

二、C++学习笔记——通讯录管理系统

2.1、系统需求

通讯录是一个可以记录亲人、好友信息的工具。

本教程主要利用C++来实现一个通讯录管理系统

系统中需要实现的功能如下：

- 添加联系人：向通讯录中添加新人，信息包括（姓名、性别、年龄、联系电话、家庭住址）最多记录1000人
- 显示联系人：显示通讯录中所有联系人信息
- 删除联系人：按照姓名进行删除指定联系人
- 查找联系人：按照姓名查看指定联系人信息
- 修改联系人：按照姓名重新修改指定联系人
- 清空联系人：清空通讯录中所有信息
- 退出通讯录：退出当前使用的通讯录

2.2、创建项目

创建项目步骤如下：

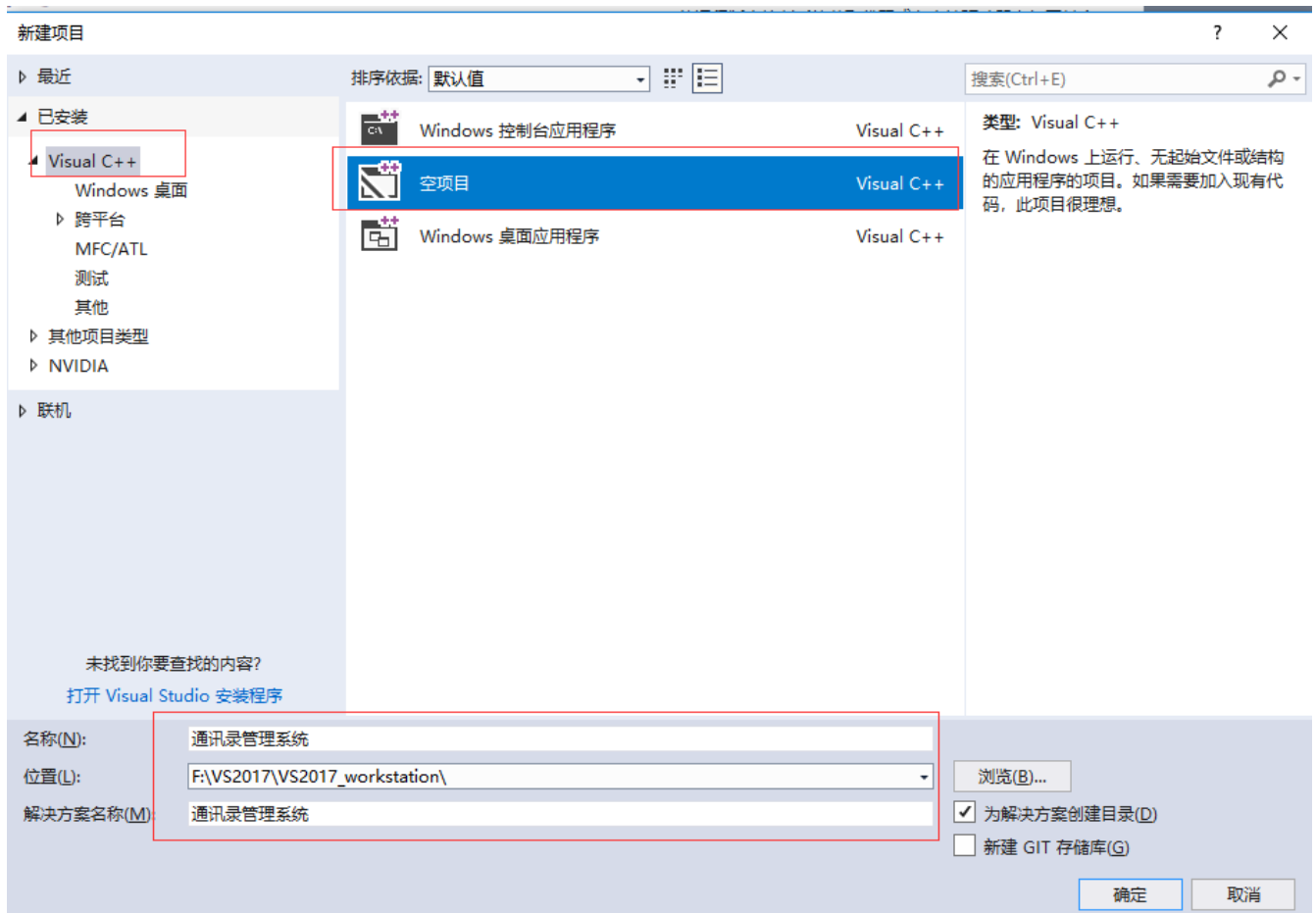
- 创建新项目
- 添加文件

2.2.1 创建项目

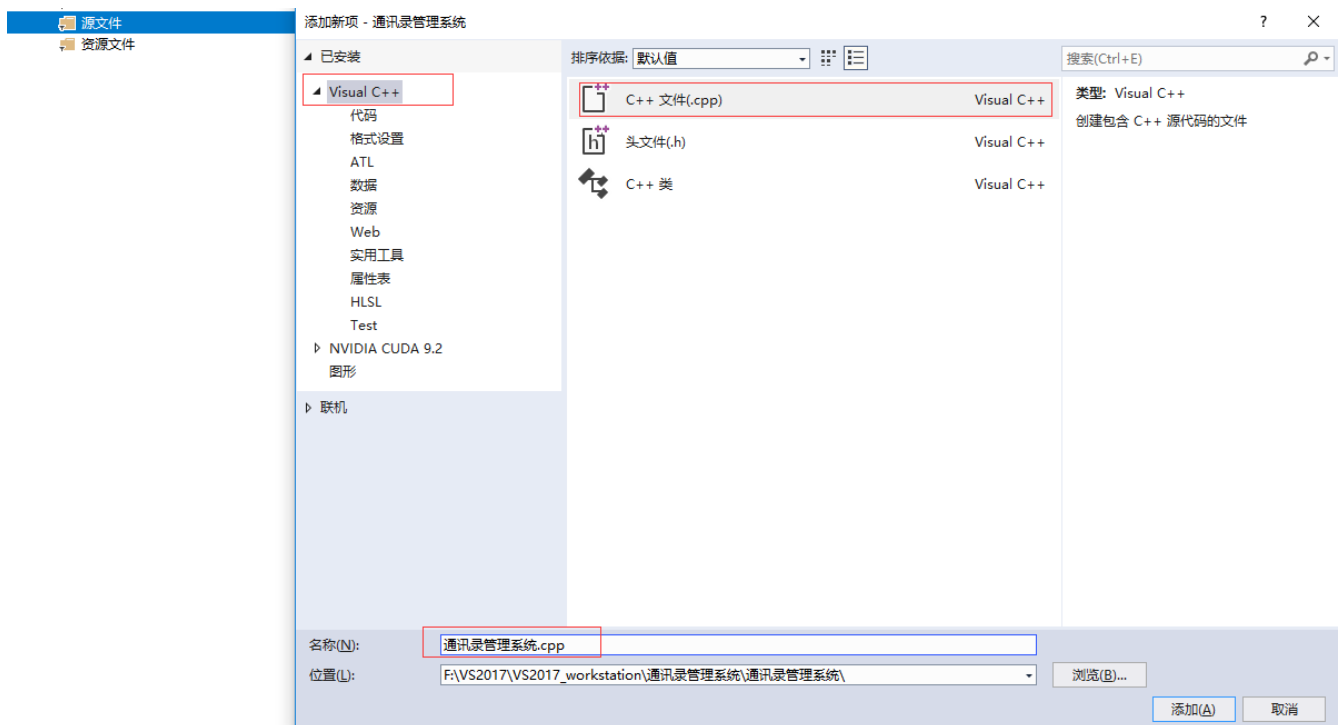
打开vs2017后，点击创建新项目，创建新的C++项目



填写项目名称，选择项目路径



2.2.2 添加文件



至此，项目已创建完毕

2.3、菜单功能

功能描述： 用户选择功能的界面

菜单界面效果如下图：

```
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
```

步骤：

- 封装函数显示该界面 如 `void showMenu()`
- 在main函数中调用封装好的函数

代码：

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4
5  //菜单界面
6  void showMenu()
7  {
8      cout << "*****" << endl;
9      cout << "***** 1、添加联系人 *****" << endl;
10     cout << "***** 2、显示联系人 *****" << endl;
11     cout << "***** 3、删除联系人 *****" << endl;
12     cout << "***** 4、查找联系人 *****" << endl;
13     cout << "***** 5、修改联系人 *****" << endl;
14     cout << "***** 6、清空联系人 *****" << endl;
15     cout << "***** 0、退出通讯录 *****" << endl;
16     cout << "*****" << endl;
17 }
18
19 int main()
20 {
