## 一、开发任务

电话簿的简单管理：编程模拟实现手机电话簿的管理。

## 需求分析

## 手机通讯录系统任务的基本功能是：

（1）通信录联系人的增加。

（2）通信录联系人的删除

（3）通信录联系人的查询。

（4）通信路联系人的修改。

（5）实现从文件读取数据和保存数据到文件。

（6）实现联系人的分组查询。

### **程序中有如下特性：**

（1）用菜单方式提供对操作的选择。

（2）实现从文件读取数据和写数据到文件，程序会默认读取Contact.txt文件里面的内容，将数据存储到内存。

## 程序设计

## **针对需求做如下设计：**

## **（1）数据结构设计**

typedef struct ADDRESS

{

char name[10];

char tele[12];

char add[50];

char group[50];

}ADDRESS;

使用struct关键字创建ADDRESS结构体，成员为name名字，tele电话，add地址，group分组。然后使用typedef创建ADDRESS类型。

## **函数模块设计**

**文件数据获取函数int Get(char \*fileName)**

使用Get()函数从文件读取数据到内存，伪代码如下,

int Get(char \*fileName)

{

打开文件fileName;

使用while循环读取文件;

将数据存储到数组;

return 1;

}

该函数使用fopen打开数据文件，使用while和fgets循环不断从文件读取数据，然后使用sscanf将数据存储到结构体中，最后使用Trans函数将数据传输到数组中。

**使用Trans()函数进行数据的传输**，伪装代码如下，

void Trans(ADDRESS a) //传输函数，将结构体ADDRESS信息存入结构体数组Capa中

{

将参数数据复制到全局数组;

排序;

}

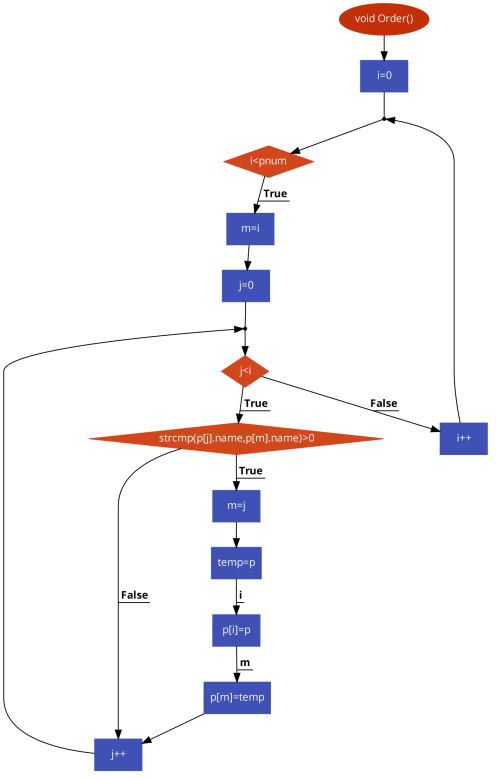
函数将参数传来的变量a，赋值给全局变量Capa数组，每次都是全局变量TotalNum自加，最后使用Order函数对数组进行排序。

**使用Order()函数对数组进行排序，该排序按照姓名排序**，

函数原型，void Order(ADDRESS \*p,int pnum) ,p为ADDRESS的指针，pnum为数组的大小。

该函数通过两个for循环进行冒泡排序，使用string.h的库函数strcmp进行比较，如果大，则进行交换。

程序流程图如下，



**使用Search\_Name函数，对指定姓名的联系人进行排序。**

函数原型int Search\_Name(char a[],int num[]) ，a为数据数组，num为数组大小。

伪代码如下，

int Search\_Name(char a[],int num[]) //按姓名逐个查找

{

for(循环扫描数组)

{

if(姓名与要查找的姓名相等)

修改数组;

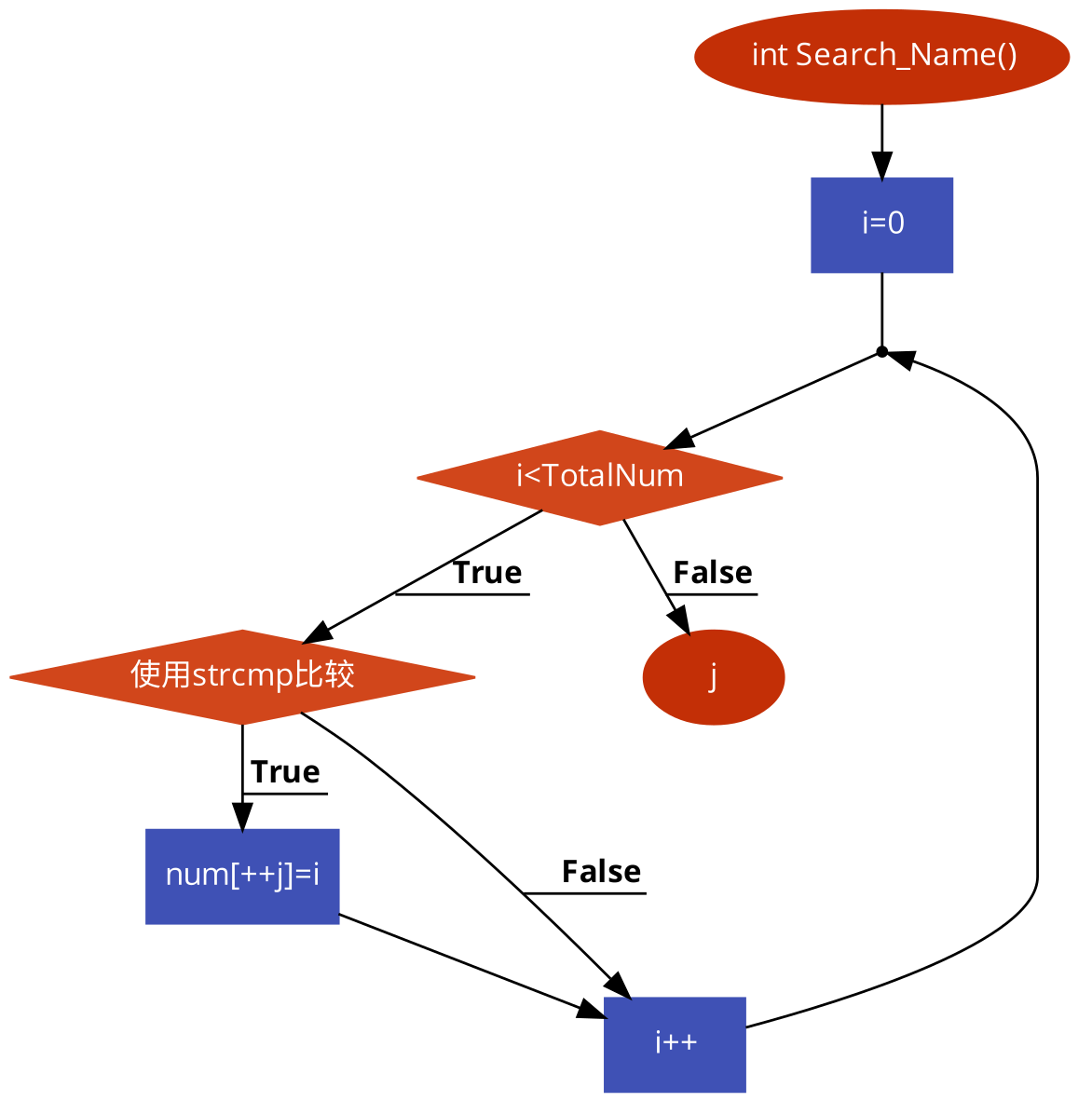
}

return 找到的下标;

