**哈尔滨工程大学**

**《计算思维二》实验报告**

**综合实践**

小组成员：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓名** | **班级** | **学号** | **分工** | **成绩** |
| 周光煜 | 2011 | 2019201131 | 60% |  |
| 赵文强 | 2011 | 2019201130 | 40% |  |

实验时间: 2020 年 6 月 29 日

**哈尔滨工程大学计算机基础课程教学中心**

**实验题目：**通讯录管理系统

**设计思想：**设计一个通讯录系统，包括联系人数据的增删改查等基本操作，程序运行后先显示如下菜单，并提示用户输入选项：

1.添加信息

2.查询信息

3.删除信息

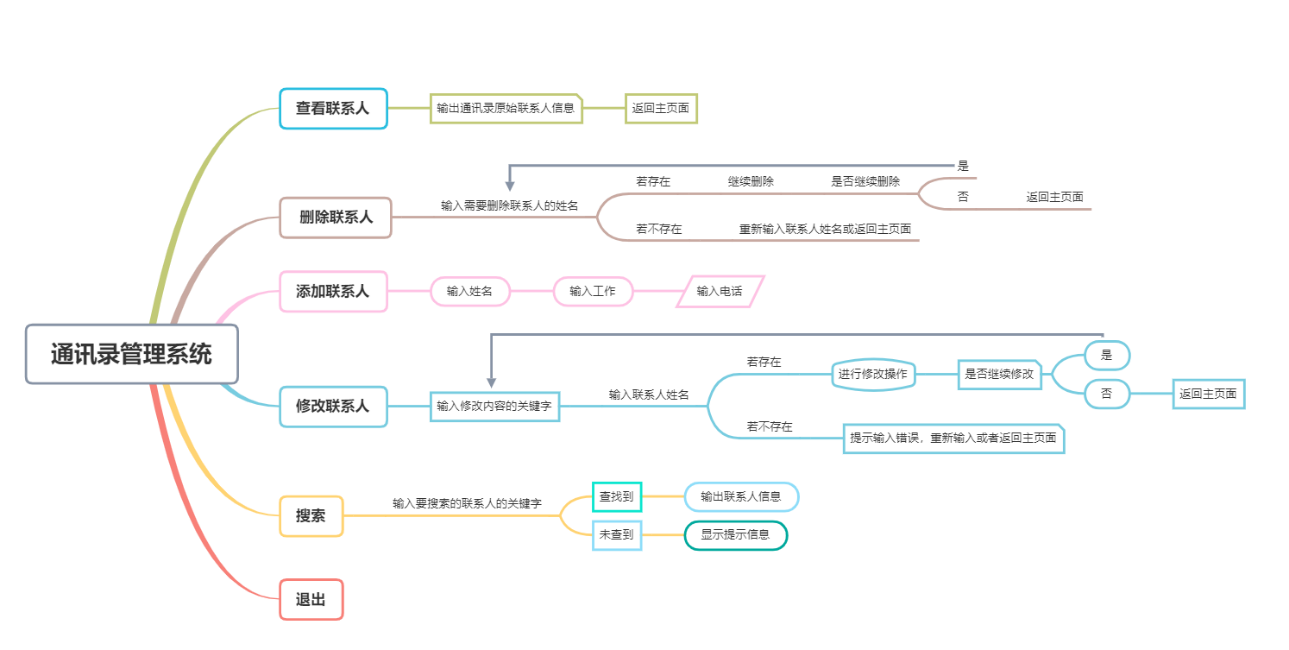
4.修改信息

5.查看通讯录

0.退出

然后根据用户输入的选项执行相应的操作。

为了方便理解，程序的总体设计思想见下图：



**成员任务分工**：

周光煜：完成主函数的构建，以及完成添加联系人、 查询联系人的操作，设计完成一些辅助函数。

赵文强：完成删除联系人、修改联系人信息和查看通讯 录的操作。

**运行环境：**操作系统：Windows 7；开发环境：Dev-C++

**编程语言：**C语言

**成员1的工作：**完成主函数的构建，以及完成添加联系人、查询联系人的操作，设计完成一些辅助函数。

**姓名：**周光煜

**学号：**2019201131

**任务分工：**完成主函数的构建，以及完成添加联系人、查询联系人的操作，并且设计完成一些辅助函数。

**设计思想：**利用while语句和switch语句设计完成主函数；在Input函数里输出主界面以及功能选择；在SumCount函数里完成对count参数（联系人的个数）的计算；在Getchar函数里，获得一个字符，此函数可以排除回车键对输入字符的影响；在Tishi函数里，提示用户输入b返回主界面；在Tips函数里，输出联系人的个数并输入一个字符；在AddInfo函数里完成对联系人的添加功能；在FindInfo函数里，完成对联系人的查找功能。

**代码行数：**1-307行，共295行(去除空行)

**实验代码及注释：**

char Input()：通过此函数输出主界面，提示用户选择功能选项，并判断用户输入的选项是否符合规定，若符合规定则继续，若不符合规定则提示输出错误。

