PRACTICAL 6

Name:	Harsh Shah	**Semester:**	VII	**Division:**	6
Roll No.:	21BCP359	**Date:**	05-09-24	**Batch:**	G11
Aim:	To Implement PoW Consensus Mechanism on your own Blockchain				

Proof of Work (PoW)

Proof of work (PoW) is a blockchain consensus mechanism that requires significant computing effort from a network of devices. The concept was adapted from digital tokens by Hal Finney in 2004 through the idea of "reusable proof of work" using the 160-bit secure hash algorithm 1 (SHA-1).

Program

```
import hashlib
import time

class Block:
    def __init__(self, index, previous_hash, data, nonce=0):
        self.index = index
        self.timestamp = time.time()
        self.previous_hash = previous_hash
        self.data = data
        self.nonce = nonce
        self.hash = self.calculate_hash()

    def calculate_hash(self):
        block_string =
f"{self.index}{self.timestamp}{self.previous_hash}{self.data}{self.nonce}".encode()
        return hashlib.sha256(block_string).hexdigest()

    def __str__(self):
        return (
            f"Block Index    : {self.index}\n"
            f"Timestamp      : {time.ctime(self.timestamp)}\n"
            f"Previous Hash  : {self.previous_hash}\n"
            f"Hash           : {self.hash}\n"
            f"Data           : {self.data}\n"
            f"Nonce          : {self.nonce}\n"
            f"{'-'*41}"
        )

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

    def create_genesis_block(self):
        return Block(0, "0", "Genesis Block")
```

```
    def get_latest_block(self):
        return self.chain[-1]

    def add_block(self, data):
        latest_block = self.get_latest_block()
        new_block = Block(len(self.chain), latest_block.hash, data)
        new_block = self.proof_of_work(new_block)
        self.chain.append(new_block)

    def proof_of_work(self, block, difficulty=4):
        while block.hash[:difficulty] != "0" * difficulty:
            block.nonce += 1
            block.hash = block.calculate_hash()
        return block

    def is_chain_valid(self):
        for i in range(1, len(self.chain)):
            current_block = self.chain[i]
            previous_block = self.chain[i - 1]
            if current_block.hash != current_block.calculate_hash():
                return False
            if current_block.previous_hash != previous_block.hash:
                return False
        return True

    def __str__(self):
        return "\n".join(str(block) for block in self.chain)


blockchain = Blockchain()

while True:
    data = input("Enter transaction data for the new block (or 'q' to quit): ")

    if data.lower() == "q":
        break

    blockchain.add_block(data)
    print("\nBlock added successfully!")

print("\nFinal Blockchain:")
print(blockchain)

print("\nBlockchain is valid:", blockchain.is_chain_valid())
```

Output

```
 Blockchain Lab  main  python -u "c:\Users\harsh\OneDrive - pdpu.ac.in\HARSH\_PDEU\SEM 7
chain_without_pow.py"
Enter transaction data for the new block (or 'q' to quit): Alice Sent Rs.100 to Bob

Block added successfully!
Enter transaction data for the new block (or 'q' to quit): Bob sent Rs.50 to John

Block added successfully!
Enter transaction data for the new block (or 'q' to quit): John recieved Rs.300 from Harry

Block added successfully!
Enter transaction data for the new block (or 'q' to quit): Harry sent Rs.410 to Alice

Block added successfully!
Enter transaction data for the new block (or 'q' to quit): q
```

```
Final Blockchain:
Block Index    : 0
Timestamp      : Thu Sep 12 14:01:58 2024
Previous Hash  : 0
Hash           : c9548f1eeb47e99d8de09654973dab1a4130c297a57f7b8b2a538ff0c22a6f8f
Data           : Genesis Block
Nonce          : 0
----------------------------------------
Block Index    : 1
Timestamp      : Thu Sep 12 14:02:11 2024
Previous Hash  : c9548f1eeb47e99d8de09654973dab1a4130c297a57f7b8b2a538ff0c22a6f8f
Hash           : 1b58b6343f46a25f9221fd926c04e51c58c0593870b1019dbb2e35da2d6c2f26
Data           : Alice Sent Rs.100 to Bob
Nonce          : 0
----------------------------------------
Block Index    : 2
Timestamp      : Thu Sep 12 14:02:15 2024
Previous Hash  : 1b58b6343f46a25f9221fd926c04e51c58c0593870b1019dbb2e35da2d6c2f26
Hash           : ff0e21f994d7f6ae120108a7b22718f54b550e0cbbdc85fd07ff284f186dcbd2
Data           : Bob sent Rs.50 to John
Nonce          : 0
----------------------------------------
```

```
Block Index    : 3
Timestamp      : Thu Sep 12 14:02:25 2024
Previous Hash  : ff0e21f994d7f6ae120108a7b22718f54b550e0cbbdc85fd07ff284f186dcbd2
Hash           : ddc612ac627c606acb1a8afa561d1d28ef481ff25564076263519a75349b6ffa
Data           : John recieved Rs.300 from Harry
Nonce          : 0
----------------------------------------
Block Index    : 4
Timestamp      : Thu Sep 12 14:02:29 2024
Previous Hash  : ddc612ac627c606acb1a8afa561d1d28ef481ff25564076263519a75349b6ffa
Hash           : bf017008971d29023da4469469a38ec3e74de2b7d8efe9429667eeee53343da2
Data           : Harry sent Rs.410 to Alice
Nonce          : 0
----------------------------------------

Blockchain is valid: True
```