操作步骤：

1. Linux环境配置：安装Vmware Workstation和Ubuntu，Deepin；

2. 了解Linux：周四上课时发现环境配置还有问题，课后根据PPT自行模仿操作；

3. 将第二次实验代码稍作修改，在vim编辑器中手动输入，适应输入基本操作；

4. 编写makefile文件并运行，模仿Test中提供的makefile进行操作，

间接帮助我理解了在VS环境下“知其然而不知其所以然”的多文件编译原理。

遇到的问题：

1. Ubuntu安装后卡在加载界面进不去：移除虚拟机并重新创建；

2. vim编辑器在命令模式和插入模式的切换：百度：ESC和i；

3. vim编辑器光标移动指令、剪切和粘贴、翻页键的使用：百度从CSCN了解

$，w/[spacebar]，b/[backspace]，e，0，[return]；d，y，P，p，[crtl-d/u/f/b]用法；

4. 安装gdb时“无法打开锁文件，权限不够”：百度：

sudo passwd root修改root密码，su登录获得操作权限；

5. makefile编写问题（makefile的文件类型与如何创建）：百度：

得知makefile不需要后缀名，同样采用vim创建，

MakeFile或makefile为默认名称，此外还可通过make –f使用其他名字；

6. freopen\_s报错问题：寻找QQ群里其他同学的类似提问和相关解答：

改为使用freopen函数并删去第一个参数&stream。

代码实现：

1. 非常淳朴的链表创建、插入、删除、排序和递归操作：

边界条件的判断和不合法操作（如空链表，长度不够）时提示信息的输出；

链表排序：两层循环嵌套、依次遍历的冒泡排序法；

两个helper function分别用于Node和TreeNode：

void reverse\_print\_helper(Node\* list\_head)（递归无法输出换行符）

在函数reverse\_print\_list中被调用；

bool is\_end(Tree\_Node\* T)（用于判断是否为叶节点）

在函数insert与postorder中被调用；

2. 因为是以前写过、debug过的代码，断点调试过程中没有太多问题。

然而，发现第二周的删除节点操作没有释放/归还被删除节点的空间；

虽然运行时没有报错，但还是值得警惕，对代码中这样的问题进行了修改；

3. 发现第二周的代码中有不必要的#include和using namespace std。

多文件编译方面之前一直不能完全理解，通过makefile的编写有了更清楚的认识。