**Recalibrating Wallet**

# Vid 69

**28:34 🡪**

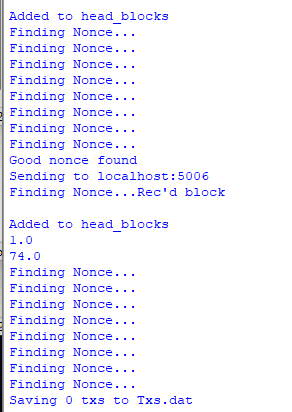
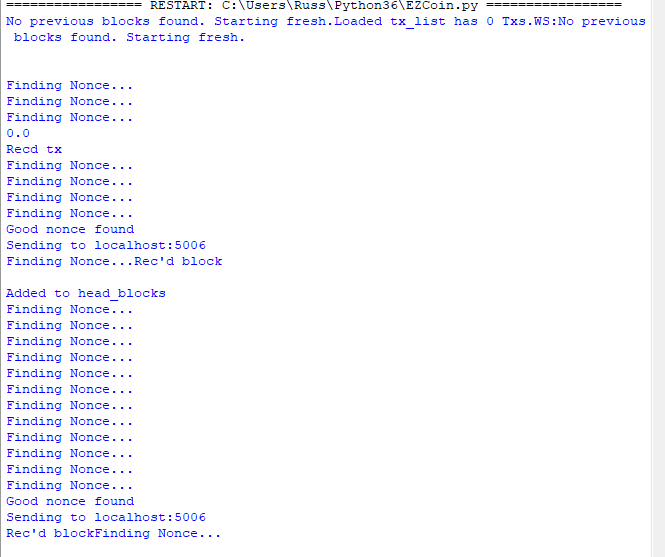
**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Added time.sleep(20) after 1st output, then cleared AllBlocks.dat and WalletBlocks.dat**



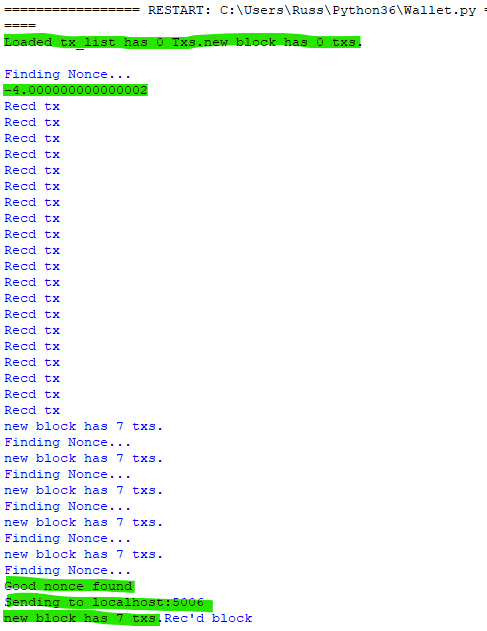
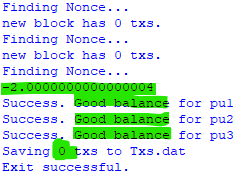
# Vid 71

3:02 🡪 everything passes (added check\_size fn, and called in is\_valid)

4:55 🡪 this block wasn’t valid, so it didn’t get added to the blockchain that the Wallet is keeping hence the wrong balances

5:30 🡪 added check\_size to nonceFinder, added removeTx to TxBlock, (7:21) added placeholder to nonceFinder

9:46 🡪 after Running GOOD BALANCES (remember to clear .dat files)



\*\* See assignment 2 folder for Wallet, Miner and TxBlock re-calibrators \*\*

\*\* MinerRecalibrator2 just has verbosity added so we can watch transactions in real time

# Vid 72

0:28 🡪 adding mining txs to the root block

Text

Description automatically generated with medium confidence

5:37 🡪 Creating and Running overspend test cases SHOULD NOT DETECT OVERSPEND yet

Graphical user interface, text

Description automatically generated

\*\* See assignment 3 for Overspend code that doesn’t work yet

# Vid 73

implement a check so that that no **pu** block spends its entire balance

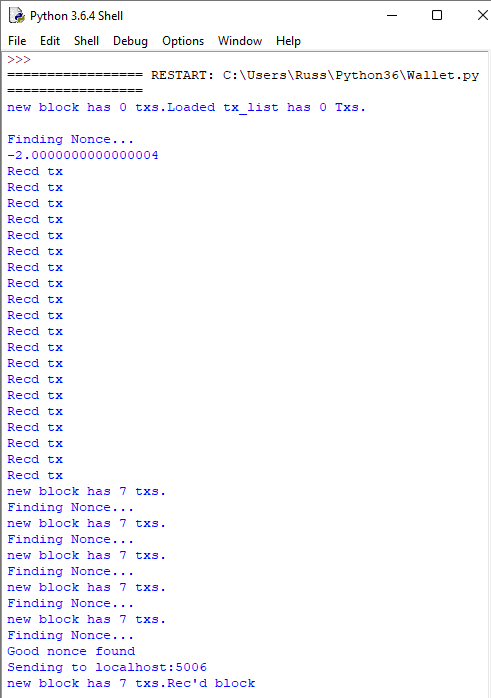
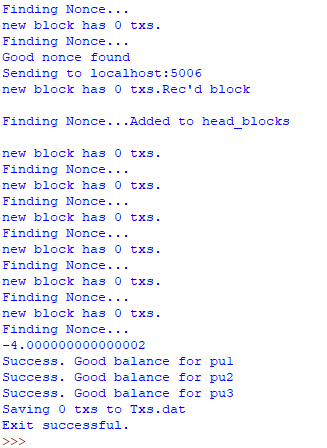
0:26 🡪 moving getBalance into TxBlock

Editing getBalance in Wallet

**Graphical user interface, text, application

Description automatically generated**

2:21 🡪 Running Wallet, still working correctly and we are getting valid blocks so the balances are still good



2:38 🡪 for TxBlock when we ask if something is\_valid, need to be checking the spending that happens in there.

Making a Dict out of the addresses and the this that users have spent.

6:26 🡪 after Running detected overspend

Change B3 inheritance to B1 because B2 is a bad block

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated