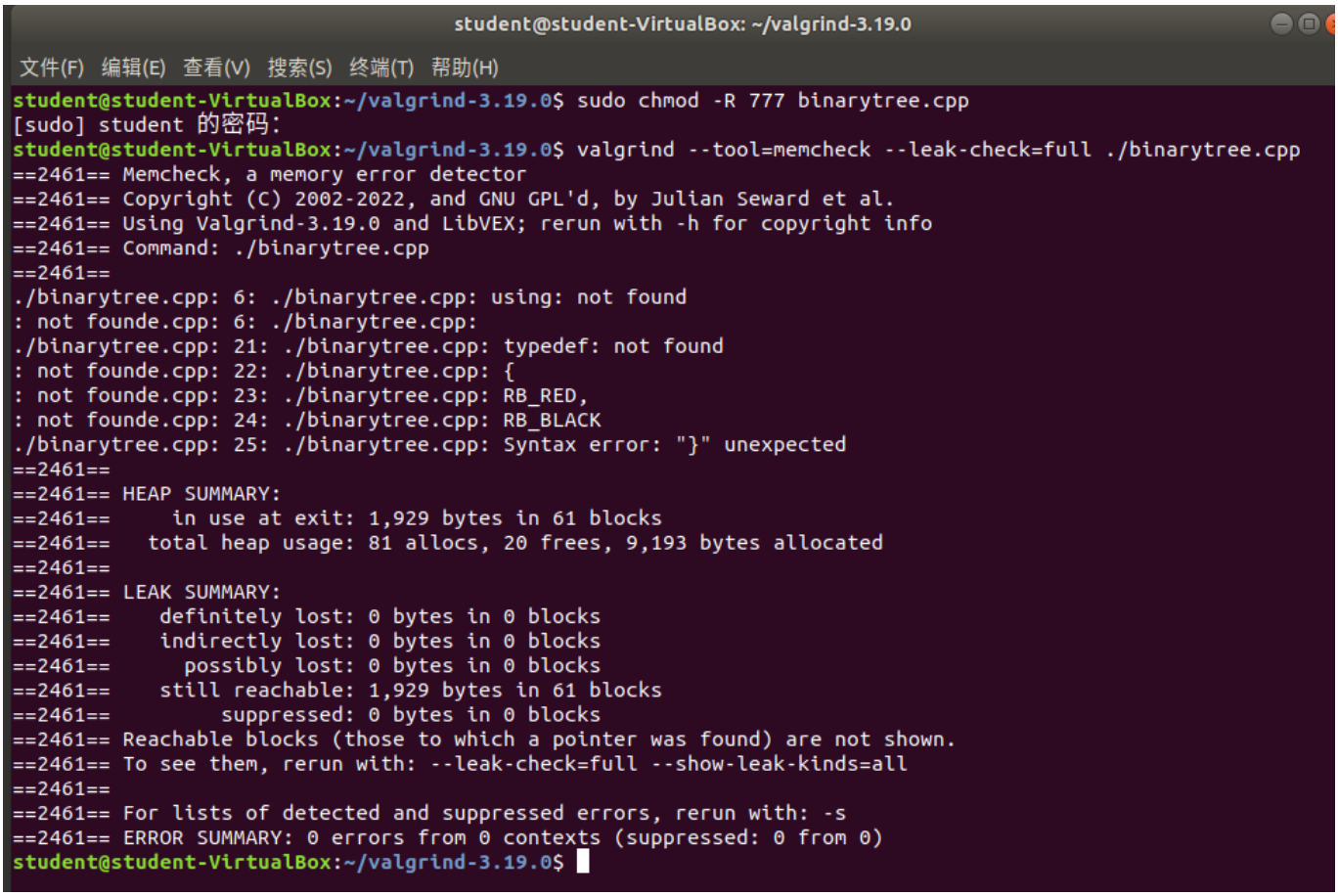
# Binary Tree 实验报告

赵宸 2021201645

本次试验基本完成实验要求，选做任务选择了实现查找兄弟节点和查找祖先节点。基本实现了对错误输入的人性化处理。

## 内存泄漏检查

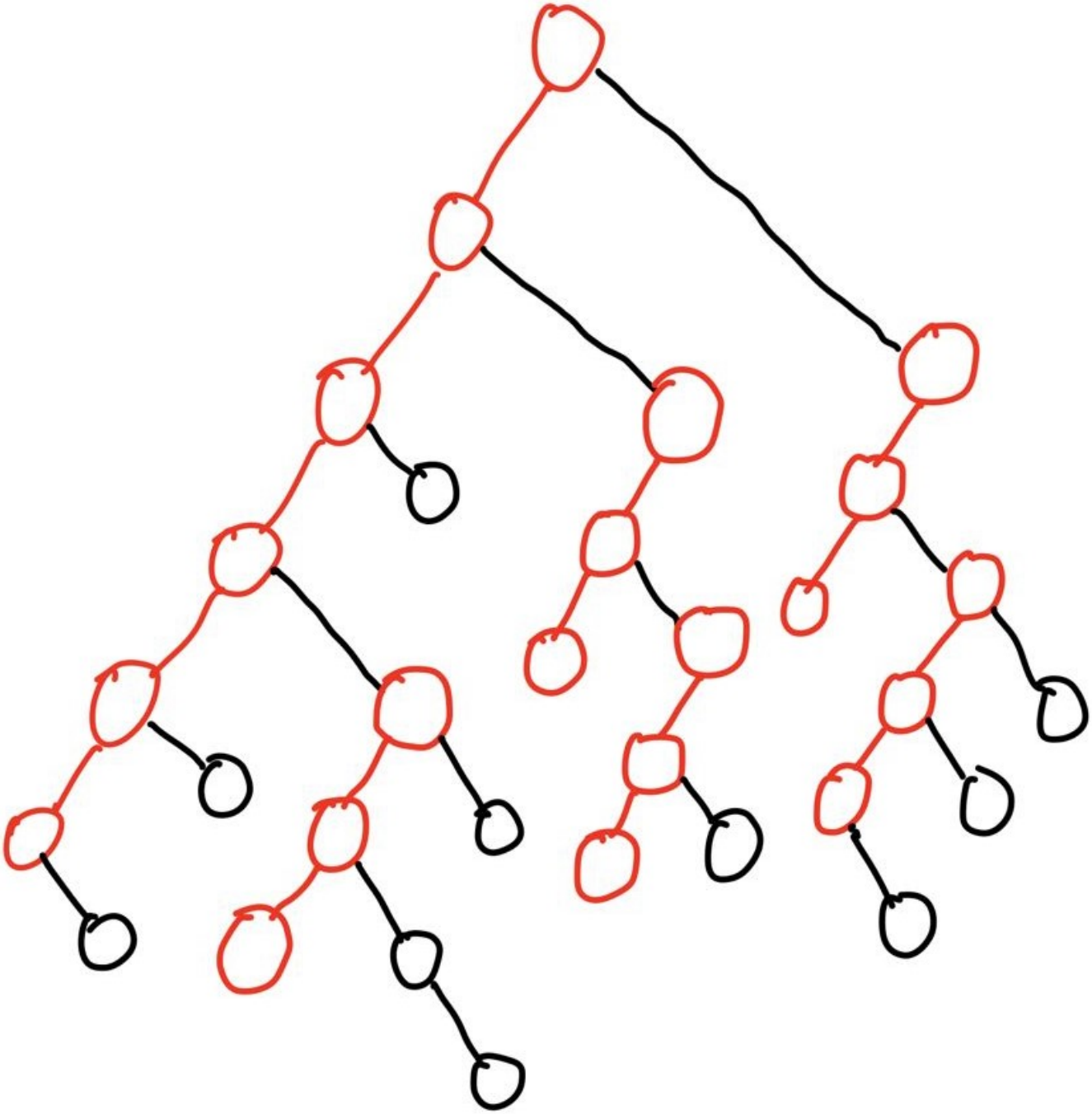
经过valgrind的内存泄漏检查，本程序没有出现内存泄漏现象,可放心食用。




## 后序遍历迭代版实现思路

由于要先遍历所有的子节点然后再搞父节点，因此我便采用了老师上课所讲的“藤缠树：模型，也就是持续的向左遍历并压栈知道nullptr；弹栈的时候先查询右节点再遍历自身；如果右节点有叶子节点，那就继续压栈。重复以上过程直到栈空为止。
为了实现这个目的，首先我要写一个向左节点延伸的“爬藤”函数，然后向右分析叶子节点，分析到非空的节点

