**实验报告五**

1. **实验内容**

Family Tree

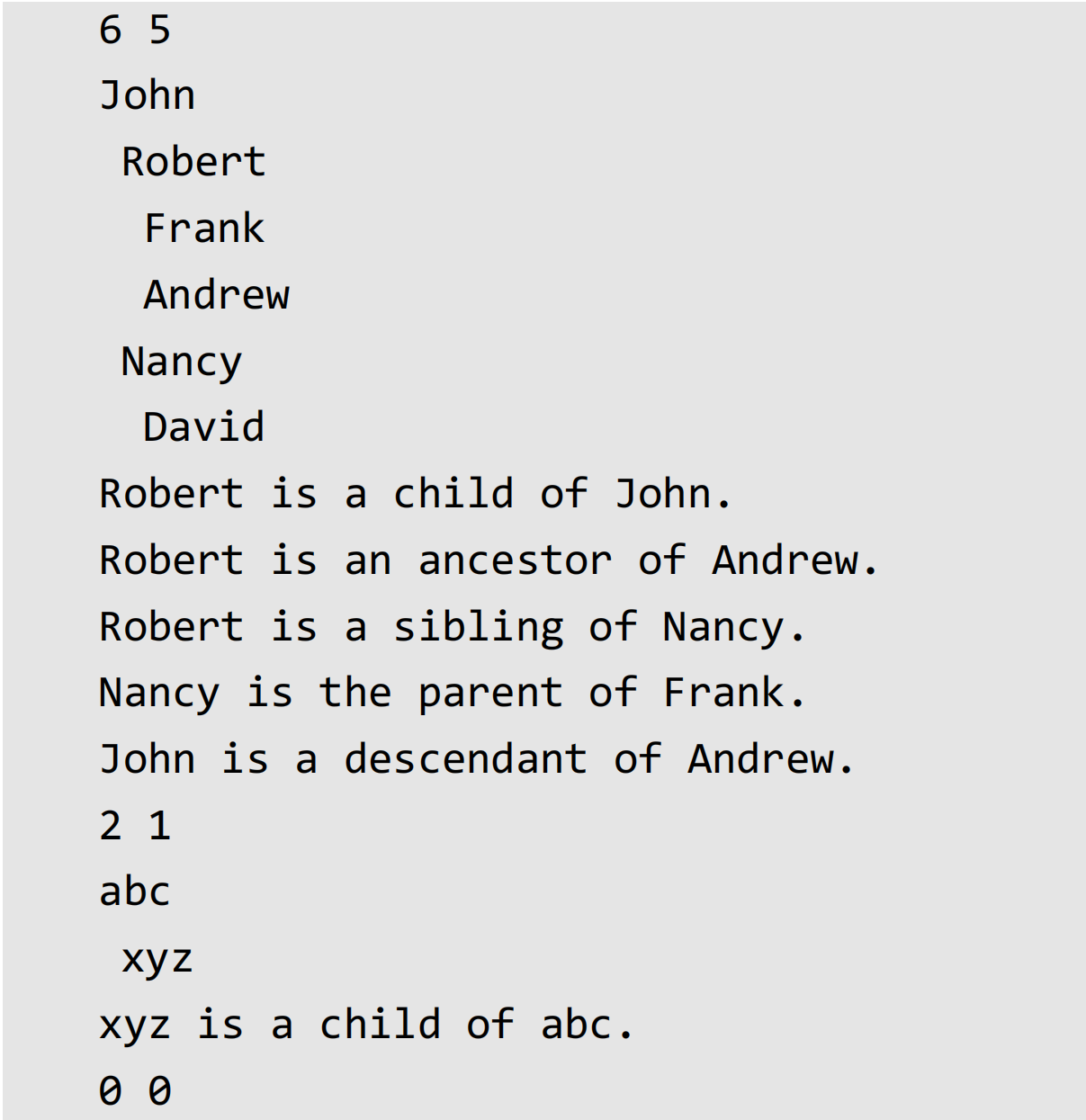
1. **数据结构设计与思路分析**
2. **本题所采用的数据结构：**

树+搜索

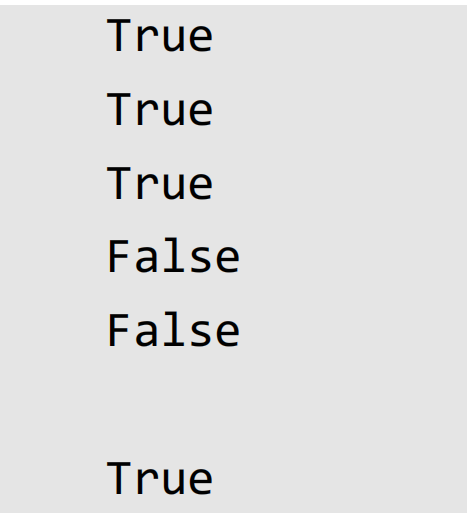
**（二）问题描述及要求：**

首先给定家族人数和命题个数。然后给定一个家族系谱树（不包括妻子），父代与子代之间相差一个空格。然后给出若干个命题，格式为 A is the relationship of B，其中，relationship 可以是child, parent, sibling, descendant, ancestor. 对于每一个命题，要求判断这个命题是 真还是假。若为真，则输出 True，否则输出 False。会有多组输入样例，以 0 0 结尾。

