ECE 579: Blockchain and Cryptocurrencies 2020
Assignment 1 Part B - ScroogeCoin

In this assignment you will continue building the ScrogeCoin using Python code. ScroogeCoin is discussed in section 1.5 of the book please read the section to understand the mechanisms of ScroogeCoin.

Design Overview:
The design has two main classes User and Scrooge. Scrooge will store the blockchain and will have the authority to create coins and accept transactions, put them into a block and add the block to the blockchain. Scrooge will contain a list to store the transaction requests and only process them to a block when Scrooge calls Mine function. This should clear the transaction list and there is no limit on the number of transactions on a block. Each transaction will consume only a single coin but can output many. Users are only allowed to create transaction requests and forward them to Scrooge for processing.

A template for the Python functions is provided. You need to implement the necessary functions to finish the project.

A simple workflow (Part A + Part B) is as follows:

1. Scrooge and Users create public and private keys.
2. Scrooge create coins for the Users, meaning it creates transactions and add it to the transaction list.
3. Scrooge mines the list to put the transactions into the blockchain.
4. Users create transactions to send coins to each other. Transactions are forwarded to Scrooge for processing.
5. Once Scrooge receives a transaction, it will check if the transaction is valid (we explain the details under the function definitions). If it is valid, it adds the transaction to the transaction list. In case transaction is not valid, it should be discarded with displaying a message on the terminal.
6. Again, once Scrooge calls mine, it puts all the transactions into a block and adds it to the blockchain.

Required Libraries:

In order to run ScroogeCoin you need the following libraries:

- **Python3: to run the script**
- **GMP: required by fastecdsa**
- **fastecdsa : crypto library for Python**

In Part B of this assignment you need to add couple more functions in order to implement the ScroogeCoin.

Summary of the functions that Scrooge uses:

Function from previous assignment:

(1) **KeyGen**: Create Public and Private Keys for Scrooge. Use curve=curve.secp256k1 for key generation.

(2) **CreateCoins**: Only Scrooge can create coins to any user. The function takes a list of input addresses and amount of coins. It will create a transaction that creates coins and adds it to the transaction list which will be later mined. The transaction should be hashed and signed by Scrooge.

(3) **Hash**: Hash can be computed for a string as

hashlib.sha256("1").hexdigest()

(4) **Sign**: Sign function that signs the input which is usually hash of a block or transaction.

(5) **Add Transaction**: Add the submitted transaction by a user to your transaction list if the transaction is valid.

Function that is provided:

(6) **Get User Transaction Position:** For a given input address, find the positions of the unspent coins for the user on the blockchain. Positions = {Block Number, Transaction Index}.

New Functions that needs to be completed:

(7) **Validate Transaction:** Check if the submitted transaction is valid. (Once user creates a transaction, it will be passed to Scrooge. Scrooge will take the transaction and check if it is valid. If it is valid, it will add the transaction to a transaction list. If the transaction is not valid, Scrooge will discard the transaction. In order to put them in to blockcain, we will use another function "Mine". Mine function is called later in the main code, so it is possible that multiple Validate Transaction calls might occur before executing Mine function.

Scrooge will do the following 4 checks to validate a transaction:

- The consumed coins are valid, that is the coins are created in previous transactions.
- The consumed coins were not already consumed in some previous transaction. That is, this is not a double-spend.
- The total value of the coins that come out of this transaction is equal to the total value of the coins that went in. That is, only Scrooge can create new value.
- The transaction is validly signed by the owner of all the consumed coins.

If there is an invalid transaction, the error should be displayed with stating the reason. For example: "Signature is not valid!".

(8) **Mine:** The function will take the transaction list and put all the items into a block. The block is hashed and signed by Scrooge and added to the blockchain. Current Transaction list should be empty after mining. There may be a situation in which transaction list is empty. In that case block should be mined with empty transaction list.

(9) **Print User Balance:** Compute the total balance of a user address. You can scan all the chain to compute all the balance. Display the amount on the terminal.

(10) **Print Block:** Display the contents of a block for a given block number. In the header, you may display, block number, previous hash, and signature. Later, you can list all the transactions with transaction number, sender, hash, location(location of the coins consumed), receiver and signature information.

Summary of the functions Users uses:

Function from previous assignment:

1. **KeyGen**: Create Public and Private Keys for User. Use curve=curve.secp256k1 for key generation.
2. **Hash**: Take the hash of a block. Hash can be computed for a string as hashlib.sha256("1").hexdigest()
3. **Sign**: Sign the hash of a block and append it at the end of the block which creates a larger block that will be added to the blockchain.
4. **Send Transaction**: Prepare a transaction to pass it to Scrooge. The transaction should point to a block and a transaction ID which shows the coin balance that belongs to the user. The coins can be sent to multiple addresses. The total amount of input coins should match the total amount of output coins. Ex: User1 can point a location that holds 50 coins. User1 can spend all the coins between two other users (25 each, 10 User2 and 40 User3, etc.). Another option is that if User1 sends 10 coins to User2, he can send 40 coins back to his address. The transaction should be hashed and signed by the User and passed to Scrooge.

Demo and report requirements:

Once you finish the implementation. Please write test codes for all the scenarios. These are:

1) Mine a valid transaction that consumes coins from a previous block.
2) Create all 4 invalid scenarios and show the error message.
3) Print a couple users balance before and after a transaction occurs between them.
4) Print a block.

Once you finished your demo, please write a small report that. You can include screenshots that shows the cases you presented in your demo.

Student 1 Name:_____ ID:_____ Mailbox:_____
Student 2 Name:_____ ID:_____ Mailbox:_____

Function	Points	Sign
Validate Transaction	55	
Mine	25	
Print User Balance	10	
Print Block	10	
Total	**100**	