void SumCount()：此函数的作用是计算出count的值，count代表的是当前通讯录中联系人的数量，函数中利用文件的知识，当指针移动到文件末尾时，将此时的值赋给count。

char Getchar()：此函数的作用是得到用户输入的字符，在调试的过程中，有时系统会将回车键误认为是输入的字符，因此这个函数的作用是排除回车键对字符输入的影响。

void TiShi()：此函数的作用是提示用户输入字符，若字符为‘b’则返回主界面，若不是则提示输入错误。

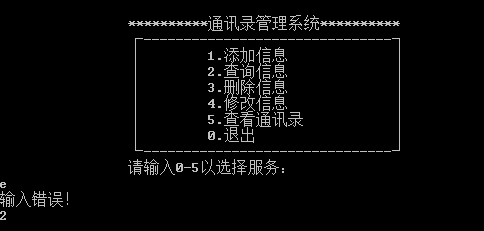
void Tips(int n)：此函数的作用是在“查看通讯录”操作中，提示用户当前有几个联系人，并提示用户如何返回主界面。

void AddInfo()：此函数的作用是新建联系人，用户只需要根据系统提示输入相应信息即可，并且系统会判断用户输入的号码是否符合规范（位数是否为11位，手机号码中是否掺杂有其他字符），并将新建联系人信息写入结构体数组和文件中，并提示用户是否继续添加。若继续则循环之前步骤，若不继续添加则退出。

void FindInfo()：此函数的作用是查询联系人信息，并且用户可以选择通过“姓名”和“手机号”进行查找，若通过姓名查找，若存在此联系人则输出联系人的其他信息，若不存在则输出“该联系人不存在”；做通过手机号查找，若存在此联系人则输出联系人的其他信息，若不存在则输出“该手机号不存在”。

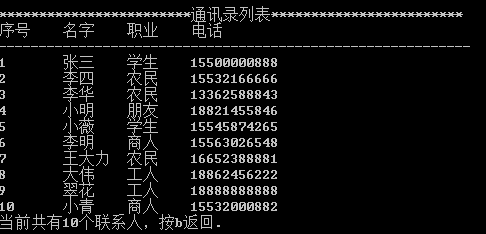
**验证与结论:**

**验证：**

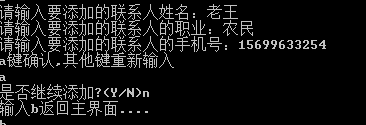
Input函数：

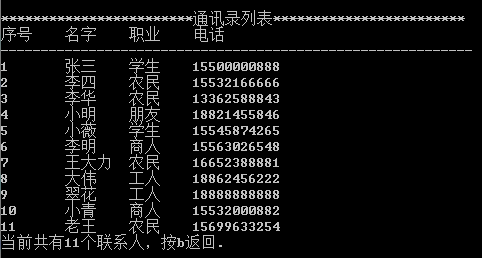
SumCount函数、Getchar函数嵌套在其他函数中，没有办法显示其函数的作用；

