**课 程 设 计 报 告**

|  |  |
| --- | --- |
| 学院、系： | 计算机科学与信息工程学院 |
| 专业名称： | 物联网工程 |
| 科 目： | C语言课程设计 |
| 学生学号： | 202103049148 |
| 学生姓名： | 黄东明 |
| 指导教师： | 陈建孝 副教授 |
| 完成时间： | 2022年6月 |

（课程设计报告正文部分）

通讯录管理系统

一、设计任务与目标

C语言现是一个应用比较广泛的语言，通过对程序的设计让我们能更好的学习和了解它的各个方面的应用，对我们今后的学习有很大的帮助。同时，在做程序的过程中锻炼了我们对程序的分析与编写能力，更锻炼了我们的实战能力，为我们以后在自己的专业取得更好的成就做了一个好的铺垫。本课程设计主要是一个简单的通讯录管理系统。通过程序中的菜单编号实现通讯录账号的注册与登录，联系人信息的添加、删除、查找、修改等操作。

通过这次课程设计更好的掌握以下知识：  
1.掌握函数的定义和使用。  
2．掌握各种基本的数据类型和结构体的定义。  
3.掌握数组的定义和使用。  
4.掌握基本的程序结构：顺序、选择、循环。

5.掌握基本使用文件管理

二、方案设计与论证

《通讯录管理系统》

登录界面：

1. 通讯录登录系统
2. 注册登录该管理系统

通讯录主界面：  
1.通讯录能存放联系人的信息  
  名字+性别+年龄+电话+地址  
2. 增加指定联系人信息  
3. 删除指定联系人信息  
4. 查找指定联系人信息  
5.修改指定联系人信息

6.保存信息到文件  
7.账号中心

（1）       注册登录功能设计

采用菜单，包含4个子功能：

1. 退出登录系统

②注册账号。该功能由Userreg ()函数实现。

1. 登录账号。该功能由UserLongin ()函数实现。

④找回密码。该功能由GetbackPassword ()函数实现。

（2）       通讯录存储结构设计

采用顺序表来存储“联系人基本信息”，基本信息包括姓名、性别、年龄、电话、地址。

首先由ContactInit ()函数来初始化通讯录，再由ContactLoad ()函数来读取文件中通讯录信息，进行通讯录联系人添加操作时会由CheckCapacity ()函数来检查空间容量是否充足，不足自动扩容，程序退出时ContactSave ()函数会保存通讯录信息到文件，最后使用ContactDestroy ()函数销毁通讯录。

（3）       通讯录功能设计

采用菜单，包含5个子功能：

1. 添加联系人。输入要添加的联系人的基本信息由ContactAdd ()函数实现。
2. 删除联系人。输入要删除的联系人的姓名或电话由ContactDel ()函数实现。
3. 查找联系人。输入要查找的联系人的基本信息由ContactFind ()函数实现。
4. 修改联系人。输入要修改的联系人的姓名或电话由ContactRevise ()函数实现。其中包含两种修改方式，批量修改和单一修改，使用if…else…语句实现。

⑤ 打印联系人。可查看通讯录中所有联系人的基本信息并统计联系人个数。该功能由ContactPrint () 函数实现。

（4）       系统主界面设计

包含登录界面与通讯录管理界面

为了实现通讯管理系统各功能的管理，设计一个含有多个菜单项的主控菜单子程序以链接系统的各项子功能，方便用户使用本系统。

注册登录：用户自创的账号（账号唯一不重复），自行设置密码。注册后可登录，便可进入通讯录管理系统，以进行用户所需要的操作。

添加：可以添加通讯录联系人的基本信息，依次进行添加联系人的姓名、性别、年龄、电话、地址（每次添加会检查通讯录容量，不够就自动扩容）。

删除：通过输入要删除的联系人的姓名或电话实现删除，欲删除的联系人不存在则不可删除。

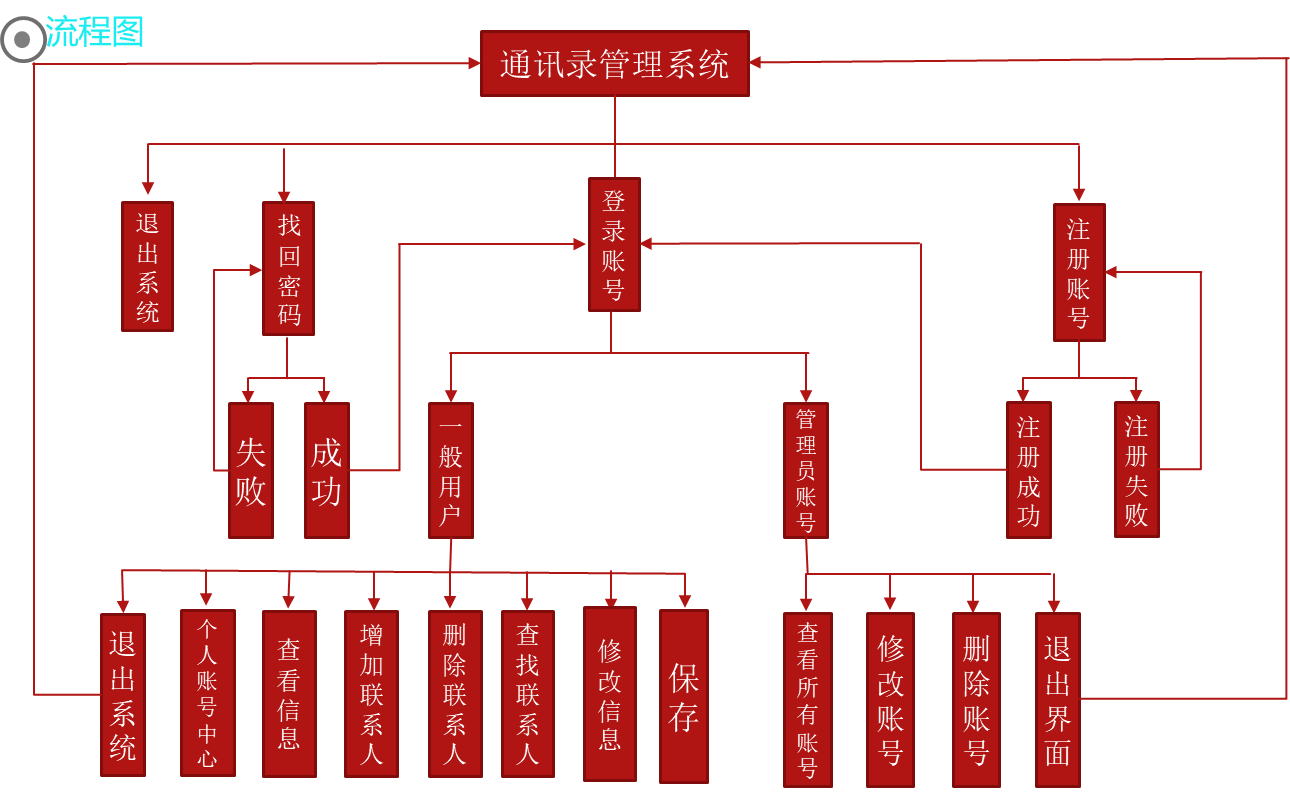
查找：可通过按姓名、性别、年龄、电话、地址查找相同类型的联系人信息，并依次打印在屏幕，通讯录信息为空则不可查找，若要查找的联系人不存在，则打印的通讯录列表为空。