}

该函数使用for循环遍历数组里面的所有成员，使用strcmp进行比较，返回对应下表。

程序流程图如下，



**使用Search\_Group函数，对指定组进行分组查询。**

函数声明int Search\_Group(char a[]) ，a为数据数组。

使用for循环遍历数组，然后使用strcmp进行对比，程序源代码如下，

int Search\_Group(char a[]) //按群组逐个查找

{

for(遍历整个数组)

{

if(字符串比较)

{

修改数组;

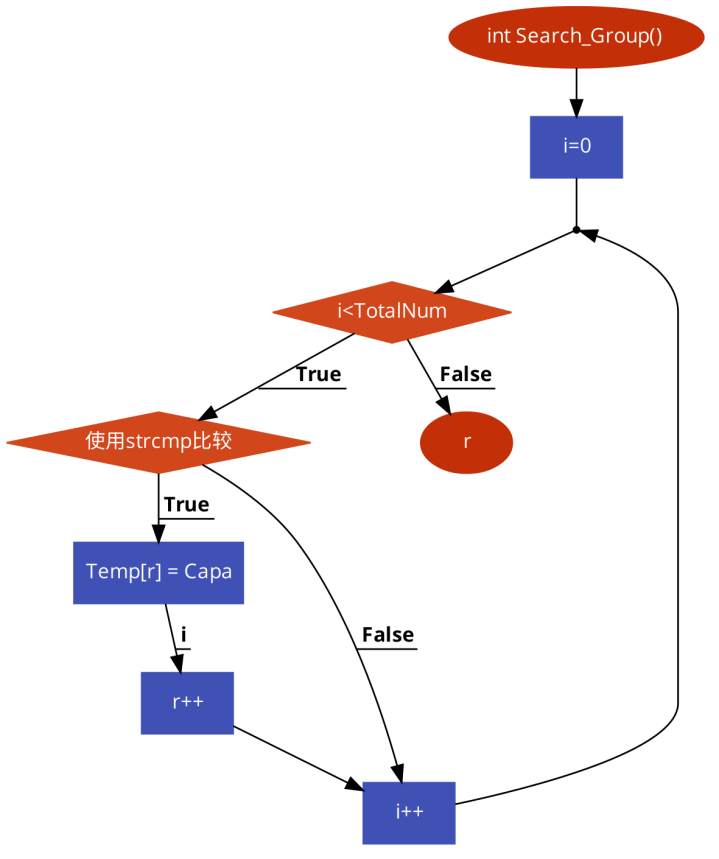
}

}

返回下标;

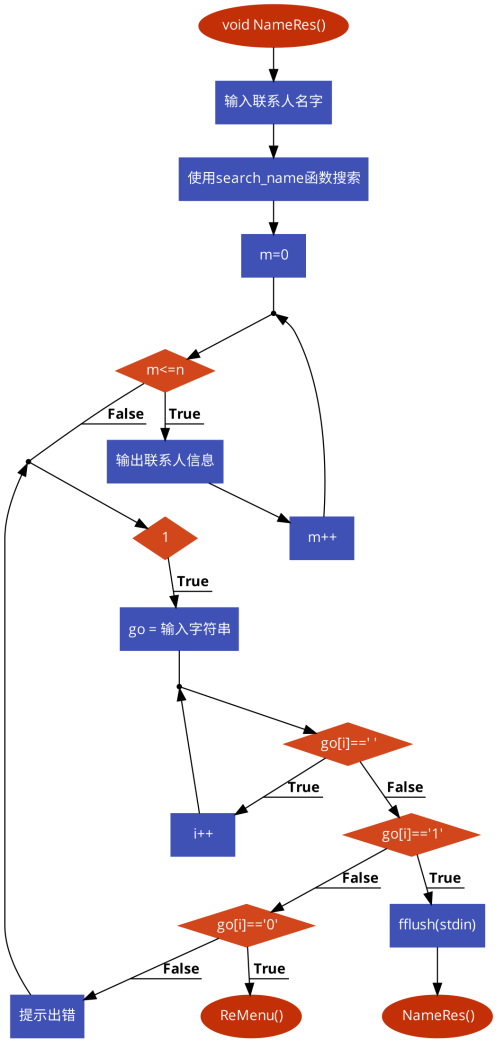
}

程序流程图如下，



使用void NameRes() 函数进行名字搜索，函数原型如下，

void NameRes() ，该函数无参数，程序流程图如下，

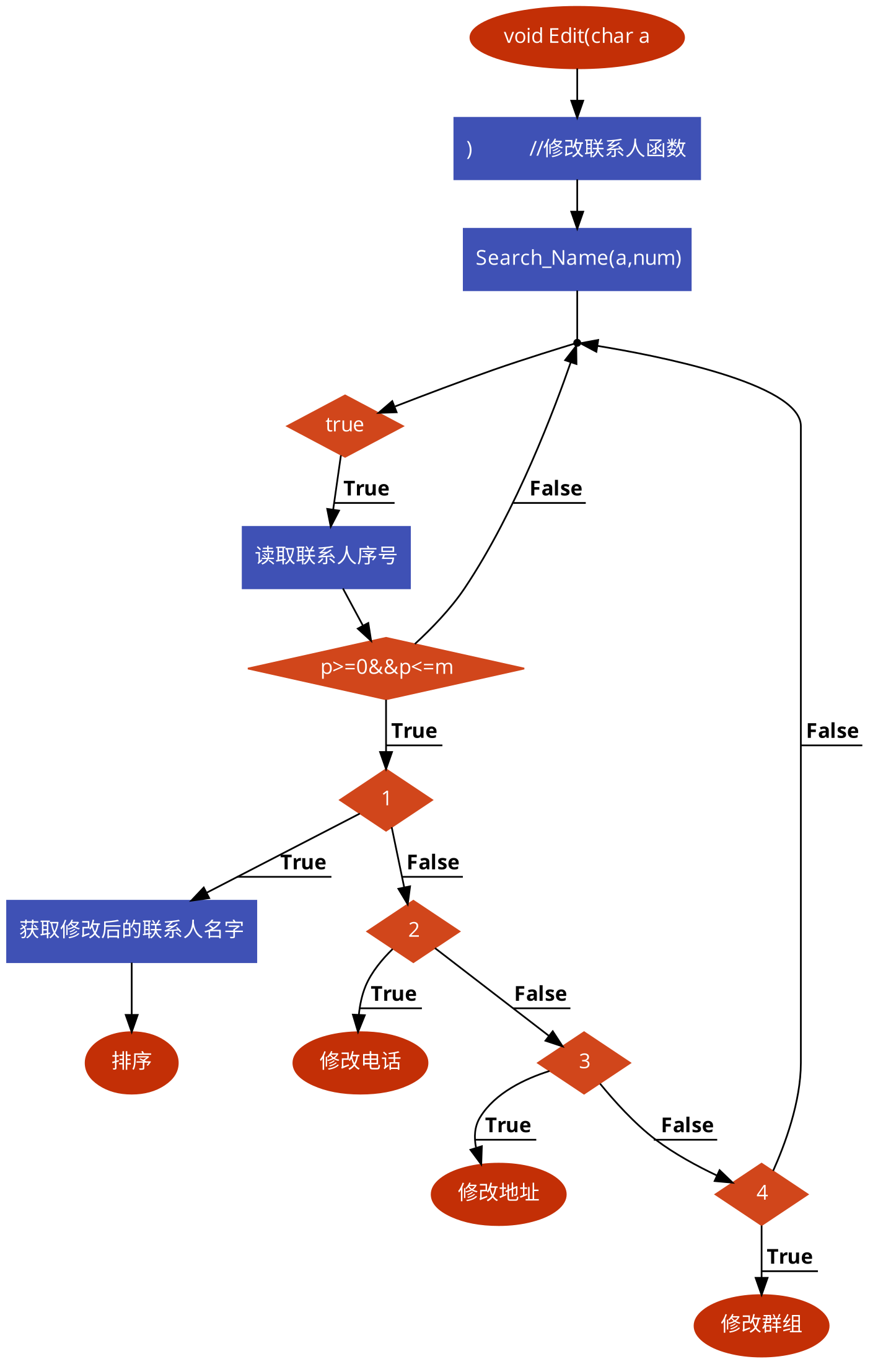


**使用edit函数进行对联系人的编辑，函数原型如下**，

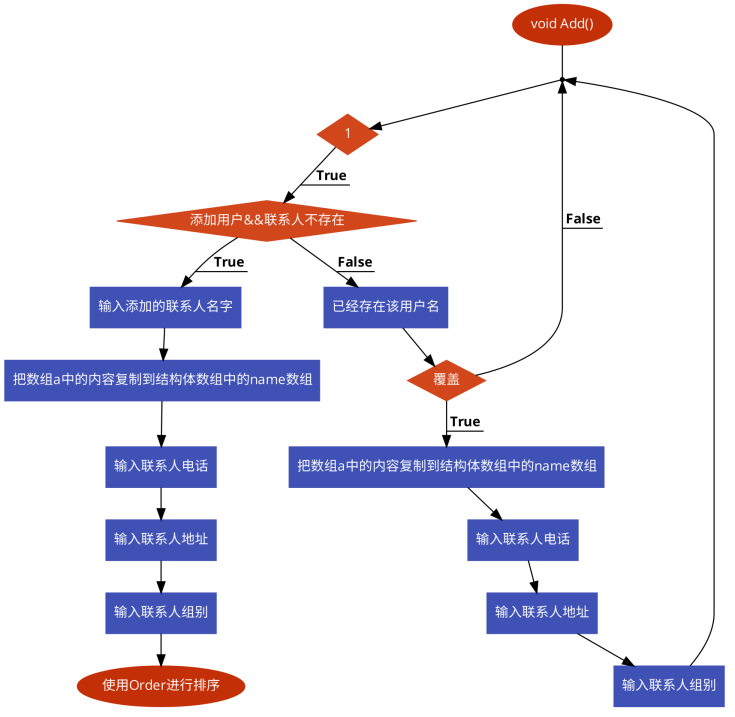
void Edit(char a[])

其中a为数据数组

该函数使用search\_name函数查找到对应用户，然后进行相应的修改，函数流程图如下，



**使用add函数进行对联系人进行添加，该函数无参数，程序流程图如下，**

****

**使用AddressList()函数输出欢迎信息，**

**int AddressList()**

**{**

**puts(" --------------------------------------------------------");**

**puts("");**

**puts(" cxh的手机通讯录 ");**

**puts("");**

**puts(" --------------------------------------------------------");**

**ShowAll(Capa,TotalNum);**

**}**

**使用Munu函数输出菜单选项，伪代码如下，**

**int Menu() //用户菜单界面**

**{**

**printf(" ｜ 功能列表如下： ｜\n");**

**printf(" ｜ 1.按照群组查询联系人信息； ｜\n");**

**printf(" ｜ 2.按照姓名查询联系人信息； ｜\n");**

**printf(" ｜ 3.添加新的联系人信息； ｜\n");**

**printf(" ｜ 4.删除指定的联系人信息； ｜\n");**

**printf(" ｜ 5.删除全部的联系人信息； ｜\n");**

**printf(" ｜ 6.查询全部联系人信息； ｜\n");**

**printf(" ｜ 7.逐个显示全部联系人信息； ｜\n");**

**printf(" ｜ 0.退出该系统并保存文件； ｜\n");**

**printf(" 按数字选择功能:");**

**输入选项;**

**If (0<选项<=7) {**

**}**

**else**

**{**

**fflush(stdin);**

**printf(" 提示：输入指令不合法，请重新输入！\n");**

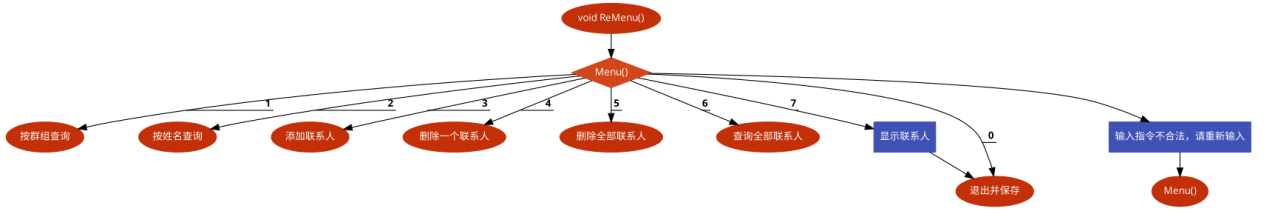
**Menu();**

**}**

**}**

**使用ReMenu函数，进行菜单操作的重选，**

**函数原型如下void ReMenu，程序流程图如下，**

****

**使用save函数进行对联系人进行保存，函数原型如下，**

int Save(char \*fileName) ，其中fileName为要保存的文件名称，

该函数使用for循环遍历所有数组，使用fprintf函数将其写入文件，伪代码如下，

int Save(char \*fileName) //保存至文件函数

{

打开文件;

if(打开文件 != NULL)

