

通讯录管理系统

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、创建项目

创建项目步骤如下：

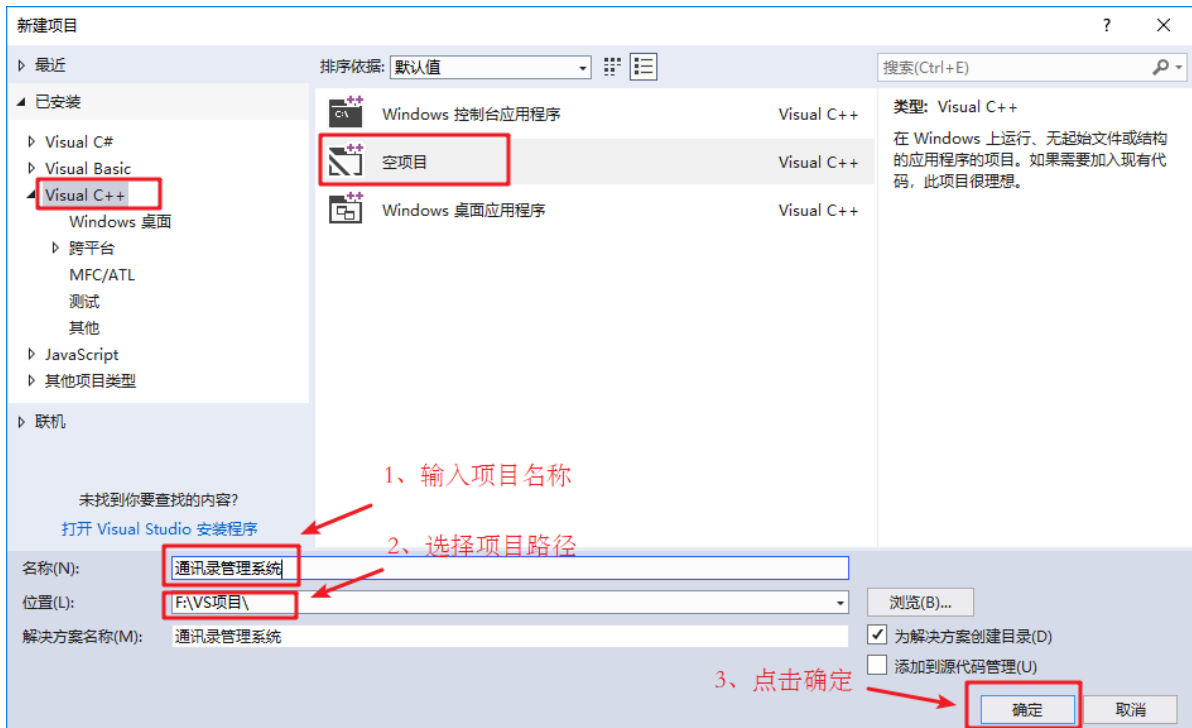
- 创建新项目