修改：修改方式分两种，首先输入联系人的姓名或电话进行下一步的修改操作。其次，一种方式是修改联系人的一系列信息（批量修改）；另一种是用户选择需要修改的信息进行修改（单一修改）。（通讯录信息不为空）若需修改的联系人不存在，则输出“要修改的人不存在”。

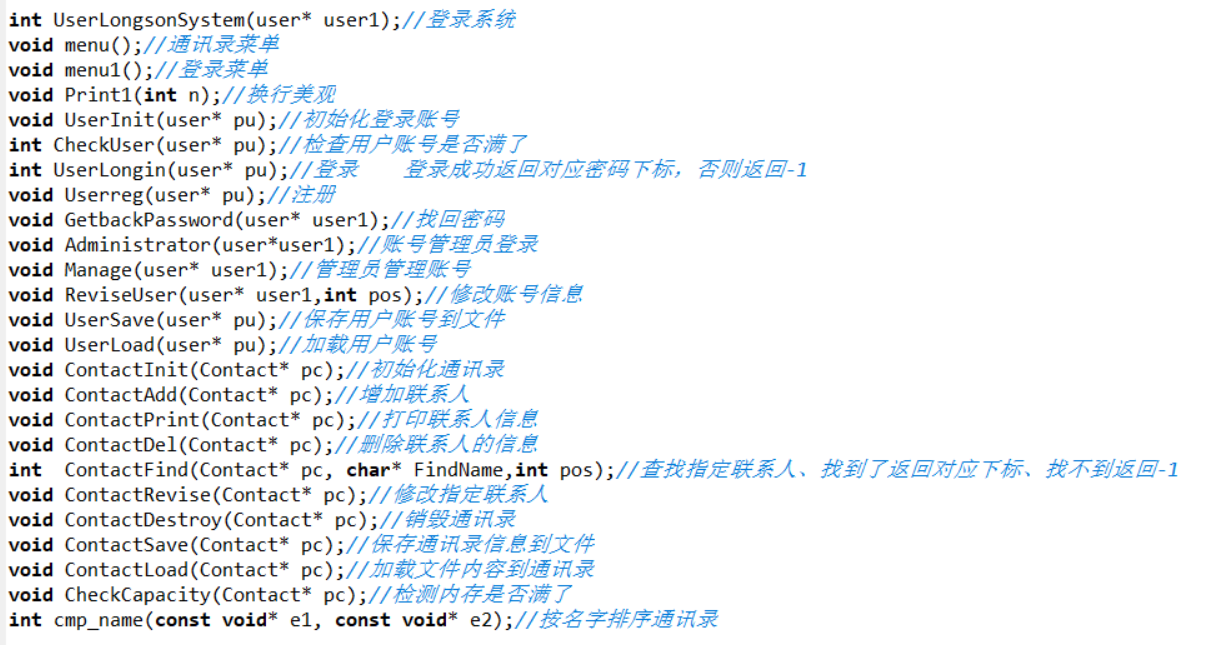
打印：选择打印的操作编号方可将通讯录内所有联系人的基本信息打印（会统计已有联系人的个数）在屏幕上。

账号中心：用户可以自行修改自己的账号信息（如密码，账号，名称），也可以查看自己账号的信息

三、程序框图或流程图，程序清单与调用关系



函数



四、全部源程序清单

/\* 一些函数的头文件 \*/

#include<stdio.h>

#include<assert.h>

#include<stdlib.h>

#include<string.h>

#include<windows.h>

#include <conio.h>

#define MAX\_NAME 20 //名字的最长个数

#define MAX\_SEX 10 //存储性别

#define MAX\_TELE 12 //电话的最长个数

#define MAX\_ADDR 30 //地址的最长个数

#define MAX\_ACC 20 //账号最长个数

#define MAX\_PASS 20 //密码的最长个数

#define N 6 //账号最多注册的个数

/\*一些函数声明和其他声明\*/

//用户账号结构体

typedef struct personnel

{

char userName[MAX\_NAME]; //账号名称

char Account[MAX\_ACC]; //账号

char password[MAX\_PASS]; //密码

}personnel;

//账号管理结构体

typedef struct User

{

struct personnel val[N];

int begin;

}user;

//联系人的信息结构体

typedef struct people

{

char name[MAX\_NAME];//姓名

char sex[MAX\_SEX]; //性别

char age[4]; //年龄

char tele[MAX\_TELE];//电话号码

char addr[MAX\_ADDR];//地址

}people;

//通讯录

typedef struct Contact

{

people\* data;

int size; //有效数据个数

int capacity;//容量

}Contact;

//通讯功能选择的枚举

enum contact

{

Exit,//退出

Print,//打印联系人

Add,//增加联系人

Del,//删除联系人

Find,//查找联系人

Revise,//修改联系人

Save,//保存信息到文件

Reviseuser//账号中心

};

int UserLongsonSystem(user\* user1);//登录系统

void menu();//通讯录菜单

void menu1();//登录菜单

void Print1(int n);//换行美观

void UserInit(user\* pu);//初始化登录账号

int CheckUser(user\* pu);//检查用户账号是否满了

int UserLongin(user\* pu);//登录 登录成功返回对应密码下标，否则返回-1

void Userreg(user\* pu);//注册

void GetbackPassword(user\* user1);//找回密码

void Administrator(user\*user1);//账号管理员登录

void Manage(user\* user1);//管理员管理账号

void ReviseUser(user\* user1,int pos);//修改账号信息

void UserSave(user\* pu);//保存用户账号到文件

void UserLoad(user\* pu);//加载用户账号

void ContactInit(Contact\* pc);//初始化通讯录

void ContactAdd(Contact\* pc);//增加联系人

void ContactPrint(Contact\* pc);//打印联系人信息

void ContactDel(Contact\* pc);//删除联系人的信息

int ContactFind(Contact\* pc, char\* FindName,int pos);//查找指定联系人、找到了返回对应下标、找不到返回-1

void ContactRevise(Contact\* pc);//修改指定联系人

void ContactDestroy(Contact\* pc);//销毁通讯录

void ContactSave(Contact\* pc);//保存通讯录信息到文件

void ContactLoad(Contact\* pc);//加载文件内容到通讯录

void CheckCapacity(Contact\* pc);//检测内存是否满了

int cmp\_name(const void\* e1, const void\* e2);//按名字排序通讯录

/\*----------------------------------------------------------------------------------------\* /

/\*通讯录整体的逻辑实现 \*/

int main()

{

system("color 6");//换字体颜色 60金黄，70白色

system("mode con cols=125 lines=38");//设定控制台窗口大小

Contact pc = { 0 };//创建通讯录

user user1 = { 0 };//创建账号数组

int pos1 = 0; //接收用户登录后返回数组的对应下标

UserInit(&user1);//初始化并加载用户账号

ContactInit(&pc);//初始化通讯录

pos1=UserLongsonSystem(&user1);//登录系统 ,登录成功就返回账号数组对应的下标，用户直接退出就返回-1

if (pos1 == -1) //终止程序条件

{

return 0;

