太原工业学院

计算机工程系

数据结构课程设计

设计题目：考试报名管理系统

专　　业： 软件工程

班　　级： 1820562

学　　号： 182056241

姓　　名： 王浪平

指导教师： 刘海静

2019 年 12 月 25 日

目录

一、 课程设计题目

考试报名管理系统

二、 需求分析

此程序是一个简单的可以用于考生报名的信息管理系统，面向群体包括报名的考生和考生信息的管理者两类人。对于考生来说，他们可以根据这个程序的提示来进行不同科目考试的报名，不需要其他人的指导，根据本程序的界面信息选择对应功能即可完成报名，并且自动生成考生的考号，除此之外，程序还提供了验证功能，即在考生报完名之后，考生可以通过查询身份证号码来确认自己的信息是否被录入到报名系统之中。

而对于考生信息的管理者来说，他们也同样可以是本程序的使用者，在界面上他们可以选择专属于信息管理者的第三个功能：对已经报名的信息进行相应的管理。考虑到信息安全的原因，在使用这个功能时，需要输入正确的管理密码才能进入，设置的初始密码是“pass”，密码可以根据后续需要在程序中做修改。目前可以对信息做的管理操作有：查看、编辑、查找、删除已录入信息、将已录入信息保存到文件中、从文件中读取已保存信息、对所有信息按照年龄排序查看。

三、 测试数据

考号	姓名	性别	年龄	身份证号	考试科目
191201	夏洛	男	33	123456789987654321	数学
191202	马冬梅	女	29	987654321123456789	语文

四、 概要设计

1. 元素类型、结点类型和指针类型：

```
typedef struct
{
    string name, gender, ID, number, subject;        //姓名、性别、身份证号、考号、考试科目
    int age;                                         //年龄
}Test_Stu;            /* 考生类的定义 */

typedef struct {
```

```
    Test_Stu* elem;
    int length;                 //顺序表当前长度
      }SqList;
```

2. 建立一个最大长度为 MaxSize + 1 的顺序表来存储信息，下标为 0 的元素为作为程序执行查找时的“哨兵”，不存放任何元素，因此第一个元素的插入是从下标为 1 的元素开始的。

3. 主要函数：

```
Status InitList(SqList&);                           //顺序表初始化
bool EmptyList(SqList&);                            //判断顺序表是否为空
istream& operator>>(istream& cin, Test_Stu& L);     //重载流插入运算符
ostream& operator<<(ostream& cout, Test_Stu& L);    //重载流提取运算符
Status InsertList(SqList& L);                       //往顺序表中插入成员

void InsertList(SqList& L, Test_Stu s);             //用于往其中插入示例信息
Status Verify(SqList& L);                           //验证考生是否报名

Status Search_number(SqList& L);                    //根据身份证号查找
Status Sort_age(SqList& L);                         //根据年龄排序
Status Delete(SqList& L);                           //删除成员
Status Edit(SqList& L);                             //编辑成员信息
Status SaveFile(SqList& L);                         //将所有信息保存到文件中
void ReadFile(SqList& L);                           //从文件中读取信息
//void Statistic(SqList& L);                        //统计各科报名人数
Status Display(SqList& L);                          //输出全部信息
void MainMenu(SqList& L);                           //显示主界面的菜单
void ManagerMenu(SqList& L);                        //适合管理者操作的第二级菜单
void main()
{
    SqList L;    InitList(L);
    Test_Stu stu_1, stu_2;
    stu_1.age = 33;   stu_1.gender = "男";   stu_1.ID = "123456789987654321";
    stu_1.name = "夏洛";   stu_1.subject = "数学";        stu_1.number = "191201";
    stu_2.age = 29;   stu_2.gender = "女";   stu_2.ID = "987654321123456789";
    stu_2.name = "马冬梅";stu_2.subject = "语文";        stu_2.number = "191202";
    InsertList(L, stu_1);          InsertList(L, stu_2);      //往其中添加的示例信
息
    while(1)
        MainMenu(L);
}
```