TiShi函数：

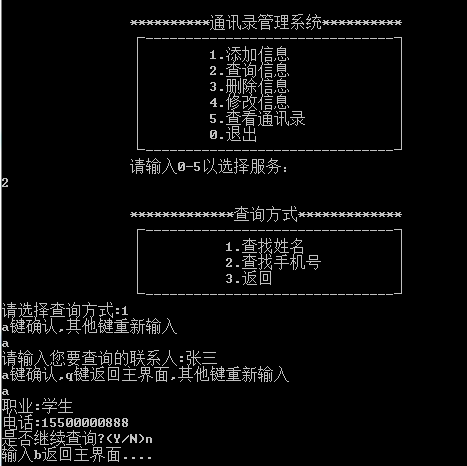
AddInfo函数：

上图为添加联系人之前





上图为添加联系人之后

FindInfo函数：

**结论：**通过调试程序，所有函数均可进行相应功能。

**成员2的工作：**完成删除联系人、修改联系人信息和查看通讯录的操作。

**姓名：**赵文强

**学号：**2019201130

**任务分工：**完成删除联系人、修改联系人信息和查看通讯 录的操作。

**设计思想：**在DeleteInfo函数中，完成对联系人的删除功能；在CreatStruct函数中，完成对通讯录的浏览功能；在ChangeInfo函数中，输出修改界面并完成对修改内容的选择；在ChangeName函数中，完成对联系人姓名修改的操作；在ChangeWork函数中，完成完成对联系人工作的修改；在ChangeTell函数中，完成对联系人电话号码的修改。

**代码行数：**309-614行，共300行（去除空行）

**实验代码及注释：**

void DeleteInfo()：此函数的作用是删除联系人的所有信息，首先用户根据系统提示输入联系人的姓名，然后系统判断该联系人是否存在，

若不存在，在用户可以重新输入或者退出并返回主界面；若存在则将联系人删除，并将“通讯录.txt”文件中的联系人信息删除。在最后还会提示用户是否继续删除。

void ChangeInfo()：此函数的作用是输出“修改信息”的主界面，并提示用户进行功能选择，然后系统根据输入的编号，进行修改操作。若输入‘1’，则运行ChangeName()函数，若输入‘2’，则运行ChangeWork()函数，若输入‘3’，则运行ChangeTell()函数，若输入‘4’，则返回主界面。

void ChangeName()：此函数的作用是修改联系人的姓名。用户根据系统提示信息输入原联系人姓名，若不存在，则输出“该联系人不存在”，若存在则输入更改之后的姓名，并进行修改，然后将修改之后的联系人信息写入文件。

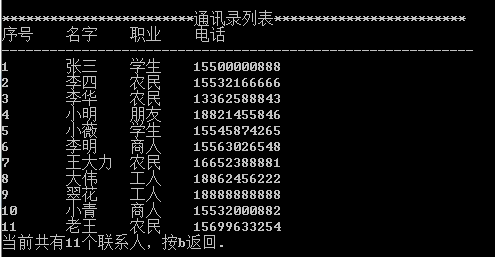
void ChangeWork()：此函数的作用是修改联系人的工作。用户根据系统提示信息输入联系人姓名，若不存在，则输出“该联系人不存在”，若存在则输入更改之后的工作，并进行修改，然后将修改之后的联系人信息写入文件。

void ChangeTell()：此函数的作用是修改联系人的电话。用户根据系统提示信息输入联系人姓名，若不存在，则输出“该联系人不存在”，若存在则输入更改之后的电话，并进行修改，然后将修改之后的联系人信息写入文件。