再次爬藤直到nullptr为止。下图是以上算法的**简略**图示：



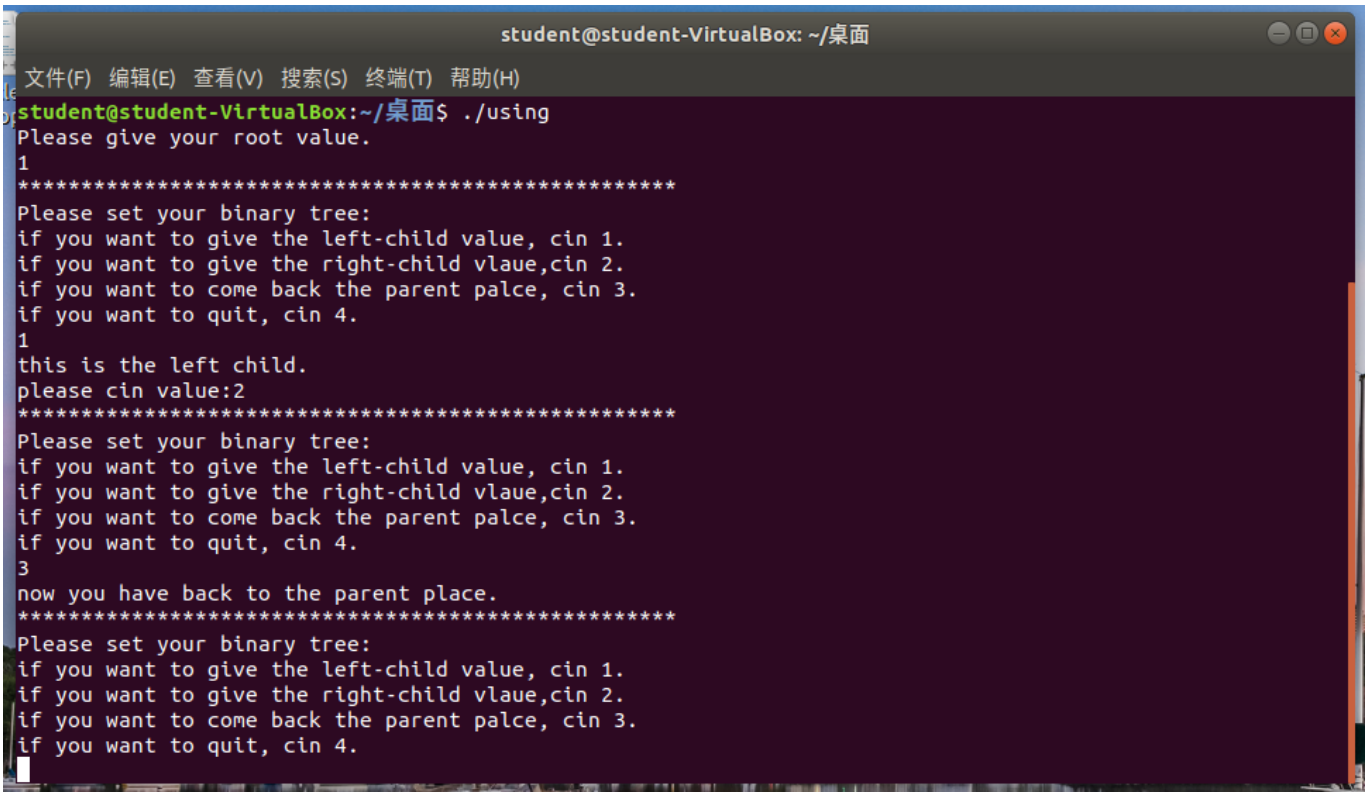
## parent和sibling的查询

由于没有特定的遍历函数要求，我在这里便使用`coutData`函数来代表遍历函数了。parent的寻找我采用递归的方法，一层一层的寻找压栈然后print，如果他根本没有parent那我就cout啥都没有来做特判；sibiling的寻找也是同理，直接找他有没有parent，如果有那再找他的左右孩子节点是否全满，最后判断哪个是我输入的x，另一个便是我兄弟。

