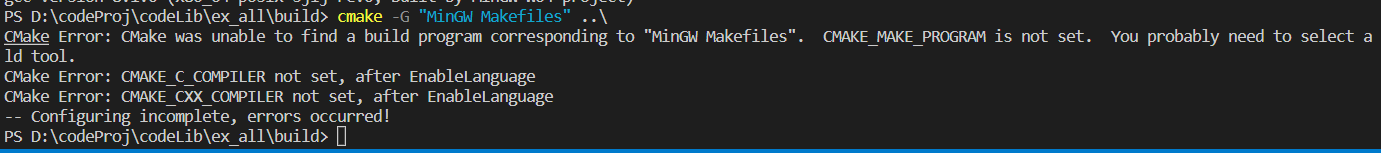
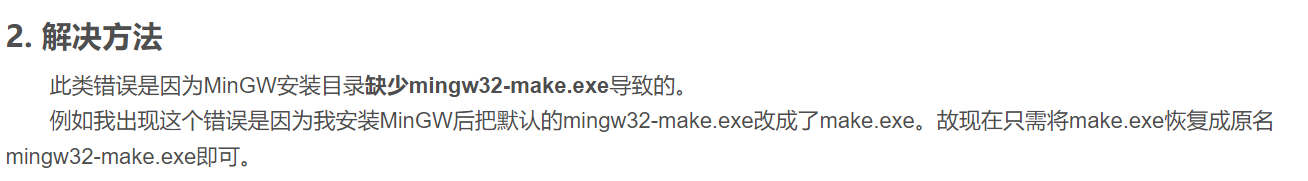
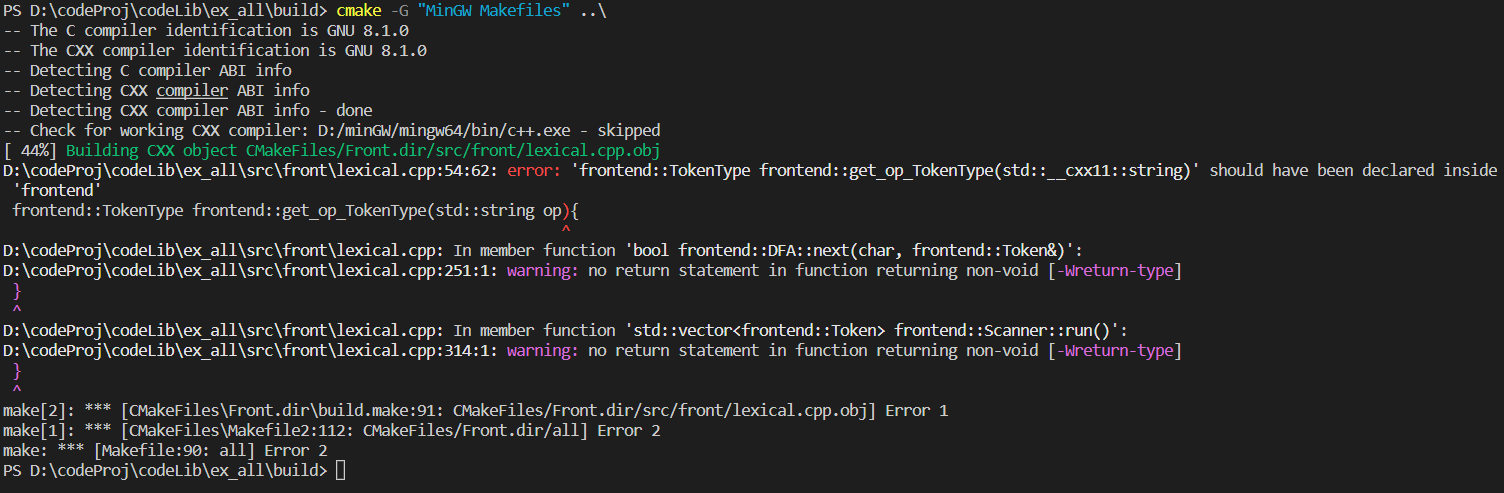
编译测试

