**项目说明文档**

**数据结构课程设计**

**——家谱管理系统**

作 者 姓 名： 戴仁杰

学 号： 1951650

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1 分析 - 4 -](#_Toc91169011)

[1.1 背景分析 - 4 -](#_Toc91169012)

[1.2 功能分析 - 4 -](#_Toc91169013)

[2 设计 - 4 -](#_Toc91169014)

[2.1 数据结构设计 - 4 -](#_Toc91169015)

[2.2 类结构设计 - 4 -](#_Toc91169016)

[2.3 成员与操作设计 - 5 -](#_Toc91169017)

[2.3.1 Vector类 - 5 -](#_Toc91169018)

[2.3.2 GenealogicalTreeNode类 - 5 -](#_Toc91169019)

[2.3.3 GenealogicalTree类 - 6 -](#_Toc91169020)

[2.3.4 GenealogicalManagementSystem类 - 6 -](#_Toc91169021)

[2.4 系统设计 - 6 -](#_Toc91169022)

[3 实现 - 7 -](#_Toc91169023)

[3.1 初始化家谱功能的实现 - 7 -](#_Toc91169024)

[3.1.1 初始化家谱功能流程图 - 7 -](#_Toc91169025)

[3.1.2 初始化家谱功能核心代码 - 7 -](#_Toc91169026)

[3.1.3 初始化家谱功能截屏示例 - 7 -](#_Toc91169027)

[3.2 完善家谱功能的实现 - 8 -](#_Toc91169028)

[3.2.1 完善家谱功能流程图 - 8 -](#_Toc91169029)

[3.2.2 完善家谱功能核心代码 - 8 -](#_Toc91169030)

[3.2.3 完善家谱功能截屏示例 - 9 -](#_Toc91169031)

[3.3 添加家庭成员功能的实现 - 10 -](#_Toc91169032)

[3.3.1 添加家庭成员功能流程图 - 10 -](#_Toc91169033)

[3.3.2 添加家庭成员功能核心代码 - 10 -](#_Toc91169034)

[3.3.3 添加家庭成员功能截屏示例 - 11 -](#_Toc91169035)

[3.4 解散局部家庭功能的实现 - 11 -](#_Toc91169036)

[3.4.1 解散局部家庭功能流程图 - 11 -](#_Toc91169037)

[3.4.2 解散局部家庭功能核心代码 - 11 -](#_Toc91169038)

[3.4.3 解散局部家庭功能截屏示例 - 12 -](#_Toc91169039)

[3.5 更改成员姓名功能的实现 - 13 -](#_Toc91169040)

[3.5.1 更改成员姓名功能流程图 - 13 -](#_Toc91169041)

[3.5.2 更改成员姓名功能核心代码 - 13 -](#_Toc91169042)

[3.5.3 更改成员姓名功能截屏示例 - 14 -](#_Toc91169043)

[4 测试 - 14 -](#_Toc91169044)

[4.1 功能测试 - 14 -](#_Toc91169045)

[4.1.1 初始化家谱功能测试 - 14 -](#_Toc91169046)

[4.1.2 完善家谱功能测试 - 14 -](#_Toc91169047)

[4.1.3 添加家庭成员功能测试 - 15 -](#_Toc91169048)

[4.1.4 解散局部家庭功能测试 - 15 -](#_Toc91169049)

[4.1.5 更改成员姓名功能测试 - 16 -](#_Toc91169050)

[4.2 边界测试 - 16 -](#_Toc91169051)

[4.2.1 给祖先成员改名 - 16 -](#_Toc91169052)

[4.2.2 解散局部家庭时解散第一代祖先家庭 - 16 -](#_Toc91169053)

[4.3 出错测试 - 17 -](#_Toc91169054)

[4.3.1 完善家谱时输入添加儿女个数为零 - 17 -](#_Toc91169055)

[4.3.2 完善家谱时新添加儿女姓名过多 - 17 -](#_Toc91169056)

[4.3.3 输入操作数孩子数目等不合法 - 18 -](#_Toc91169057)

1 分析

1.1 背景分析

家谱，又称族谱、宗谱等。是一种以表谱形式，记载一个家族的世系繁衍及重要人物事迹的书。家谱是一种特殊的文献，就其内容而言，是中华文明史中具有平民特色的文献，记载的是同宗共祖血缘集团世系人物和事迹等方面情况的历史图籍。家谱属珍贵的人文资料，对于历史学、民俗学、人口学、社会学和经济学的深入研究，均有其不可替代的独特功能。

以往的家谱都是用纸张来进行记载的，这种记载方式不仅修改起来比较麻烦，而且还很容易丢失，复制起来也十分不方便。如果制作一个家谱管理系统，用计算机来进行管理，那么如果需要修改家谱的话，只需要按操作输入修改信息，计算机就可以自动进行修改，十分便捷。同时家谱保存在计算机中，也较为安全，不易丢失。因此，开发一个基于计算机操作的家谱管理系统是十分有必要的。

1.2 功能分析

作为一个家谱管理系统，首先要能够初始化输入家族的祖先，这样才能进行之后的操作。其次，家谱系统还需要能够不断添加新的家庭成员，完善整个家谱。如果家庭内部出现离异等情况，还需要将离异的家庭解散掉。最后，家庭内部如果有人改名，系统应当也可以对应修改此人的名字。为了方便用户操作，家谱管理系统可以设置菜单栏来指引用户。

综上所述，该家谱管理系统需要有初始化，完善家谱，添加家庭成员，解散家庭，更改成员姓名，退出系统的功能。

2 设计

2.1 数据结构设计

如上功能分析所述，该家谱管理系统要求大量增加，删除操作。增加操作要求在一个双亲下添加多名子女，删除操作要求将目标结点以及目标的所有子女全部删除。家谱的结构类似于一种树的结构。增加时在一个树枝上加上叶子，删除时将树枝和上面的叶子一起砍掉。因此决定采用树结构来储存家谱。由于每个双亲结点的子女个数不确定，因此使用多叉树数据结构，并采用左子女-右兄弟链表表示法来存储。

2.2 类结构设计

由于添加子女的时候可能会一次添加多个，但是添加的时候需要将它们输入完，比对是否有重名后再保存进家谱，因此首先设计了Vector向量类来保存输入数据。为了能够正常的储存和查找家谱信息，本系统设计了一个家族树（GenealogicalTree）类，以及它的结点类：家族树结点（GenealogicalTreeNode）类。为了使Vector类与树更具有泛用性，本系统将Vector类，GenealogicalTree类，GenealogicalTreeNode类都设计为了模板类。除此之外，为了方便管理，并丰富与用户的交互性，以及增强输入错误处理能力，本系统还设置了家谱管理系统（GenealogicalManagementSystem）类，给用户提供完善家谱，添加家庭成员，解散家庭，更改成员姓名等功能的接口。

2.3 成员与操作设计

2.3.1 Vector类

Vector类为向量类，类内实现了下标索引，插入元素，删除元素，获得数组大小等操作。通过动态内存申请调整大小，避免直接声明数组造成的空间浪费。



2.3.2 GenealogicalTreeNode类

GenealogicalTreeNode类为家谱树结点类，采用了类模板提高泛用性。提供默认构造函数、带参构造函数以及与外接数据交流的一系列get与set函数。类内数据为data数据域、孩子指针、兄弟指针以及双亲指针域，存储结点信息并与孩子兄弟双亲结点建立联系。