- 添加文件

2.1 创建项目

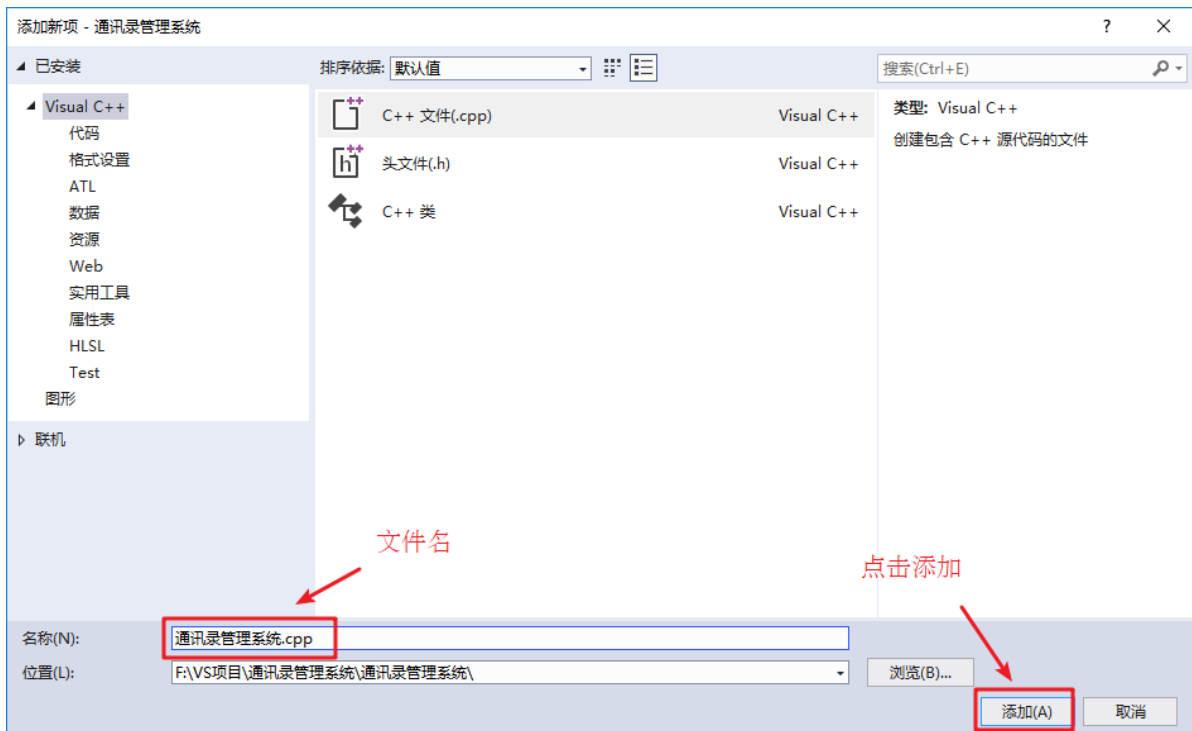
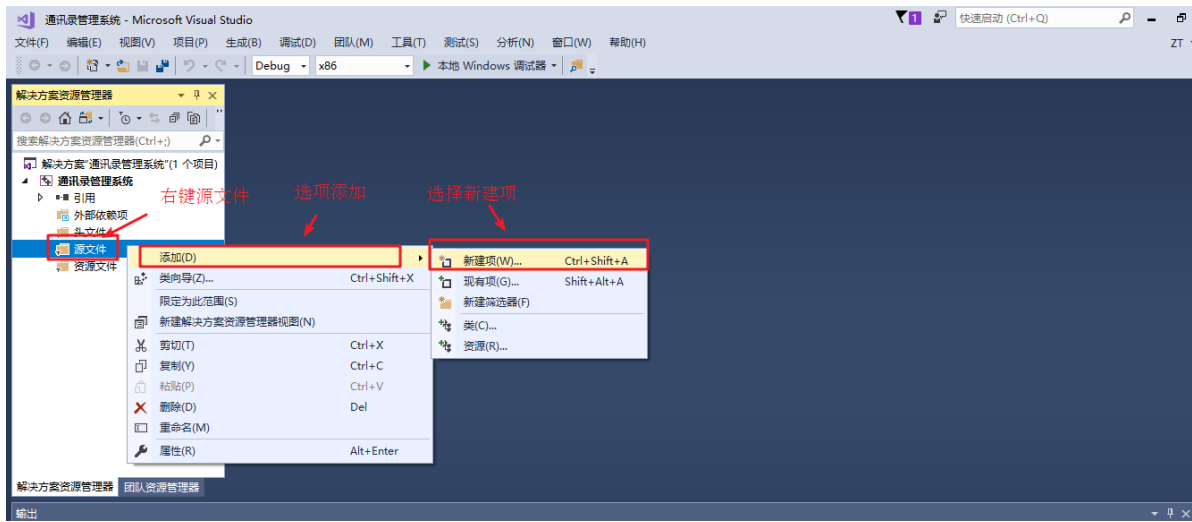
打开vs2017后，点击创建新项目，创建新的C++项目