}

Sleep(300);//让程序停500ms

//通讯录管理系统逻辑实现

int input = 0; //接收用户选择的功能

int pos = 0; //接收查找联系人的下标

int flag = 0; //用于控制退出的开关

int Q = 0; //统计选择功能的次数

char FindObject[MAX\_NAME] = { 0 };//接收用户搜索联系人的信息

int pos2 = 0; //用于查找多个联系人的中间值

int n = 0; //接收查找联系人时的退出条件

do

{

if (Q != 0) //这个是已经选择了某一功能之后的选项，可以提醒用户随时退出，或者转到起始界面

{

printf("\n\n请按1退出此界面，按0退出系统....................\n");

scanf("%d", &flag);

if (flag == 0)

{

input = 0;

qsort(pc.data, pc.size, sizeof(pc.data[0]), cmp\_name);//按名字排序通讯录

ContactSave(&pc);//用户选择退出，就先把通讯录的信息保存在文件中

UserSave(&user1);

ContactDestroy(&pc);

printf("\n退出成功\n");

break;

}

}

system("cls");

Print1(7);

printf("\t\t\t\t\t 欢迎%s来到通讯录管理系统\t\t\t\t\t\n", user1.val[pos1].userName);

menu();//打印菜单

Print1(3);//换行美观

printf("\n\t\t\t\t\t请选择>:");

scanf("%d", &input);//接收功能选项

switch (input)

{

case Print: //打印全部联系人信息

system("cls"); //清屏

Print1(7); //换行美观

qsort(pc.data, pc.size, sizeof(pc.data[0]), cmp\_name);//按名字排序通讯录

ContactPrint(&pc);

break;

case Add: //增加联系人

system("cls");

ContactAdd(&pc);

break;

case Del: //删除联系人

system("cls");

ContactPrint(&pc);

ContactDel(&pc);

break;

case Find: //查找相关联系人

system("cls");

Print1(3);

n = 0; //n作为是否找到联系人的开关

printf("\n\t\t\t\t\t\t\t\tPS:可以是名字,性别，电话号码，地址，年龄\n");

printf("\n\t\t\t\t\t请输入你要查找联系人的信息:");

scanf("%s", FindObject); //接收相关信息

pos = ContactFind(&pc, FindObject, pos); //进入查找，找到了返回对应下标，找不到返回-1

pos2 = pos;

while (1)

{

if (pos2 != -1) //不为-1，则打印该下标的元素

{

n++;

if (n == 1)

printf("\n%-20s %-10s %-8s %-18s %-30s\n", "姓名", "性别", "年龄", "电话号码", "地址");

printf("\n%-20s %-10s % -8s % -18s %-30s\n", pc.data[pos2].name,

pc.data[pos2].sex, pc.data[pos2].age, pc.data[pos2].tele, pc.data[pos2].addr);

//然后再把已经找到的元素的下一个的下标传过去进行查找，直到找完所有的相关元素就停止

pos2 = ContactFind(&pc, FindObject, pos2+1);

}

else

{

if (n == 0) //如果一开始n就为0，则说明一个也没有找到，也就不存在该联系人

printf("\t\t\t\t\t无此联系人\n");

break;

}

}

break;

case Revise: //修改联系人信息

system("cls");

ContactPrint(&pc); //修改前先打印联系人信息，以方便用户选择修改对象

ContactRevise(&pc); //修改联系人信息函数实现

break;

case Save: //用户也可以手动保存信息到文件

system("cls");

qsort(pc.data, pc.size, sizeof(pc.data[0]), cmp\_name);//按名字排序通讯录

ContactSave(&pc);

UserSave(&user1);

printf("\t\t\t\t\t保存成功\n");

break;

case Exit: //退出前先保存通讯录信息和账号信息到文件里，最后再销毁开辟的空间

system("cls");

ContactSave(&pc);

UserSave(&user1);

ContactDestroy(&pc);

printf("\t\t\t\t\t退出成功\n");

break;

case Reviseuser: //

ReviseUser(&user1, pos1);

break;

default: //选择错误提醒用户重新输入

system("cls");

printf("\t\t\t\t\t无此选项，请重新输入\n");

break;

}

Q++;

} while (input);

return 0;

}

/\*----------------------------------------------------------------------------------------\* /

/\*所有的函数实现 \*/

//换行美观

void Print1(int n)

{

int i = 0;

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

{

printf("\n");

}

}

//菜单

void menu()

{

printf("\t\t\t\t==================================================\t\t\t\t\t\n");

printf("\t\t\t\t\* 1.查看联系人信息 2.增加联系人 \*\t\t\t\t\t\n");

printf("\t\t\t\t\* 3.删除联系人 4.查找联系人 \*\t\t\t\t\t\n");

printf("\t\t\t\t\* 5.修改联系人 6.保存 \*\t\t\t\t\t\n");

printf("\t\t\t\t\* 7.账号中心 0.退出 \*\t\t\t\t\t\n");

printf("\t\t\t\t===================================================\t\t\t\t\t\n");

}

//登录菜单

void menu1()

{

Print1(7);

printf("\t\t\t\t\t=========欢迎来到通讯录登录系统=========\t\t\t\t\t\n");

printf("\t\t\t\t\t========================================\t\t\t\t\t\n");

printf("\t\t\t\t\t\* 1.登录 \*\t\t\t\t\t\n");

printf("\t\t\t\t\t\* 2.注册 \*\t\t\t\t\t\n");

printf("\t\t\t\t\t\* 3.找回密码 \*\t\t\t\t\t\n");

printf("\t\t\t\t\t\* 4.管理登录 \*\t\t\t\t\t\n");

printf("\t\t\t\t\t\* 0.退出 \*\t\t\t\t\t\n");

printf("\t\t\t\t\t========================================\t\t\t\t\t\n");

}

//初始化通讯录

void ContactInit(Contact\* pc)

{

pc->data = NULL;

pc->capacity = pc->size = 0;

ContactLoad(pc);//加载通讯录信息

}

//检测内存是否满了

void CheckCapacity(Contact\* pc)

{

//如果通讯录容量为空，就先赋值4个空间，满了就扩容两倍

if (pc->capacity == pc->size)

{

int newcapacity = pc->capacity == 0 ? 4 : pc->capacity \* 2;

people\* tem = (people\*)realloc(pc->data,sizeof(people) \* newcapacity);

if (tem == NULL)

{

printf("扩容失败\n");

exit(-1);

}

pc->data = tem;

pc->capacity = newcapacity;

}

}

//增加联系人

void ContactAdd(Contact\* pc)

{

//断言，判断pc是否为NULL

assert(pc);

//增加联系人前先检查容量

CheckCapacity(pc);

printf("\n\t\t\t\t\t请输入姓名:");

scanf("%s", pc->data[pc->size].name);

printf("\n\t\t\t\t\t请输入性别:");

scanf("%s", pc->data[pc->size].sex);

printf("\n\t\t\t\t\t请输入年龄:");

scanf("%s", pc->data[pc->size].age);

printf("\n\t\t\t\t\t请输入电话号码:");

scanf("%s", pc->data[pc->size].tele);

printf("\n\t\t\t\t\t请输入地址:");

scanf("%s", pc->data[pc->size].addr);

pc->size++;

printf("\t\t\t\t\t添加成功\n");

}

//销毁通讯录

void ContactDestroy(Contact\* pc)