2.3.3 GenealogicalTree类

GenealogicalTree类为家谱树类，采用了类模板提高泛用性。提供构造函数新建家谱树，析构函数释放内存。提供寻找树中成员的函数、清空家谱树的函数、删除某结点及其孩子兄弟节点的函数。此外，提供一系列get和set函数与外界进行数据交流。类内数据为家谱树根节点。



2.3.4 GenealogicalManagementSystem类

GenealogicalManagementSystem类为家谱管理系统类。对家谱树类进行封装增强，丰富了交互提示信息，增强了输入错误、边界错误处理性能。并封装了运行家谱管理系统的函数runGenealogicalManagementSystem()，增强了外部调用的友好性。类内数据为一个家谱树类型变量。



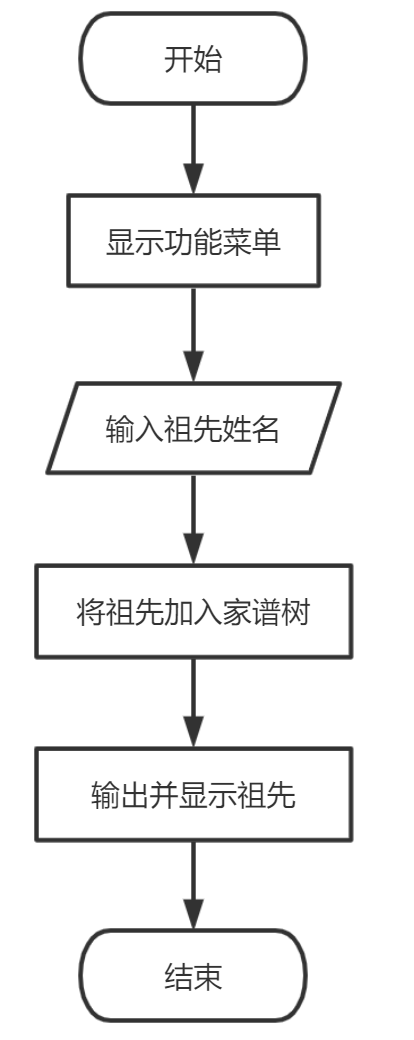
2.4 系统设计

系统在使用时，首先要输入祖先姓名来初始化家谱，随后就可以根据菜单，输入对应的操作码来执行相应的操作。功能有完善家谱，添加家庭成员，解散家庭，更改成员姓名，退出系统等等。