## 使用方法及运行结果

### 建立二叉树

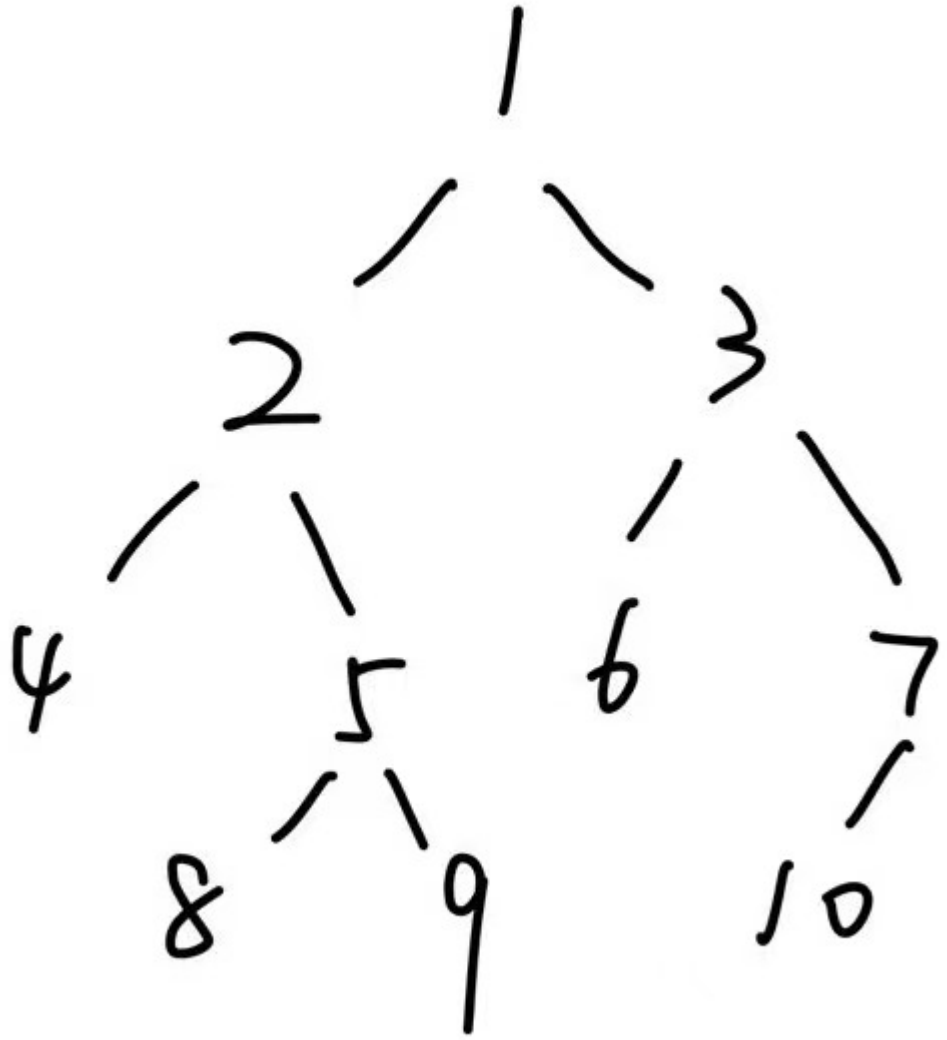
首先要输入根节点的数值，然后会允许你选择：延伸左节点，延伸右节点，返回父节点，如果左右的data输入到已有元素的节点中会报错并退出。运行截图如下：




下面我将以图中的树为例，介绍各个功能的运行，输入数据：

```
1
1 2
1 4
3
2 5
1 8
3
2 9
3
3
3
2 3
1 6
3
2 7
1 10
4
```

