项目说明文档

数据结构课程设计

——家谱管理系统

作 者 姓 名： 董震宇

学 号： 1852143

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**

[一、项目分析 4](#_Toc27396183)

[1. 项目内容 4](#_Toc27396184)

[2. 项目要求 4](#_Toc27396185)

[二、使用手册 4](#_Toc27396186)

[1.运行程序 4](#_Toc27396187)

[2.完善家谱 4](#_Toc27396188)

[3.添加家庭成员 5](#_Toc27396189)

[4.解散局部家庭 5](#_Toc27396190)

[5.更改家庭成员姓名 5](#_Toc27396191)

[6.退出程序 5](#_Toc27396192)

[三、设计 6](#_Toc27396193)

[1.数据结构设计 6](#_Toc27396194)

[2.类结构设计 6](#_Toc27396195)

[3.成员与操作设计 6](#_Toc27396196)

[（1）家庭成员设计 6](#_Toc27396197)

[（2）家谱类操作设计 6](#_Toc27396198)

[四、具体实现 7](#_Toc27396199)

[1.总体思路 7](#_Toc27396200)

[2.操作A：完善家谱 7](#_Toc27396201)

[（1）简要思路 7](#_Toc27396202)

[（2）算法流程图 7](#_Toc27396203)

[（3）核心代码 8](#_Toc27396204)

[3.操作B：添加家庭成员 9](#_Toc27396205)

[（1）简要思路 9](#_Toc27396206)

[（2）算法流程图 9](#_Toc27396207)

[（3）核心代码 10](#_Toc27396208)

[4.操作C：解散局部家庭 11](#_Toc27396209)

[（1）简要思路 11](#_Toc27396210)

[（2）算法流程图 11](#_Toc27396211)

[（3）核心代码 12](#_Toc27396212)

[5.操作D：更改家庭成员姓名 12](#_Toc27396213)

[（1）简要思路 12](#_Toc27396214)

[（2）代码 12](#_Toc27396215)

[6.查找家庭成员函数 12](#_Toc27396216)

[（1）简要思路 12](#_Toc27396217)

[（2）代码 13](#_Toc27396218)

[五、出错测试 13](#_Toc27396219)

[1.查无此人 13](#_Toc27396220)

[2.已经建立过家庭 13](#_Toc27396221)

[3.尚未成立家庭 14](#_Toc27396222)

[4.Child\_num=0 14](#_Toc27396223)

[5.注意事项 14](#_Toc27396224)

[六、自我小结 14](#_Toc27396225)

一、项目分析

1. 项目内容

家谱是一种以表谱形式，记载一个以血缘关系为主体的家族世袭繁衍和重要任务事迹的特殊图书体裁。家谱是中国特有的文化遗产，是中华民族的三大文献（国史，地志，族谱）之一，属于珍贵的人文资料，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。本项目兑对家谱管理进行简单的模拟，以实现查看祖先和子孙个人信息，插入家族成员，删除家族成员的功能。

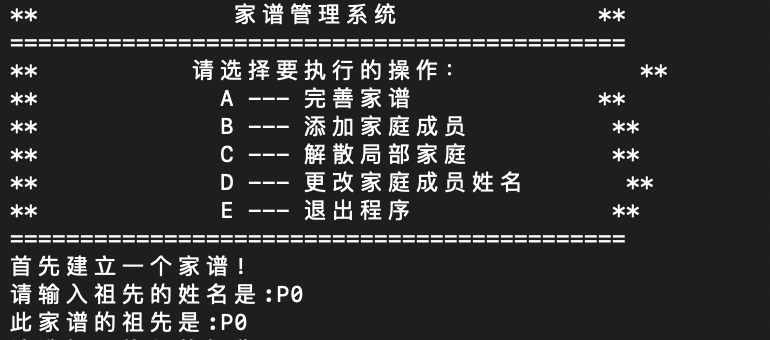
2. 项目要求

本项目的实质是完成兑家谱成员信息的建立，查找，插入，修改，删除等功能，可以首先定义家族成员数据结构，然后将每个功能作为一个成员函数来完成对数据的操作，最后完成主函数以验证各个函数功能并得到运行结果。