3 实现

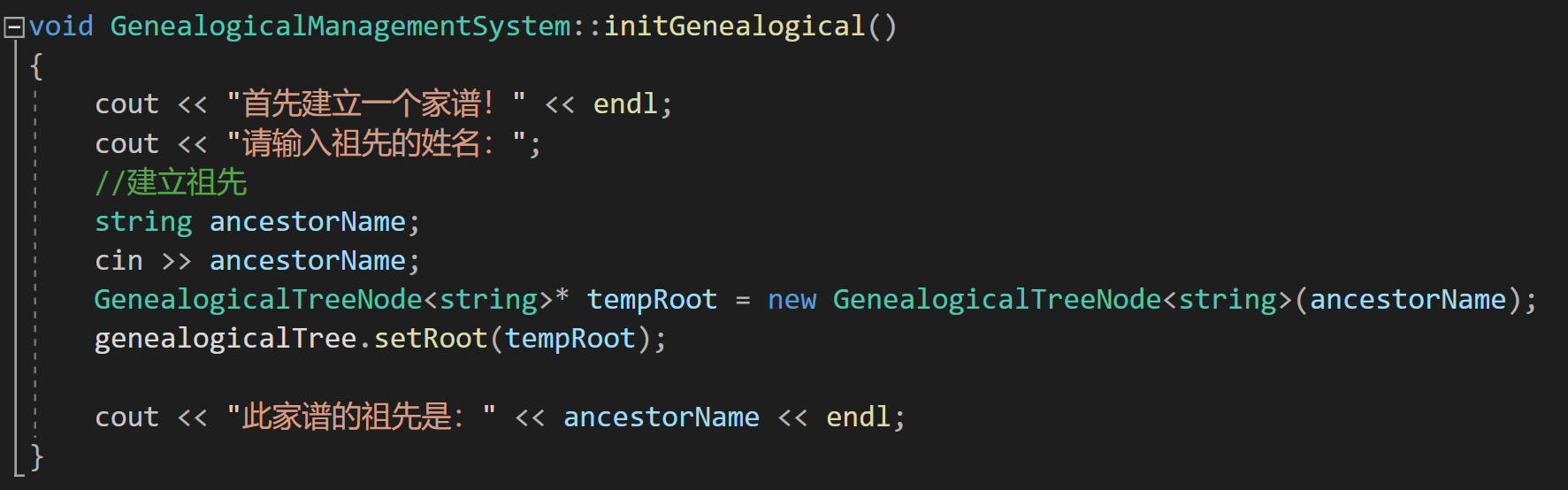
3.1 初始化家谱功能的实现

3.1.1 初始化家谱功能流程图

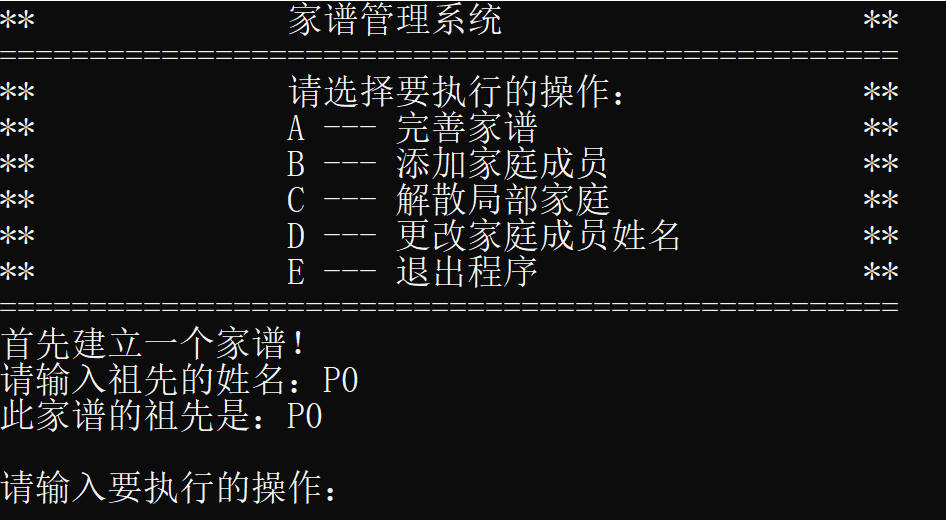


3.1.2 初始化家谱功能核心代码

GenealogicalManagementSystem类中的initGenealogical()函数

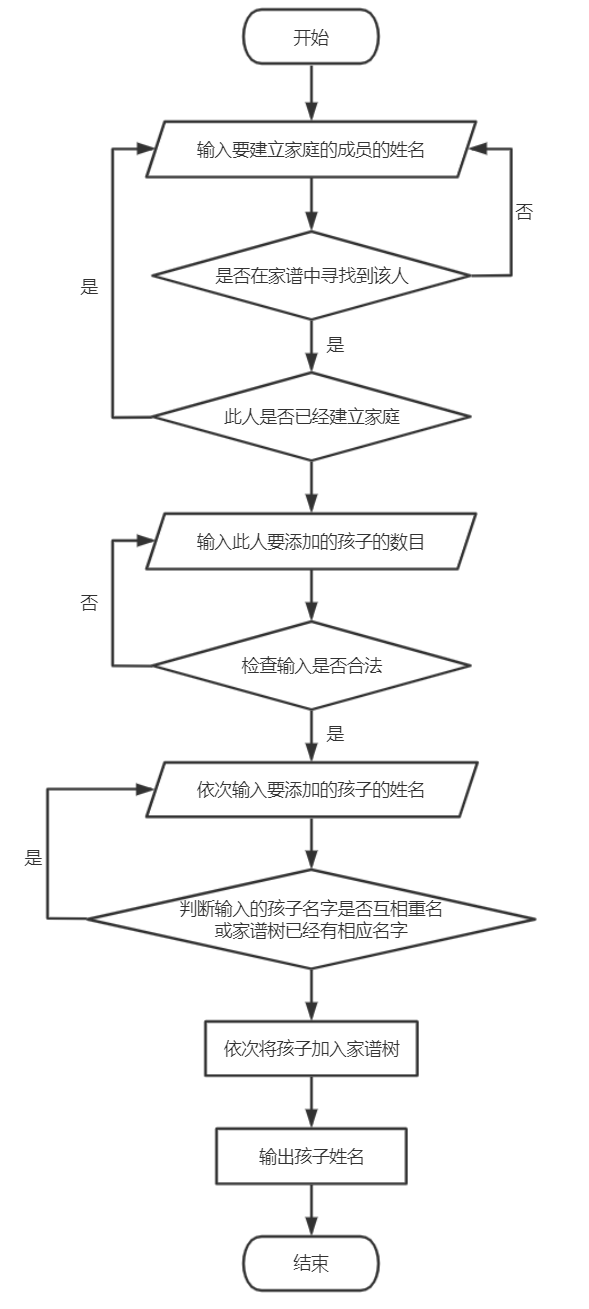


3.1.3 初始化家谱功能截屏示例



3.2 完善家谱功能的实现

3.2.1 完善家谱功能流程图

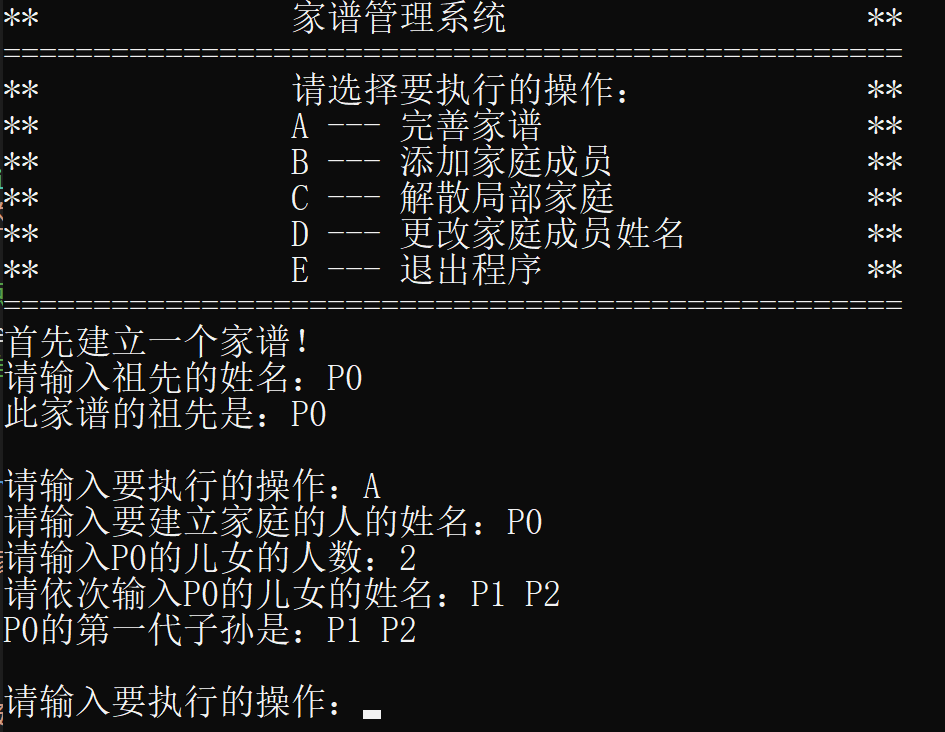


3.2.2 完善家谱功能核心代码

GenealogicalManagementSystem类中的initGenealogical()函数

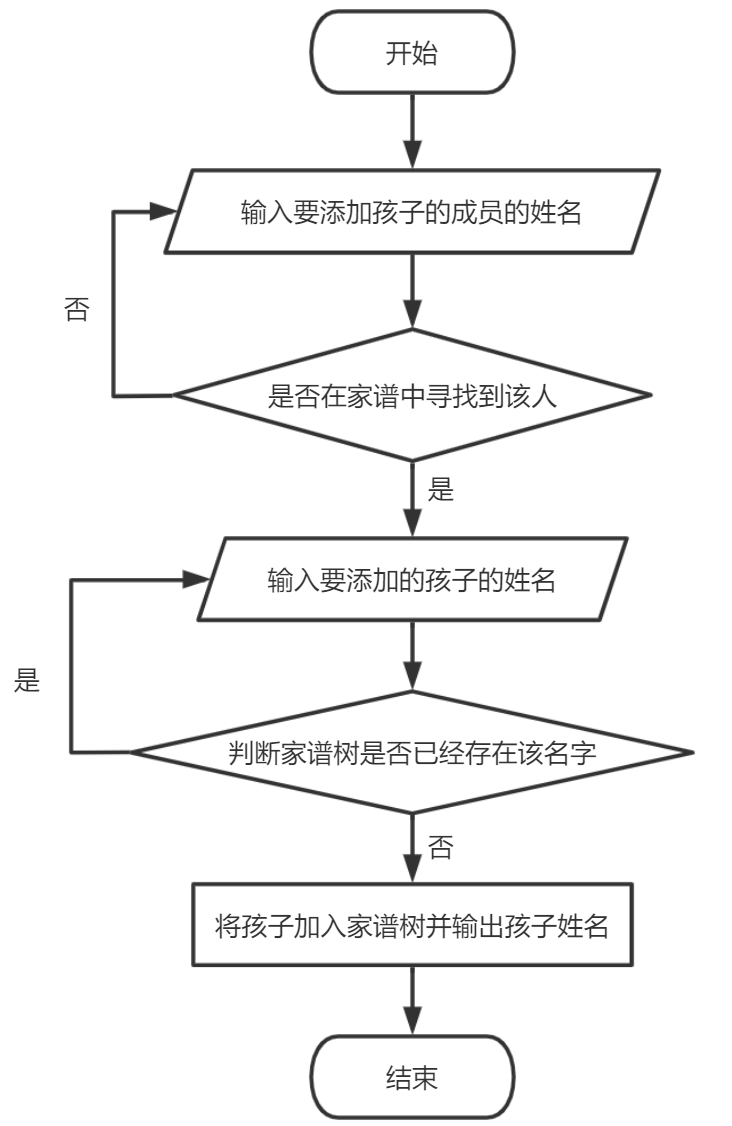


3.2.3 完善家谱功能截屏示例



3.3 添加家庭成员功能的实现

3.3.1 添加家庭成员功能流程图

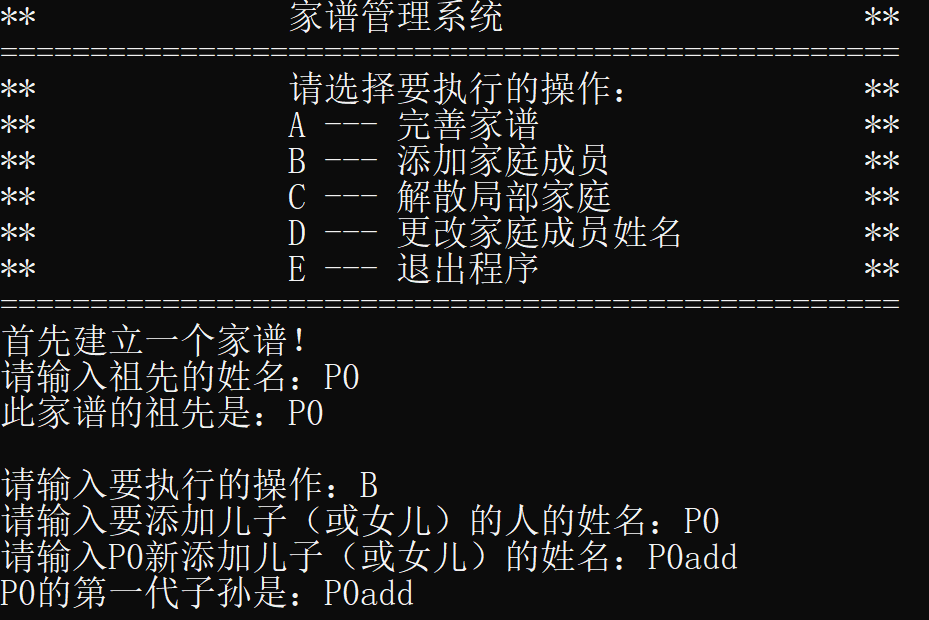


3.3.2 添加家庭成员功能核心代码

GenealogicalManagementSystem类中的addFamilyMember()函数

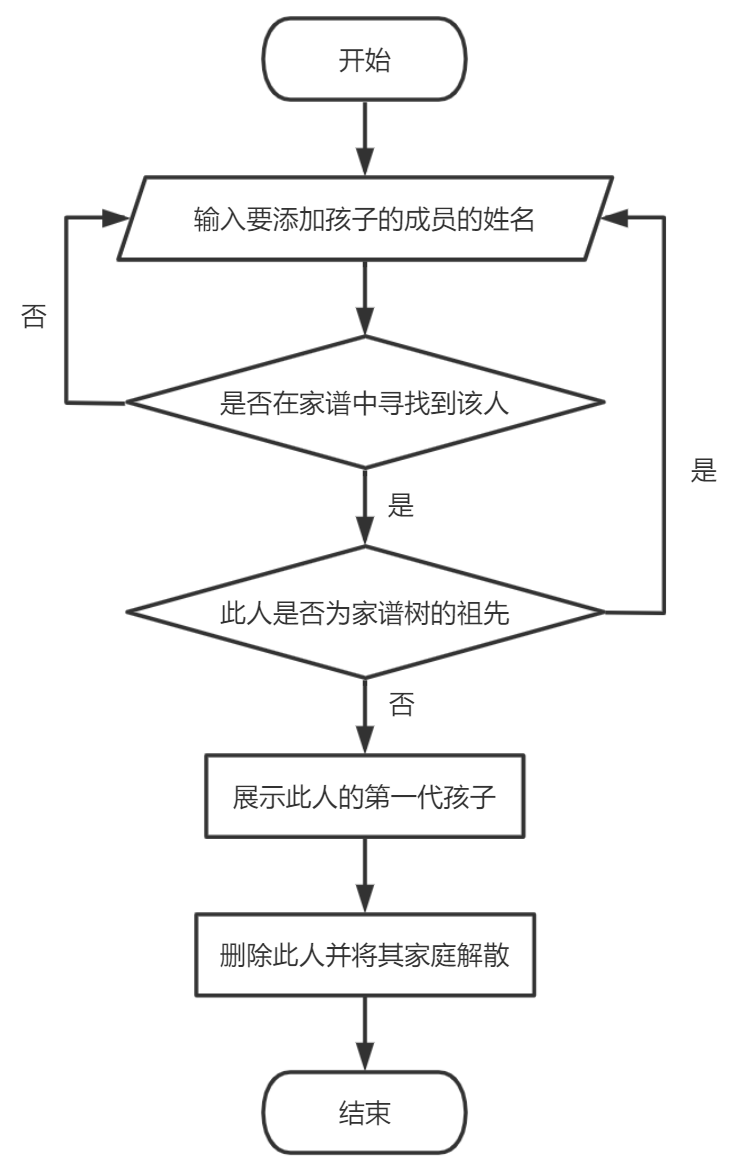


3.3.3 添加家庭成员功能截屏示例



## 3.4 解散局部家庭功能的实现

### 3.4.1 解散局部家庭功能流程图

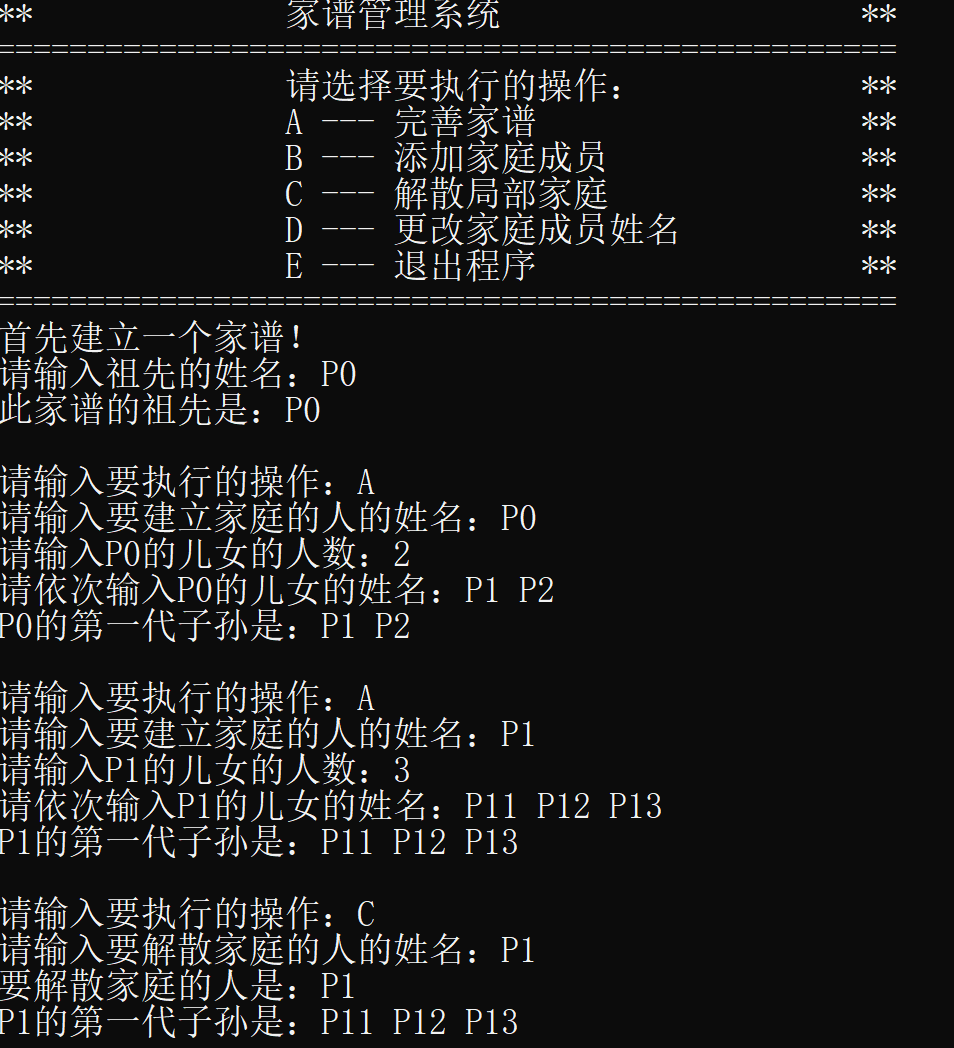


3.4.2 解散局部家庭功能核心代码