{

return 0;

}

int i = 0;

while(i<TotalNum)

{

数据写在文件中;

i++;

}

关闭文件;

return 1;

}

**使用showAll函数进行对联系人显示，函数原型如下，**

void ShowAll(ADDRESS \*p,int pnum) //显示全部联系人信息

其中，p是ADDRESS类型的指针，指向数组的数据，pnum是数组的大小，

该函数通过for循环依次输出数组里面的数据，伪代码如下，

void ShowAll(ADDRESS \*p,int pnum) //显示全部联系人信息

{

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

{

输出联系人信息p[i];

}

}

使用ShowOne函数查询联系人，函数原型如下，

void ShowOne(ADDRESS \*p,int pnum), 伪代码如下，

void ShowOne(ADDRESS \*p,int pnum)

{

输入要删除的联系人名字;

For( int i = 0; i < pnum; i++) {

If (p[pnum] == name) {

输出联系人信息;

}

}

}

使用DelectAll函数删除所有联系人，函数原型如下，

void DeleteAll() ， 伪代码如下，

Void DeleteAll()

{

将联系人数组清空;

将pnum重置为0;

}

**使用DelectOne函数进行对联系人删除，函数原型如下，**

void DelectOne()

{

请输入要删除的联系人的姓名;

n = 搜索用户Search\_Name;

if(n==-1)

{

不存在该联系人;

break;

}

else

{

if (确认删除) {

将数组的每个数据往前移动;

数组数量-1;

}

}

**该程序由main函数驱动，main函数使用switch case语句进行用户输入的判断，然后执行相应的函数**，main函数源代码如下，

int main()

{

获取文件数据;

对数据进行排序;

输出所有联系人;

switch()

{

case 1:

按群组查找;

break;

case 2:

按姓名查询;

break;

case 3:

添加联系人;

break;

case 4:

删除指定联系人;

break;

case 5:

删除全部联系人;

break;

case 6:

显示全部联系人

break;

case 7:

逐个显示全部联系人;

break；

case 0:

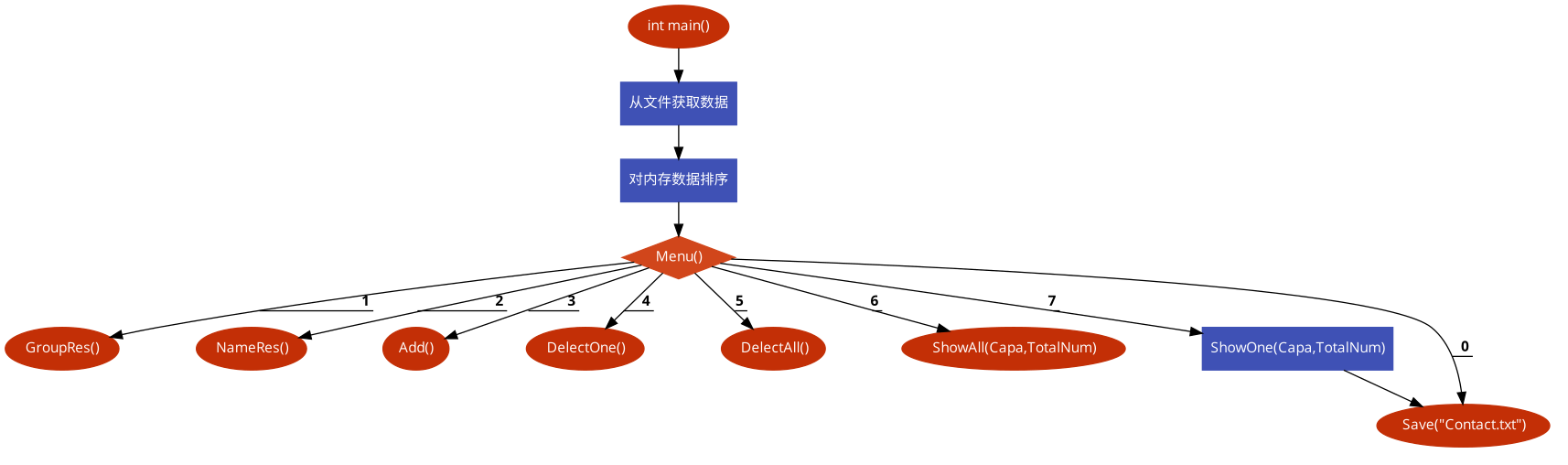
退出并保存;

break;