21     showMenu(); // 菜单调用
22
23     system("pause");
24     return 0;
25 }
```

2.4、退出功能

功能描述：退出通讯录系统

思路：根据用户不同的选择，进入不同的功能，可以选择switch分支结构，将整个架构进行搭建

当用户选择0时候，执行退出，选择其他先不做操作，也不会退出程序

代码：

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4
5  //菜单界面
6  void showMenu()
7  {
8      cout << "*****" << endl;
9      cout << "*****  1、添加联系人  *****" << endl;
10     cout << "*****  2、显示联系人  *****" << endl;
11     cout << "*****  3、删除联系人  *****" << endl;
12     cout << "*****  4、查找联系人  *****" << endl;
13     cout << "*****  5、修改联系人  *****" << endl;
14     cout << "*****  6、清空联系人  *****" << endl;
15     cout << "*****  0、退出通讯录  *****" << endl;
16     cout << "*****" << endl;
17 }
18
19
20 int main()
21 {
22     int select = 0; //创建一个用户选择输入的变量
23     while (true) // 并不是真正意义上的死循环
24     {
25         showMenu(); // 菜单调用
26         cin >> select;
27         switch (select)
28         {
29             case 1: //1、添加联系人
30                 break;
31             case 2: //2、显示联系人
32                 break;
33             case 3: //3、删除联系人
34                 break;
35             case 4: //4、查找联系人
36                 break;
37             case 5: //5、修改联系人