* cmake -G "MinGW Makefiles" ..\命令执行失败





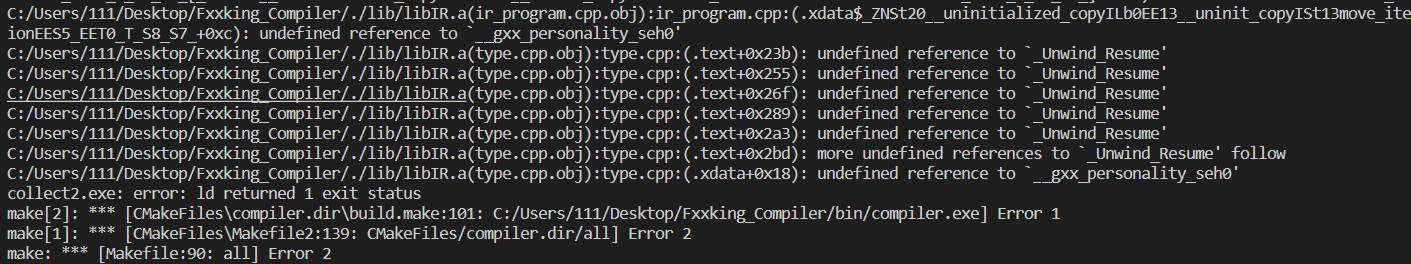
* 检查很久后发现是.h声明函数和.cpp实现函数不对应导致，很愚蠢

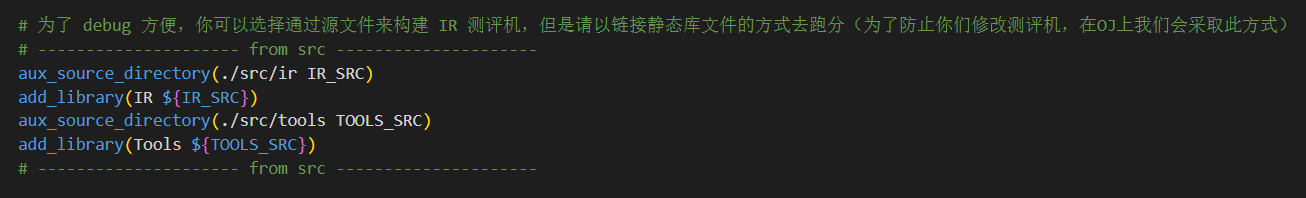


* Output tk与ref tk一致但是score为0，而且单独拿两个出来测试diff命令，返回值都是0，不知道为什么score.py执行时返回值为1.

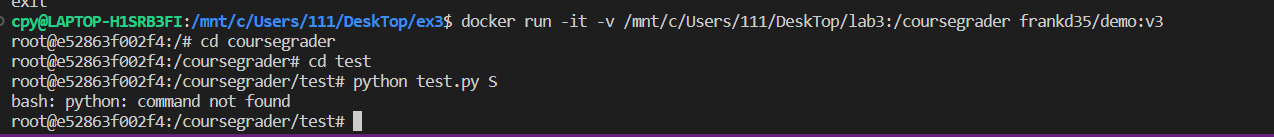
进行使用单独的指令分别比较返回值都是0，但是运行score.py就全为1，当时不知原因，但是第二天电脑重启后就正常了，很奇怪。

* 实验二make命令报错，根据群内答疑发现如果要自测需要取消注释，取消注释后make 命令执行成功

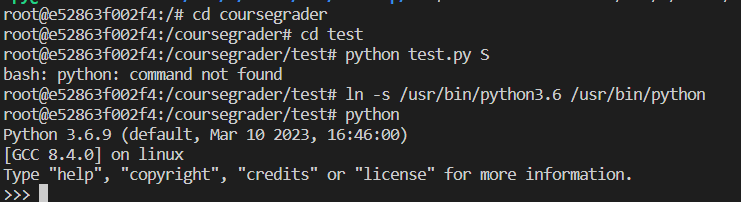




* Docker运行python脚本，显示没有python语法。



解决方法：

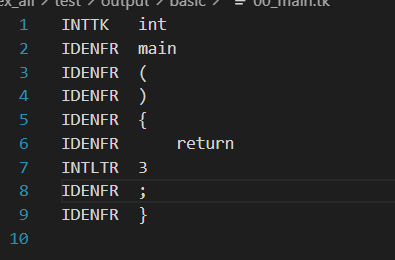


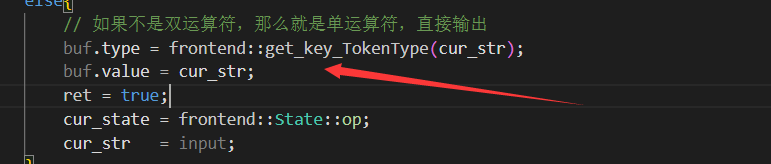
词法分析

* 执行后发现没有写入.tk文件中

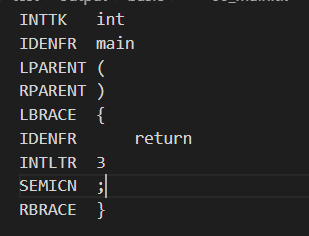
..\bin\compiler.exe .\testcase\basic\00\_main.sy -s0 -o .\output\basic\00\_main.tk