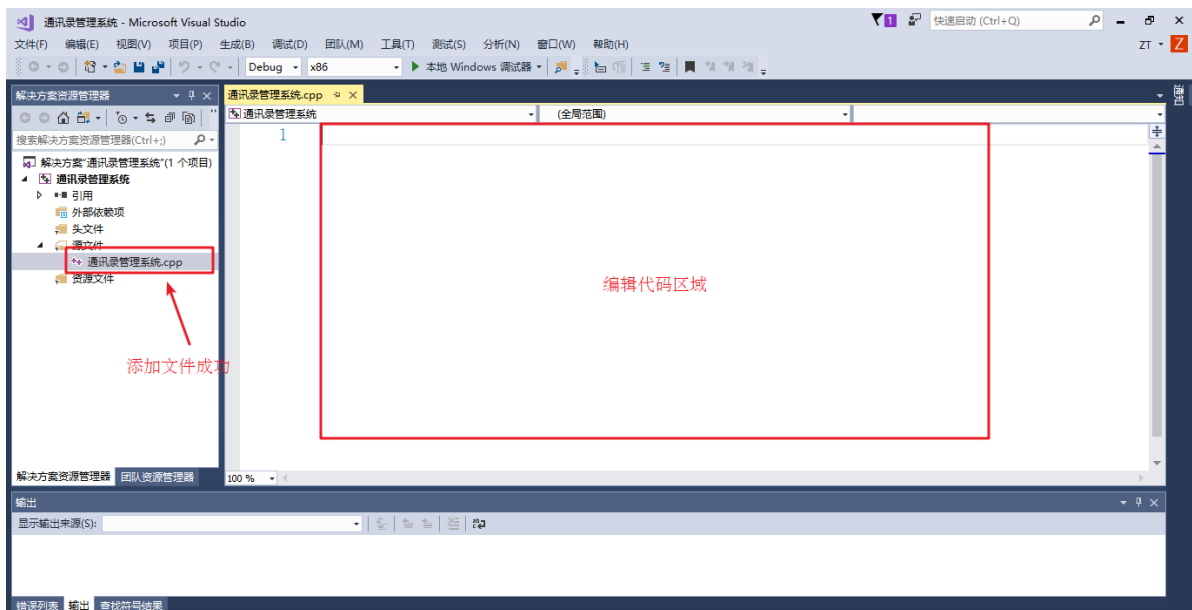
填写项目名称，选择项目路径



2.2添加文件



添加成功后，效果如图：



至此，项目已创建完毕

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  using namespace std;
3
4  //菜单界面
5  void showMenu()
6  {
7      cout << "*****" << endl;
8      cout << "***** 1、添加联系人 *****" << endl;
9      cout << "***** 2、显示联系人 *****" << endl;
10     cout << "***** 3、删除联系人 *****" << endl;
11     cout << "***** 4、查找联系人 *****" << endl;
12     cout << "***** 5、修改联系人 *****" << endl;
13     cout << "***** 6、清空联系人 *****" << endl;
14     cout << "***** 0、退出通讯录 *****" << endl;
15     cout << "*****" << endl;
16 }
```

```
17
18  int main() {
19
20      showMenu();
21
22      system("pause");
23
24      return 0;
25  }
```

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：

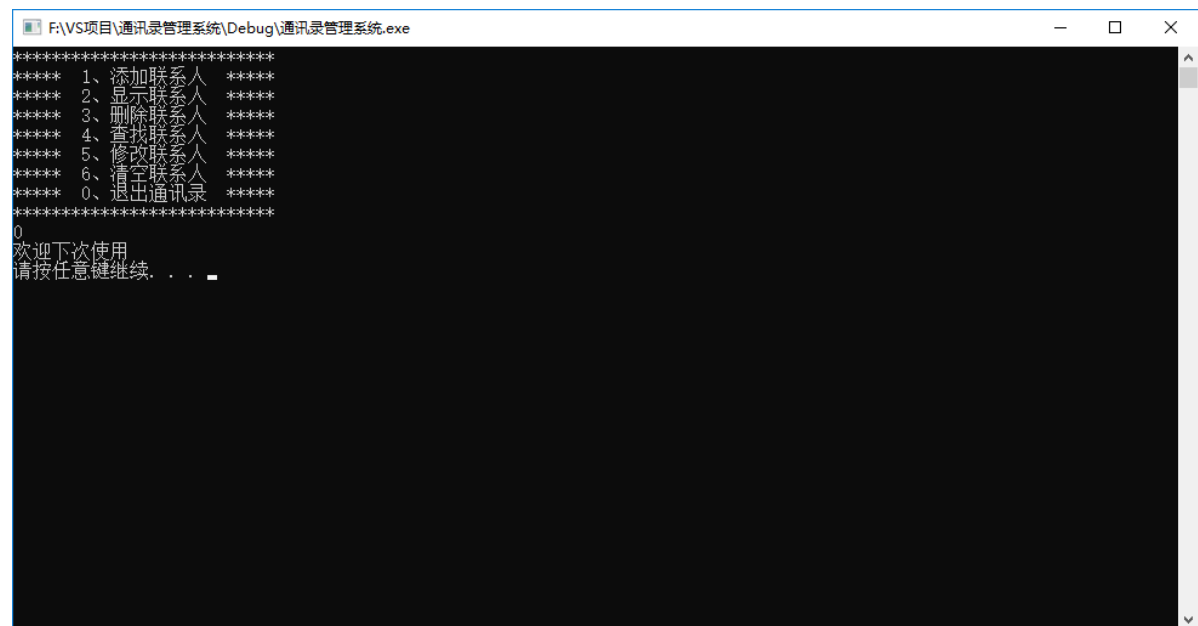
```
1  int main() {
2
3      int select = 0;
4
5      while (true)
6      {
7          showMenu();
8
9          cin >> select;
10
11         switch (select)
12         {
13             case 1: //添加联系人
14                 break;
15             case 2: //显示联系人
16                 break;
17             case 3: //删除联系人
18                 break;
19             case 4: //查找联系人
20                 break;
21             case 5: //修改联系人
22                 break;
23             case 6: //清空联系人
24                 break;
25             case 0: //退出通讯录
26                 cout << "欢迎下次使用" << endl;
```

```

27         system("pause");
28         return 0;
29         break;
30     default:
31         break;
32     }
33 }
34
35 system("pause");
36
37 return 0;
38 }

```

效果图：



```

F:\VS项目\通讯录管理系统\Debug\通讯录管理系统.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 *****
*****
0
欢迎下次使用
请按任意键继续. . .

```

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

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>  //string头文件
2  //联系人结构体
3  struct Person
4  {
5      string m_Name; //姓名
6      int m_Sex; //性别: 1男 2女
7      int m_Age; //年龄
8      string m_Phone; //电话
9      string m_Addr; //住址
10 };
```

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

设计如下

```
1  #define MAX 1000 //最大人数
2
3  //通讯录结构体
4  struct Addressbooks
5  {
6      struct Person personArray[MAX]; //通讯录中保存的联系人数组
7      int m_Size; //通讯录中人员个数
8  };
```