二、使用手册

1.运行程序

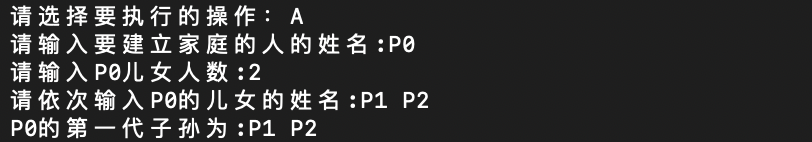
第一步，输入祖先的姓名。

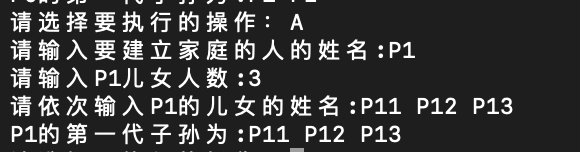


2.完善家谱

首先输入要建立家庭的人的姓名，然后输入此人儿女的个数，再分别输入儿女的姓名。

之后会输出该人的第一代子女。

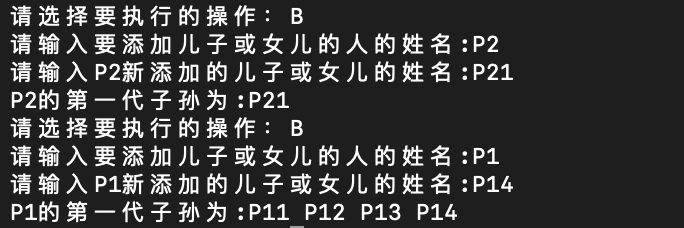




3.添加家庭成员

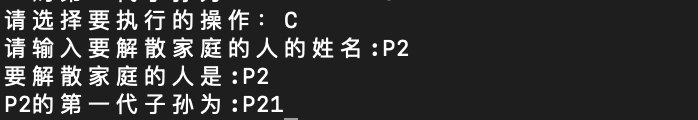
首先输入要添加儿女的人的姓名，然后再输入新添加儿女的姓名。

之后会输出该人第一代全部的子女。



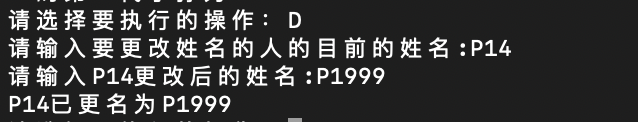
4.解散局部家庭

输入要解散家庭的人的姓名。

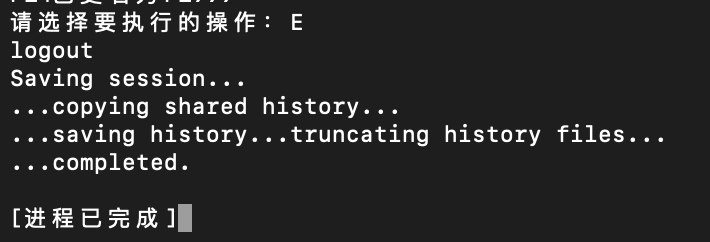


5.更改家庭成员姓名

首先输入要改名的成员的姓名，再输入更改后的姓名即可完成改名。



6.退出程序



三、设计

1.数据结构设计

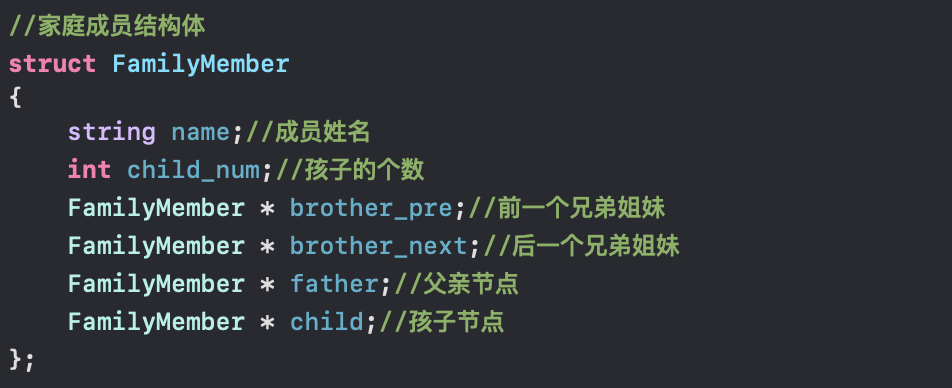
考虑到需要对家谱进行大量增加、删除、修改、查找等操作，而链表的操作特点正好符合要求，所以采用链表数据结构作为基础。家谱可以看成是一颗多叉树，所以本题最终用树的数据结构来实现功能。