38                 break;
39             case 6: //6、清空联系人
```

```

40         break;
41     case 0: //0、退出通讯录
42         cout << "欢迎下次使用" << endl;
43         system("pause");
44         return 0;
45         break;
46     default:
47         break;
48     }
49
50 }
51 system("pause");
52 return 0;
53 }

```

2.5、添加联系人

功能描述：

实现添加联系人功能，联系人上限为1000人，联系人信息包括（姓名、性别、年龄、联系电话、家庭住址）

添加联系人实现步骤：

- 设计联系人结构体
- 设计通讯录结构体
- main函数中创建通讯录
- 封装添加联系人函数
- 测试添加联系人功能

2.5.1 设计联系人结构体

联系人信息包括：姓名、性别、年龄、联系电话、家庭住址

设计如下：

```

1  #include<string>
2  //设计联系人结构体
3  struct Person
4  {
5      // 姓名、性别、年龄、电话、住址
6      string m_Name;
7      int m_Sex;
8      int m_Age;
9      string m_Phone;
10     string m_Addr;
11 };

```

2.5.2 设计通讯录结构体

设计时候可以在通讯录结构体中，维护一个容量为1000的存放联系人的数组，并记录当前通讯录中联系人数量
设计如下

```
1 #define MAX 1000
2 //设计通讯录结构体
3 struct Addressbooks
4 {
5     //通讯录中保存的联系人数组
6     struct Person personArray[MAX];
7     //通讯录中当前记录联系人个数
8     int m_Size;
9 };
```

2.5.3 main函数中创建通讯录

添加联系人函数封装好后，在main函数中创建一个通讯录变量，这个就是我们需要一直维护的通讯录

```
1 main函数起始位置添加：
2     //创建通讯录
3     struct Addressbooks abs;
4     //初始化通讯录中人数
5     abs.m_Size = 0;
```