GenealogicalManagementSystem类中的removeFamily()函数

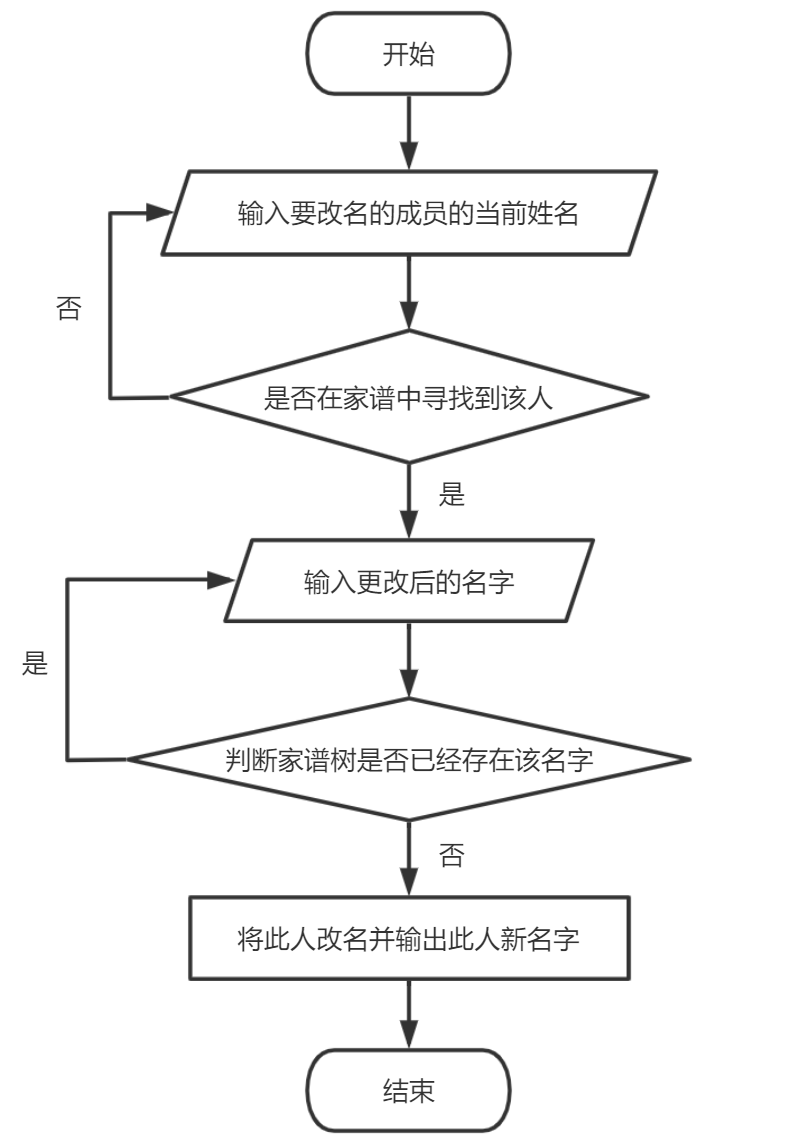


3.4.3 解散局部家庭功能截屏示例



3.5 更改成员姓名功能的实现

3.5.1 更改成员姓名功能流程图

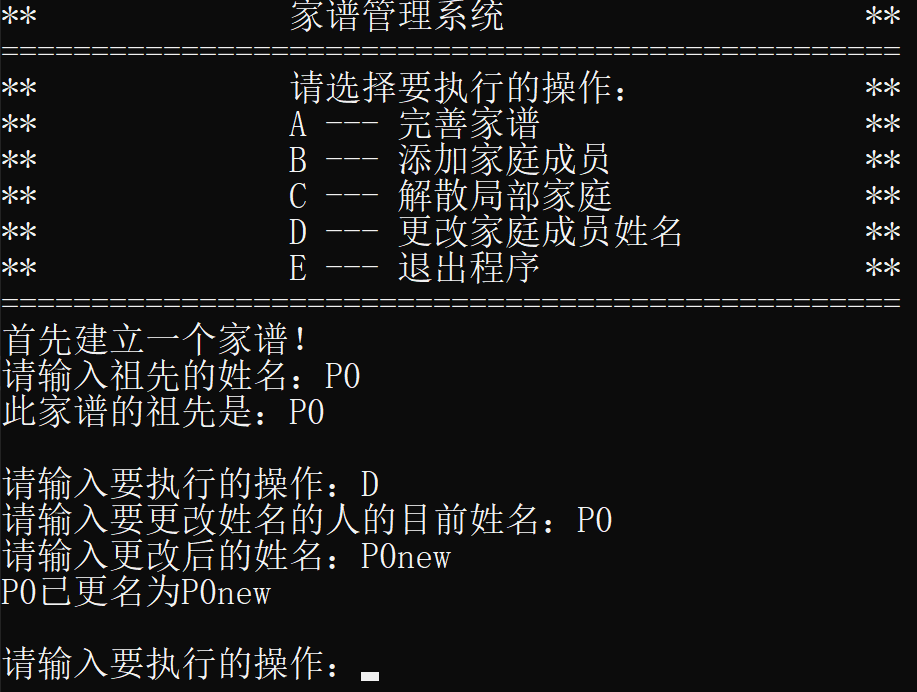


3.5.2 更改成员姓名功能核心代码

GenealogicalManagementSystem类中的changeMemberName()函数



3.5.3 更改成员姓名功能截屏示例



# 4 测试

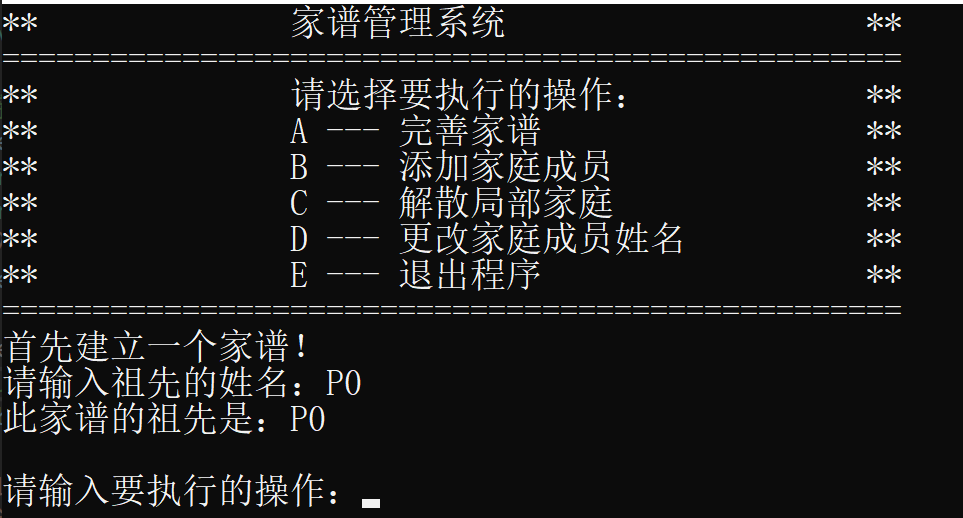
4.1 功能测试

4.1.1 初始化家谱功能测试