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

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、添加联系人信息
2 void addPerson(Addressbooks *abs)
3 {
4     //判断电话本是否满了
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        //姓名
13        string name;
14        cout << "请输入姓名: " << endl;
15        cin >> name;
16        abs->personArray[abs->m_Size].m_Name = name;
17
18        cout << "请输入性别: " << endl;
19        cout << "1 -- 男" << endl;
20        cout << "2 -- 女" << endl;
21
22        //性别
23        int sex = 0;
24        while (true)
25        {
26            cin >> sex;
27            if (sex == 1 || sex == 2)
28            {
29                abs->personArray[abs->m_Size].m_Sex = sex;
30                break;
31            }
32            cout << "输入有误，请重新输入";
33        }
34
35        //年龄
36        cout << "请输入年龄: " << endl;
37        int age = 0;
38        cin >> age;
39        abs->personArray[abs->m_Size].m_Age = age;
40
41        //联系电话
42        cout << "请输入联系电话: " << endl;
43        string phone = "";
```



```

44     cin >> phone;
45     abs->personArray[abs->m_Size].m_Phone = phone;
46
47     //家庭住址
48     cout << "请输入家庭住址: " << endl;
49     string address;
50     cin >> address;
51     abs->personArray[abs->m_Size].m_Addr = address;
52
53     //更新通讯录人数
54     abs->m_Size++;
55
56     cout << "添加成功" << endl;
57     system("pause");
58     system("cls");
59 }
60 }

```

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

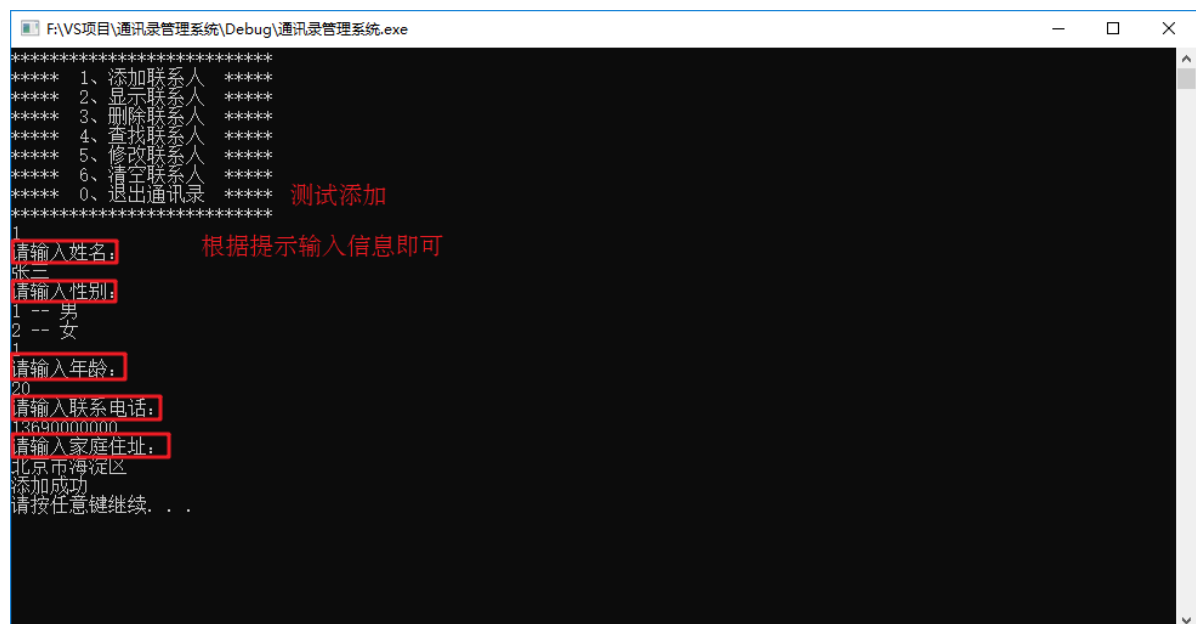
在switch case 语句中，case1里添加：

```

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

```

测试效果如图：



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

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、显示所有联系人信息
2 void showPerson(Addressbooks * abs)
3 {
4     if (abs->m_Size == 0)
5     {
6         cout << "当前记录为空" << endl;
7     }
8     else
9     {
10         for (int i = 0; i < abs->m_Size; i++)
11         {
12             cout << "姓名: " << abs->personArray[i].m_Name << "\t";
13             cout << "性别: " << (abs->personArray[i].m_Sex == 1 ? "男" : "女")
14             << "\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     system("pause");
21     system("cls");
22 }
23 }
```

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：



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

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 //判断是否存在查询的人员，存在返回在数组中索引位置，不存在返回-1
2 int isExist(Addressbooks * abs, string name)
3 {
4     for (int i = 0; i < abs->m_Size; i++)
5     {
6         if (abs->personArray[i].m_Name == name)
7         {
8             return i;
9         }
10    }
11    return -1;
12 }
13
```

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、删除指定联系人信息
2 void deletePerson(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     int ret = isExist(abs, name);
9     if (ret != -1)
10    {
11        for (int i = ret; i < abs->m_Size; i++)
12        {
13            abs->personArray[i] = abs->personArray[i + 1];
14        }
15        abs->m_Size--;
16    }
17 }
```

```

16         cout << "删除成功" << endl;
17     }
18     else
19     {
20         cout << "查无此人" << endl;
21     }
22
23     system("pause");
24     system("cls");
25 }