2.5.4 封装添加联系人函数

思路：添加联系人前先判断通讯录是否已满，如果满了就不再添加，未满情况将新联系人信息逐个加入到通讯录

添加联系人代码：

```
1 //1、添加联系人
2 void addPerson(struct Addressbooks * abs)
3 {
4     //判断通讯录是否已满，如果满了就不在添加
5     if (abs->m_Size==MAX)
6     {
7         cout << "通讯录已满，无法添加!" << endl;
8         return;
9     }
10    else
11    {
12        //添加具体联系人
13        //姓名
14        string name;
15        cout << "请输入姓名：" << endl;
16        cin >> name;
17        abs->personArray[abs->m_Size].m_Name = name;
18    }
```

```

19 //性别
20 cout << "请输入性别: " << endl;
21 cout << "1---男" << endl;
22 cout << "2---女" << endl;
23 int sex = 0;
24 while (true)
25 {
26     //如果输入的是 1 或者 2 可以退出循环
27     //如果输入的有误, 重新输入
28     cin >> sex; //输入性别
29     if (sex == 1 || sex == 2)
30     {
31         abs->personArray[abs->m_Size].m_Sex = sex;
32         break;
33     }
34     cout << "输入有误, 请重新输入" << endl;
35 }
36
37
38 //年龄
39 cout << "请输入年龄: " << endl;
40 int age = 0;
41 cin >> age;
42 abs->personArray[abs->m_Size].m_Age = age;
43
44 //电话
45 cout << "请输入联系电话: " << endl;
46 string phone = "";
47 cin >> phone;
48 abs->personArray[abs->m_Size].m_Phone = phone;
49
50 //住址
51 cout << "请输入家庭住址: " << endl;
52 string address;
53 cin >> address;
54 abs->personArray[abs->m_Size].m_Addr = address;
55
56 abs->m_Size++;
57 cout << "添加成功" << endl;
58 system("pause");
59 system("cls"); //清屏操作
60 }
61 }

```

2.5.5 测试添加联系人功能

选择界面中, 如果玩家选择了1, 代表添加联系人, 我们可以测试下该功能

在switch case 语句中, case1里添加:

```

1 case 1: //添加联系人
2     addPerson(&abs);
3     break;

```

测试效果图：

```

C:\WINDOWS\system32\cmd.exe - a.exe
F:\VS2017\VS2017_workstation\通讯录管理系统\通讯录管理系统>g++ 通讯录管理系统.cpp
F:\VS2017\VS2017_workstation\通讯录管理系统\通讯录管理系统>a.exe
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
1
请输入姓名：
贾继康
请输入性别：
1---男
2---女
1
请输入年龄：
100
请输入联系电话：
11111111
请输入家庭住址：
甘肃省庆阳市
添加成功
请按任意键继续. . .

```

2.6、显示联系人

功能描述：显示通讯录中已有的联系人信息

显示联系人实现步骤：

- 封装显示联系人函数
- 测试显示联系人功能

2.6.1 封装显示联系人函数

思路：判断如果当前通讯录中没有人员，就提示记录为空，人数大于0，显示通讯录中信息

显示联系人代码：

```

1 //2、显示所有联系人
2 void showPerson(struct Addressbooks * abs)
3 {
4     //判断通讯录人数是否为0，如果为0，提示记录为空；如果不为0，显示记录的联系人信息
5     if (abs->m_Size==0)
6     {
7         cout << "当前的记录为空" << endl;
8     }
9     else
10    {

```



```

11     for (int i = 0; i < abs->m_Size; i++)
12     {
13         cout << "姓名: " << abs->personArray[i].m_Name << "\t";
14         cout << "性别: " << (abs->personArray[i].m_Sex == 1 ? "男" : "女") << "\t";
15         cout << "年龄: " << abs->personArray[i].m_Age << "\t";
16         cout << "电话: " << abs->personArray[i].m_Phone << "\t";
17         cout << "住址: " << abs->personArray[i].m_Addr << endl;
18     }
19 }
20 }
21 system("pause");
22 system("cls"); //清屏操作
23 }
24

```

2.6.2 测试显示联系人功能

在switch case语句中，case 2 里添加

```

1 case 2: //显示联系人
2     showPerson(&abs);
3     break;

```

测试效果如图：

```

*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
2
姓名: 贾继康    性别: 女        年龄: 2  电话: 2222    住址: 23434
请按任意键继续. . .