五、 调用关系图

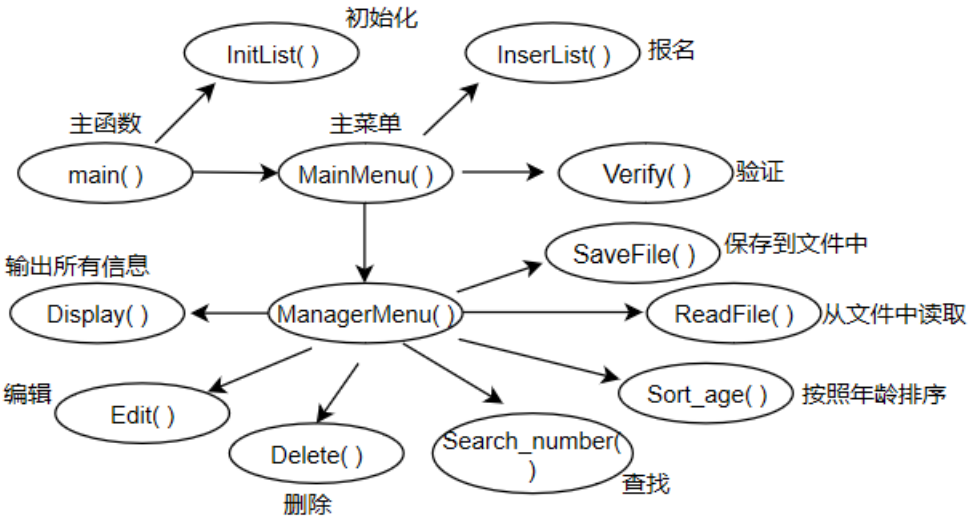

图 1

六、 程序代码

程序总共分为三个部分：Test_Manager.h（定义部分）、Test_Manager.cpp（实现部分）、main.cpp（调用部分）

1. Test_Manager.h（定义部分）

```
#pragma once
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
#define Status int
#define OK 1
#define ERROR 0
#define Maxsize 1000        //设置顺序表中的最大考生数为1000人

typedef struct
{
    string name, gender, ID, number, subject;        //姓名、性别、身份证号、考号、考试科目
    int age;                                         //年龄
}Test_Stu;            /* 考生类的定义 */

typedef struct {
    Test_Stu* elem;
    int length;                //顺序表当前长度
}SqList;

Status InitList(SqList&);                            //顺序表初始化
bool EmptyList(SqList&);                             //判断顺序表是否为空
```

```
istream& operator>>(istream& cin, Test_Stu& L);       //重载流插入运算符
ostream& operator<<(ostream& cout, Test_Stu& L); //重载流提取运算符
Status InsertList(SqList& L);                         //往顺序表中插入成员

void InsertList(SqList& L, Test_Stu);                 //用于往其中插入示例信息
Status Verify(SqList& L);                             //验证考生是否报名

Status Search_number(SqList& L);                      //根据身份证号查找
Status Sort_age(SqList& L);                               //根据年龄排序
Status Delete(SqList& L);                             //删除成员
Status Edit(SqList& L);                                   //编辑成员信息
Status SaveFile(SqList& L);                               //将所有信息保存到文件中
void ReadFile(SqList& L);                             //从文件中读取信息
//void Statistic(SqList& L);                          //统计各科报名人数
Status Display(SqList& L);                            //输出全部信息
void MainMenu(SqList& L);                             //显示主界面的菜单
void ManagerMenu(SqList& L);                          //适合管理者操作的第二级菜单
```