* 写入tk文件错误1，发现是函数写错！

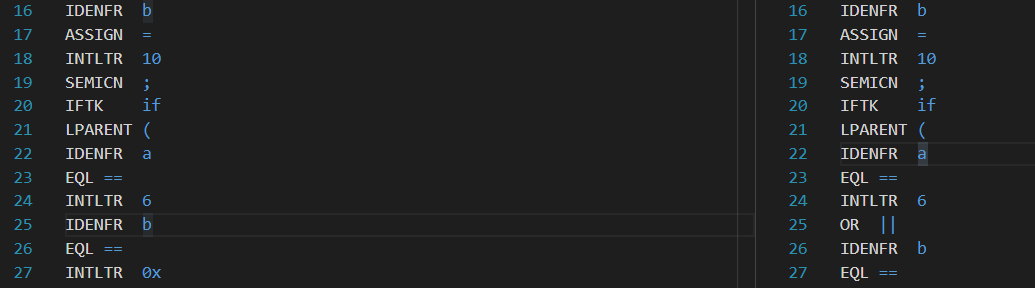




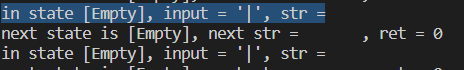
* 错误2 str是对的，但是tokentype错误，预测应该是在Op和Empty状态对空格处理不当导致，修改后正常，但是由于经常忘记修改后重新make，导致以为一直有bug从而浪费了很多时间



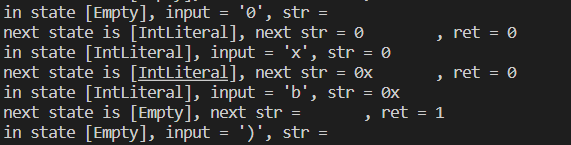
* 错误3 双运算符未判断成功，且16进制数匹配错误

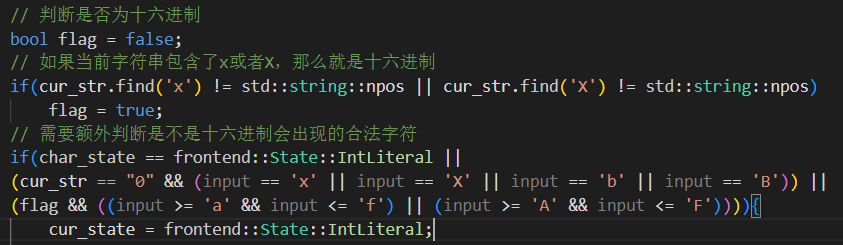


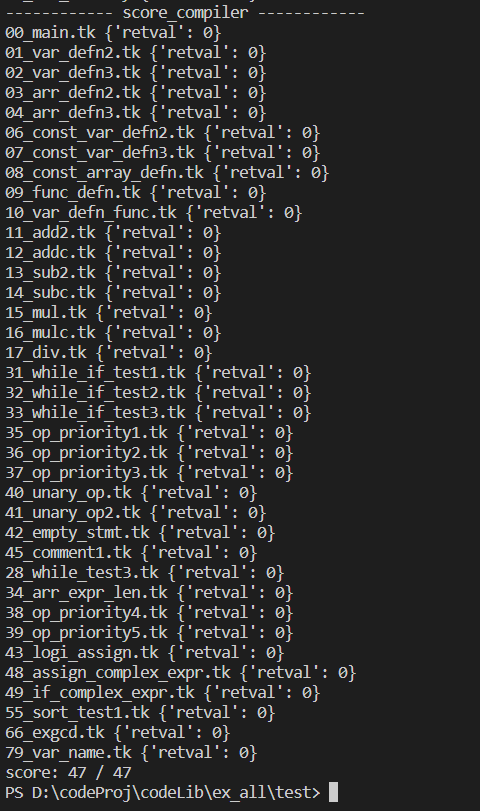
调试发现，由于||中的|并不属于单预算符，导致判断为Empty状态而丢失了||，需要在set中添加|和&。