{

//断言，判断pc是否为NULL

assert(pc);

free(pc->data);

pc->data = NULL;

pc->capacity = pc->size = 0;

}

//打印联系人信息

void ContactPrint(Contact\* pc)

{

//断言，判断pc是否为NULL

assert(pc);

int i = 0;

printf("%-20s %-10s %-8s %-18s %-30s\n\n", "姓名", "性别", "年龄", "电话号码", "地址");

for (i = 0; i < pc->size; i++)

{

printf("%-20s %-10s %-8s %-18s %-30s\n", pc->data[i].name, pc->data[i].sex, pc->data[i].age, pc->data[i].tele, pc->data[i].addr);

}

printf("\n\n总计有%d个联系人\n", pc->size);

}

//查找指定联系人 ，找到了返回对应下标，找不到返回-1

int ContactFind(Contact\* pc, char\* FindObject,int pos)

{

//断言，判断pc是否为NULL

assert(pc);

int i = 0;

for (i = pos; i < pc->size; i++)//搜索所有满足条件的元素进行返回

{

//对所有联系人的名字，性别，电话，地址，年龄进行一一比较

if (strcmp(FindObject, pc->data[i].name) == 0 || strcmp(FindObject,pc->data[i].sex) == 0 || strcmp(FindObject, pc->data[i].tele) == 0

|| strcmp(FindObject, pc->data[i].age) == 0 || strcmp(FindObject, pc->data[i].addr) == 0)

{

return i;

}

}

return -1;

}

//删除联系人的信息

void ContactDel(Contact\* pc)

{

assert(pc);

printf("\n\t\t\t\t\t你输入要删除的联系人名字:");

char DelName[MAX\_NAME] = { 0 };

scanf("%s", DelName);

int del = ContactFind(pc, DelName,0); //对用户要删除的名字进行搜索，有该联系人就进行删除

if (del == -1)

{

printf("\n\t\t\t\t\t该联系人不存在\n");

return;

}

int i = 0;

for (i = del; i < pc->size; i++) //删除

{

pc->data[i] = pc->data[i + 1];

}

pc->size--;

printf("\n\t\t\t\t\t删除成功\n");

}

//修改指定联系人

void ContactRevise(Contact\* pc)

{

assert(pc);

int input = 0;

printf("\n\n\t\t\t\t\t请输入你要修改联系人的名字:");

char RevName[MAX\_NAME] = { 0 };

scanf("%s", RevName);

int pos = ContactFind(pc, RevName,0); //对用户要修改的名字进行搜索，有该联系人就进行修改

if (pos == -1)

{

printf("\n\t\t\t\t\t该联系人不存在!");

return;

}

system("cls");

//修改前先打印该联系人信息

printf("\n\n%-20s %-10s %-8s %-18s %-30s\n", "姓名", "性别", "年龄", "电话号码", "地址");

printf("\n%-20s %-10s % -8s % -18s %-30s\n", pc->data[pos].name,

pc->data[pos].sex, pc->data[pos].age, pc->data[pos].tele, pc->data[pos].addr);

//让用户选择修改的对象

do

{

printf("\n\n\t\t请选择要修改的信息!\n");

printf("\n\t\t1.姓名 2.性别 3.年龄 4.电话号码 5.地址 6.全部重新修改 0.退出\n");

printf("\n\t\t请选择:>");

scanf("%d", &input);

switch(input)

{

case 1:

printf("\n\t\t请输入姓名:");

scanf("%s", pc->data[pos].name);

printf("\n\t\t修改成功\n");

printf("\n\t\t是否进行修改,按\"1\"继续，按\"0\"退出\n");

scanf("%d", &input);

break;

case 2:

printf("\n\t\t请输入性别:");

scanf("%s", pc->data[pos].sex);

printf("\n\t\t修改成功\n");

printf("\n\t\t是否进行修改,按\"1\"继续，按\"0\"退出\n");

scanf("%d", &input);

break;

case 3:

printf("\n\t\t请输入年龄:");

scanf("%s", pc->data[pos].age);

printf("\n\t\t修改成功\n");

printf("\n\t\t是否进行修改,按\"1\"继续，按\"0\"退出\n");

scanf("%d", &input);

break;

case 4:

printf("\n\t\t请输入电话号码:");

scanf("%s", pc->data[pos].tele);

printf("\n\t\t修改成功\n");

printf("\n\t\t是否进行修改,按\"1\"继续，按\"0\"退出\n");

scanf("%d", &input);

break;

case 5:

printf("\n\t\t请输入地址:");

scanf("%s", pc->data[pos].addr);

printf("\n\t\t修改成功\n");

printf("\n\t\t是否进行修改,按\"1\"继续，按\"0\"退出\n");

scanf("%d", &input);

break;

case 6:

printf("\n\t\t请输入姓名:");

scanf("%s", pc->data[pos].name);

printf("\n\t\t请输入性别:");

scanf("%s", pc->data[pos].sex);

printf("\n\t\t请输入年龄:");

scanf("%s", pc->data[pos].age);

printf("\n\t\t请输入电话号码:");

scanf("%s", pc->data[pos].tele);

printf("\n\t\t请输入地址:");

scanf("%s", pc->data[pos].addr);

printf("\n\t\t修改成功\n");

input = 0;

break;

case 0:

printf("退出成功\n");

break;

default:

printf("\n\t\t\t\t\t无此选项，请重新输入\n");

break;

}

} while (input);

}

//保存通讯录信息到文件

void ContactSave(Contact\* pc)

{

assert(pc);

FILE\* pf = fopen("people.data", "wb");

if (pf == NULL)

{

printf("ContactSave::打开文件失败\n");

exit(-1);

}

int i = 0;

for (i = 0; i < pc->size; i++) //保存通讯录信息到文件

{

fwrite(&pc->data[i], sizeof(people), 1, pf);

}

fclose(pf);

pf = NULL;

}

//加载文件内容到通讯录

void ContactLoad(Contact\* pc)

{

//读文件信息

FILE\* pf = fopen("people.data", "rb");

if (pf == NULL) //如果pf==NULL，则文件没有，进去此地方 先预打开一个文件就达到了 自动创建一个文件的目标

{

pf = NULL;

pf = fopen("people.data", "wb");//创建文件

if (pf == NULL)

{

printf("ContactLoad::打开文件失败\n");

exit(-1);

}

fclose(pf);

pf = NULL;

pf = fopen("people.data", "rb");//再重新打开文件

if (pf == NULL)

{

printf("ContactLoad::失败\n");

exit(-1);

}

}

people tmp = { 0 }; //用来临时接收文件内的信息

while (fread(&tmp, sizeof(people), 1, pf)) //从文件加载通讯录信息

{

CheckCapacity(pc);

pc->data[pc->size] = tmp;

pc->size++;

}

fclose(pf); //关闭文件

pf = NULL;

}

//初始化登录账号

void UserInit(user\* pu)

{

assert(pu);

pu->begin = 0;

UserLoad(pu); //加载账号信息到数组里

}

//注册

void Userreg(user\* pu)

{

assert(pu);