2. Test_Manager.cpp（实现部分）

```
#include "Test_Manager.h"

Status InitList(SqList& L)                          /* 顺序表初始化 */
{
    L.elem = new Test_Stu[Maxsize + 1];   /* 给顺序表分配大小为MaxSize+1
                                                的空间,下标为0的元素作为哨兵 */
    L.length = 0;                         //初始化时顺序表当前长度为0
    return OK;
}

bool EmptyList(SqList& L)                           /* 判断顺序表是否为空 */
{
    if (L.length == 0)
        return true;                      //若为空则返回true
    return false;
}

istream& operator>>(istream& cin, Test_Stu& stu)
{
    cout << "姓名：";          cin >> stu.name;
    cout << "性别：";          cin >> stu.gender;
    cout << "年龄：";          cin >> stu.age;
    cout << "身份证号：";      cin >> stu.ID;
    cout << "考试科目：";      cin >> stu.subject;
    return cin;
}
```

```
ostream& operator<<(ostream& cout, Test_Stu& stu)
{
    cout << "\t————————————————————————————————————————————————————————————————————\n";
    cout << "\t" << stu.number << "\t\t" << stu.name << "\t\t" << stu.gender << "\t\t"
        << stu.age << "\t\t" << stu.ID << "\t\t" << stu.subject << "\n";
    return cout;
}

Status InsertList(SqList& L)                                    /* 考生报名，自动为考生生成考号
*/
{
    if (L.length == Maxsize + 1)
    {
        cout << "\t报名人数已满！";
        return ERROR;
    }
    cout << "\t请输入考生信息：" << endl;
    Test_Stu stu;      cin >> stu;
    string str;
    str = std::to_string((long double)L.length + 1);        //to_string()函数将long double
型转化为字符串
    stu.number = "19120" + str;
    L.elem[L.length + 1] = stu;
    L.length++;
    cout << "\t报名成功！"<<endl;
    cout << "您的考号为“" << L.elem[L.length].number << "”，请牢记您的考号！"<<endl;
    return OK;
}

void InsertList(SqList& L, Test_Stu stu)
{
    L.elem[L.length + 1] = stu;
    L.length++;
}

Status Search_number(SqList& L)                                 /* 根据身份证号查找 */
{
    cout << "请输入要查找考生的考生号：";
    string key;        cin >> key;            //输入身份证号
    L.elem[0].number = key;              //把要查找元素的值赋给哨兵
    int num;
    for (num = L.length; L.elem[num].number != key; num--);
```

```
                                        //从后往前依次寻找
    if (num == 0)
    {
        cout << "\t————————————————————————————————————————————————————————
————————————————————————\n";
        cout << "\t" << "考号\t\t姓名\t\t性别\t\t年龄\t\t身份证号\t\t\t考试科目\n";
        cout << "\t————————————————————————————————————————————————————————
————————————————————————\n";
        cout << "\t很抱歉，未找到该考生信息！" << endl;
        return ERROR;
    }
    else {
        cout << "\t您要查找的考生生信息如下：" << endl;
        cout << "\t————————————————————————————————————————————————————————
————————————————————————\n";
        cout << "\t" << "考号\t\t姓名\t\t性别\t\t年龄\t\t身份证号\t\t\t考试科目\n";
        cout << L.elem[num];
        cout << "\t————————————————————————————————————————————————————————
————————————————————————\n";
    }
    return OK;
}

Status Verify(SqList& L)                                /* 验证考生是否报名 */
{
    cout << "请输入要考生姓名：";
    string key;         cin >> key;             //输入姓名
    L.elem[0].name = key;                   //把要查找元素的值赋给哨兵
    int num;
    for (num = L.length; L.elem[num].name != key; num--);
                                        //从后往前依次寻找
    if (num == 0)
        cout << "\t很抱歉，未找到您的信息！" << endl;
    else
        cout << "\t您已报名！" << endl;
    return OK;
}

Status Sort_age(SqList& L)                              /* 根据年龄排序 */
{
    Test_Stu Stu[Maxsize]; int j;
    for (int i = 0; i <= L.length; i++)
        Stu[i] = L.elem[i];
    for (int i = 2; i <= L.length; i++)
```

```