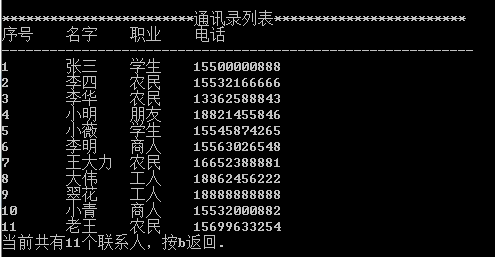
void CreatStruct()：此函数的作用是输出通讯录列表。该函数首先将“通讯录.txt”文件中的信息写入结构体数组，在通过结构体数组，输出联系人的信息。

**验证与结论：**

**验证：**

CreatStruct函数：

Delete函数：

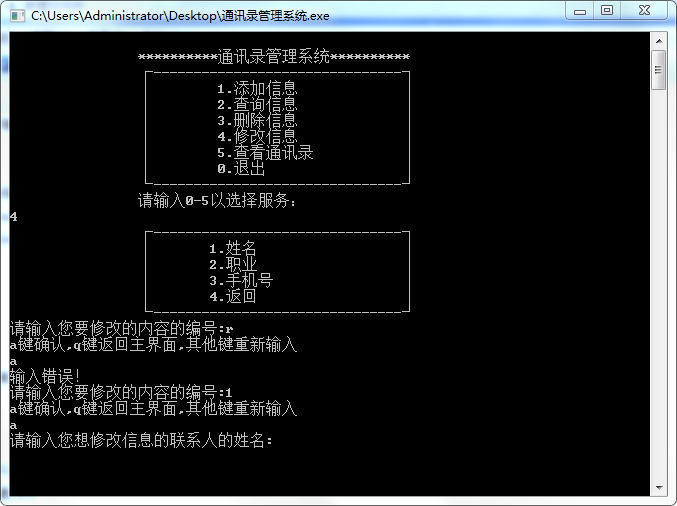


上图为删除之前的列表



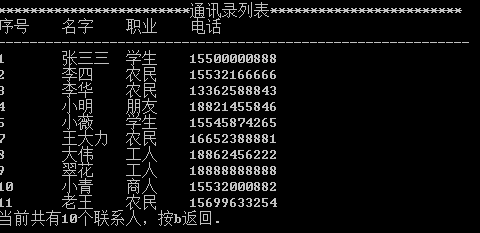
上图为执行删除联系人“李明”操作之后的列表

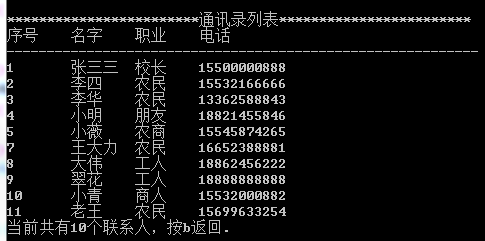
ChangeInfo函数：



ChangeName函数：

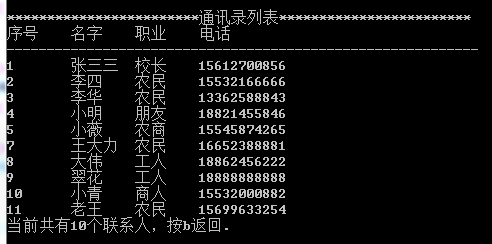
下图为将联系人“张三”改名为“张三三”的结果



ChangeWork函数：

以上是对 “张三三”的职业更改之后的结果

ChangeTell函数：



以上的是对“张三三”更改手机号之后的结果

**结论：**通过调试程序，所有函数均可进行相应功能。

**源码：**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#define N 100 //便于修改结构体的大小

FILE \*fp; //定义文件指针

int count=0; //定义一个count，count代表通讯录中联系人的数量

struct node //定义一个结构体

{

int num;

char name[20];

char work[20];

char tel[11];

};

typedef struct node list;

list person[N];

void CreatStruct();

char Input();

void TiShi();

void Tips(int n);

char Getchar();

void AddInfo();

void SumCount();

void ChangeInfo();

void ChangeName();

void ChangeWork();

void ChangeTell();

void FindInfo();

void DeleteInfo();

int main()