Print1(7);

printf("\n\t\t\t\t\t\t\t\tPS:此界面任意时刻输入\"exit\"即可退出此界面\n");

printf("\n\t\t\t\t\t请开始注册账号\n");

printf("\n\t\t\t\t\t账号:");

scanf("%s", pu->val[pu->begin].Account); //接收用户输入的账号

if (strcmp(pu->val[pu->begin].Account, "exit") == 0) //接收用户的退出条件

{

system("cls");//清屏

return;

}

printf("\n\t\t\t\t\t密码:");

scanf("%s", pu->val[pu->begin].password); //接收密码

if (strcmp(pu->val[pu->begin].password, "exit") == 0) //接收用户的退出条件

{

system("cls");

return;

}

printf("\n\t\t请给你的账号起个漂亮的名称,它将是你以后进来的名字！同时也是你找回密码的根据！！\n");

printf("\n\t\t\t\t\t");

scanf("%s", pu->val[pu->begin].userName);

if (strcmp(pu->val[pu->begin].userName, "exit") == 0)//接收用户的退出条件

{

system("cls");

return;

}

pu->begin++;

system("cls");

printf("\t\t\t\t\t注册成功\n");

}

//密码输入处理，对屏幕回显‘\*’

void Getpassword(char\* pass, int sz)

{

assert(pass);

char c = 0;

int i = 0;

while ((c = getch()) != 13)

{

pass[i] = c;

i++;

putchar('\*'); //屏幕回显‘\*’

if (i > sz - 2)

{

printf("\n密码过长\n");

break;

}

}

pass[i] = '\0';

}

//登录 登录成功返回对应密码下标，否则返回-1

int UserLongin(user\* pu)

{

char UName[MAX\_NAME] = { 0 };

char Pass[MAX\_PASS] = { 0 };

int i = 0;

int flag = 0;//用于没有系统不存在该账号时提醒用户的开关

system("cls");

printf("\n\n\t\t\t\t\t=============================\n");

printf("\t\t\t\t\t\* 登录界面 \*\n");

printf("\t\t\t\t\t=============================\n");

printf("\n\t\t\t\t\t账号:");

scanf("%s", UName);

for (i = 0; i <= pu->begin; i++) //对用户输入的账号进行和文件中获取的账号逐一对比

{

if (strcmp(UName, pu->val[i].Account) == 0) //有相同的账号则让用户输入密码

{

while (strcmp(Pass, "exit") != 0)

{

printf("\n\t\t\t\t\t密码:");

Getpassword(Pass, MAX\_PASS); //接收密码，同时对屏幕回显‘\*’

if (strcmp(Pass, pu->val[i].password) == 0) //密码输入正确就返回改账号的数组下标，然后进入通讯

{

system("cls");

printf("\n\n\n\n登录成功,正在加载通讯录................................\n");

return i;

}

else//输入错误可以提醒用户是否进行输入

{

printf("\t\t\t\t\t\n密码错误,可以输入exit退出,继续输入按1\n");

scanf("%s", Pass);

if (strcmp(Pass, "exit") == 0)

{

flag = 1;

break;

}

else

continue;

}

}

}

}

if (flag == 0) //如果账号对不上就提醒用户

{

system("cls");

printf("\t\t\t\t\t无此账号\n");

Sleep(300);

}

return -1;

}

//保存用户账号

void UserSave(user\* pu)

{

assert(pu);

FILE\* pf = fopen("user.data", "wb");

if (pf == NULL)

{

printf("UserSave::打开文件失败\n");

exit(-1);

}

int i = 0;

for (i = 0; i < pu->begin; i++) //保存账号信息到文件

{

fwrite(&pu->val[i], sizeof(personnel), 1, pf); //把信息写进文件里

}

fclose(pf);

pf = NULL;

}

//加载用户账号

void UserLoad(user\* pu)

{

//读文件信息

FILE\* pf = fopen("user.data", "rb");

if (pf == NULL) //如果pf==NULL，则文件没有，进去此地方 先预打开一个文件就达到了 自动创建一个文件的目标

{

pf = NULL;

pf = fopen("user.data", "wb");//创建文件

if (pf == NULL)

{

printf("UserSave::打开文件失败\n");

exit(-1);

}

fclose(pf); //再关闭

pf = NULL;

pf = fopen("user.data", "rb");//再重新打开文件

if (pf == NULL)

{

printf("UserLoad::失败\n");

exit(-1);

}

}

//读文件信息

int i = 0;

personnel tem = {0};

while (fread(&tem, sizeof(personnel), 1, pf))//从文件一个一个读到数组中

{

pu->val[i] = tem;

i++;

pu->begin++;

}

fclose(pf);

pf = NULL;

}

//检查用户账号是否满了

int CheckUser(user\* pu)

{

assert(pu);

if (pu->begin >= N)

{

printf("\t\t\t\t\t用户账号已满\n");

return -1;

}

return 1;

}

//修改账号信息

void ReviseUser(user\* user1,int pos)

{

assert(user1);

system("cls");

Print1(7);

printf("\n\t\t\t\t1.修改名称 2.修改账号 3.修改密码 4.查看账号信息 5.退出此界面\n");

printf("\n\t\t\t\t请选择:");

int input = 0;

scanf("%d", &input); //接收用户选择修改对象的功能

switch (input)

{

case 1:

printf("\n\t\t\t\t原名称:%s", user1->val[pos].userName);

printf("\n\t\t\t\t请修改:");

scanf("%s", user1->val[pos].userName);

break;

case 2:

printf("\n\t\t\t\t原账号:%s", user1->val[pos].Account);

printf("\n\t\t\t\t请修改:");

scanf("%s", user1->val[pos].Account);

break;

case 3:

printf("\n\t\t\t\t原账号:%s", user1->val[pos].password);

printf("\n\t\t\t\t请修改:");

scanf("%s", user1->val[pos].password);

break;

case 4:

printf("\n\t\t\t\t名称:%s", user1->val[pos].userName);

printf("\n\t\t\t\t账号:%s", user1->val[pos].Account);

printf("\n\t\t\t\t密码:%s", user1->val[pos].password);

case 5:

break;

default:

printf("\n\t\t\t\t无此选项\n");

break;

}

}

//管理员管理

void Manage(user\* user1)

{

assert(user1);

int input = 0;

int i = 0;

int n = 0;

do

{

printf("\n\t\t\t\t1.查看所有账号 2.修改账号信息 3.删除账号 0.退出\n");

printf("\n\t\t\t\t\t请选择:");

scanf("%d", &input);

switch (input)

{

case 1: //查看所有账号

for (i = 0; i < user1->begin; i++)

{

printf("\n\t\t\t\t%d.名称:%s",i,user1->val[i].userName);

printf("\n\t\t\t\t 账号:%s", user1->val[i].Account);

printf("\n\t\t\t\t 密码:%s\n\n", user1->val[i].password);

}

break;

case 2: //修改账号信息

system("cls");

for (i = 0; i < user1->begin; i++)

{

printf("\n\t\t\t\t%d.名称:%s", i, user1->val[i].userName);

printf("\n\t\t\t\t 账号:%s", user1->val[i].Account);

printf("\n\t\t\t\t 密码:%s\n\n", user1->val[i].password);

}

printf("\n\t\t\t\t请选择要修改的序号:>");

scanf("%d", &n);

if(n>=0&& n<user1->begin)

{

ReviseUser(user1, n); //修改账号信息函数

UserSave(user1); //修改完就保存同步到文件

}

else

{

printf("\n无效选择\n");