**测试用例**：P0

**预期结果**：程序正常运行不崩溃，输出“此家谱的祖先为：P0”

**实验结果：**

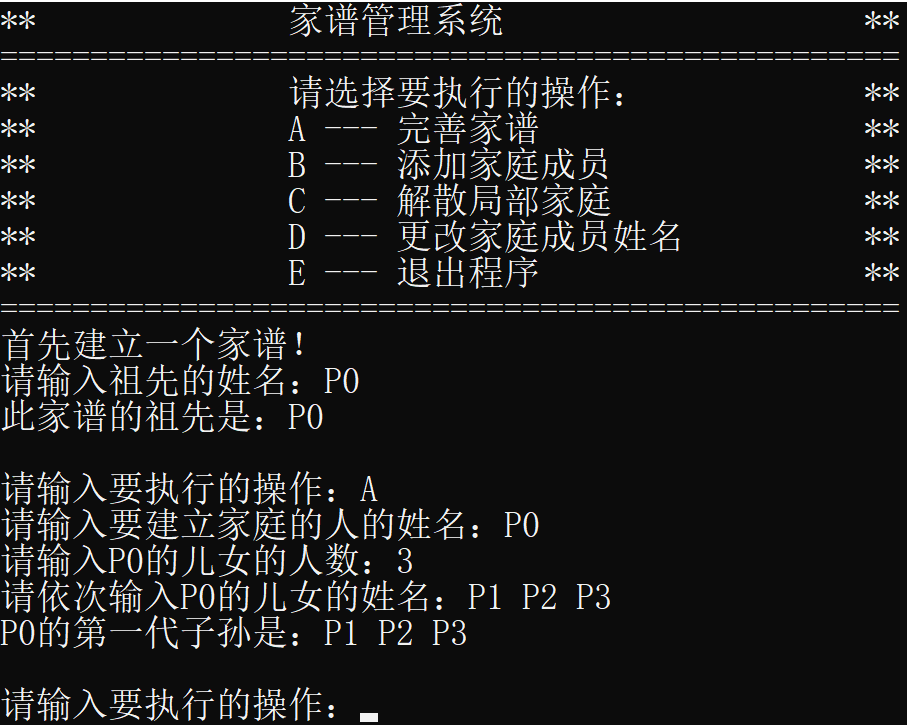


4.1.2 完善家谱功能测试

**测试用例：**祖先P0插入3个孩子P1 P2 P3

**预期结果**：程序正常运行不崩溃,输出“P0的第一代子孙是：P1 P2 P3”

**实验结果：**

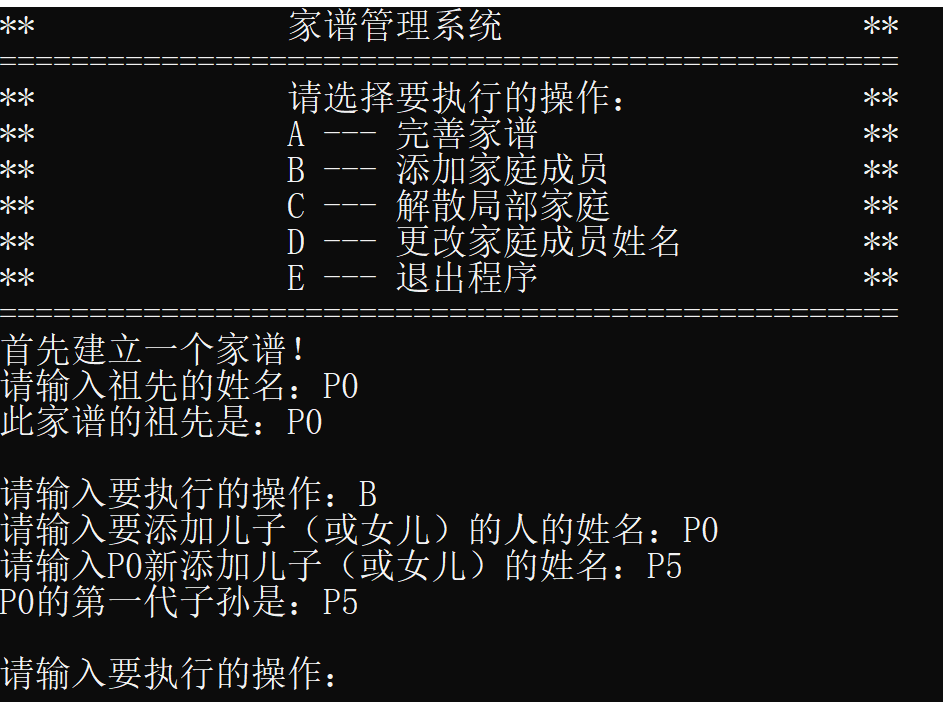


4.1.3 添加家庭成员功能测试

**测试用例：**祖先P0添加孩子P5

**预期结果**：程序正常运行不崩溃,输出“P0的第一代子孙是：P5”

**实验结果：**

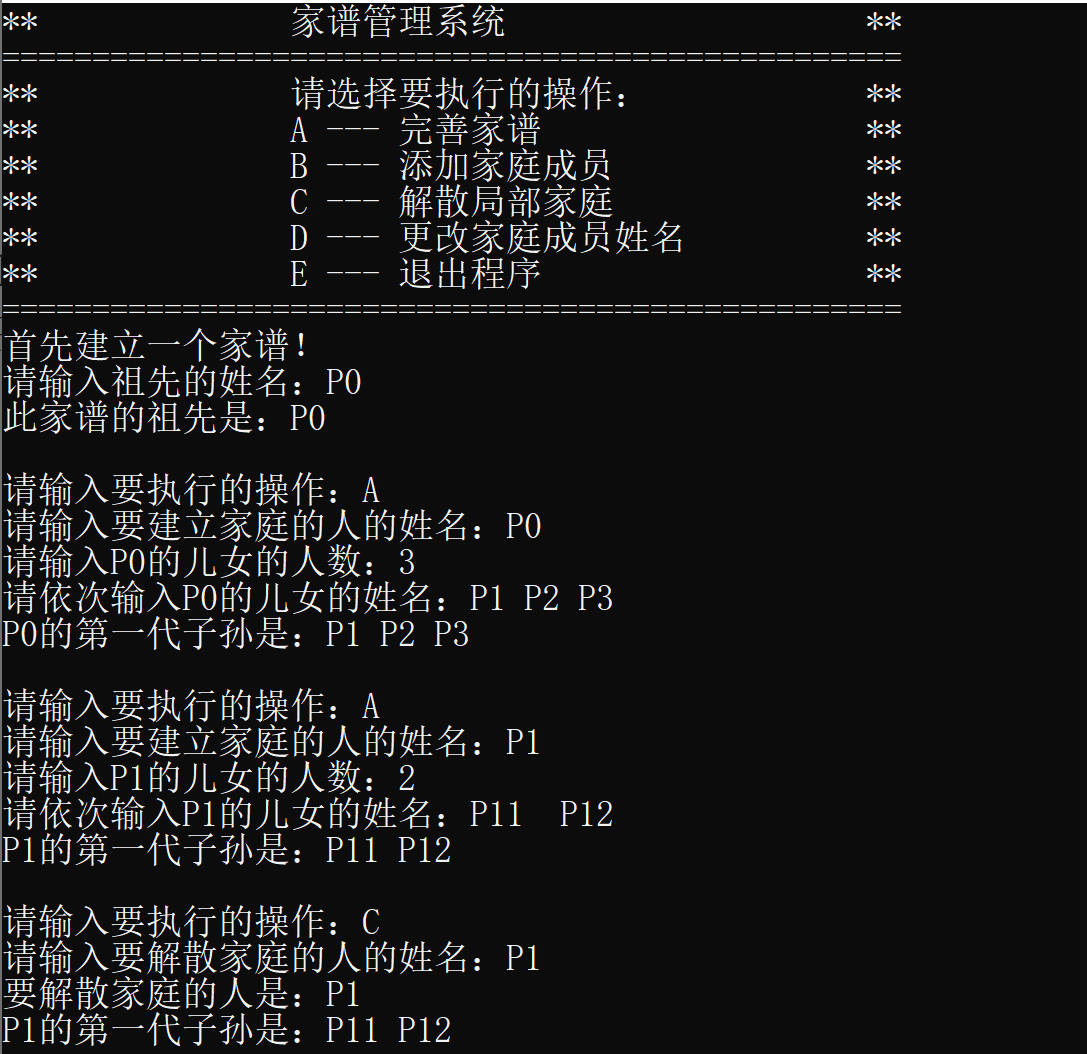


4.1.4 解散局部家庭功能测试