```

2.7、删除联系人

功能描述：按照姓名进行删除指定联系人

删除联系人实现步骤：

- 封装检测联系人是否存在
- 封装删除联系人函数
- 测试删除联系人功能

2.7.1 封装检测联系人是否存在

设计思路：

删除联系人前，我们需要先判断用户输入的联系人是否存在，如果存在删除，不存在提示用户没有要删除的联系人

因此我们可以把检测联系人是否存在封装成一个函数中，如果存在，返回联系人在通讯录中的位置，不存在返回-1

检测联系人是否存在代码：

```
1 //参数1: 通讯录 参数2: 对比的姓名
2 int isExist(struct Addressbooks * abs, string name)
3 {
4     for (int i = 0; i < abs->m_Size; i++)
5     {
6         //找到用户输入的姓名
7         if (abs->personArray[i].m_Name == name)
8         {
9             return i; // 找到了，返回这个人在数组中下标
10        }
11    }
12    return -1; //如果遍历结束都没有找到，返回-1
13 }
14
15 }
```

2.7.2 封装删除联系人函数

根据用户输入的联系人判断该通讯录中是否有此人

查找到进行删除，并提示删除成功

查不到提示查无此人。

```
1 //3、删除指定联系人
2 void deletePerson(struct Addressbooks * abs)
3 {
4     cout << "请输入您要删除的联系人" << endl;
5     string name;
6     cin >> name;
7
8     //ret == -1 表示无人，
9     //ret != -1 查到了
10    int ret = isExist(abs,name);
11    if (ret != -1)
12    {
13        //查找到人，要进行删除操作
14        for (int i = ret; i < abs->m_Size; i++)
15        {
16            //数据迁移
17            abs->personArray[i] = abs->personArray[i + 1];
18        }
19        abs->m_Size--; //更新一下通讯录中的人员数
20    }
```

```

20         cout << "删除成功" << endl;
21
22     }
23     else
24     {
25         cout << "查无此人" << endl;
26
27     }
28
29     system("pause");
30     system("cls");
31
32 }

```

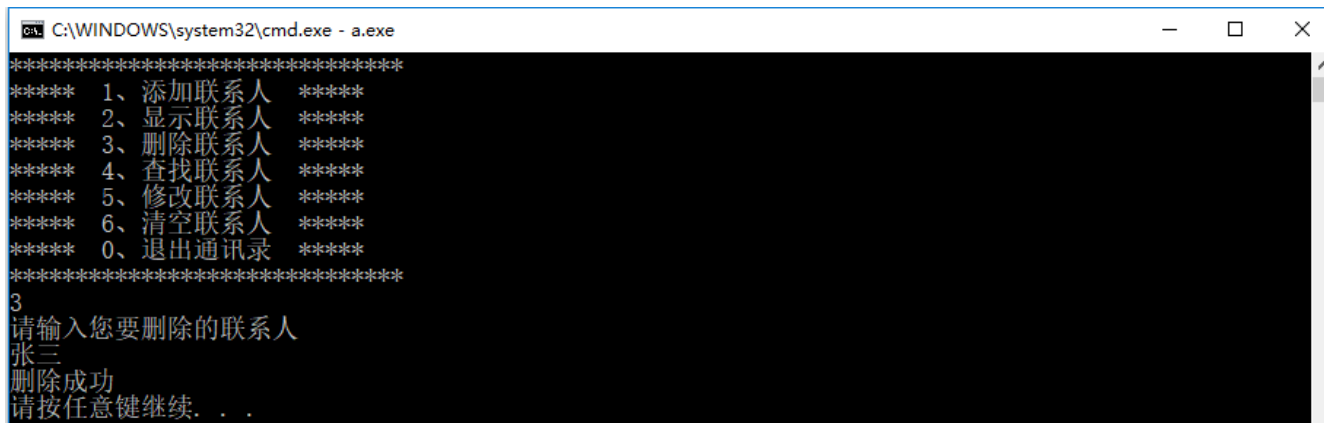
2.7.3 测试删除联系人功能

在switch case 语句中，case3里添加：

```

1 case 3: //删除联系人
2     deletePerson(&abs);
3     break;