2.类结构设计

设计两个类（结构体），每个家庭成员的详细信息存储在结构体中，而对于家谱的增删查改操作则通过家谱类来实现。

3.成员与操作设计

（1）家庭成员设计



（2）家谱类操作设计



四、具体实现

1.总体思路

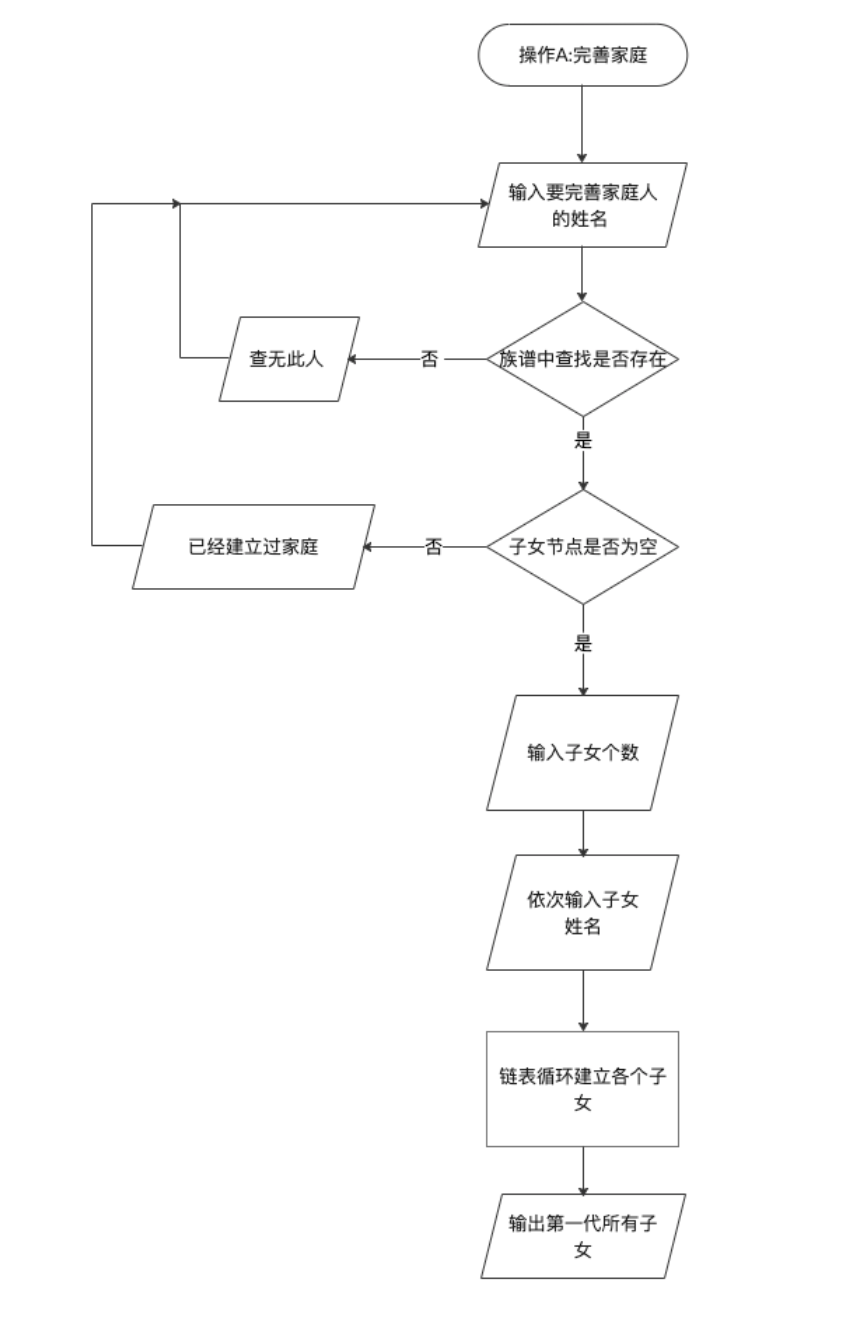
采用子女—兄弟链表表示法的二叉链表实现。Family成员结构体，对应有父指针和子指针以及前后兄弟指针，将整个家庭构造成树的结构。在添加新家庭成员时，只需在原链表尾处添加新节点并标记其父指针为原表尾。同样，解散家庭成员时，根据该成员所处位置，将其前后兄弟指针相连，并减少父节点的子孙数后删除该节点。完善家庭成员时，先输入子女个数，之后为每个子女建立新节点并与父节点相连，每个子女节点之间用兄弟指针相连，便于查找家庭成员。