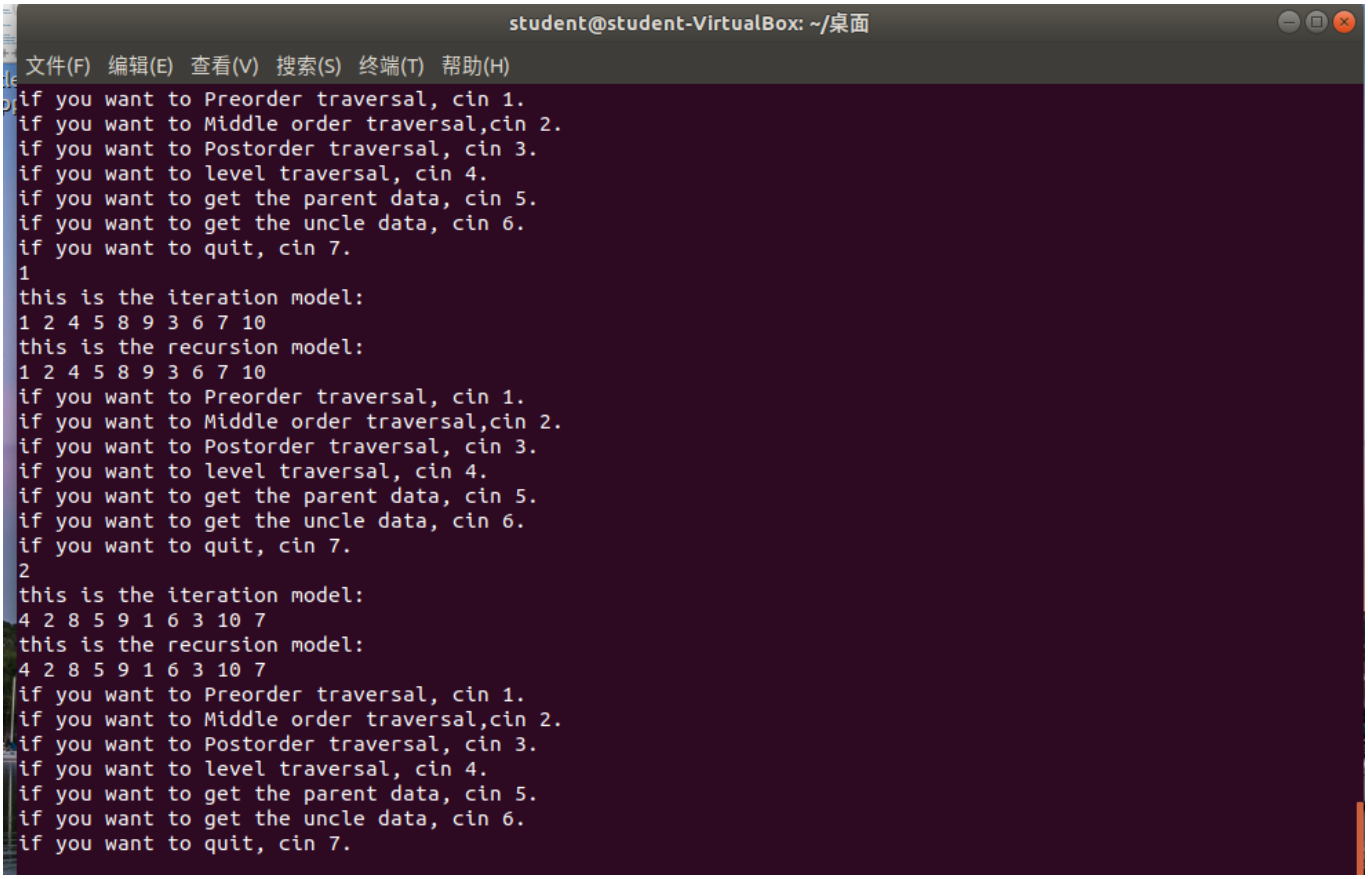
其中1表示延伸左节点，后边的数据为Data；2表示右节点，3表示返回父节点，4表示退出二叉树编辑。 以此我就得到了这样一颗二叉树：



