Homework 4

The preferable format of the solution is IPython notebook.

Pre-requisite:

1. Generate 3 testnet addresses (address_1, address_2, address_3; should start with either m or n)
2. From prerequisite 1, generate 2-3 MultiSignature address (2 e.g. private_keys for {addess_1, address_2, address_3} out of three are required to spend Bitcoins). Note that your 2-3 MultiSignature address will start with 2.
3. If you face any problems with bitcoin library, try
   - Install version 1.1.39:
     !pip uninstall bitcoin
     !pip install bitcoin==1.1.39
   - Or replace bitcoin.* with bitcoin.transaction.* For example,
     bitcoin.mk_multisig_script(pub_keys,m)->
     bitcoin.transaction.mk_multisig_script(pub_keys,m)
   - Or use **bitcoinlib** (https://github.com/1200wd/bitcoinlib)

Tasks:

1. (1 point) Calculate double Hash_256 of your name. Take two inputs from the user, i.e., first name and last name. Concatenate string and take SHA256(SHA256(<first name><lastname>)).

2. (1 point) Get some testnet Bitcoins to your testnet address_1 from any source. (For example: https://testnet-faucet.mempool.co/, https://coinfaucet.eu/en/btc-testnet/ or search "bitcoin testnet faucet"). Provide the Transaction_ID.

3. (2 points) Create a new transaction from address_1 and provide Transaction_ID. Make 2 outputs:

Output_1 : Some bitcoins to address_2.
Output_2: Data transfer with first 4 Bytes double Hash_256 of your name. You send 0 bitcoins in this output.

Note. Be aware of small transaction fees. You may find https://blockstream.info/testnet/tx/push useful.

4. (2 points) From address_2, Send some bitcoins to your 2-of-3 MultiSignature Address.

5. (4 points) From 2-of-3 MultiSignature Address. Send some bitcoins back to address_1.

6. (bonus) Write a function to
   - (1* point) compute a transaction hash taking a transaction hex string as an input. It should work at least for P2PKH inputs. Provide an execution example.
   - (2* points) check sigscript validity for given UTXO, transaction hash and sigscript as an input for P2PKH input. You should check formats of UTXO and sigscript, and verify signature. Provide an execution example.
7. (2* points bonus) Generate P2WPKH address, send transaction to it and from it. Feel free to use **bitcoinlib** for all the steps (https://github.com/1200wd/bitcoinlib). Provide transaction IDs.

Guidelines:

1. At any point above, if bitcoins UTXO are not sufficient for the transaction. Get more coins from any arbitrary source.
Write only outputs, i.e., generated addresses and created multi-sig address for prerequisite in your IPython notebook.
2. Provide an IPython notebook with the solution code and corresponding transactions ids.