WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 14

Title: Assignment on Blockchain databases

Program:

```
import hashlib
import json
from time import time

class Blockchain(object):
    def __init__(self):
        self.chain = []
        self.pending_transactions = []

        self.new_block(proof=88)

# Create a new block listing key/value pairs of block information in a JSON
object. Reset the list of pending transactions & append the newest block to
the chain.

    def new_block(self, proof, previous_hash=None):
        block = {
            'index': len(self.chain) + 1,
            'proof': proof,
        }
        self.chain.append(block)

        return block

#Search the blockchain for the most recent block.

    @property
    def last_block(self):

        return self.chain[-1]

# Add a transaction with relevant info to the 'blockpool' - list of pending
tx's.

    def new_transaction(self, sender, recipient):
        transaction = {
            'sender': sender,
            'recipient': recipient,
        }
```

WALCHAND INSTITUTE OF TECHNOLOGY, SOLAPUR
INFORMATION TECHNOLOGY
2021-22 SEMESTER –I

Advanced Database System

Name: Alaikya S Yemul Roll No: 62

ASSIGNMENT NO: 14

Title: Assignment on Blockchain databases

```
        return self.last_block['index'] + 1

# receive one block. Turn it into a string, turn that into Unicode (for
hashing). Hash with SHA256 encryption, then translate the Unicode into a
hexidecimal string.

    def hash(self, block):
        string_object = json.dumps(block, sort_keys=True)
        block_string = string_object.encode()

        raw_hash = hashlib.sha256(block_string)
        hex_hash = raw_hash.hexdigest()

        return hex_hash

blockchain = Blockchain()
t1 = blockchain.new_transaction("Alaikya", "Santosh")
t2 = blockchain.new_transaction("Santosh", "Sudhakar")
t3 = blockchain.new_transaction("Sudhakar", "Walchand")
blockchain.new_block(789)

t4 = blockchain.new_transaction("Walchand", "Alaikya")
t5 = blockchain.new_transaction("Alaikya", "Savita")
t6 = blockchain.new_transaction("Savita", "Santosh")
blockchain.new_block(564)


print("Genesis block: ")
for i in blockchain.chain:
    print(i,sep="\n")
```

Screenshots:

```
Genesis block:
{'index': 1, 'proof': 88}
{'index': 2, 'proof': 789}
{'index': 3, 'proof': 564}
```