Name: Jash Mavani

Subject : Blockchain Technology

Batch : EL2

Practical 3

Aim: To perform thorough study and installation of Anaconda 5 0 1 and Python 3 6 and perform proof of work (consensus mechanism Also, notice the changes in mining rewards and nonce requirement

```
import datetime # To get the time and date
import hashlib # To hash the data
import·random·#·To·generate·a·random·value

def·computeHash(data):
····'''Returns·the·hash·value·of·the·given·data.'''
    return hashlib.sha256(data.encode('utf-8')).hexdigest()

def ProofOfWork(block):
    nonce = 1
    difficulty = 4
    s=''
    print('Difficulty: ', difficulty)
    for i in range(difficulty):
        s+='0'
    while True:
        data = str(block.index) + str(block.timestamp) + str(block.data) + str(block.pr
        h = computeHash(data)
        # print(h)
        if h[:difficulty]==s:
            break
        nonce+=1
    return [h, nonce]


class Block:
    def __init__(self, index, timestamp, data, previousHash):
        '''Block Constructor initiating the Block attribute values.'''
        self.index=index # Index of the Block
        self.timestamp=timestamp # Timestampof the block
        self.data=data # Data of th block
        self.previousHash=previousHash # Hash value of the previous block
        self.nonce=1

        difficulty = 4
        s=''
```

```
        for i in range(difficulty):
            s+='0'
        while True:
            data = str(self.index) + str(self.timestamp) + str(self.data) + str(self.pr
            h = computeHash(data)
            # print(h)
            if h[:difficulty]==s:
                break
            self.nonce+=1

        self.currentHash=computeHash(data) # Hash value of the current block
        if self.index==0:
            self.previousHash=''
            for i in range(len(self.currentHash)):
                self.previousHash+='0'



    def printBlock(self):
        '''Prints the values of the block.'''
        print()
        print('Index:\t\t ', self.index)
        print('TimeStamp:\t ', self.timestamp[:-7])
        print('Data:\t\t ', self.data)
        print('Previous Hash:\t ', self.previousHash)
        print('Current Hash:\t ', self.currentHash)
        print('Nonce:\t\t ', self.nonce)


class BlockChain:
    def __init__(self):
        '''Blockchain Constructor initiating the blockchain with a genesis block.'''
        genesis = Block(0, str(datetime.datetime.now()), 'This is Genesis block', 0) #
        self.chain = [genesis] # Blockchain initiated
        self.total_reward = random.randint(100, 100000) # Total initial reward
        self.balance = self.total_reward # Balance of the reward

    def addBlock(self, data, reward):
        '''This function adds a new block to the blockchain.'''
        if reward>self.balance: # Check if there is sufficient balance to add a new blo
            print('No more blocks can be added')
            return -1
        index = len(self.chain)
        timestamp = str(datetime.datetime.now())
        previousHash = self.chain[-1].currentHash
        a = Block(index, timestamp, data, previousHash) # New block is created
        self.chain.append(a) # New block is appended
        self.balance -= reward # Reward is subtracted from the balance
        input('Press ENTER to start Mining... ')
        b = ProofOfWork(a)
        a.currentHash = b[0]
        a.nonce = b[1]
        return reward

    def printBlockChain(self):
```

```
        '''This function prints the complete blockchain.'''
        for block in self.chain:
            block.printBlock()


# Initiating Blockchain
blockchain = BlockChain()
print()
print('Initiating Blockchain...')
reward = 0.1


while True:
    # Menu-driven
    print()
    print('1.   Add Block')
    print('2.   Print Blockchain')
    print('3.   Print specific Block')
    print('4.   Print Total Reward')
    print('0.   Exit')
    print()

    # Input from the user
    a = int(input('Enter your choice: '))
    if a==0:
        # Exit
        break
    elif a==1:
        # Adding a new block
        val = input('Enter the data: ')
        reward = blockchain.addBlock(val, reward)
        if reward>0:
            print('Block Added Successfully...')
            print('Reward: ', reward)
            reward*=2
    elif a==2:
        # Printing the complete Blockchain's blocks
        blockchain.printBlockChain()
    elif a==3:
        # Printing the specific indexed block
        val=int(input('Enter index number: '))
        if val>=len(blockchain.chain) or val<=0:
            print('Wrong index')
            continue
        blockchain.chain[val].printBlock()
    elif a==4:
        # Printing Blockchain balance
        print('Blockchain Initial Balance: {:.2f}'.format(blockchain.total_reward))
        print('Blockchain Balance: {:.2f}'.format(blockchain.balance))

```

```
Initiating Blockchain...

1.   Add Block
```

```
2.   Print Blockchain
3.   Print specific Block
4.   Print Total Reward
0.   Exit

Enter your choice: 2

Index:            0
TimeStamp:        2022-10-18 09:19:24
Data:             This is Genesis block
Previous Hash:    0000000000000000000000000000000000000000000000000000000000000000
Current Hash:     00003fe773ec7162c504dc1d452f60089559277f87fed109f0f60ae50ca57958
Nonce:            243074

1.   Add Block
2.   Print Blockchain
3.   Print specific Block
4.   Print Total Reward
0.   Exit

Enter your choice: 1
Enter the data: Hi I'm Jash
Press ENTER to start Mining...
Difficulty:  4
Block Added Successfully...
Reward:  0.1

1.   Add Block
2.   Print Blockchain
3.   Print specific Block
4.   Print Total Reward
0.   Exit

Enter your choice: 1
Enter the data: Nirma University
Press ENTER to start Mining...
Difficulty:  4
Block Added Successfully...
Reward:  0.2

1.   Add Block
2.   Print Blockchain
3.   Print specific Block
4.   Print Total Reward
0.   Exit

Enter your choice: 2

Index:            0
TimeStamp:        2022-10-18 09:19:24
Data:             This is Genesis block
Previous Hash:    0000000000000000000000000000000000000000000000000000000000000000
Current Hash:     00003fe773ec7162c504dc1d452f60089559277f87fed109f0f60ae50ca57958
Nonce:            243074
```

1

✓ 59s completed at 2:50 PM