2.操作A：完善家谱

（1）简要思路

得到家庭成员姓名后，在家谱中查找该成员。如果没有找到，则报错：“查无此人，请重新输入！”。如果找到，则对其孩子节点进行判断，如果孩子节点不为空，则说明已经建立过家庭；孩子节点为空则需要输入孩子的个数和各个孩子的名字，然后再循环添加子女到树中。

（2）算法流程图



（3）核心代码

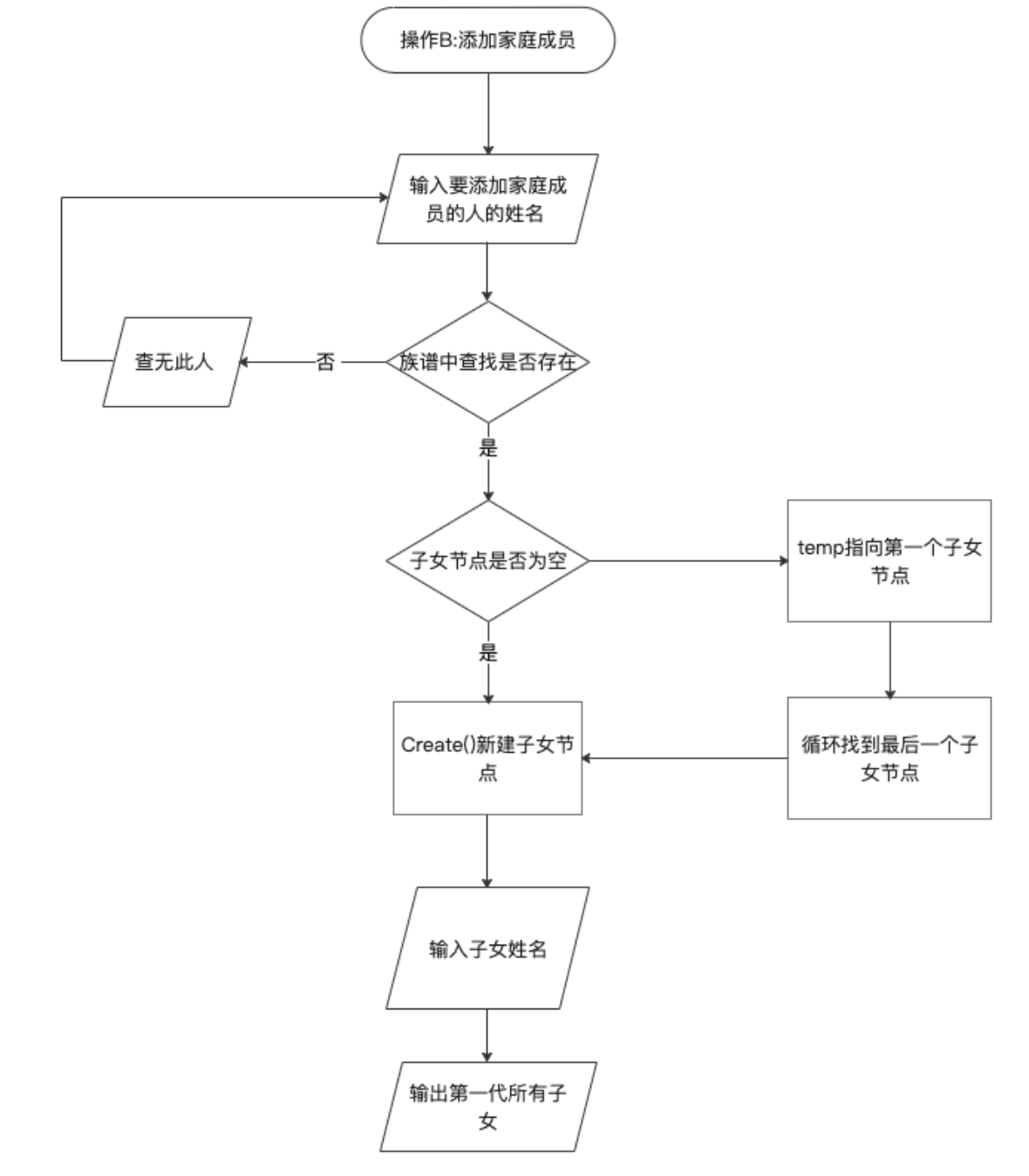