        if (Stu[i].age < Stu[i - 1].age)
        {
            Stu[0] = Stu[i];
            Stu[i] = Stu[i - 1];
            for (j = i - 2; Stu[0].age < Stu[j].age; j--)
                Stu[j + 1] = Stu[j];
            Stu[j + 1] = Stu[0];
        }
    cout << "\t————————————————————————————————————————————————————————
————————————————————\n";
    cout << "\t" << "考号\t\t姓名\t\t性别\t\t年龄\t\t身份证号\t\t\t考试科目\n";
    for (int i = 1; i <= L.length; i++)                          //输出排好序的元素
        cout << Stu[i];
    cout << "\t————————————————————————————————————————————————————————
————————————————————\n";
    return OK;
}

Status Delete(SqList& L)                                         /* 删除指定信息 */
{
    if(EmptyList(L) == true)
    {       cout << "\t暂无数据，无法删除！" << endl;     return ERROR;     }
    cout << "请输入要删除的人员考生号：";
    string key;       cin >> key;
    L.elem[0].number = key;              //把要查找元素的值赋给哨兵
    int num;
    for (num = L.length; L.elem[num].number != key; num--);
                                                 //从后往前依次寻找
    if (num == 0)
        cout << "\t未找到该考生信息！请检查您是否输入正确"<<endl;
    else {
        cout << "\t•————————————————————————————————————————•" << endl;
        cout << "\t|          该人员信息如下，是否要确认删除该人员信息？      |" << endl;
        cout << "\t|                  确认请按Y，取消请按N                    |" << endl;
        cout << "\t|                                                          |" << endl;
        cout << "\t|          ┌      ┐                ┌      ┐          |" <<
endl;
        cout << "\t|            是（Y）                  否（N）          |" << endl;
        cout << "\t|          └      ┘                └      ┘          | " <<
endl;
        cout << "\t•————————————————————————————————————————•" << endl;
        cout << "\t————————————————————————————————————————————————————————
————————————————————\n";
        cout << "\t" << "考号\t\t姓名\t\t性别\t\t年龄\t\t身份证号\t\t\t考试科目\n";
```

```
        cout << L.elem[num];
        cout << "\t————————————————————————————————————————————————————————————————————————\n";
        char ch; cin >> ch;
        while (ch != 'y' && ch != 'Y' && ch != 'n' && ch != 'N')              //判断用户输入是否合法
        {
            cout << "您的输入有误，请重新输入！";
            cin >> ch;
        }
        switch (ch)
        {
        case'y':
        case'Y':
            for (int i = num; i < L.length + 1; i++)
                L.elem[i] = L.elem[i + 1];
            L.length--;                                  //删除完成后长度减1
            cout << "\t删除成功！" << endl;
            break;
        case'n':
        case'N':cout << "\t已退出！" << endl;break;
        }
    }
    return OK;
}
Status Edit(SqList& L)                                                  /* 编辑指定信息 */
{
    if (EmptyList(L) == true)
    {
        cout << "暂无数据，无法删除！" << endl;        return ERROR;
    }
    cout << "请输入要编辑的人员考生号：";
    string key;        cin >> key;
    L.elem[0].number = key;              //把要查找元素的值赋给哨兵
    int num;
    for (num = L.length; L.elem[num].number != key; num--);
                                         //从后往前依次寻找
    if (num == 0)
        cout << "未找到该考生信息！请检查您是否输入正确"<<endl;
    else {
        cout << "\t•————————————————————————————————————————•" << endl;
        cout << "\t|          该人员信息如下，是否要确认修改该人员信息？       |" << endl;
        cout << "\t|                  确认请按Y，取消请按N                     |" << endl;
        cout << "\t|                                                           |" << endl;
```

```
        cout << "\t┃          ┌      ┐                  ┌      ┐          ┃" <<
endl;