```

7.3 测试删除联系人功能

在switch case 语句中，case3里添加：

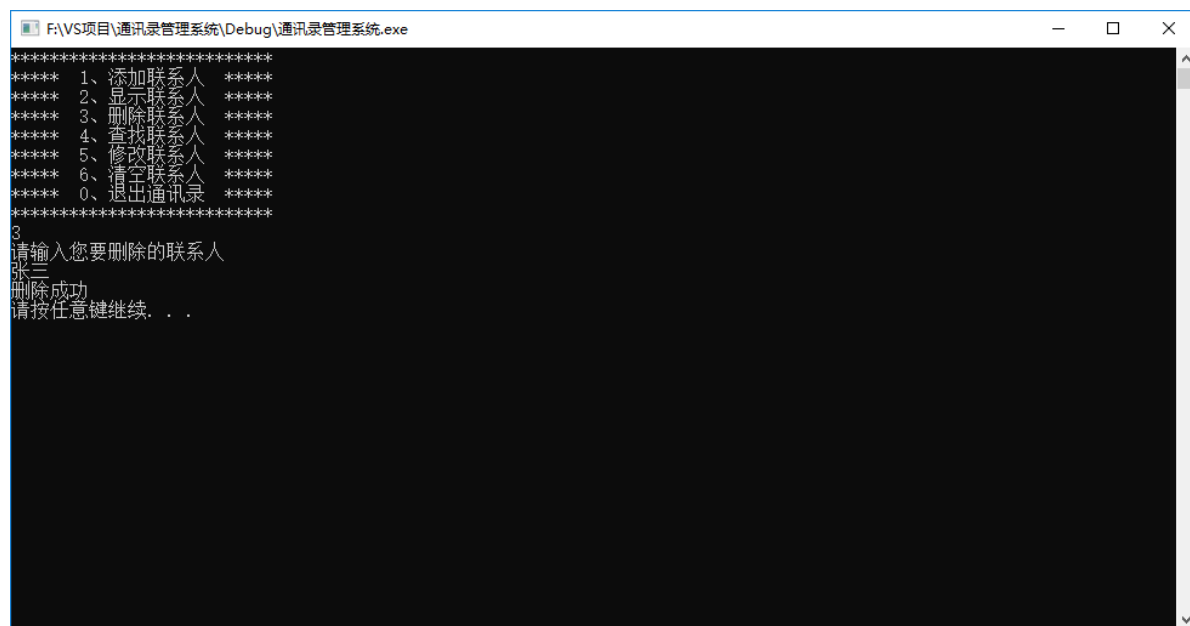
```

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

```

测试效果如图：

存在情况：

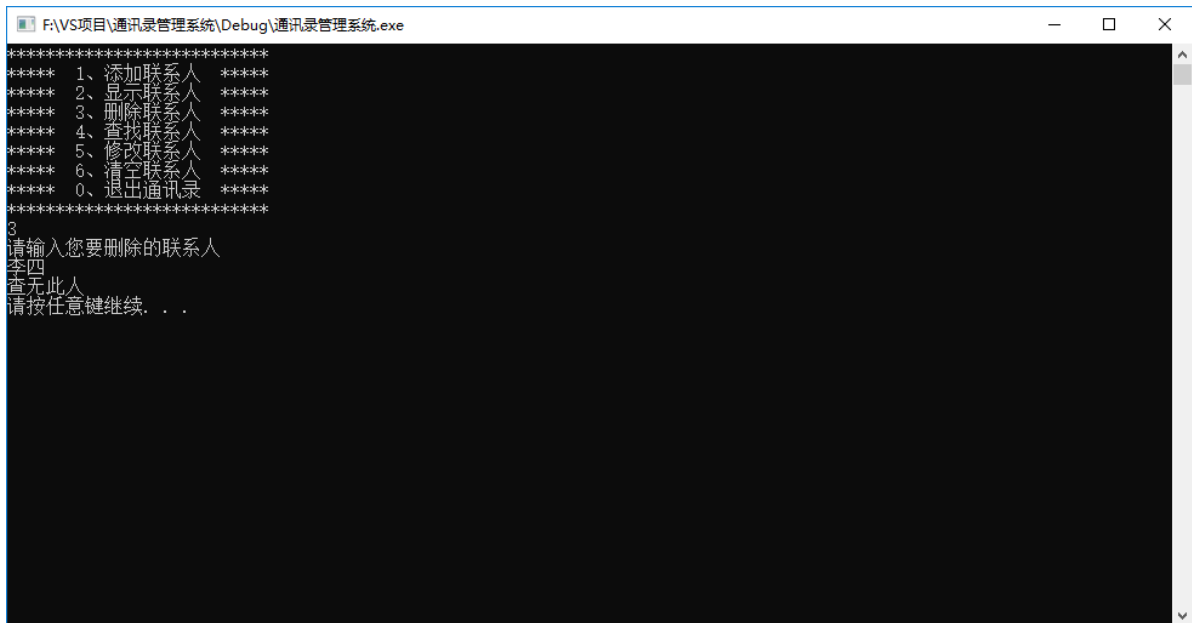


```

F:\VS项目\通讯录管理系统\Debug\通讯录管理系统.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. . .