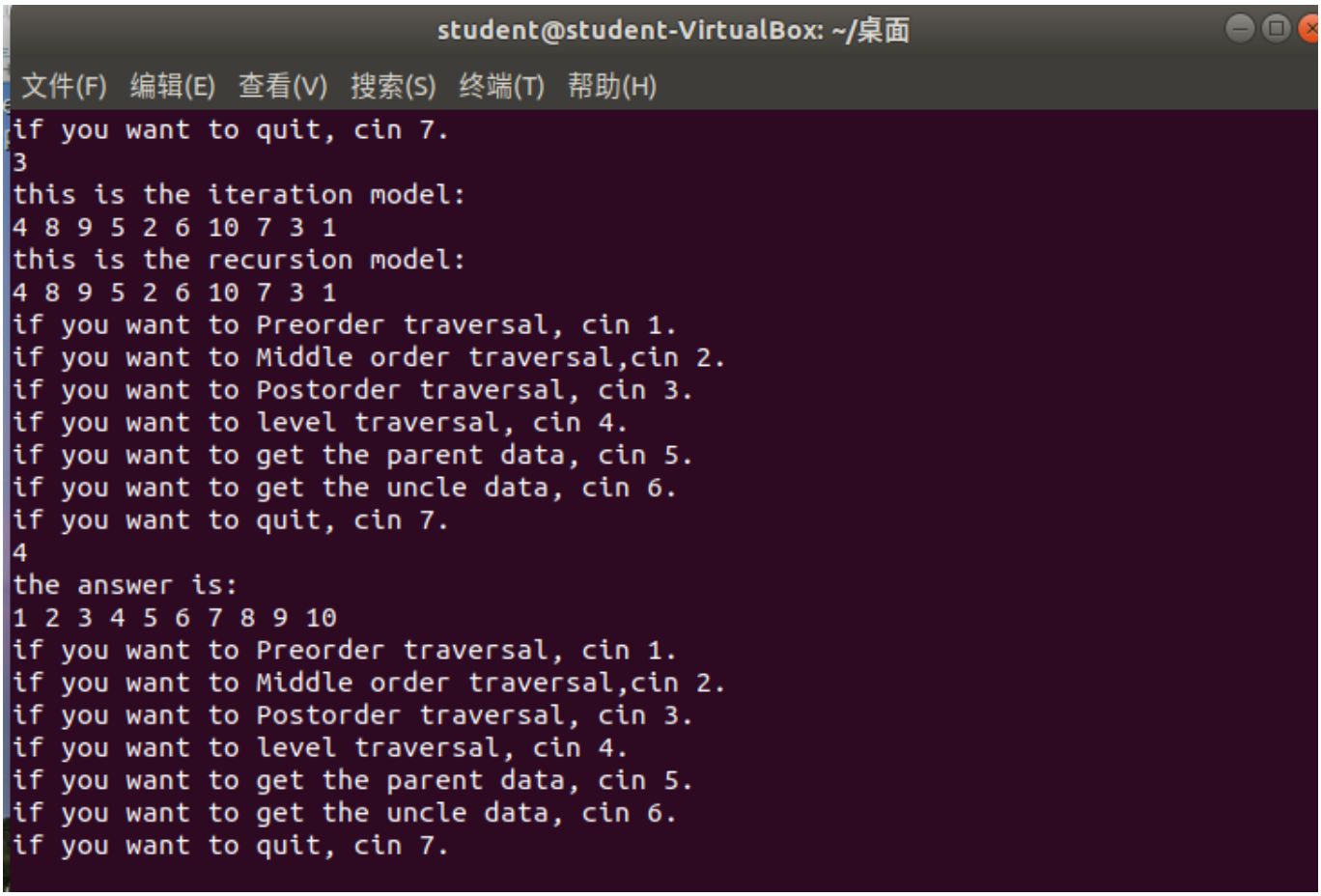

我将以此树为样例演示操作。

## 先序遍历和中序遍历

先输入迭代版，再输出递归版。