        cout << "\t┃            是（Y）                  否（N）          ┃" << endl;
        cout << "\t┃          └      ┘                  └      ┘          ┃" <<
endl;
        cout << "\t•————————————————————————————————————————————————•" << endl;
        cout << "\t————————————————————————————————————————————————————
————————————————————————\n";
        cout << "\t" << "考号\t\t姓名\t\t性别\t\t年龄\t\t身份证号\t\t\t考试科目\n";
        cout << L.elem[num];
        cout << "\t————————————————————————————————————————————————————
————————————————————————\n";
        char ch; cin >> ch;
        while (ch != 'y' && ch != 'Y' && ch != 'n' && ch != 'N')              //判断用户输
入是否合法
        {
            cout << "您的输入有误，请重新输入！";
            cin >> ch;
        }
        switch (ch)
        {
        case'y':
        case'Y':
            cin >> L.elem[num];
            cout << "\t编辑成功！" << endl;
            break;
        case'n':
        case'N':cout << "\t已退出！" << endl;break;
        }
    }
    return OK;
}

Status SaveFile(SqList& L)
{
    if (EmptyList(L) == true)
    {       cout << "您还未添加数据，无法保存到文件！"<<endl;       return ERROR;      }
    ofstream outfile;                                //建立ofstream类的对象outfile
    outfile.open("Manager.txt", ios::app);           //与文件Manager.txt建立关联
    for (int i = 1; i < L.length+1; i++)                 //向文件内写入数据
        outfile << "\n" << L.elem[i].number << "\t\t" << L.elem[i].name << "\t\t" <<
L.elem[i].gender << "\t\t" << L.elem[i].age << "\t\t"
        << L.elem[i].ID << "\t\t" << L.elem[i].subject << "\n";
    outfile.close();                                 //取消与文件的关联
```

```
    cout << "\t信息保存成功！" << endl;;
    return OK;
}

void ReadFile(SqList& L)
{
    ifstream infile;                                    //建立对象
    infile.open("Manager.txt",ios::in);                 //与文件Manager.txt建立关联
    for(;;)
    {
        string number,name,gender,age,ID,subject;
        infile>>number>>name>>gender>>age>>ID>>subject;          //读取文件中的信息
        if(number=="\0"&&name=="\0"&&gender=="\0"&&age=="\0"&&ID=="\0"&&subject=="\0")
break;
                                                                 //用于判断读取的信息是
否为空，若信息为空，则结束循环
        else{
            cout << "\t-------------------------------------------------------------
-----------------------------\n";
            cout << "\t" << number << "\t\t" << name << "\t\t" << gender << "\t\t"
                 << age << "\t\t" << ID << "\t\t" << subject << "\n";
        }
    }
    cout << "\t-------------------------------------------------------------------
----------------------\n";
}

Status Display(SqList& L)                                    /* 输出全部信息 */
{
    if (L.length == 0)
    {
        cout << "暂无信息！" << endl;        return ERROR;
    }
    cout << "\t-------------------------------------------------------------------
----------------------\n";
    cout << "\t" << "考号\t\t姓名\t\t性别\t\t年龄\t\t身份证号\t\t\t考试科目\n";
    for (int i = 1; i <= L.length; i++)
        cout << L.elem[i];
    cout << "\t-------------------------------------------------------------------
----------------------\n";
    return OK;
}

void MainMenu(SqList& L)                                     /* 实现主界面菜单
```

```
的现实及相应操作 */
{
        cout << "\t•*****************************•" <<
endl;
        cout << "\t*                    ┏                  ┓                  *" <<
endl;
        cout << "\t*                     简陋的考生报名系统                  *" <<
endl;
        cout << "\t*                    ┗                  ┛                  *" <<
endl;
        cout << "\t*                      1.开始报名                          *" <<