```

不存在情况：



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

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(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     int ret = isExist(abs, name);
9     if (ret != -1)
10    {
11        cout << "姓名: " << abs->personArray[ret].m_Name << "\t";
12        cout << "性别: " << abs->personArray[ret].m_Sex << "\t";
13        cout << "年龄: " << abs->personArray[ret].m_Age << "\t";
14        cout << "电话: " << abs->personArray[ret].m_Phone << "\t";
15        cout << "住址: " << abs->personArray[ret].m_Addr << endl;
16    }
17    else
18    {
```

```

19         cout << "查无此人" << endl;
20     }
21
22     system("pause");
23     system("cls");
24
25 }

```

8.2 测试查找指定联系人

在switch case 语句中，case4里添加：

```

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

```

测试效果如图

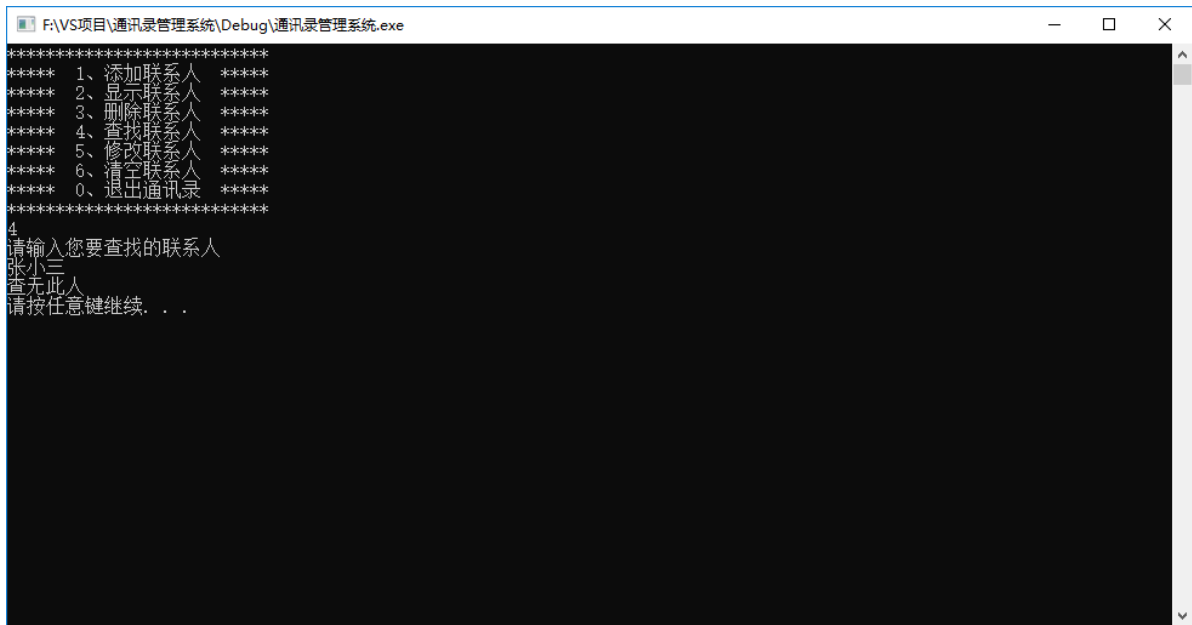
存在情况：

```

F:\VS项目\通讯录管理系统\Debug\通讯录管理系统.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
张三
姓名：张三      性别：1 年龄：20      电话：13690000000      住址：北京市海淀区
请按任意键继续. . .

```

不存在情况：



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

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、修改指定联系人信息
2 void modifyPerson(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)
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     cout << "请输入性别: " << endl;
18     cout << "1 -- 男" << endl;
19     cout << "2 -- 女" << endl;
20
21     //性别
22     int sex = 0;
23     while (true)
24     {
25         cin >> sex;
26         if (sex == 1 || sex == 2)
27         {
28             abs->personArray[ret].m_Sex = sex;
29             break;
30         }
31         cout << "输入有误, 请重新输入";
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     //联系电话
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     //家庭住址
47     cout << "请输入家庭住址: " << endl;
48     string address;
49     cin >> address;
50     abs->personArray[ret].m_Addr = address;
51
52     cout << "修改成功" << endl;
53 }
54 else
55 {
56     cout << "查无此人" << endl;
57 }
58
59 system("pause");
60 system("cls");
61
62 }