输入样例：



输出样例：



**（三）算法设计思路：**

考虑家族系谱的特点，可以使用数据结构——树来建立每一个人之间的关系。根据输入

的命题的特点，需要判断的两个人的关系只有孩子、父亲、兄弟、后代和祖先，可以通过双亲表示法来建立这棵树。

分别对以下三大种情况进行讨论：

1. 如果 A 和 B 两个人是父子关系，那么只需要判断 A 是否为 B 的父亲或者 B 是否为 A 的父亲即可；
2. 如果两个人是兄弟关系，那么只需要判断 A 和 B 的父亲是否是同一个人即可；
3. 如果两个人是后代和祖先的关系，那么只需要以 A（或 B）为起点，往上搜索，查看是否能搜索到 B（或 A）即可。满足上述条件的命题即为真，否则为假。

输入实现方法：

由于本题输入较为复杂。为将输入的数据转换成一棵树，这里选择使用类似于栈的思想，

通过定义一个移动的指针，指向上一个人。依据名字前面的空格数，决定将这个节点挂到哪 一个父节点下。具体思路如下：

1. 读入一个名字，计算这个名字前面有多少个空格（代表第几代）

2. 创建节点

3. 将计算得到的空格数与上一个名字的空格数进行比较

a) 如果当前的空格数比上一个名字的空格数大，这说明这个人是上一个人的孩子，那么将这个节点挂到上一个人的节点下面

b) 否则，计算当前空格数和上一个名字的空格数的差，表示这个人和上一个人相差了几代。然后将移动的指针向上移动对应的次数，移动结束后这个指针将指向这个人的父亲节点，然后将这个节点挂到父亲节点的下面。

4. 将指针移动到当前这个人的节点上

5. 回到步骤 1，进入下一次循环，直到把所有人读完

由于树是通过双亲表示法来建立的，在判断两个人的关系的时候，无法从根开始往下遍

历查找两个人分别所在的位置。因此在建树的时候，另开一个向量，用于存储每一个人的节

点地址。这样需要查找某个人的位置的时候，直接遍历这个向量即可。

**（四）算法分析：**

Node\* search(string name)

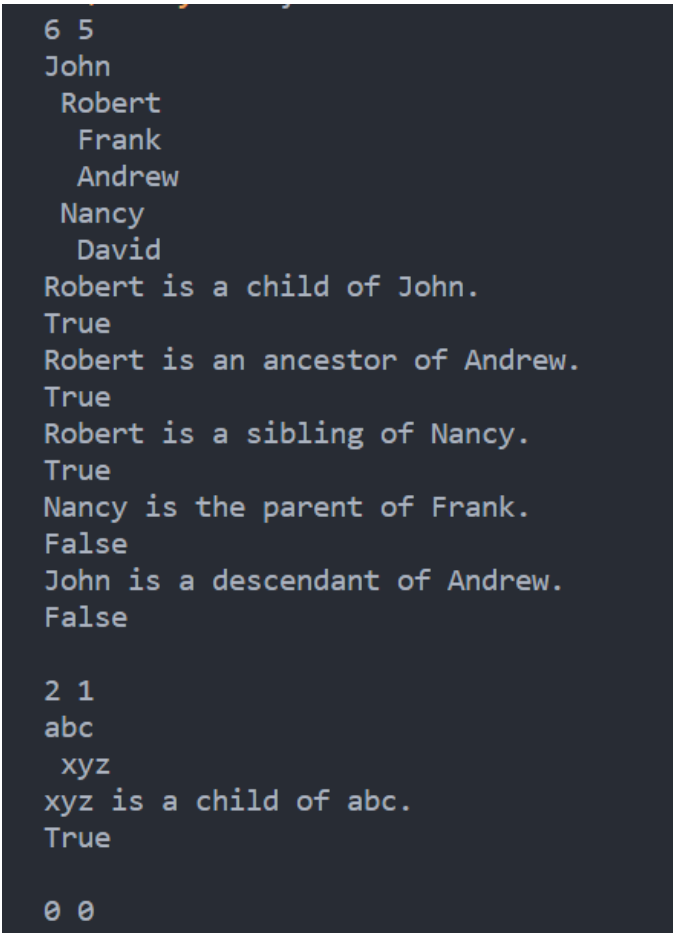
搜索方法是线性搜索具有给定名称的节点。时间复杂度为O(n)，其中n为家族树的节点数。

bool readAndJudge()

该方法读取输入关系并执行比较。它没有嵌套循环或递归调用，因此每次调用的时间复杂度为0(1)。但是，如果在主循环中调用m次，则总时间复杂度变为O(m)。

1. **实验结果与分析**

采用题中的测试样例得到的结果：



1. **实验小结**

通过本次实验，我对树这种数据结构有了更加深刻的认识，了解了树的真实应用场景，

并通过选择数据结构和设计算法，实现了树的应用。提高了自己的程序设计能力，增加了自己的经验。