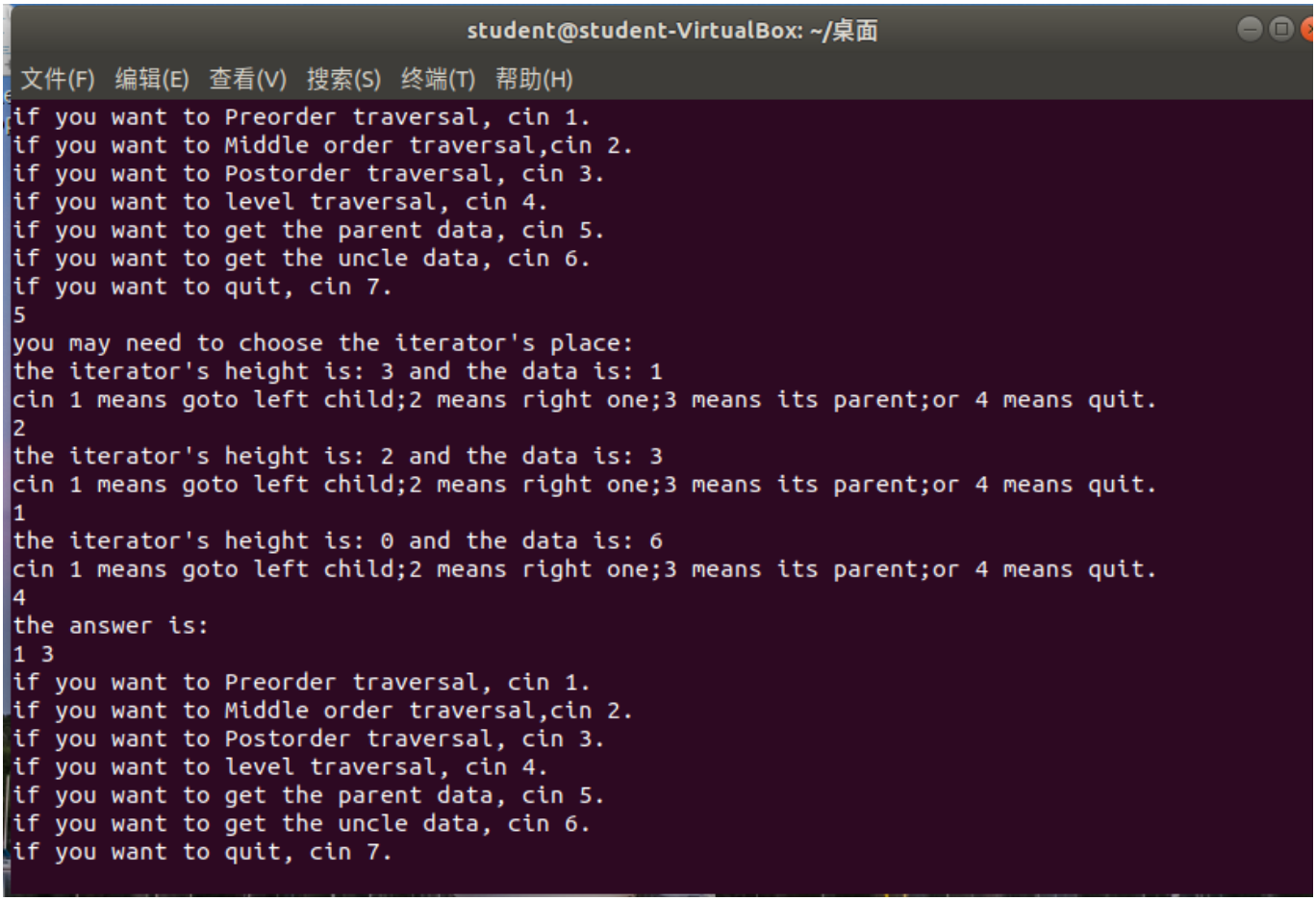



后序遍历和层次遍历



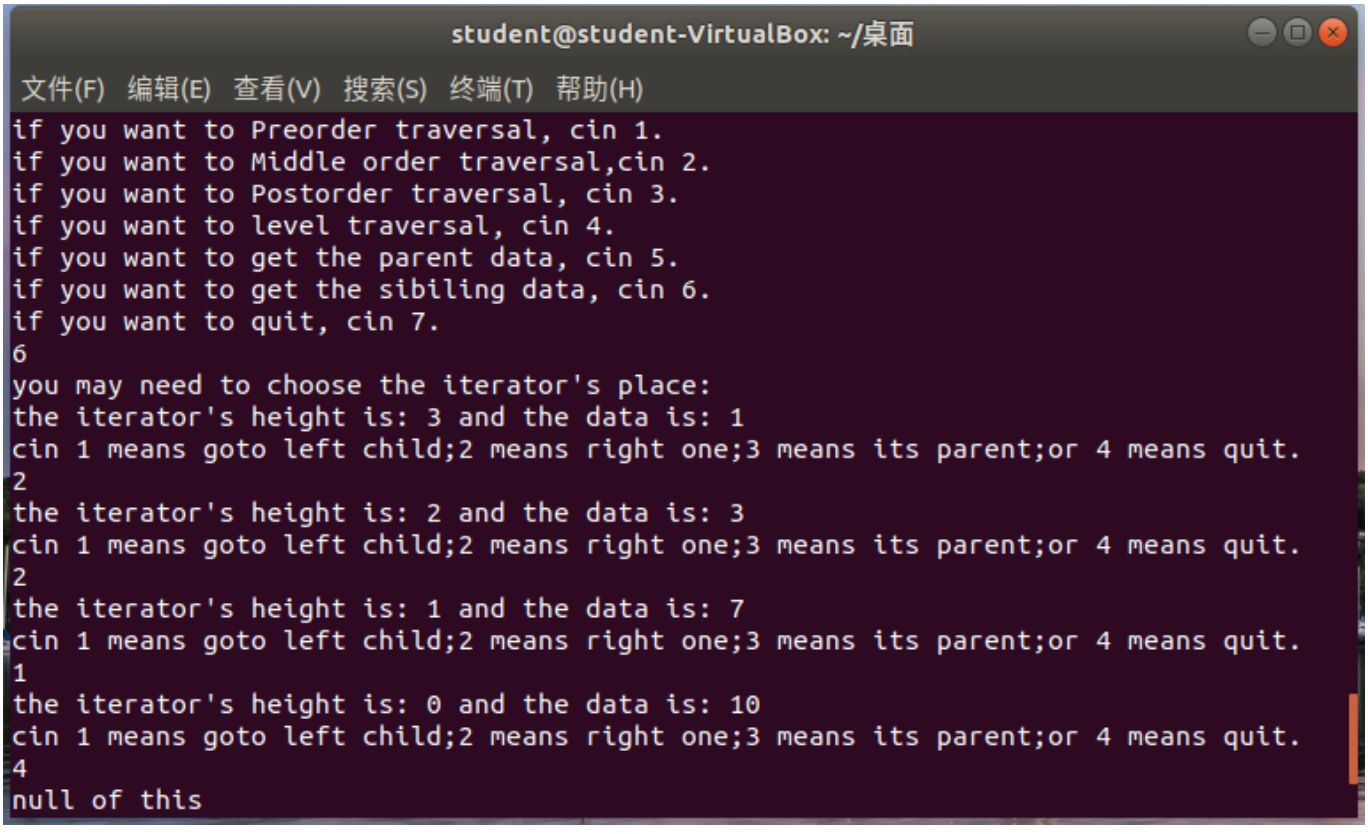

parent查找

以6为例，显然祖先为1和3.