```

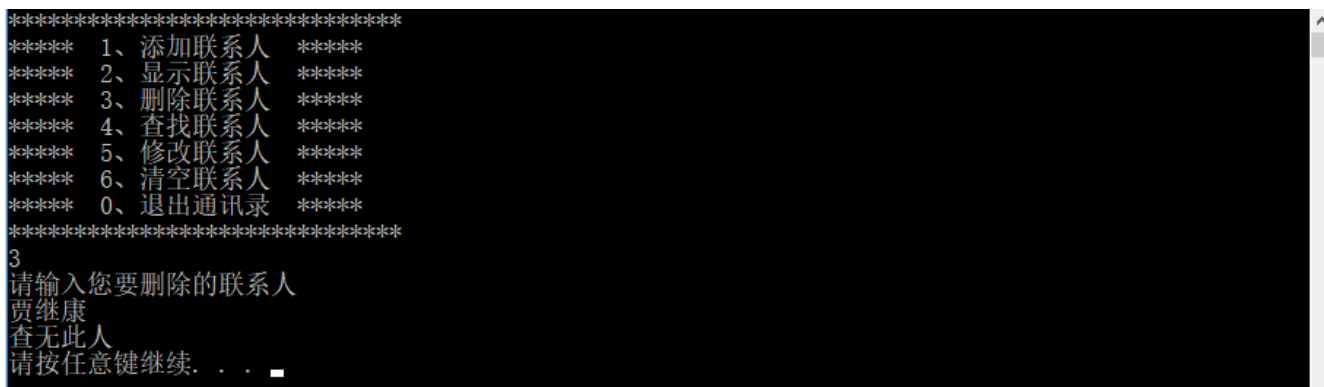


```

C:\WINDOWS\system32\cmd.exe - a.exe
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
3
请输入您要删除的联系人
张三
删除成功
请按任意键继续. . .

```

不存在的情况：



```

*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
3
请输入您要删除的联系人
贾继康
查无此人
请按任意键继续. . .

```

2.8、查找联系人

功能描述：按照姓名查看指定联系人信息

查找联系人实现步骤

- 封装查找联系人函数
- 测试查找指定联系人

2.8.1 封装查找联系人函数

实现思路：判断用户指定的联系人是否存在，如果存在显示信息，不存在则提示查无此人。

查找联系人代码：

```
1 //4、查找指定联系人信息
2 void findPerson(struct Addressbooks * abs)
3 {
4     cout << "请输入您要查找的联系人：" << endl;
5     string name;
6     cin >> name;
7
8     //判断指定的联系人是否存在通讯录中
9     int ret = isExist(abs, name);
10    if (ret != -1) //找到联系人
11    {
12        cout << "姓名：" << abs->personArray[ret].m_Name << "\t";
13        cout << "性别：" << abs->personArray[ret].m_Sex << "\t";
14        cout << "年龄：" << abs->personArray[ret].m_Age << "\t";
15        cout << "电话：" << abs->personArray[ret].m_Phone << "\t";
16        cout << "家庭住址：" << abs->personArray[ret].m_Addr << endl;
17    }
18    else
19    {
20        cout << "查无此人！" << endl;
21    }
22    system("pause");
23    system("cls");
24 }
```

2.8.2 测试查找指定联系人

在switch case 语句中，case4里添加：

```
1 case 4: //4、查找联系人
2         findPerson(&abs);
3         break;
```

测试效果如图

比存在情况

```
C:\WINDOWS\system32\cmd.exe - a.exe
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
4
请输入您要查找的联系人：
小三
查无此人！
请按任意键继续. . .
```

存在情况：

```
C:\WINDOWS\system32\cmd.exe - a.exe
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
4
请输入您要查找的联系人：
贾继康
姓名：贾继康 性别：1 年龄：11 电话：11111 家庭住址：甘肃省
请按任意键继续. . .
```