**测试用例：**祖先P0有孩子P1 P2 P3，P1有孩子P11 P12 解散P1的家庭

**预期结果：**程序正确输出“要解散家庭的人是P1，P1的第一代子孙是P11”

**实验结果：**

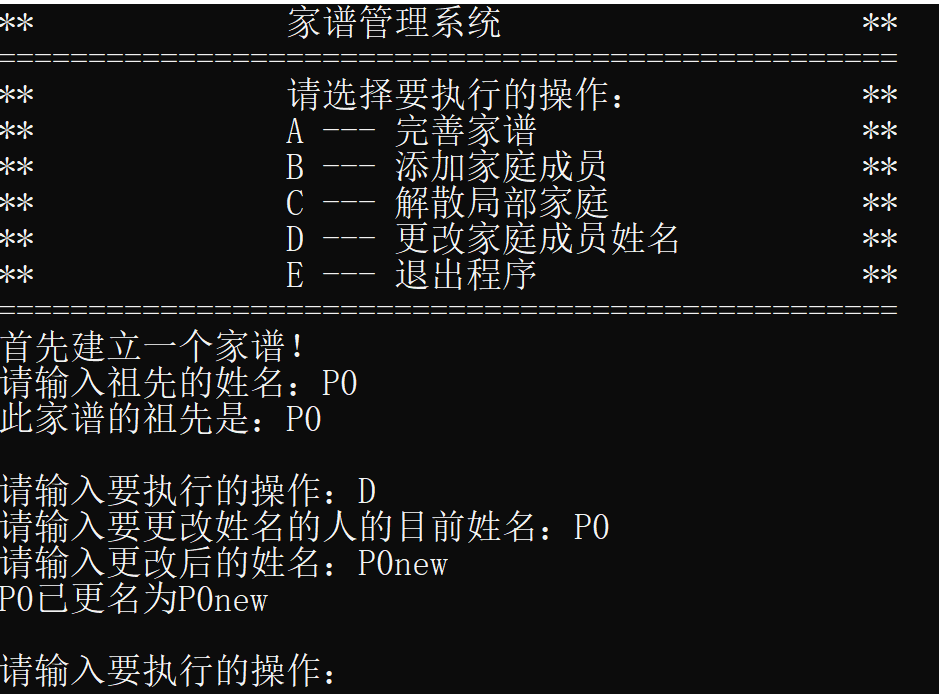


4.1.5 更改成员姓名功能测试

**测试用例：**祖先P0改名为P0new

**预期结果：**程序正常运行不崩溃,输出“P0已更名为P0new”

**实验结果：**



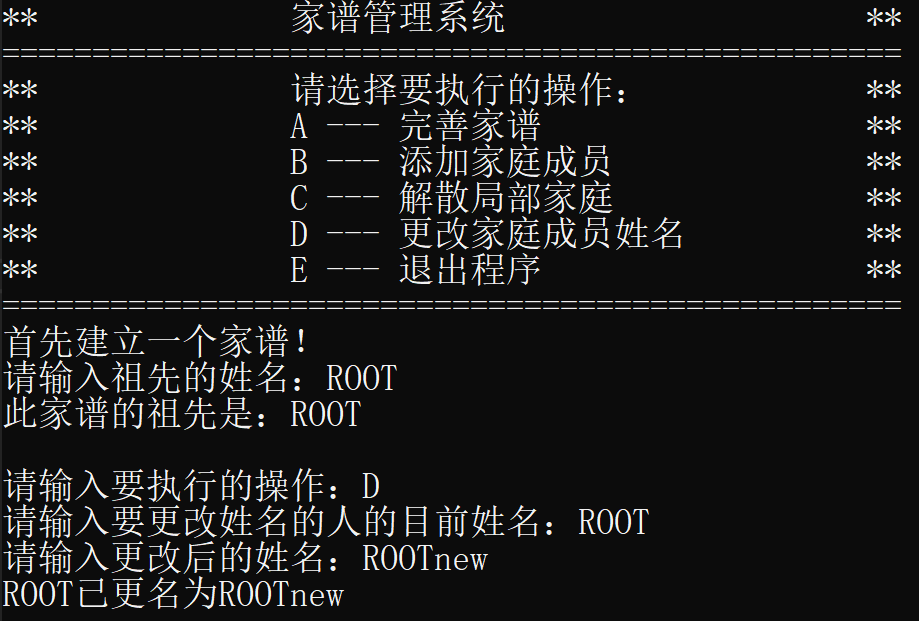
4.2 边界测试

4.2.1 给祖先成员改名

**测试用例：**ROOT为祖先，给ROOT改名

**预期结果：**程序正常运行并成功改名

**实验结果：**

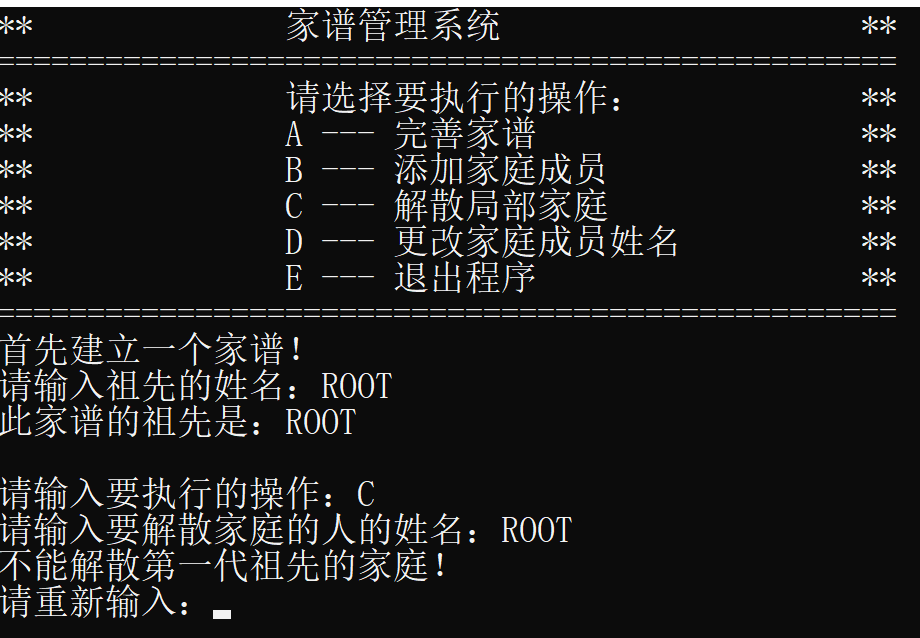


4.2.2 解散局部家庭时解散第一代祖先家庭