3.操作B：添加家庭成员

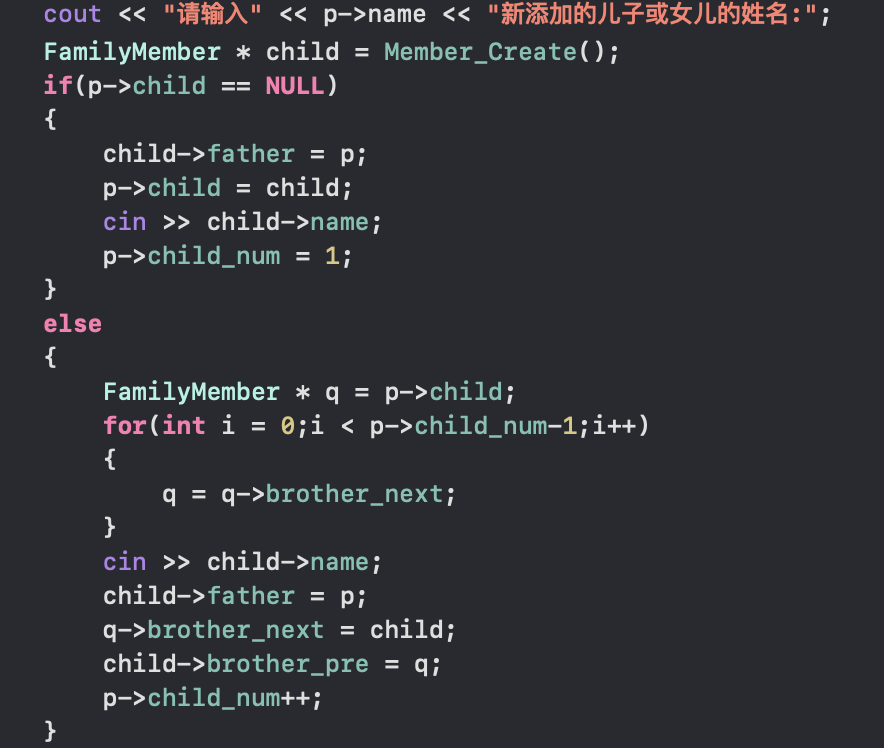
（1）简要思路

首先输入要添加儿子或女儿的人的姓名，在家谱中查找此人，如果未找到则报错，找到的话需要进行进一步判断，如果此人孩子节点为空，则通过Member\_create()函数新建孩子节点并添加到族谱中；如果此人子节点不为空，则对其孩子节点操作，循环找到子节点最后一个，然后加入新节点。