}

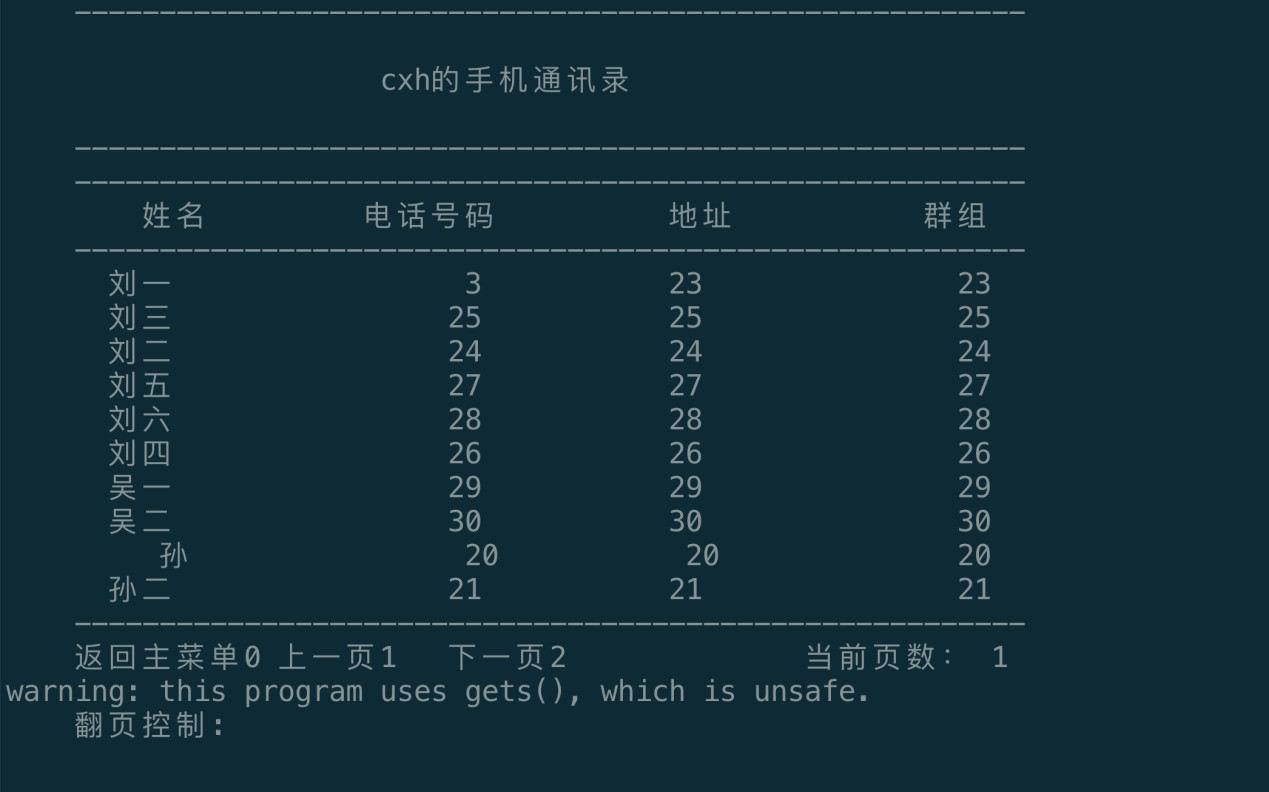
}

main函数程序流程图如下，

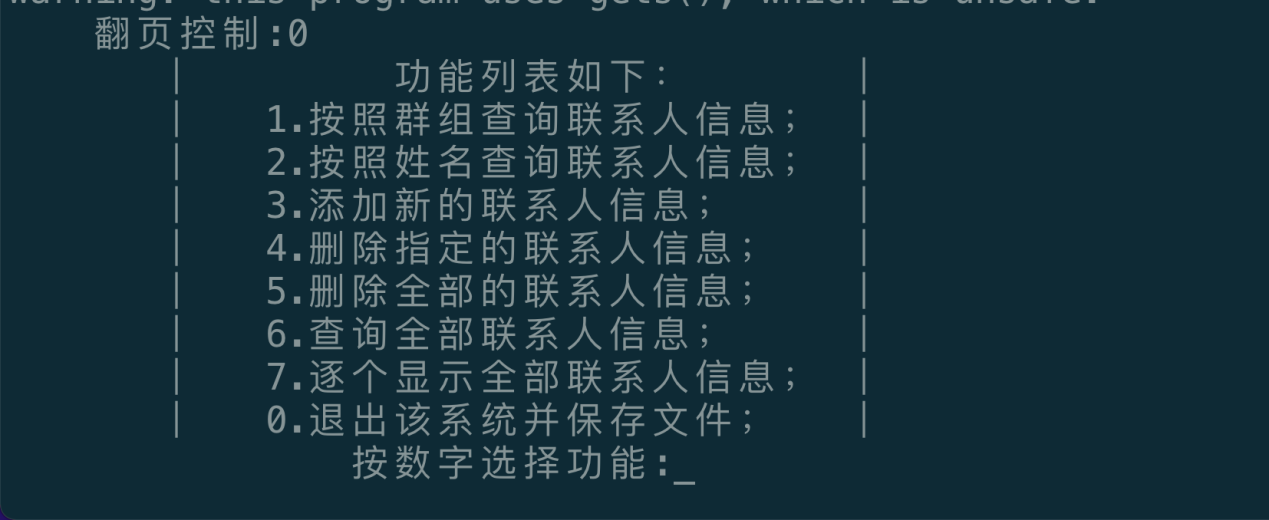


**四、运行结果**

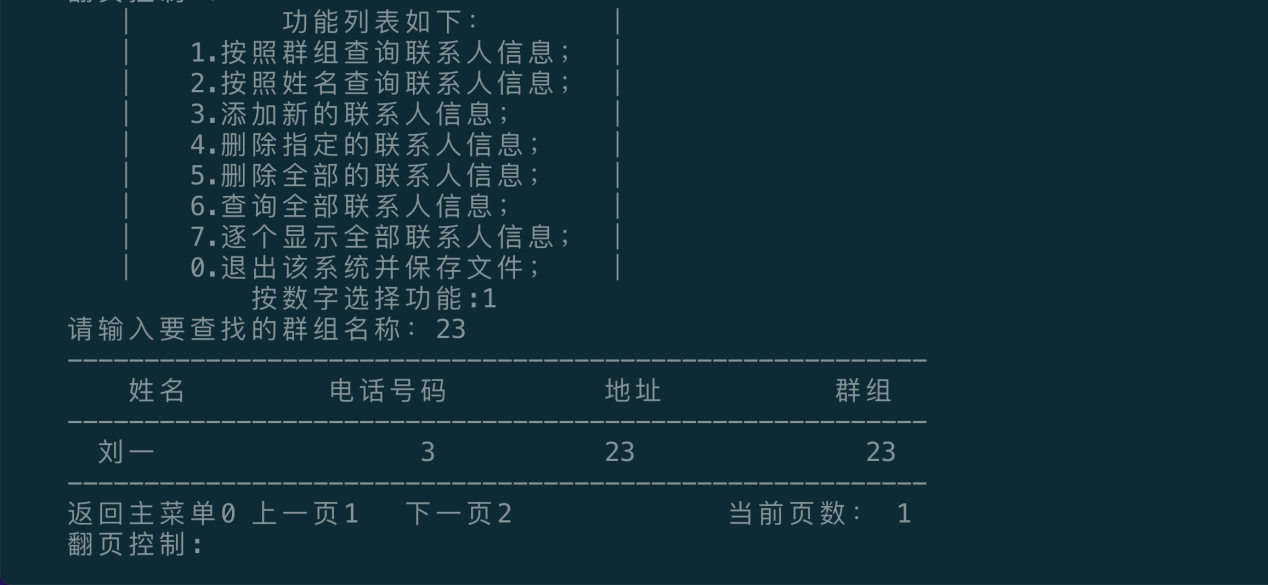
（1）程序主界面



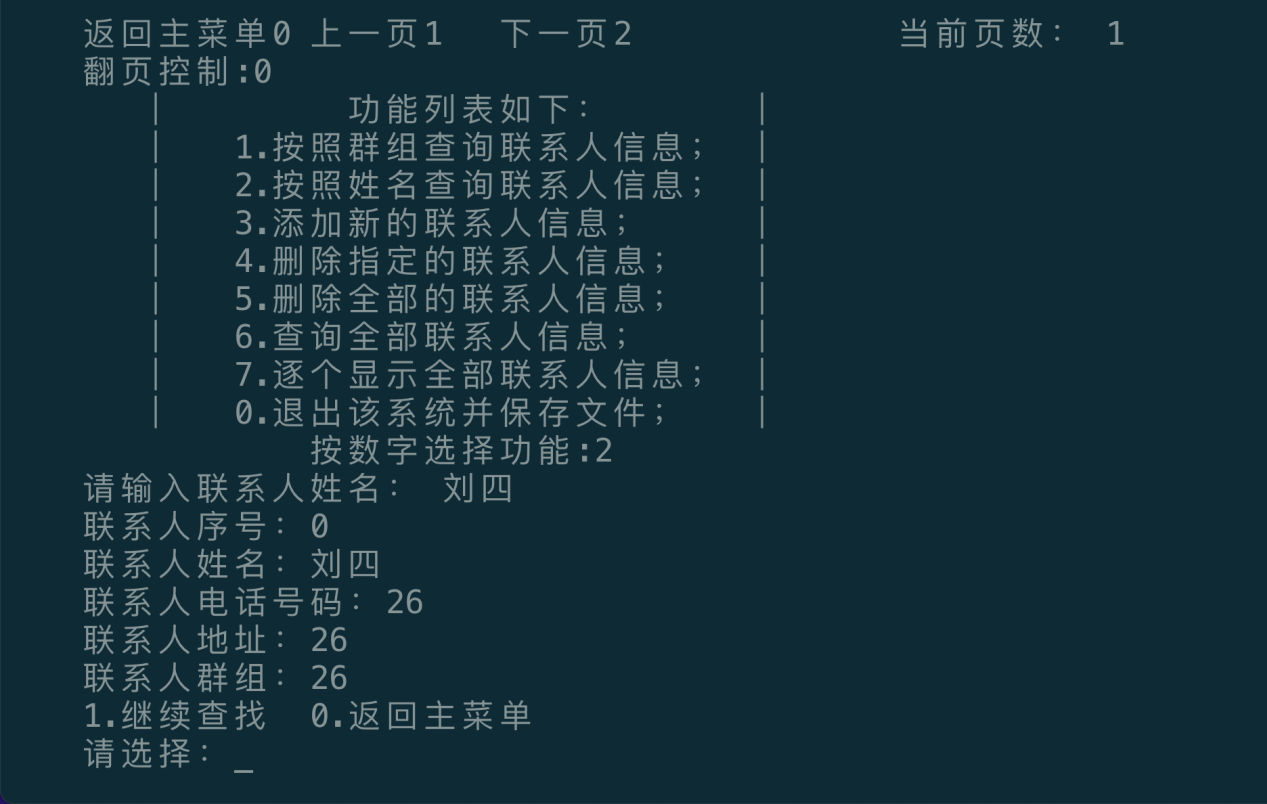