```

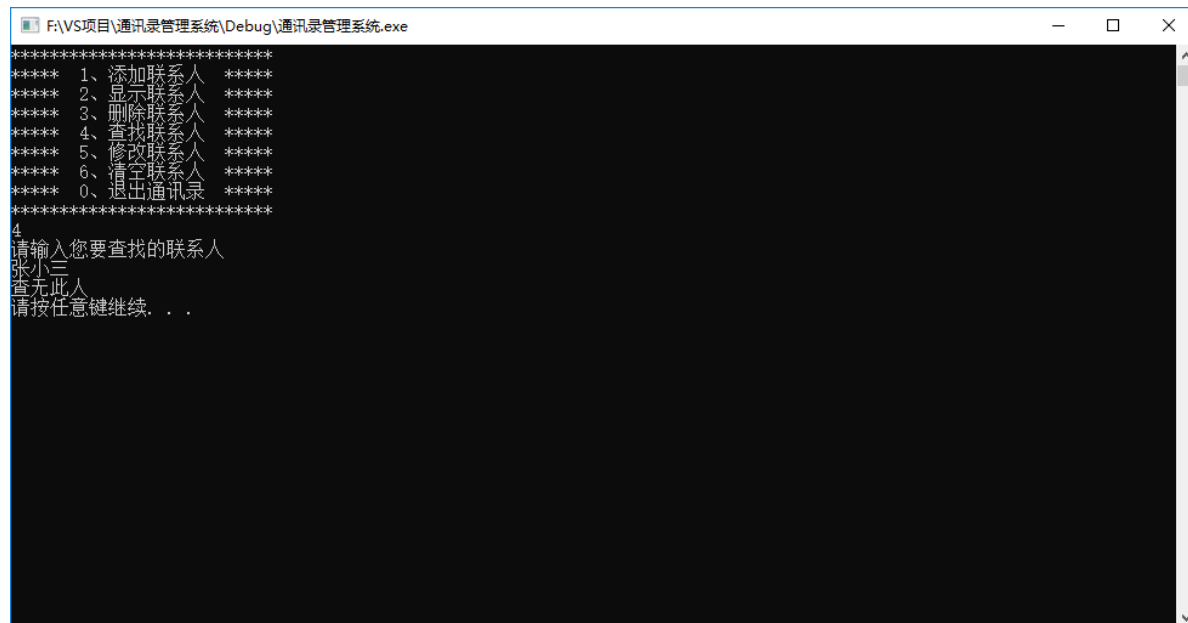
9.2 测试修改联系人功能

在switch case 语句中，case 5里添加：

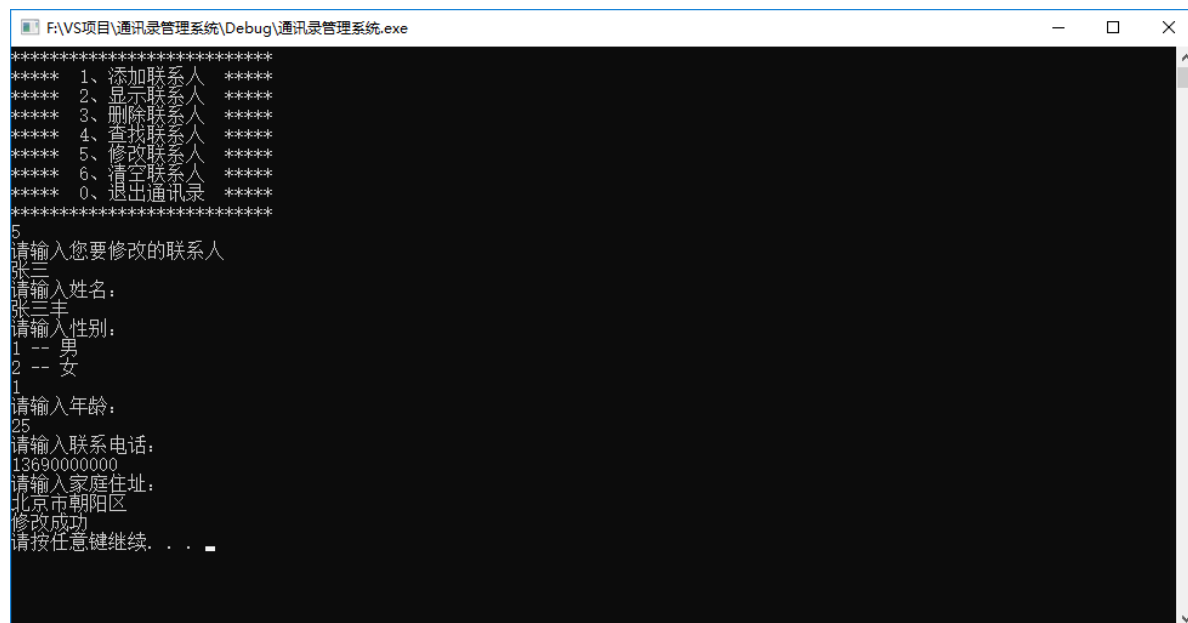
```
1 case 5: //修改联系人
2     modifyPerson(&abs);
3     break;
```

