "Enter password: ";

cin >> Spassw;

while (strstr(librarian\_arr2[i].password, Spassw))

{

menu();

count2++;

break;

}

count1++;

break;

}

}