}

break;

case 3: //删除账号

system("cls");

for (i = 0; i < user1->begin; i++)

{

printf("\n\t\t\t\t%d.名称:%s", i, user1->val[i].userName);

printf("\n\t\t\t\t 账号:%s", user1->val[i].Account);

printf("\n\t\t\t\t 密码:%s\n\n", user1->val[i].password);

}

printf("\n\t\t\t\t请选择要删除的序号:>");

scanf("%d", &n);

if (n >= 0 && n < user1->begin)

{

for (i = 0; i < user1->begin; i++)

{

if (n+1 == user1->begin) //如果只有一个，或者删除最后一个直接对数组总数减一即可

{

break;

}

else //覆盖删除

{

strcpy(user1->val[i].userName, user1->val[i + 1].userName);

strcpy(user1->val[i].Account, user1->val[i + 1].Account);

strcpy(user1->val[i].password, user1->val[i + 1].Account);

}

}

user1->begin--;

UserSave(user1); //修改完就保存同步到文件

}

else

{

printf("\n无效选择\n");

}

break;

case 0: //退出

break;

default:

printf("\n无此选项，请重新选择\n");

break;

}

} while (input);

}

//账号管理员登录

void Administrator(user\* user1)

{

assert(user1);

char account[MAX\_ACC] = {0};

char password[MAX\_PASS] = {0};

system("cls");

printf("\n\n\t\t\t\t\t管理员登录界面\n");

printf("\n\t\t\t\t\t账号:");

scanf("%s", account);

if (strcmp(account, "admin") == 0) //管理员账号是唯一的，是程序内定的

{

while (strcmp(password,"exit")!=0)

{

printf("\n\t\t\t\t\t密码:");

Getpassword(password, MAX\_PASS);

if (strcmp(password, "123") == 0) //密码也是唯一的

{

Manage(user1); //管理员管理界面

break;

}

else

{

printf("\n\t\t\t\t\t密码错误，忘记请联系开发人员\n");

}

}

}

else

{

printf("\n\t\t\t\t\t账号错误,忘记请联系开发人员\n");

}

}

////找回密码

void GetbackPassword(user\* user1)

{

assert(user1);

char account[MAX\_ACC] = {0};

//system("cls");

printf("\n\t\t\t\t\t输入你的账号:");

scanf("%s", account);

int input = 1;

int i = 0;

for (i = 0; i < user1->begin; i++) //查找对应的账号

{

if (strcmp(account, user1->val[i].Account) == 0)

{

char UserName[MAX\_NAME] = { 0 };

input = 1; //接收退出条件

while (input)

{

//system("cls");

printf("\n\t\t\t\t\t请输入你注册时的名称:"); //注册时的名称作为找回密码的准则

scanf("%s", UserName);

if (strcmp(UserName, user1->val[i].userName) == 0)

{

system("cls");

Print1(7);

printf("\n\t\t\t\t\t账号:");

printf("%s", user1->val[i].Account);

printf("\n\t\t\t\t\t密码为:");

printf("%s", user1->val[i].password);

printf("\t\t\tPS:按0退出此界面\n");

printf("\n\t\t\t\t\t");

scanf("%d", &input);

system("cls");

break;

}

else

{

printf("\n\t\t\t\t\t跟你刚注册时的名称不符\n");

Print1(7);

printf("\t\t\tPS:按1继续，按0退出此界面\n");

scanf("%d", &input);

}

}

}

if (input == 0)

{

system("cls");

return;

}

}

system("cls");

printf("\n\t\t\t\t\t无此账号\n");

Sleep(400); //让程序停400ms

}

//登录系统

int UserLongsonSystem(user\* user1)

{

int p = 0;

int input = 0;

int pos1 = 0;

do {

system("cls");

menu1();

Print1(3);

printf("\t\t\t\t\t请选择>:");

scanf("%d", &input); //接收用户选择的功能

switch (input)

{

case 1: //登录

pos1 = UserLongin(user1);

if (pos1 == -1)

{

continue;

}

else

{

input = 0;

}

break;

case 2: //注册

system("cls");

printf("\n\t\t\t已有%d个账号在系统，最多可以再注册%d个账号，满了请联系管理员增加\t\t\t\t\t\n\n", user1->begin, N - user1->begin);

p = CheckUser(user1);

if (p != -1)//账号未满进入注册

{

Userreg(user1);

UserSave(user1);

}

break;

case 3: //找回密码

GetbackPassword(user1);

break;

case 4: //管理员登录

Administrator(user1);

break;

case 0: //退出程序

UserSave(user1);

printf("\t\t\t\t\t退出成功\n");

return -1;

default: //选择错误提醒

system("cls");

printf("\t\t\t\t\t无此选项，请重新输入\n");

break;

}

} while (input);

return pos1;

}

//按名字排序通讯录

int cmp\_name(const void\* e1, const void\* e2)

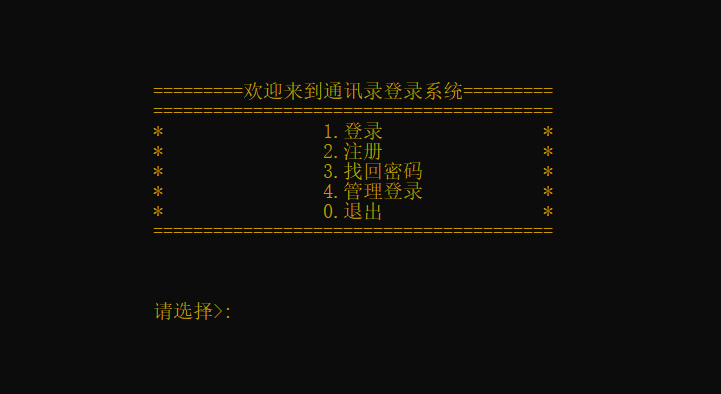
{

return strcmp(((people\*)e1)->name, ((people\*)e2)->name);

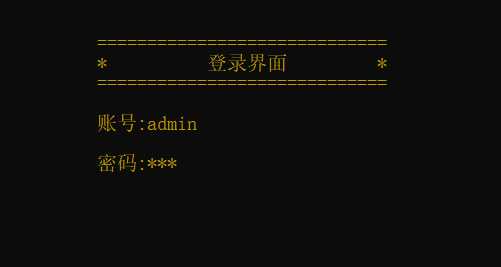
}

五、程序运行的测试与分析

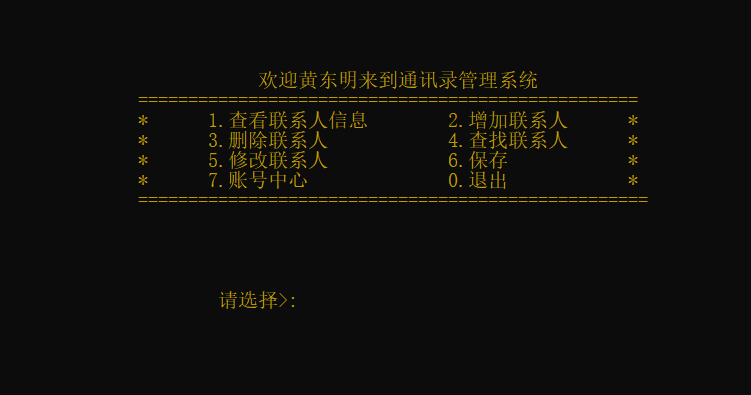
初始界面

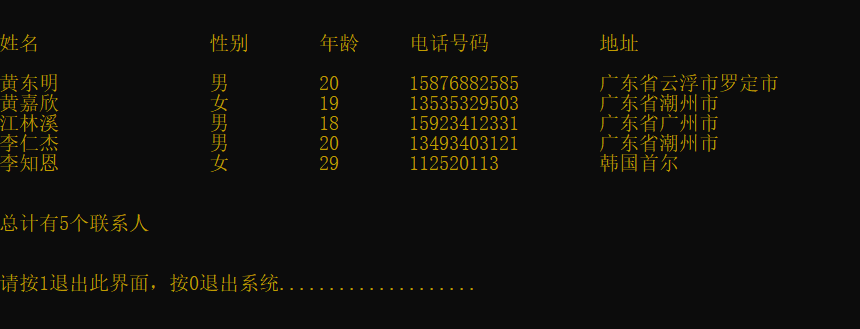


登录界面

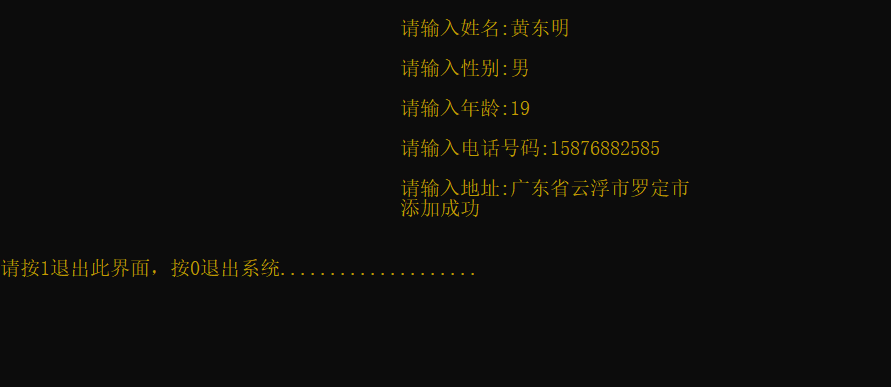


主界面

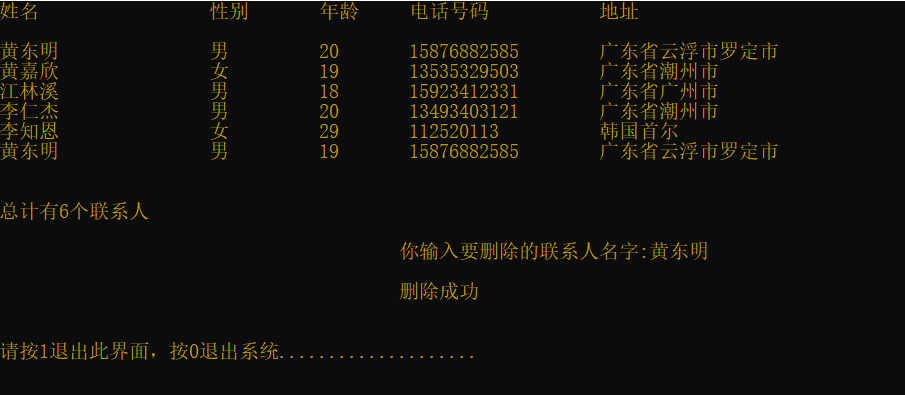


功能1（打印联系人信息）：

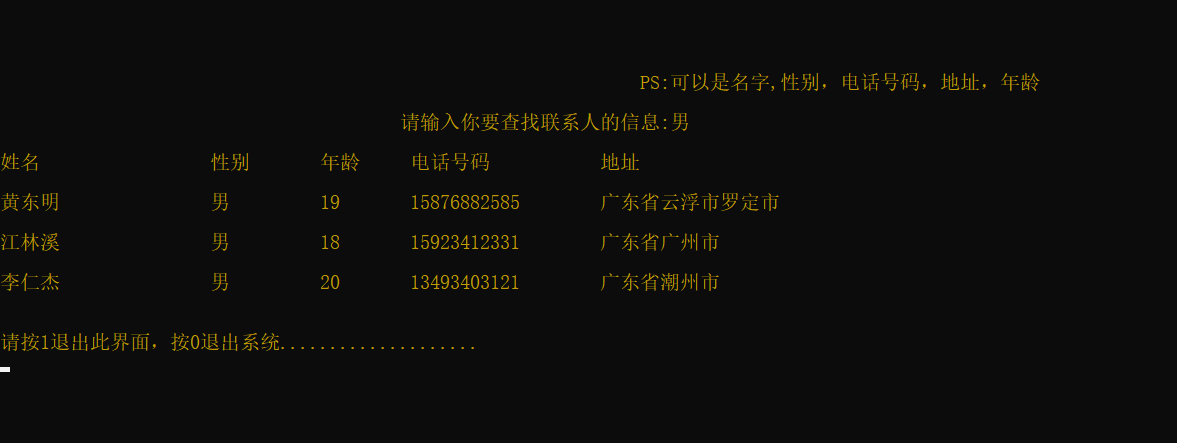
功能2（增加联系人）：



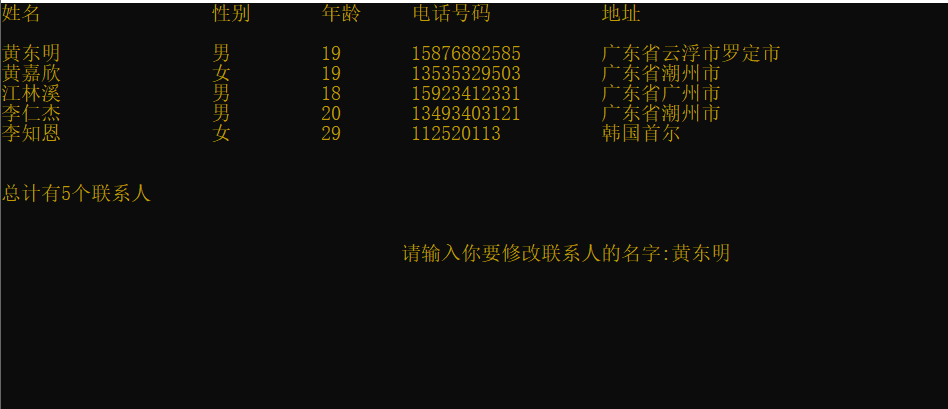
功能3（删除联系人）

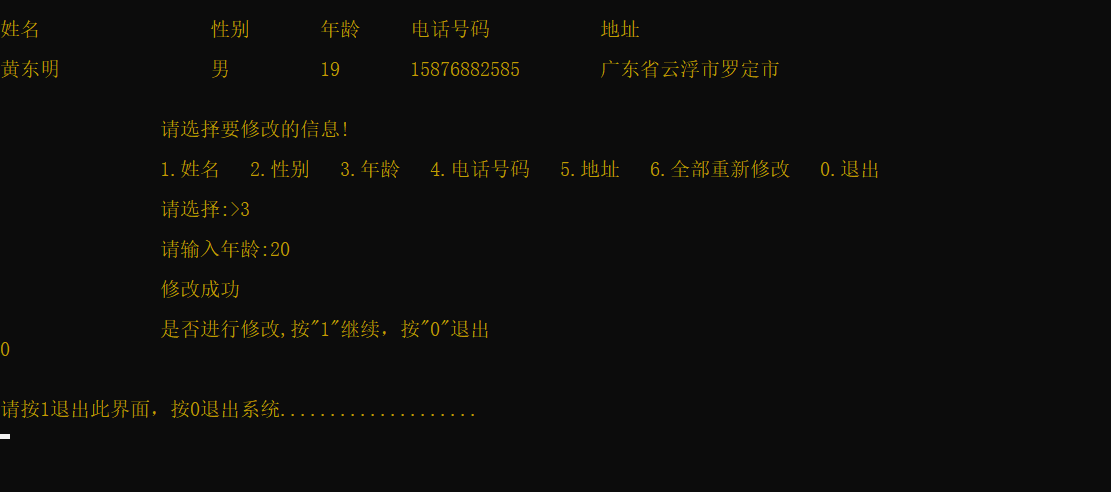


功能4（查找联系人）：

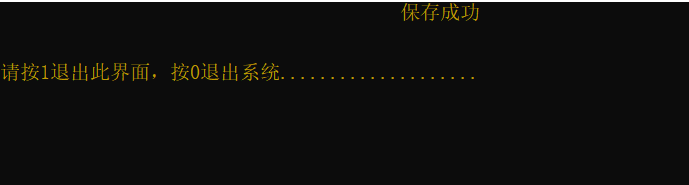


功能5（修改联系人）：

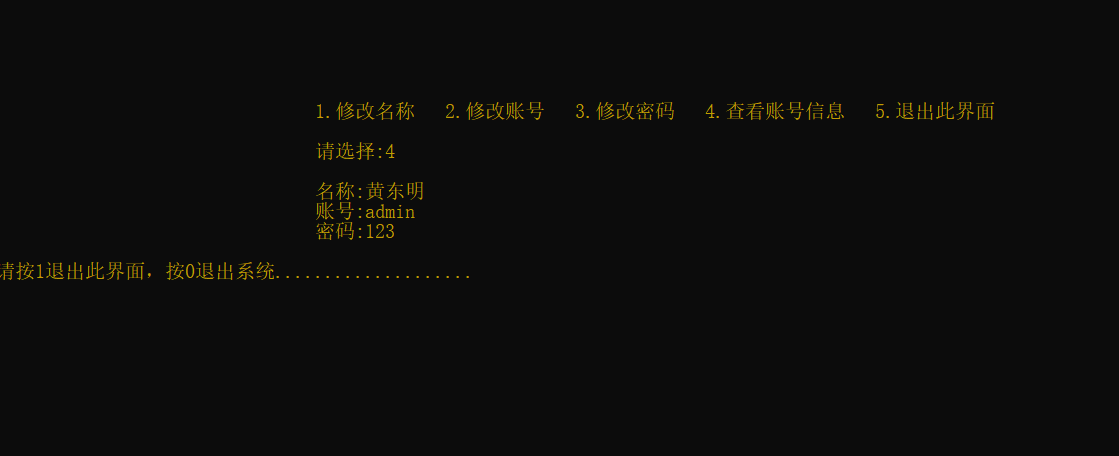