1. 操作菜单栏



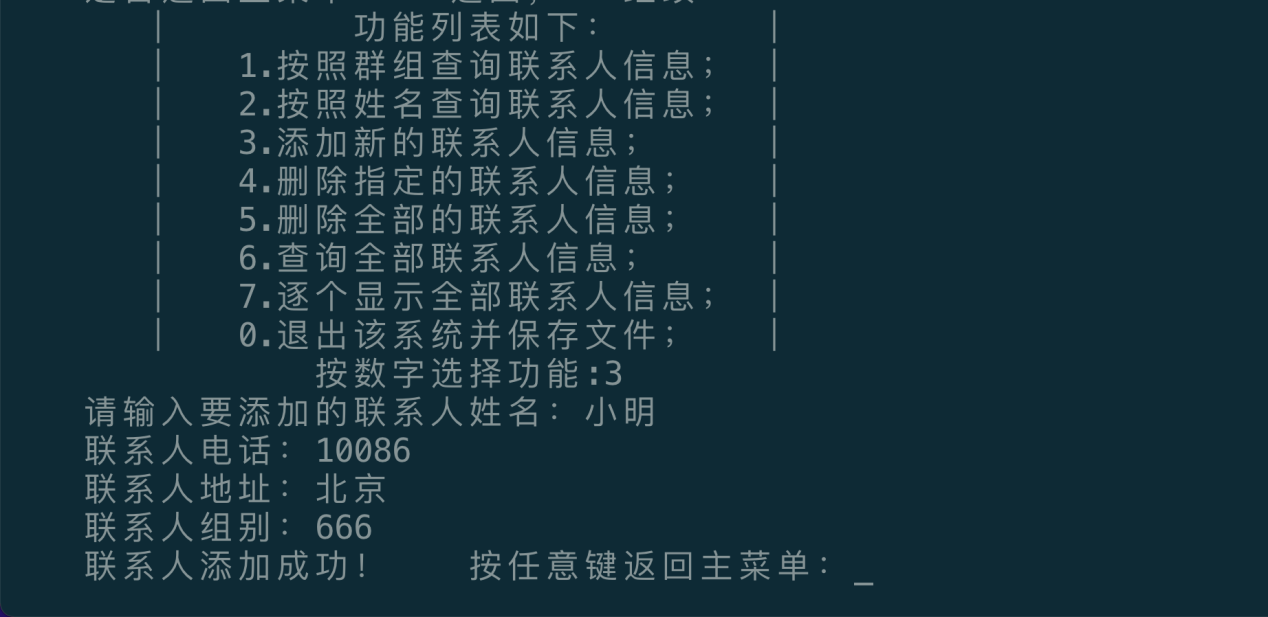
1. 按组查询



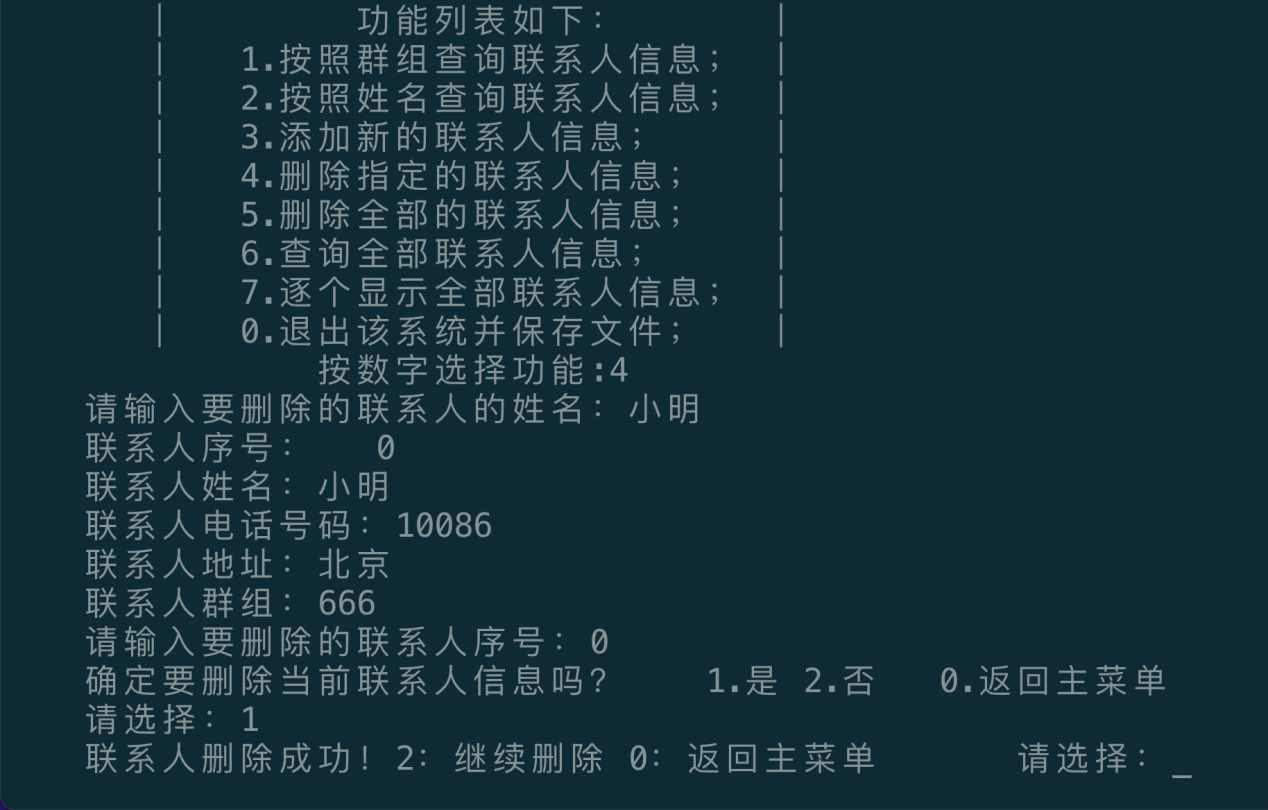
1. 按姓名查询



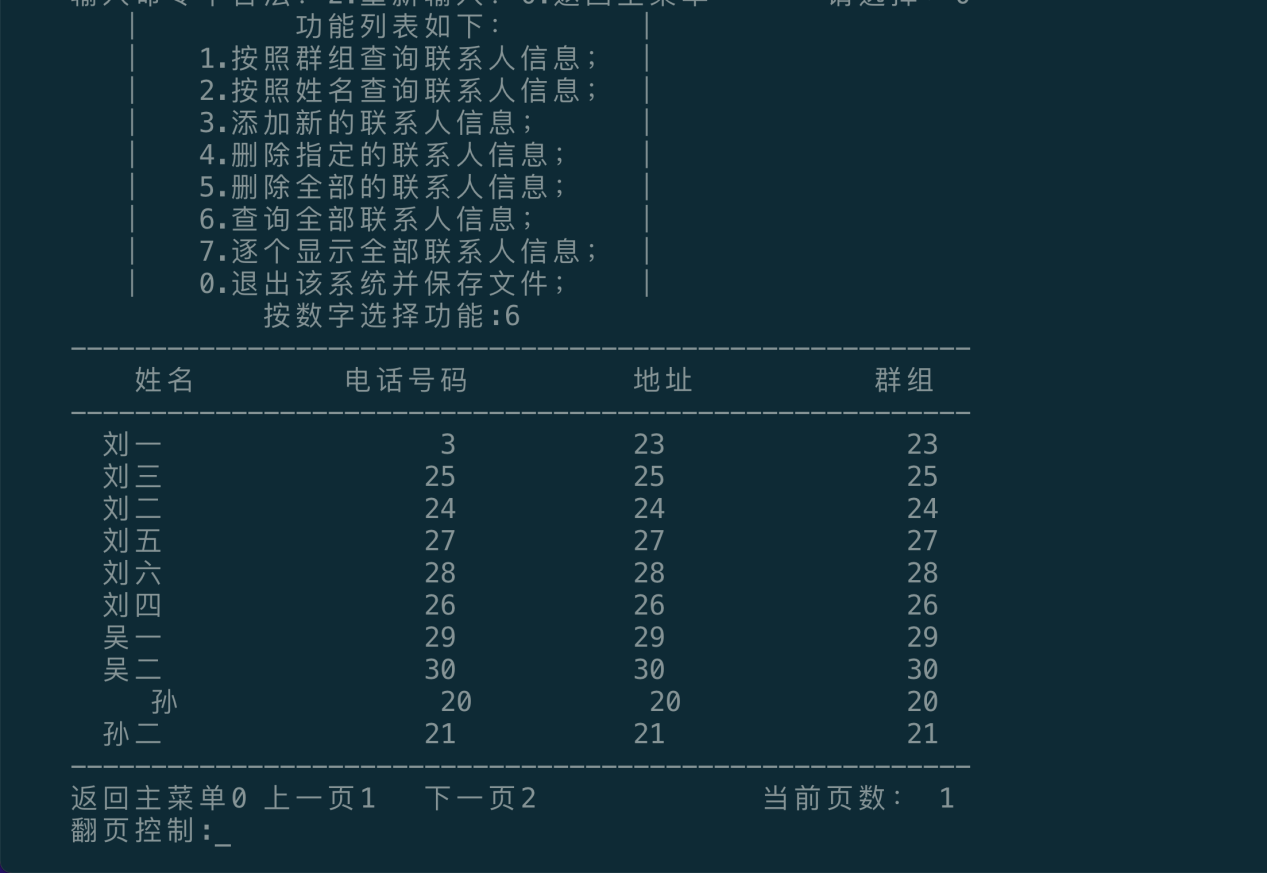
1. 新建联系人



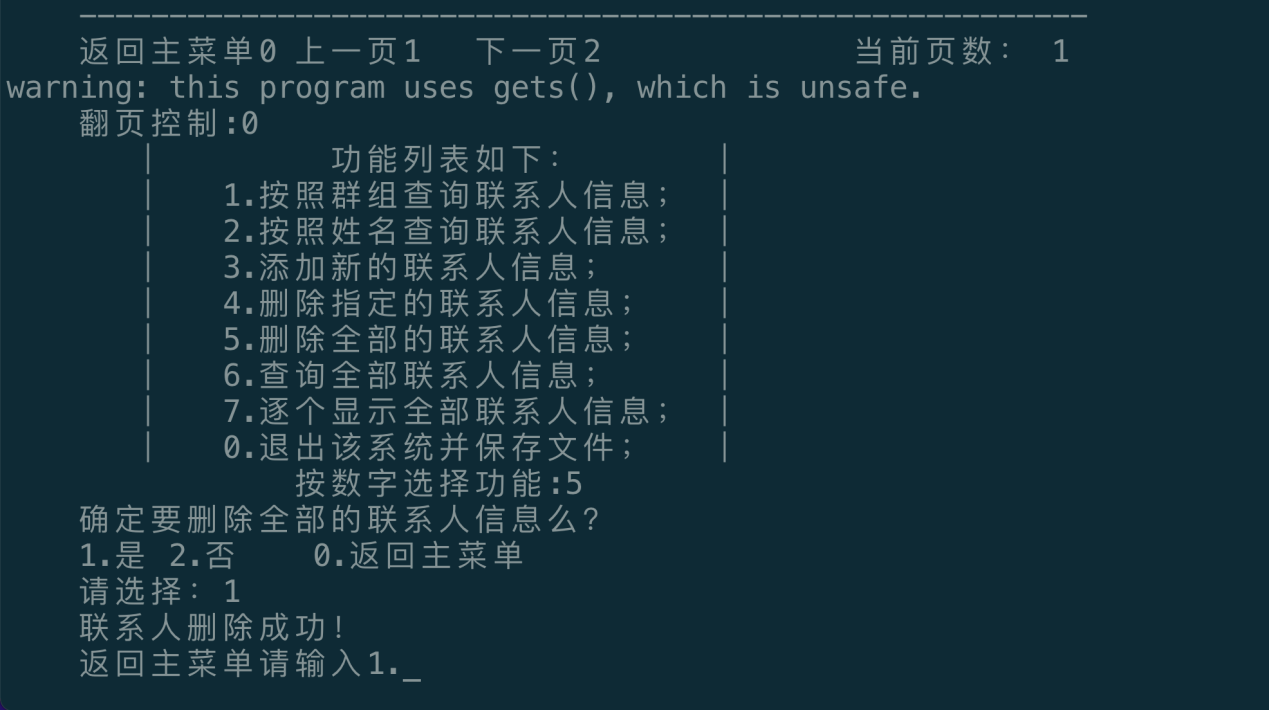