{

while(1)

{

SumCount();

char i;

i=Input();

switch(i)

{

case '0': //退出

return 0;

break;

case '1': //添加信息

AddInfo();

break;

case '2': //查询信息

FindInfo();

break;

case '3': //删除信息

DeleteInfo();

break;

case '4': //修改信息

ChangeInfo();

break;

case '5': //查看通讯录

CreatStruct();

break;

default:

printf("输入错误，请重新输入：\n");

break;

}

}

}

char Input()

{

char i;

printf("\n\t\t\*\*\*\*\*\*\*\*\*\*通讯录管理系统\*\*\*\*\*\*\*\*\*\*\n"); //输出界面，使使用人更加清楚、明白

printf("\t\t┌-------------------------------┐\n");

printf("\t\t│ 1.添加信息 │\n");

printf("\t\t│ 2.查询信息 │\n");

printf("\t\t│ 3.删除信息 │\n");

printf("\t\t│ 4.修改信息 │\n");

printf("\t\t│ 5.查看通讯录 │\n");

printf("\t\t│ 0.退出 │\n");

printf("\t\t└-------------------------------┘\n");

printf("\t\t请输入0-5以选择服务：\n");

while(1)

{

i=Getchar();

if(i=='0'||i=='1'||i=='2'||i=='3'||i=='4'||i=='5') //判断输入是否符合规范

return i;

else

printf("输入错误！\n");

}

}

void SumCount() //此函数的作用是计算出count的值 ，count代表的是当前通讯录中的联系人的数量

{

int i,res=0;

if((fp=fopen("通讯录.txt","r"))==NULL) //判断文件是否能打开

{

printf("cannot open file!\n");

exit(0);

}

for(i=0;i<N&&!feof(fp);i++) //当读到文件末尾时，停止读取

{

res=fscanf(fp,"%d\t%s\t%s\t%s",&person[i].num,person[i].name,person[i].work,person[i].tel); //将联系人信息写入结构体中

if(res==EOF)

break;

}

count=i;

fclose(fp); //关闭文件

}

char Getchar() //当输入某个字符时，系统有时会将回车键误认为输入，所以此函数的作用是消除回车键对输入字符的影响

{

char c,z;

int flag=1;

while((c=getchar())!='\n'||flag==1)

{

z=c;

flag=0;

}

return z;

}

void TiShi() //此函数的作用是提示使用人返回主界面

{

printf("输入b返回主界面....\n");

char tip;

while(1)

{

tip=Getchar();

if(tip=='b') //判断输入的字符是否符合要求

return;

else

printf("输入错误！\n");

}

}

void Tips(int n) //此函数的作用是在“查看通讯录 ”时，提示使用人有多少个联系人

{

printf("当前共有%d个联系人，按b返回.\n",n);

char tip;

while(1)

{

tip=Getchar();

if(tip=='b') //判断输入是否符合要求

return;

else

printf("输入错误！\n");

}

}

void AddInfo() //此函数的作用是添加联系人

{

char key='y';

int k,t;

while(key!='n'&&key!='N') //用key判断是否继续添加联系人

{

person[count].num=person[count-1].num+1; //使新添联系人的序号比前一位大1

list \*p; //运用指针，使程序更高效

p=person+count;

while(1) // 输入新建联系人的信息

{

printf("请输入要添加的联系人姓名：");

scanf("%s",p->name);

printf("请输入要添加的联系人的职业：");

scanf("%s",p->work);

printf("请输入要添加的联系人的手机号：");

scanf("%s",p->tel);

printf("a键确认,其他键重新输入\n"); //此条语句的作用是确认一下信息是否输入正确，否则重新输入

if(Getchar() == 'a')

break;

}

do

{

k=0;

if(strlen(p->tel)!=11) //判断手机号码是否符合格式，否则重新输入

{

k=1;

printf("您输入的手机号码格式不存在,请重新输入:");

scanf("%s",p->tel);