2.9、修改联系人

功能描述：按照姓名重新修改指定联系人

修改联系人实现步骤

- 封装修改联系人函数
- 测试修改联系人功能

2.9.1 封装修改联系人函数

实现思路：查找用户输入的联系人，如果查找成功进行修改操作，查找失败提示查无此人

修改联系人代码：

```
1 //5、修改指定人信息
2 void moidfyPerson(struct Addressbooks * abs)
3 {
4     cout << "请输入您要修改的联系人" << endl;
5     string name;
6     cin >> name;
7
8     int ret = isExist(abs, name);
9     if (ret != -1) // 找到联系人
10    {
11        //姓名
```

```

12     string name;
13     cout << "请输入姓名: " << endl;
14     cin >> name;
15     abs->personArray[ret].m_Name = name;
16
17     //性别
18     cout << "请输入性别: " << endl;
19     cout << "1--男" << endl;
20     cout << "2--女" << endl;
21     int sex = 0;
22     while (true)
23     {
24         cin >> sex;
25         if (sex == 1 || sex == 2)
26         {
27             //输入正确, 退出循环输入
28             abs->personArray[ret].m_Sex = sex;
29             break;
30         }
31         cout << "输入错误, 请重新输入" << endl;
32     }
33
34     //年龄
35     cout << "请输入年龄: " << endl;
36     int age = 0;
37     cin >> age;
38     abs->personArray[ret].m_Age = age;
39
40     //电话
41     cout << "请输入联系电话: " << endl;
42     string phone;
43     cin >> phone;
44     abs->personArray[ret].m_Phone = phone;
45
46     //住址
47     cout << "请输入家庭住址: " << endl;
48     string address;
49     cin >> address;
50     abs->personArray[ret].m_Addr = address;
51     cout << "修改成功" << endl;
52 }
53 else
54 {
55     cout << "查无此人" << endl;
56 }
57
58 //按任意键后清屏
59 system("pause");
60 system("cls");
61 }

```

2.9.2 测试修改联系人功能

在switch case 语句中，case 5里添加：

```
1 case 5: //修改联系人
2     modifyPerson(&abs);
3     break;
```

测试效果如图：

查不到指定联系人情况：

```
F:\VS2017\VS2017_workstation\通讯录管理系统\通讯录管理系统>a.exe
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
4
请输入您要查找的联系人：
小三
查无此人！
请按任意键继续. . .
```

查找到联系人，并修改成功：

```
C:\WINDOWS\system32\cmd.exe - a.exe
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
5
请输入您要修改的联系人
张三
请输入姓名：
贾继康
请输入性别：
1--男
2--女
1
请输入年龄：
12
请输入联系电话：
122
请输入家庭住址：
甘肃省
修改成功
请按任意键继续. . .
```

再次查看通讯录，确认修改完毕

```
C:\WINDOWS\system32\cmd.exe - a.exe
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
2
姓名: 贾继康    性别: 男        年龄: 12        电话: 122        住址: 甘肃省
请按任意键继续. . .
```

2.10、清空联系人

功能描述：清空通讯录中所有信息

清空联系人实现步骤

- 封装清空联系人函数
- 测试清空联系人

2.10.1 封装清空联系人函数

实现思路：将通讯录所有联系人信息清除掉，只要将通讯录记录的联系人数量置为0，做逻辑清空即可。

清空联系人代码：

```
1 //6、清空所有联系人
2 void cleanPerson(struct Addressbooks * abs)
3 {
4     cout << "请确认是否要清空所有联系人,1: 确认清空; 2: 放弃清空" << endl;
5     int ret = 0;
6     while (true)
7     {
8         cout << "请输入是否清空按钮: ";
9         cin >> ret;
10
11         if (ret == 1)
12         {
13             abs->m_Size = 0; //将当前记录联系人数量置为0, 做逻辑清空操作
14             cout << "通讯录已经清空" << endl;
15             break;
16         }
17         else if (ret == 2)
18         {
19             cout << "放弃清空完成" << endl;
20             break;
21         }
22         else
23         {
24             cout << "输入有误, 请重新输入" << endl;
```



```

25     }
26 }
27     system("pause");
28     system("cls");
29 }

```

2.10.2 测试清空联系人

在switch case 语句中，case 6 里添加：

```

1  case 6: //6、清空联系人
2      cleanPerson(&abs);
3      break;

```

测试效果如图：

清空通讯录：其实一开始还需要判断一下，是否有人，要是没有就没有清空的必要

F:\VS2017\VS2017_workstation\通讯录管理系统\通讯录管理系统>a.exe

```

*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****

```

```

6
请确认是否要清空所有联系人, 1: 确认清空; 2: 放弃清空
请输入是否清空按钮: 1
通讯录已经清空
请按任意键继续. . .

```

再次查看信息，显示记录为空

```

C:\WINDOWS\system32\cmd.exe - a.exe
*****
***** 1、添加联系人 *****
***** 2、显示联系人 *****
***** 3、删除联系人 *****
***** 4、查找联系人 *****
***** 5、修改联系人 *****
***** 6、清空联系人 *****
***** 0、退出通讯录 *****
*****
2
当前的记录为空
请按任意键继续. . .

```

至此，通讯录管理系统完成