（2）算法流程图



（3）核心代码

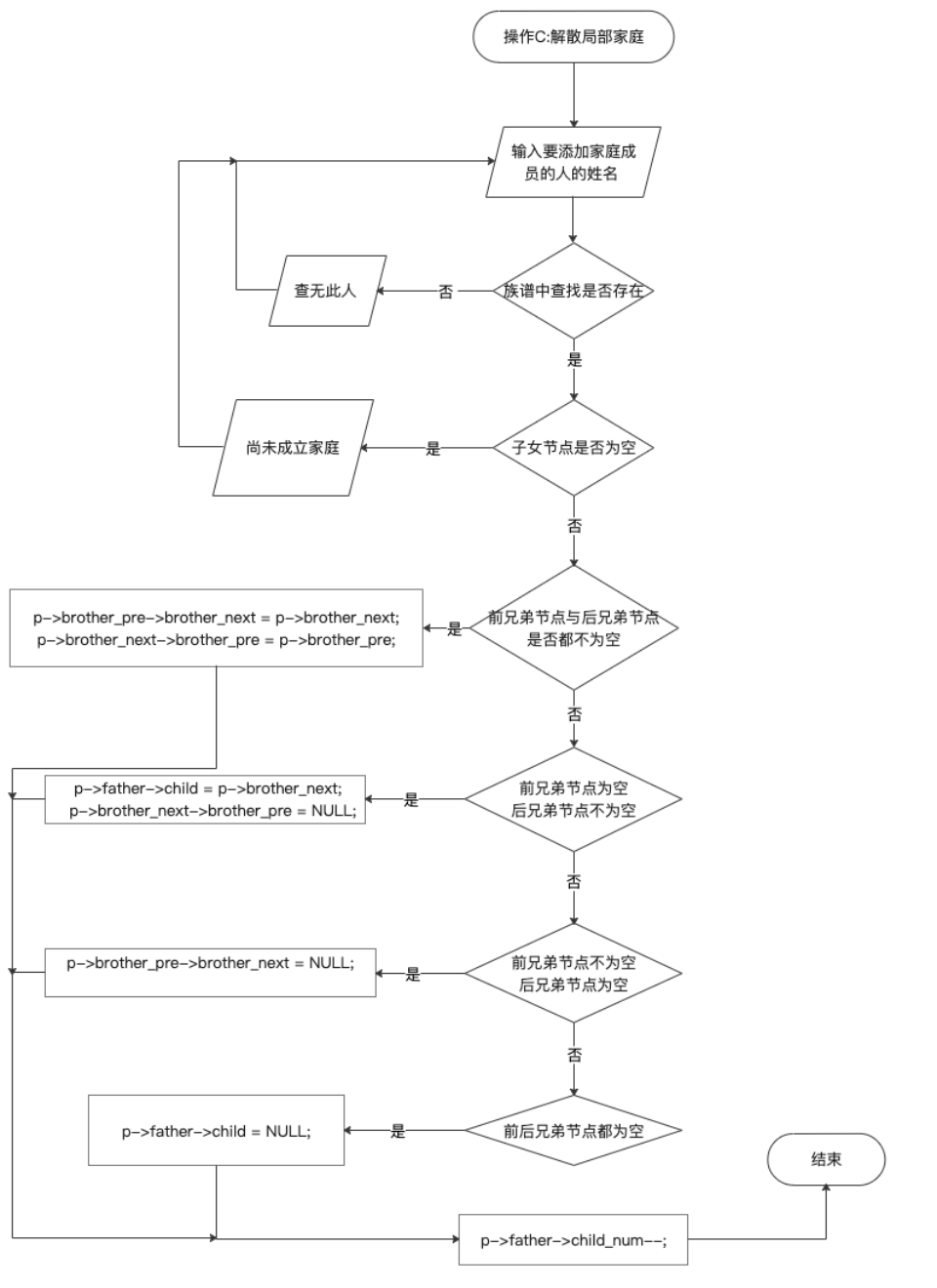


4.操作C：解散局部家庭

（1）简要思路

同前面操作，在族谱中找到要解散家庭的人。如果他的孩子节点为空，则输出“尚未建立家庭！”，如果孩子节点不为空，需要分情况继续讨论，判断此人是否有兄弟指针，让前后兄弟指针相连接或置为空，并让父节点的孩子数减一，再删除该节点。