对于16进制匹配错误，调试发现0x都是对的，但在x后接收b导致提前输出。是因为源代码只考虑了0后x和b的情况，而x后的字母abcdef没有考虑到，添加后该点通过，至此词法分析调试结束





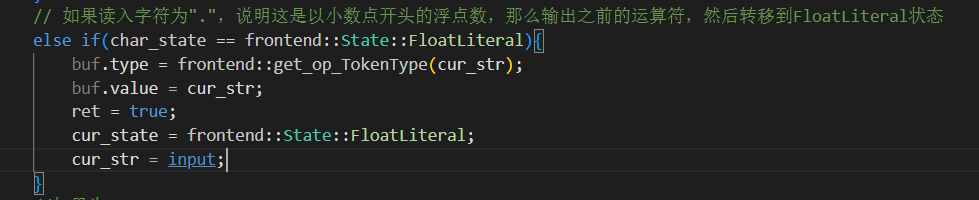


* 切换到实验二的项目，发现95号测试点未通过

问题出现浮点数的判断，没有考虑以”.”开头的字符串







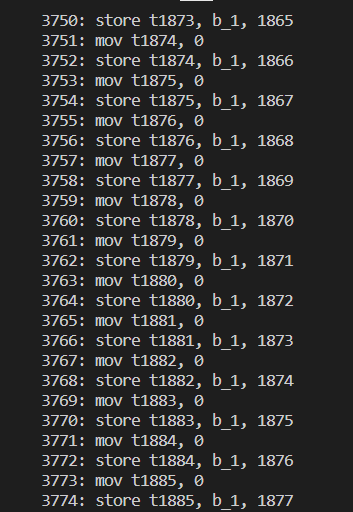
语法分析

* 切换到实验二的项目，发现95号测试点未通过

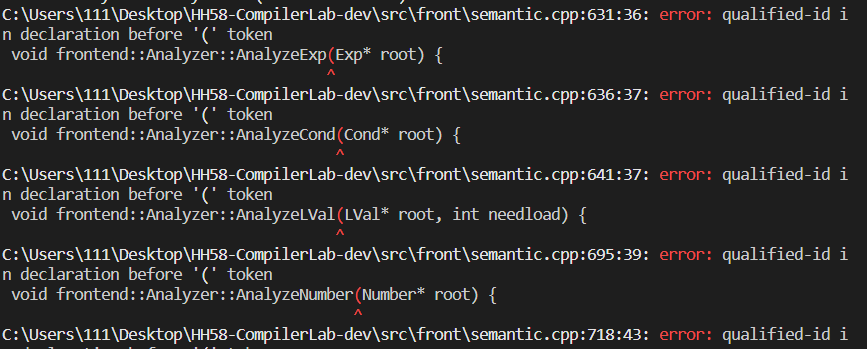
修改词法分析后，语法分析也通过

IR

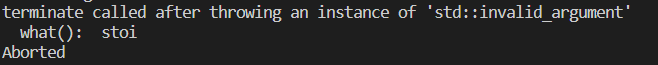
* 添加函数是添加到全局还是当前
* ConstDef中，最后一步如果是直接用的children的name会导致找不到ste，因为生成ste时用的时初始名字
* ConstDef中，数组声明问题，顺序不一致导致8号测试点过不去
* ConstInitVal中，如果判断ste的维度，会导致一个测试点过不去，但是判断下一个token就可以；
* ConstInitVal中，改变ste的level和len后，需要将其复原，否则八号测试点过不去
* 添加全局变量初始化指令后，需要加入return，否则无法得到正确结果
* VarDecl中必须先对子节点VarDef的t赋值，再对其分析，因为变量声明已经规定了他的类型，不用根据后面判断
* VarDef中条件判断需要具体，否则出错，与ConstDef不一样
* InitVal中
* TODO：改写.h文件



* Stmt结构比较复杂，编写的时候缺少了一个大括号



* Lval需要考虑取值和存值两种情况
* Number需要分辨进制 51测试点
* Primary不能直接向上传值，需要先判断类型
* 估计忽略了部分类型转换，导致stoi出错



LAB3

* FuncDef分析block时需要先将该block加入分析域再分析参数，否则导致89号测试点过不去