**测试用例：**ROOT为根节点，解散ROOT

**预期结果：**程序给出提示信息，程序正常运行不崩溃。

**实验结果：**



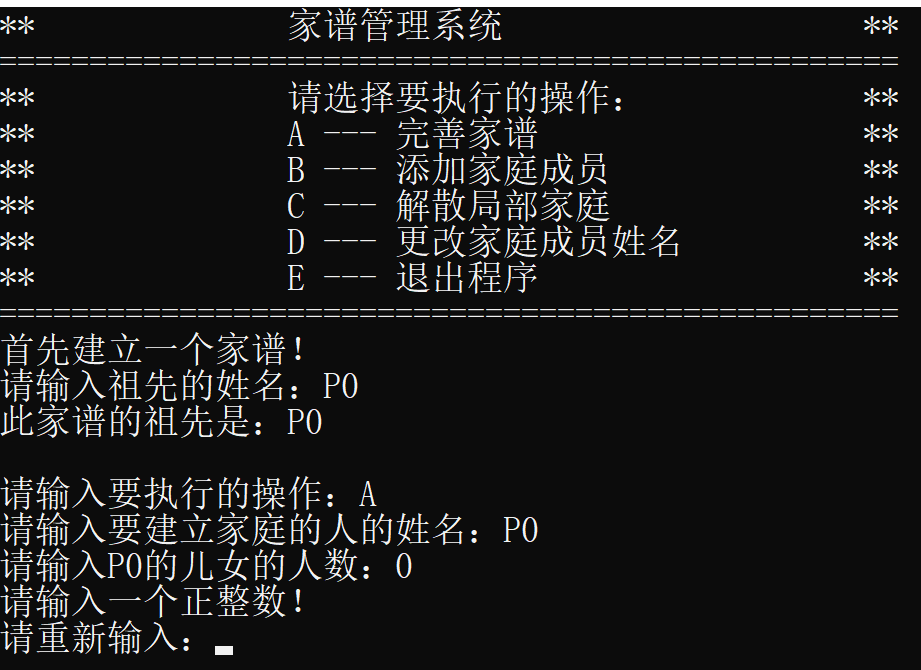
4.3 出错测试

4.3.1 完善家谱时输入添加儿女个数为零

**测试用例：**祖先P0孩子个数为0

**预期结果：**程序给出提示信息，程序正常运行不崩溃。

**实验结果：**

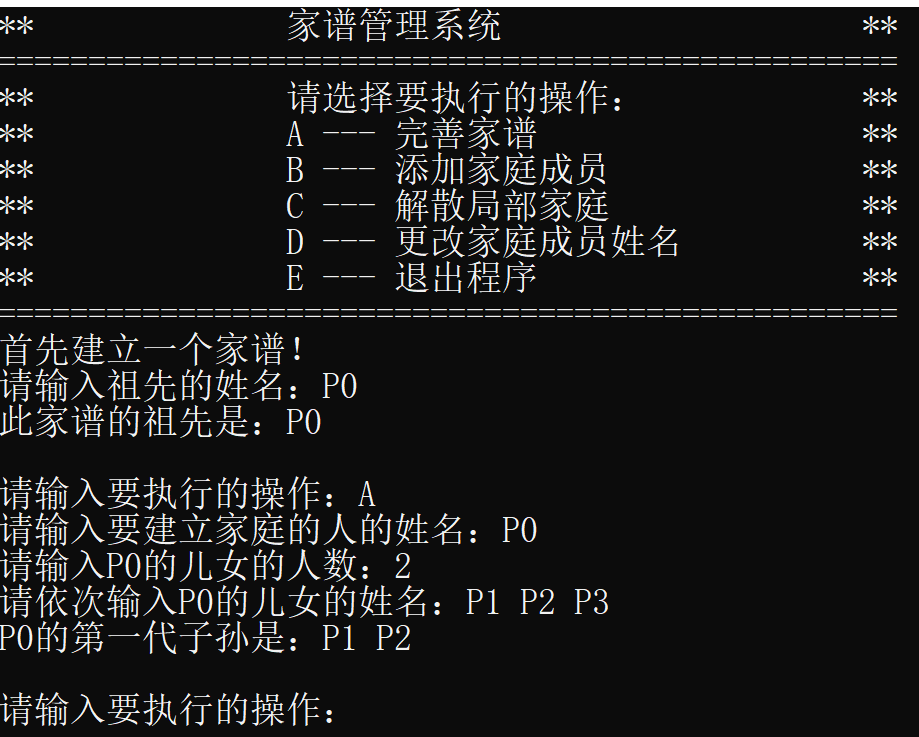


4.3.2 完善家谱时新添加儿女姓名过多

**测试用例：**给P0添加2个孩子，但是输入3个名字

**预期结果：**程序忽略多余的姓名，程序正常运行不崩溃。

**实验结果：**



4.3.3 输入操作数孩子数目等不合法

**测试用例：**输入错误的操作数，孩子数目等输入负数或者字符

**预期结果：**程序给出提示信息，程序正常运行不崩溃。

**实验结果：**