}

else

{

for(t=0;t<11;t++)

{

if(person[count].tel[t]<'0'||person[count].tel[t]>'9') //判断输入的手机号码中是否有不合规范的字符

{

k=1;

printf("您输入的手机号码格式不合理,请重新输入:");

scanf("%s",p->tel);

break;

}

}

}

}while(k);

if((fp=fopen("通讯录.txt","a"))==NULL) //判断文件是否能够打开

{

printf("cannot open file!\n");

exit(0);

}

fseek(fp,count\*sizeof(list),0); //将文件的指针指向文件末尾，从而将新建联系人写入文件

fprintf(fp,"%d\t%s\t%s\t%s\n",person[count].num,person[count].name,person[count].work,person[count].tel);

rewind(fp); //将文件指针重新返回文件的开头

fclose(fp); //关闭文件

count+=1; //每添加一次 联系人的数量加一

printf("是否继续添加?(Y/N)"); //判断是否继续新建联系人

key=Getchar();

}

TiShi();

}

void FindInfo() //此函数的作用是查询联系人信息

{

char key='y';

int m,k,i;

while(key!='n'&&key!='N') //用key判断是否继续运行

{

printf("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*查询方式\*\*\*\*\*\*\*\*\*\*\*\*\*\n"); //输出界面，让使用人更清楚、明白接下来需要干什么

printf("\t\t┌-------------------------------┐\n");

printf("\t\t│ 1.查找姓名 │\n");

printf("\t\t│ 2.查找手机号 │\n");

printf("\t\t│ 3.返回 │\n");

printf("\t\t└-------------------------------┘\n");

char i;

while(1)

{

while(1) //输入查询方式，并检查是否符合规范

{

printf("请选择查询方式:");

i=Getchar();

printf("a键确认,其他键重新输入\n");

if(Getchar() == 'a')

break;

}

if(i=='1'||i=='2'||i=='3')

{

break;

}

else

printf("输入错误！\n");

}

if(i=='1') //用 姓名 查找联系人信息

{

char nm[20];

do

{

k=1;

while(1)

{

printf("请输入您要查询的联系人:");

scanf("%s",nm);

printf("a键确认,q键返回主界面,其他键重新输入\n"); //使用人若输入错误，还有更改的机会

char w=Getchar();

if(w=='q') //如果使用人不想继续查询，或者没有要查询的联系人，输入q返回主界面

return;

if(w == 'a')

break;

}

for(i=0;i<count;i++) //判断所查联系人是否存在

{

if(strcmp(nm,person[i].name)==0)

{

k=0;

break;

}

}

if(k==1)

printf("该联系人不存在，");

}while(k);

printf("职业:%s\n",person[i].work); //若存在输出其信息

printf("电话:%s\n",person[i].tel);

}

if(i=='2') //用 手机号 查找联系人

{

char te[20];

do

{

k=1;

while(1)

{

printf("请输入您要查询的手机号码:");

scanf("%s",te);

printf("a键确认,q键返回主界面,其他键重新输入\n"); //使用人若输入错误，还有更改的机会

char w=Getchar();

if(w=='q') //如果使用人不想继续查询，或者没有要查询的联系人，输入q返回主界面

return;

if(w == 'a')

break;

}

for(i=0;i<count;i++) //判断所查手机号是否存在

{

if(strcmp(te,person[i].tel)==0)

{

k=0;

break;

}

}

if(k==1)

printf("该手机号不存在，");

}while(k);

printf("姓名:%s\n",person[i].name); //若存在输出其信息

printf("职业:%s\n",person[i].work);

}

if(i=='3') //返回主界面

return;

printf("是否继续查询?(Y/N)");

key=Getchar();

}

TiShi(); //输出提示信息

}

void DeleteInfo() //此函数的作用是删除联系人信息

{

char nm[20];

char key='y';

while(key!='n'&&key!='N') //用key判断是否继续运行

{

int i,j,k;

do

{

k=1;

while(1)