endl;
        cout << "\t*                      2.查询是否报名                      *" <<
endl;
        cout << "\t*                      3.管理已报名信息                    *" <<
endl;
        cout << "\t*                      0.关闭窗口                          *" <<
endl;
        cout << "\t*                                                          *" <<
endl;
        cout << "\t*      请选择进行的操作（0到3）：                          *" <<
endl;
        cout << "\t*                                                          *" <<
endl;
        cout << "\t•*****************************•" <<
endl;
        int choose;        cin >> choose;
        while (choose < 0 || choose > 3) //判断用户输入是否合法
        {
            cout << "您的输入有误，请重新输入！";
            cin >> choose;
        }
        switch (choose)
        {
        case 1:InsertList(L);  break;                              //报名
        case 2:Verify(L);      break;
        case 3:ManagerMenu(L); break;        //访问子菜单
        case 0:cout << "\t感谢您的使用^-^" << endl;        exit(0);
        }
        system("pause");
        system("cls");
}
void ManagerMenu(SqList& L)                                  /* 管理者菜单 */
{
```

```
    string password;
    cout << "\t请输入管理密码：";              cin >> password;
    if (password == "pass")
    {
        cout << "\t密码正确！" << endl;
        system("pause");        system("cls");
        while (1)
        {
            cout << "\t•…………………………………………………………•" <<
endl;
            cout << "\t:                    ┏              ┓               :"
<< endl;
            cout << "\t:                      管理菜单                   :"
<< endl;
            cout << "\t:                    ┗              ┛               :"
<< endl;
            cout << "\t:                                                  :"
<< endl;
            cout << "\t:              1.查看已报名考生信息               :"
<< endl;
            cout << "\t:              2.删除指定人员信息                 :"
<< endl;
            cout << "\t:              3.根据考号查找人员信息             :"
<< endl;
            cout << "\t:              4.编辑指定人员信息                 :"
<< endl;
            cout << "\t:              5.保存人员信息到文件               :"
<< endl;
            cout << "\t:              6.读取文件内的信息                 :"
<< endl;
            cout << "\t:              7.按年龄排序查看所有人员信息       :"
<< endl;
            //cout << "\t:              8.统计各科报名人数                 :"
<< endl;
            cout << "\t:              0.返回上级菜单                     :"
<< endl;
            cout << "\t:      请选择进行的操作（0到7）：                 :"
<< endl;
            cout << "\t:                                                  :"
<< endl;
            cout << "\t•…………………………………………………………•" <<
endl;
            int choose;        cin >> choose;
            string password;
```

```
            while (choose < 0 || choose > 7) //判断用户输入是否合法
            {
                cout << "您的输入有误，请重新输入！";
                cin >> choose;
            }
            switch (choose)
            {
            case 1:Display(L);      break;
            case 2:Delete(L);       break;
            case 3:Search_number(L);    break;
            case 4:Edit(L);             break;
            case 5:SaveFile(L);         break;
            case 6:ReadFile(L);         break;
            case 7:Sort_age(L);         break;
                //case 8:Statistic(L); break;
            case 0:break;
            }
            if (choose == 0)   break;              //返回上一级菜单
            system("pause");
            system("cls");
        }
    }
    else
        cout << "\t密码错误，拒绝您的访问！" << endl;
}
```