功能6（保存到文件中）：



功能7（账号中心）：



分析：经过测试，该程序未发现严重错误。能以菜单方式工作，能够正常且无误地完成账号的注册与登录、信息录入、信息删除、信息查询、信息修改、信息浏览、系统退出等一系列基本功能。

六、结论与心得

该系统能完成一些基础功能，大部分功能是没有问题的，但可能还存在一些潜在问题暂时没有显露出来。

虽然采用的数组是动态内存开辟空间，但在内存不够进行扩容时可能会出现部分空间浪费。然后虽然是具有登录功能，但却是多对一的方式进行管理，也就是通讯录的数据在各个账号是统一的，是多个账号进行管理一个通讯录系统，往后写类似的操作时尽量做到一对一管理，让每个账号都能有一个自己管理独自数据的系统。

最后一个就是最能通过键盘输入的方式进行操作，对用户来说并不友好。

心得体会：

巩固了C语言语法基础

理解和学会程序模块化思想

提高了对代码的编写能力

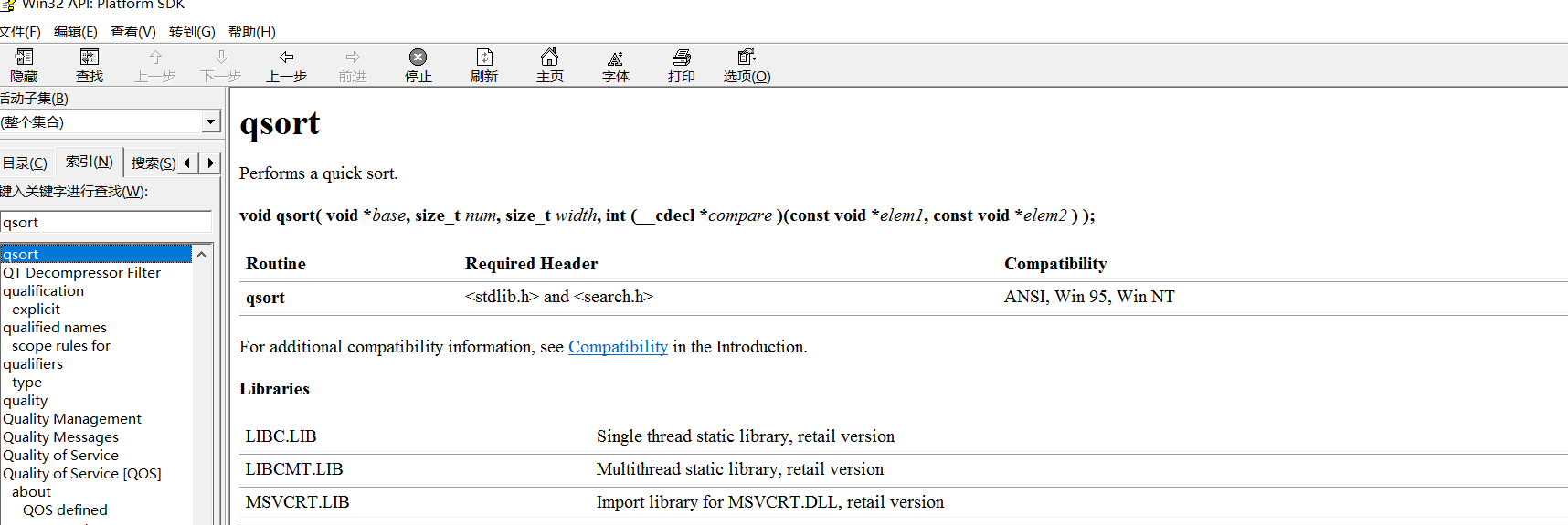
学会和理解文件操作

加深自己在程序设计的思想

七、参考资料

MSDN（一款可以查看库函数使用方法介绍的软件）

如：C语言库函数中的qsort函数使用介绍



八、致谢

感谢陈建孝老师的辛苦指导