{

printf("请输入您要删除的联系人姓名:");

scanf("%s",nm);

printf("a键确认,q键返回主界面,其他键重新输入\n"); //若使用人回心转意，可以输入q返回主界面

char w=Getchar();

if(w=='q')

return;

if(w=='a')

break;

}

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

{

if(strcmp(nm,person[i].name)==0) //判断联系人是否存在

{

k=0;

break;

}

}

if(k==1)

printf("该联系人不存在,");

}while(k);

if((fp=fopen("通讯录.txt","w"))==NULL) //判断文件是否能打开

{

printf("cannot open file!\n");

exit(0);

}

for(j=0;j<count;j++)

{

if(j!=i)

fprintf(fp,"%d\t%s\t%s\t%s\n",person[j].num,person[j].name,person[j].work,person[j].tel); //将指定联系人删除后的信息写入文件

}

rewind(fp); //将文件指针移到文件开头

fclose(fp); //关闭文件

printf("该联系人已删除!\n");

printf("是否继续删除?(Y/N)"); //是否继续

key=Getchar();

}

TiShi();

}

void ChangeInfo() //此函数的作用是修改联系人信息

{

char key='y';

while(key!='n'&&key!='N') //用key判断是否继续

{

printf("\t\t┌-------------------------------┐\n"); //使界面更加友好

printf("\t\t│ 1.姓名 │\n");

printf("\t\t│ 2.职业 │\n");

printf("\t\t│ 3.手机号 │\n");

printf("\t\t│ 4.返回 │\n");

printf("\t\t└-------------------------------┘\n");

int k;

char i;

while(1)

{

while(1)

{

printf("请输入您要修改的内容的编号:");

i=Getchar();

printf("a键确认,q键返回主界面,其他键重新输入\n"); //若使用人回心转意，可以输入q返回主界面

char w=Getchar();

if(w=='q')

return;

if(w=='a')

break;

}

if(i=='1'||i=='2'||i=='3'||i=='4') //判断输入字符是否符合规范

break;

else

printf("输入错误！\n");

}

if(i=='1')

ChangeName(); //修改名字

if(i=='2')

ChangeWork(); //修改工作

if(i=='3')

ChangeTell(); //修改电话

if(i=='4') //返回主界面

return;

printf("是否继续修改?(Y/N)");

key=Getchar();

}

}

void ChangeName() //此函数的作用是修改联系人的名字

{

char nm[20],nam[20];

int i,j,k;

do

{

k=1;

while(1)

{

printf("请输入您想修改信息的联系人的姓名:");

scanf("%s",nm);

printf("a键确认,q键返回主界面,其他键重新输入\n"); //再次确认是否继续修改

char w=Getchar();

if(w=='q')

return;

if(w=='a')

break;

}

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

{

if(strcmp(nm,person[i].name)==0) //判断输入的联系人是否存在

{

k=0;

break;

}

}

if(k==1)

printf("该联系人不存在，");

}while(k);

while(1)

{

printf("请输入更改之后的联系人的姓名:");

scanf("%s",nam);

printf("a键确认,q键返回主界面,其他键重新输入\n"); //再次确认信息

char w=Getchar();

if(w=='q')

return;

if(w=='a')

break;

}

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

{

if(strcmp(nm,person[i].name)==0)

{

for(j=0;j<strlen(nam);j++)

{

person[i].name[j]=nam[j]; //找到所要修改的联系人，并修改信息

}

}

}

if((fp=fopen("通讯录.txt","w"))==NULL) //判断文件是否能打开

{

printf("cannot open file!\n");

exit(0);

}

for(j=0;j<count;j++)

{

fprintf(fp,"%d\t%s\t%s\t%s\n",person[j].num,person[j].name,person[j].work,person[j].tel); //将修改之后的联系人信息写入文件

