**实验报告**

一、需求分析

读入区块、交易、输入输出数据

找到冲突交易，删除冲突交易及之后所有交易

能将所有区块按高度顺序组织成链表，打印链表长度。

输出以下统计信息：区块总数、合法交易总数、不合法的交易总数。

从键盘输入区块高度，输出该区块内容。

从键盘输入交易txID，输出该交易内容。

二、设计思想：概要设计

实现四个存储数据的类：Block、Transaction、Input、Output

实现了读入及解决冲突的函数：

//文件读入，用双向循环链表存储

void blockinit(FILE \*fp, Blockptr &bp, int &nowsize)

void transactioninit(FILE \*fp, Blockptr &bp, int &valid)

void inputsinit(FILE \*fp, Blockptr &bp)

void outputsinit(FILE \*fp, Blockptr &bp)

//冲突查找辅助函数

Transactionptr t\_find(Transactionptr tp\_now,char\* txid)//在transaction数组中查找

Outputptr o\_find(Outputptr op\_now, int index)//在output数组中查找

void tp\_free(Transactionptr &tp\_now,Transactionptr &tp\_move)//释放一个无效交易

//冲突查找主函数，沿着block依次查询交易有效性，删除无效交易，统计数据

void validate(Blockptr &bp, const int &block\_count, int &valid, int &invalid)

三、调试分析

1. 遇到在释放无效交易时，当无效交易为coinbase，删除时可能使block的transaction数组失效，解决方法就是判断交易指针是否为链表头指针，并按情况更改头指针指向

2. 在malloc申请内存后可能出现sigsegv信号，经搜索可能为空间未初始化导致，因此改为calloc申请内存并自动初始化

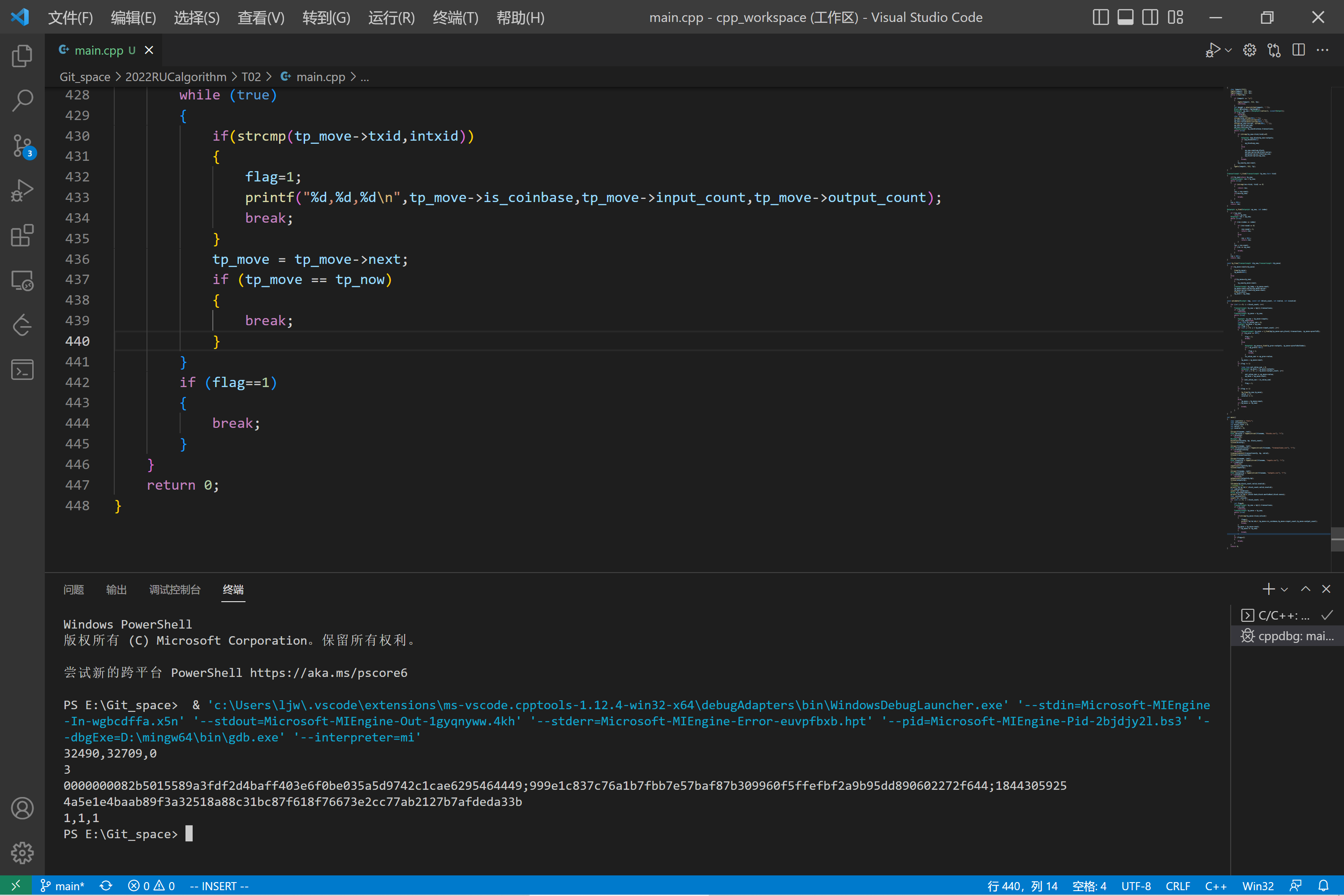
四、测试

full/中数据，

输入

3，

4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b



五、实验总结/心得

经过本次实验，我学习了文件读入与内存管理的方法，对链表的概念、模型及代码实现都有了深入的认识