测试效果如图：

查不到指定联系人情况：



查找到联系人，并修改成功：



再次查看通讯录，确认修改完毕



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

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(Addressbooks * abs)
3 {
4     abs->m_Size = 0;
5     cout << "通讯录已清空" << endl;
6     system("pause");
7     system("cls");
8 }
```

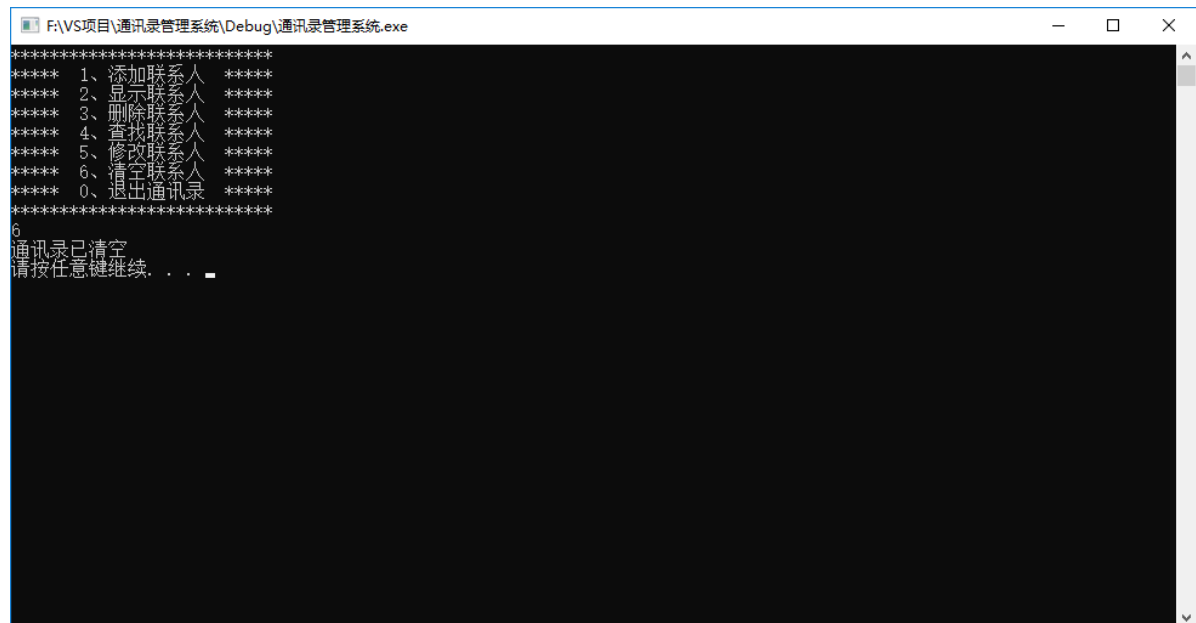
10.2 测试清空联系人

在switch case 语句中，case 6 里添加：

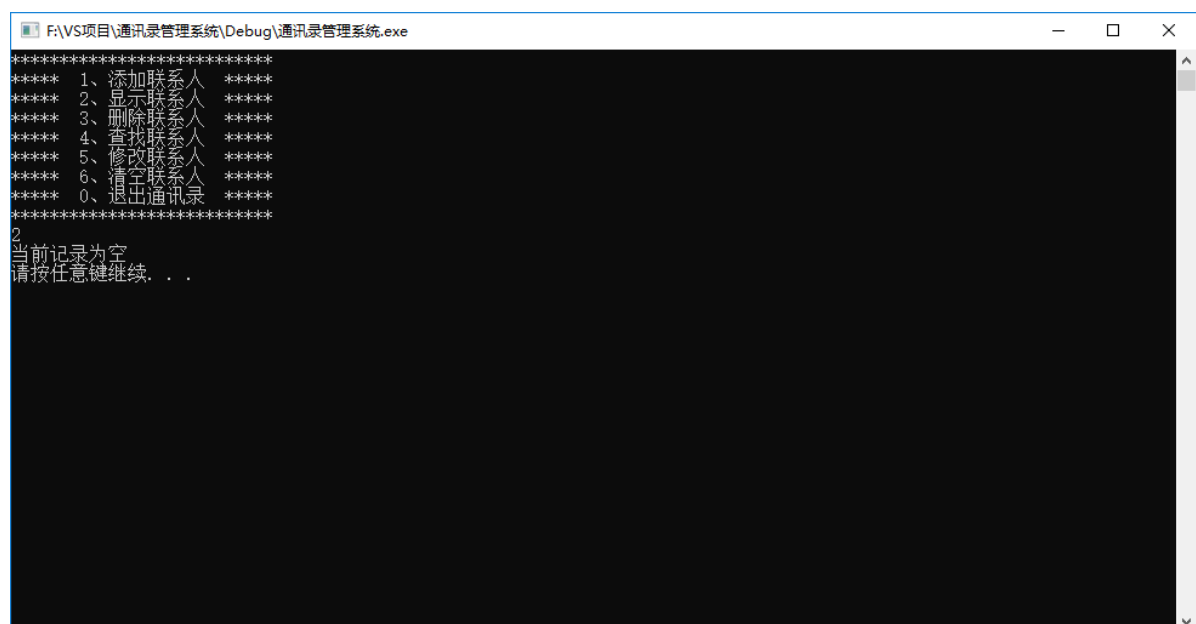
```
1 case 6: //清空联系人
2     cleanPerson(&abs);
3     break;
```

测试效果如图：

清空通讯录



再次查看信息，显示记录为空



至此，通讯录管理系统完成！