（2）算法流程图



（3）核心代码



5.操作D：更改家庭成员姓名

（1）简要思路

找到家庭成员并直接更改其姓名即可。

（2）代码



6.查找家庭成员函数

（1）简要思路

先判断传入的member是否为要查找的人，再对其兄弟节点和孩子节点进行递归查找。

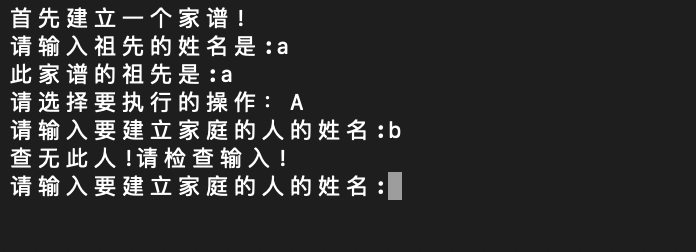
（2）代码

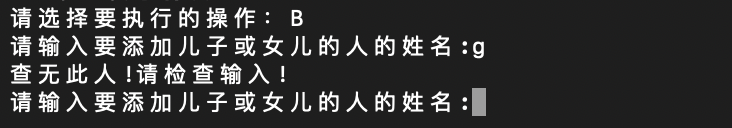


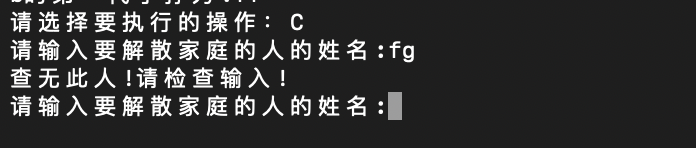
五、出错测试

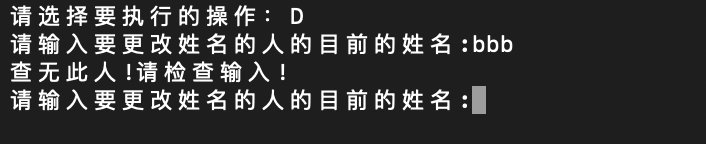
1.查无此人

如果在进行A、B、C、D操作时输入了家谱中不存在的成员姓名，则会报错：“查无此人！”



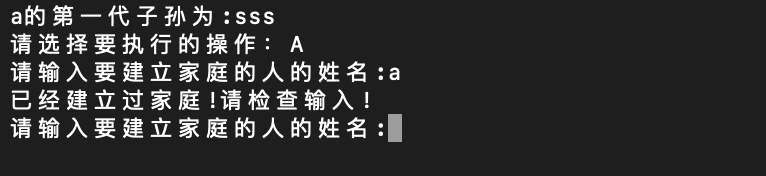






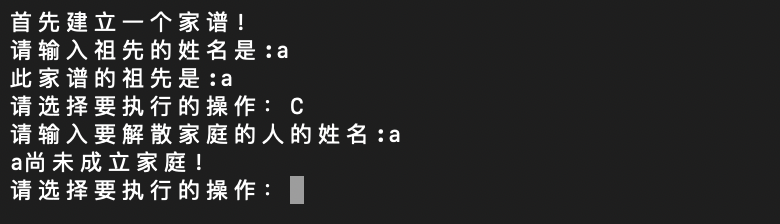
2.已经建立过家庭

完善家庭操作只适用于没有建立家庭的成员，建立家庭后只可以通过增加家庭成员或者解散后重建来实现更改。



3.尚未成立家庭

在进行解散操作是，如果A尚未建立家庭，则不可以进行解散操作，会有提示信息：“A尚未建立家庭！”



4.Child\_num=0

完善家庭时，如果输入的孩子个数为0，则会有提示信息：“没有子女！”



5.注意事项

（1）在解散局部家庭时不可以解散“祖先”的局部家庭。

（2）输出子孙信息只可以输出第一代。

六、自我小结

家谱管理系统的完成使我感到了一丝成就感，因为在初次拿到设计题目时，我是一筹莫展的。又一次亲手设计了数据结构与类，实现起来十分方便，美中不足的是并不能像生活中的家谱那样全部输出展示出来，这用到了很多没有学到的知识。

在设计过程中，经过同学的一些提醒，更正了不少细节的问题，比如解散局部家庭时对于兄弟节点的多种情况进行讨论。使用链表时，时刻要注意它的指向各个节点的指针是否为空，“牵一发而动全身”，删除此节点需要在链表中把与该节点所有相关的信息都进行更改，以后处理时需要更加仔细。

本次设计中族谱建立使用了“兄弟-子女”表示法，源于数据结构的教材，在我初学树这种数据结构时，我并没有想到这种方法，现在也很有收获。

另一个存在的问题就是成员的查找——用递归实现。即先找兄弟节点，若兄弟节点中没有找到再对子女节点进行递归。这里是要区分递归与循环的区别的。递归是对下一级的同类型的数据进行递归，而循环是对同级的数据在一定范围内进行循环，一开始我不理解为什么子女节点要递归而兄弟节点要循环。在写递归或者循环代码的时候，伪代码是十分重要的，根据伪代码来写，很快就会有结果，而空想就很抽象。这一点在后面排序算法的比较中我也深有体会。