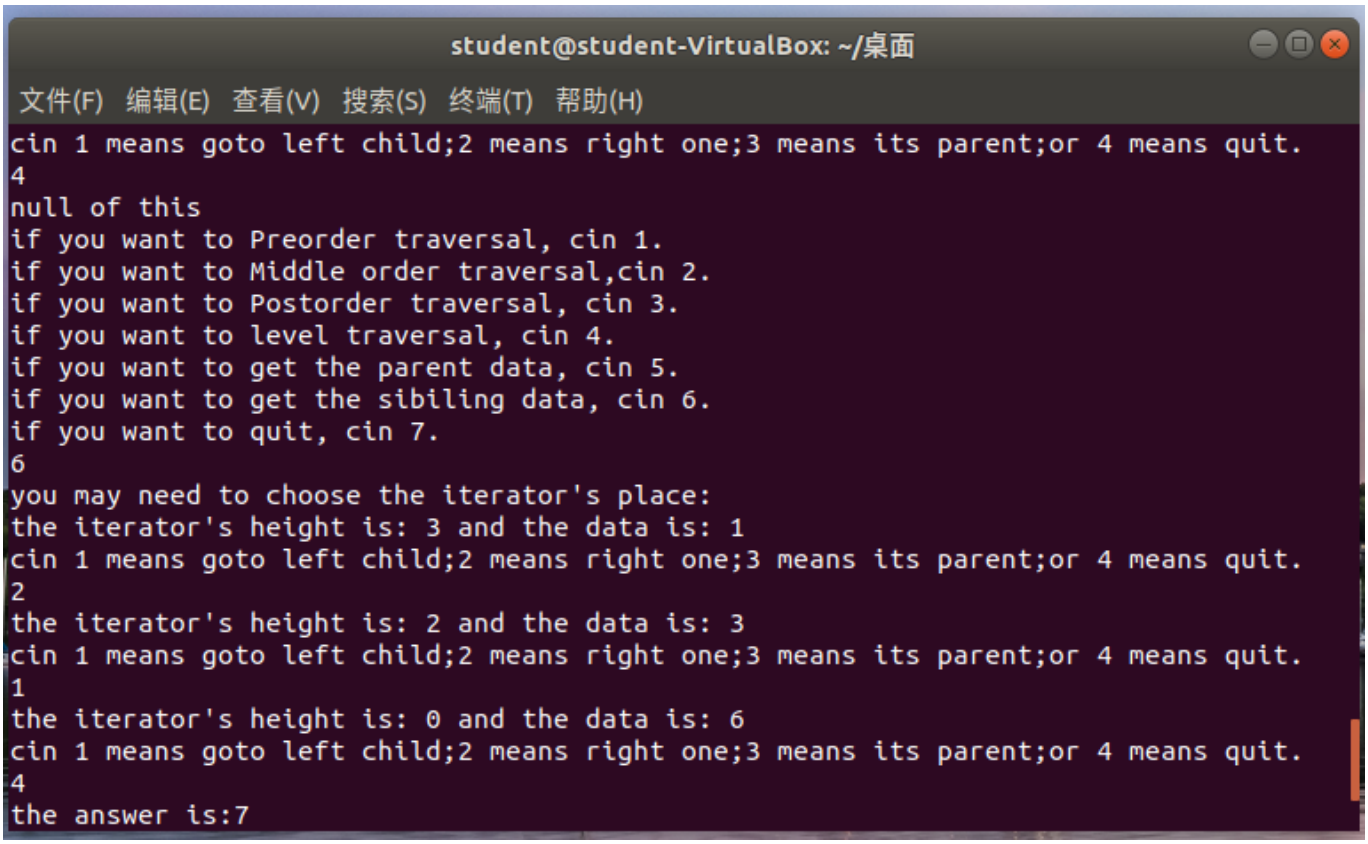



## uncle查找

如果没有兄弟节点，返回null of this：




正常返回值：




以上就是本次实验的全部内容，通过本次lab，我加深了对二叉树的理解，实验很有意义。