}

fclose(fp); //关闭文件

}

void ChangeWork() //此函数的作用是修改联系人的工作

{

char nm[20];

char nam[20];

int i,j,k;

do

{

k=1;

while(1)

{

printf("请输入您想修改信息的联系人的姓名:");

scanf("%s",nm);

printf("a键确认,q键返回主界面,其他键重新输入\n"); //再次确认信息

char w=Getchar();

if(w=='q')

return;

if(w=='a')

break;

}

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

{

if(strcmp(nm,person[i].name)==0) //判断所要修改的联系人是否存在

{

k=0;

break;

}

}

if(k==1)

printf("该联系人不存在,");

}while(k);

while(1)

{

printf("请输入更改之后的联系人职业：");

scanf("%s",nam);

printf("a键确认,q键返回主界面,其他键重新输入\n"); //再次确认

char w=Getchar();

if(w=='q')

return;

if(w== 'a')

break;

}

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

{

if(strcmp(nm,person[i].name)==0)

{

for(j=0;j<strlen(nam);j++)

{

person[i].work[j]=nam[j]; //找到所要修改的联系人，并修改

}

}

}

if((fp=fopen("通讯录.txt","w"))==NULL) //判断文件是否能够打开

{

printf("cannot open file!\n");

exit(0);

}

for(j=0;j<count;j++)

{

fprintf(fp,"%d\t%s\t%s\t%s\n",person[j].num,person[j].name,person[j].work,person[j].tel); //将修改之后的联系人信息重新写入文件

}

fclose(fp); //关闭文件

}

void ChangeTell() //此函数的作用是更改联系人的电话号码

{

char nm[20],nam[20];

int i,j,k;

do

{

while(1)

{

printf("请输入您想修改信息的联系人的姓名:");

scanf("%s",nm);

printf("a键确认,q键返回主界面,其他键重新输入\n"); //再次确认是否继续

char w=Getchar();

if(w=='q')

return;

if(w == 'a')

break;

}

k=1;

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

{

if(strcmp(nm,person[i].name)==0) //判断所要修改的联系人是否存在

{

k=0;

break;

}

}

if(k==1)

printf("该联系人不存在,");

}while(k);

while(1)

{

printf("请输入更改之后的联系人手机号：");

scanf("%s",nam);

printf("a键确认,q键返回主界面,其他键重新输入\n"); //再次确认

char w=Getchar();

if(w=='q')

return;

if(w == 'a')

break;

}

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

{

if(strcmp(nm,person[i].name)==0)

{

for(j=0;j<strlen(nam);j++)

{

person[i].tel[j]=nam[j]; //找到所要修改的联系人，并修改信息

}

}

}

if((fp=fopen("通讯录.txt","w"))==NULL) //判断文件是否能够打开

{

printf("cannot open file!\n");

exit(0);

}

for(j=0;j<count;j++)

{

fprintf(fp,"%d\t%s\t%s\t%s\n",person[j].num,person[j].name,person[j].work,person[j].tel); //将修改之后的信息重新写入文件

}

fclose(fp); //关闭文件

}

void CreatStruct() //此函数的作用是输出通讯录列表

{

int i,res=0;

if((fp=fopen("通讯录.txt","r"))==NULL) //判断文件是否能够打开

{

printf("cannot open file!\n");

exit(0);

}

for(i=0;i<N&&!feof(fp);i++) //判断是否到了文件的末尾

{

res=fscanf(fp,"%d\t%s\t%s\t%s",&person[i].num,person[i].name,person[i].work,person[i].tel); //将文件内的数据赋给结构体

if(res==EOF)

break;

}

count=i;

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*通讯录列表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"); //使界面更友好

printf("序号\t名字\t职业\t电话\n");

printf("-----------------------------------------------------------\n");

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

printf("%d\t%s\t%s\t%s\n",person[i].num,person[i].name,person[i].work,person[i].tel); //将文件内的信息输出到屏幕上

fclose(fp); //关闭文件

Tips(count); //输出提示信息

}