1. 删除联系人



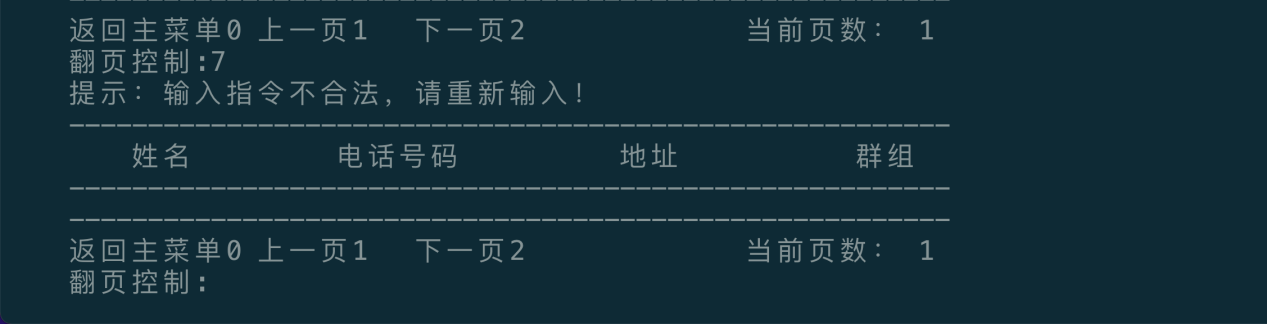
1. 查询所有联系人



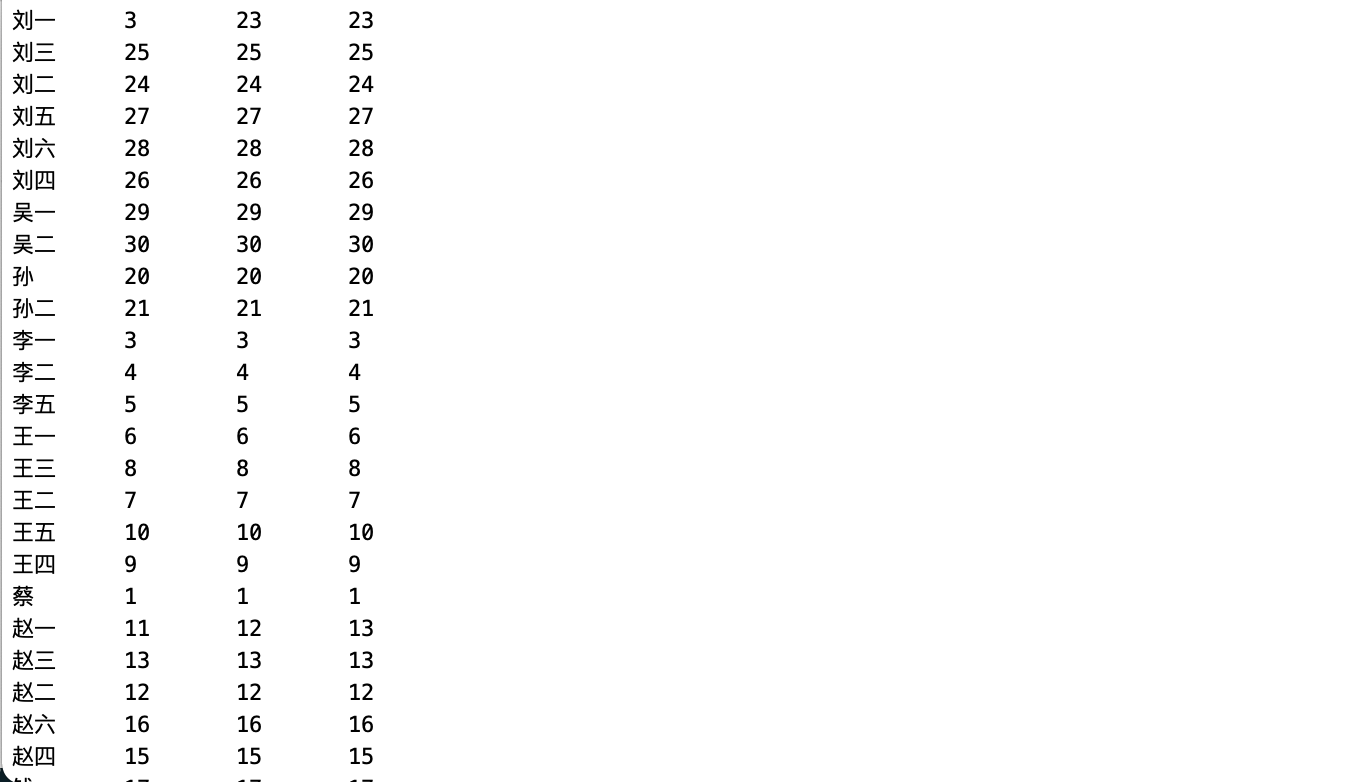
删除全部联系人



逐个显示全部联系人



1. 保存到文件



**五、设计小结**

本任务设计的代码完成基本功能，但是有几个缺点：

（1）电话号码簿信息太简单

现在的手机功能越来越全，人们之间的联系方式也原来越多，电话号码簿内容更丰富，

如：姓名，手机号码，家庭号码，办公电话，邮箱地址，QQ 号码等。

（2）界面不友好

程序应该设计成具有图形界面的程序，单纯的命令行的程序，一般用户比较难操作，我们可以使用qt或者mfc等框架设计出GUI的程序。

（3）安全性问题

程序的所有数据都是明文保存，且没有账号保护机制，最好的情况是设置登录界面，然后将数据进行AES或者DES加密保存。