3. main.cpp（调用部分）

```
#include "Test_Manager.h"
void main()
{
    SqList L;InitList(L);
    Test_Stu stu_1, stu_2;
    stu_1.age = 33;             stu_1.gender = "男";        stu_1.ID =
"123456789987654321";
    stu_1.name = "夏洛";    stu_1.subject = "数学";     stu_1.number = "191201";
    stu_2.age = 29;             stu_2.gender = "女";        stu_2.ID =
"987654321123456789";
    stu_2.name = "马冬梅"; stu_2.subject = "语文";     stu_2.number = "191202";
    InsertList(L, stu_1);           InsertList(L, stu_2);       //往其中添加的示例信息
    while(1)
        MainMenu(L);
}
```

七、 测试结果

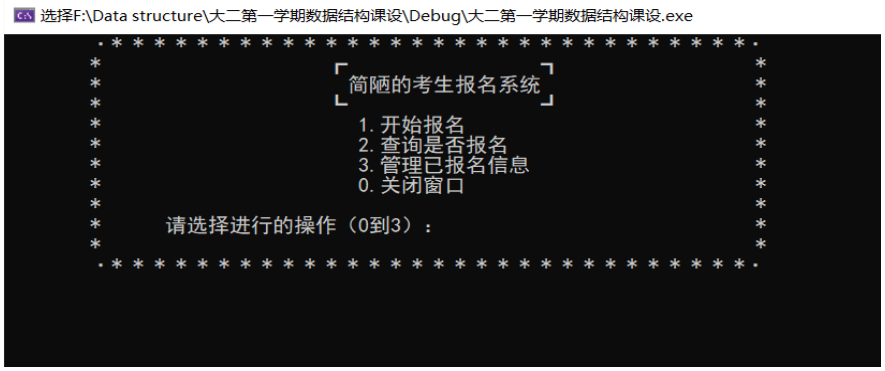

图 2 主界面

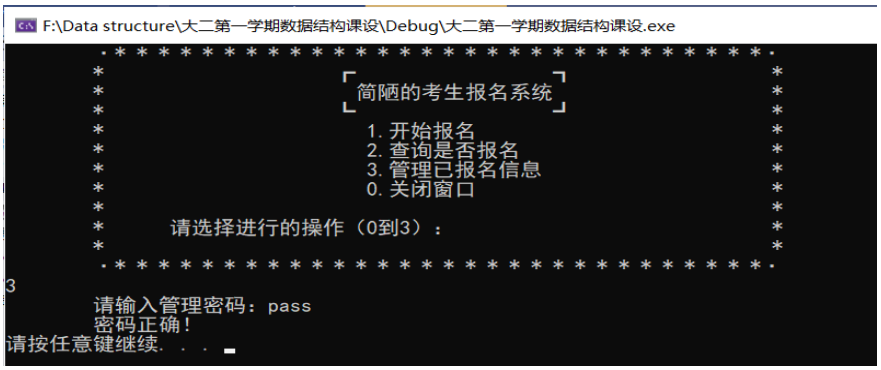

图 3 进入管理界面需要输入密码

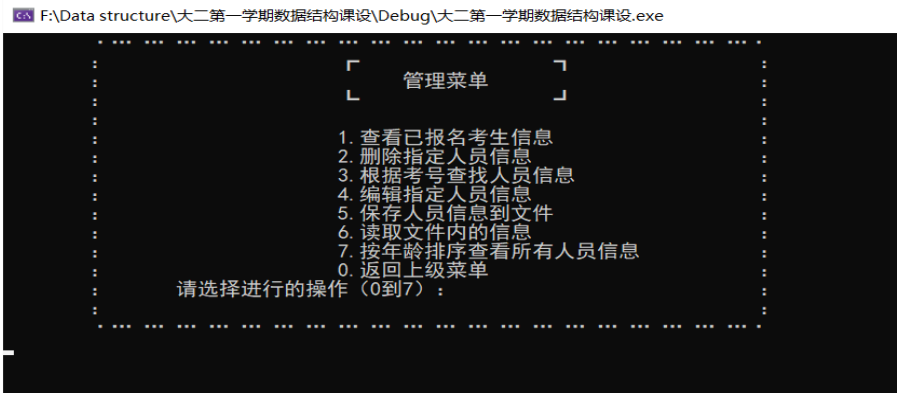

图 4 子菜单

图 5 查看所有信息

八、 心得体会及总结

总的来说，这次写完这个管理系统收获颇多，不仅是代码量的累积，更多的是编程经验，查错能力以及对知识点的熟练程度都有了很大的提高。

其实将管理系统要用到的每一个知识点（比如顺序表的增删改查、运算符的重载、文件的读写操作等）分别拿出来写一个小程序的话，是很简单的，但如果要将这些知识点整合到一起写出这个管理系统，那就不一样了。所以我认为这个管理系统的难度就是将所有简单知识点的整合到一起的能力，从另一层面来说也是对编程经验的考验；

在写的过程中，前面几遍测试写好的功能时总会有莫名其妙的问题出现。虽然大部分都还是能通过检查代码或者上网搜索错误代码来找到错误来源，但总有那么几个错误无论如何也是找不到解决办法，让人很头疼，通过多种手段，还是想办法解决了。所以这次课程设计也算是收获颇丰：

1、 巩固和加深了对数据结构的理解，提高综合运用本课程所学知识的能力。
2、 培养了我选用参考书，查阅手册及文献资料的能力。培养独立思考，深入研究，分析问题、解决问题的能力。
3、 通过实际编译系统的分析设计、编程调试，掌握应用软件的分析方法和工程设计方法。
4、 通过课程设计，培养了我严肃认真的工作作风，逐步建立正确的生产观念、经济观念和全局观念。

虽然说以前非常不懂这门语言，在它上面花费了好多心思，觉得它很难，是需用花费了大量的时间编写出来的。现在真正的明白了一些代码的应用，每个程序都有一些共同点，通用的结构，相似的格式。只要努力去学习，就会灵活的去应用它。