Blockchain Technologies 1

Assignment 1

Adilet Seilkanov  
SE-2303

**hash(text)**

* Implements SHA-256 manually, following the FIPS PUB 180-4 standard.
* Appends a single 0x80 byte, then zeroes, then the original length in bits, then processes 512-bit blocks.
* Uses \_right\_rotate for the internal compression function.

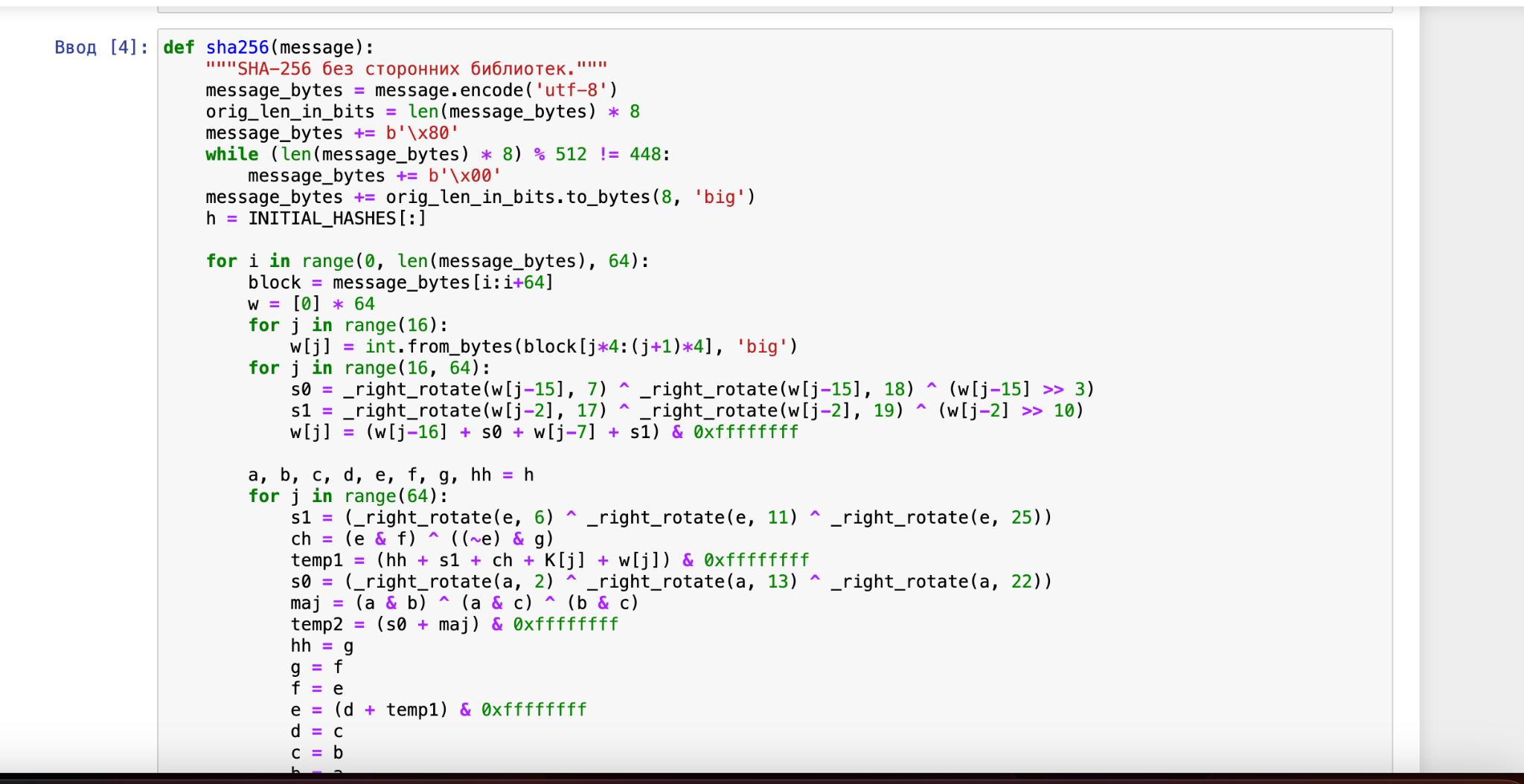
**Transaction(sender, receiver, amount)**

* A simple data structure representing a transaction.
* \_\_str\_\_() returns a string like "Dimash Kudaibergen|Kairat Nurtas|10", which is then hashed.

**build\_merkle\_root(transactions)**

* Takes the SHA-256 hash of each transaction.
* Concatenates adjacent hashes and re-hashes until a single Merkle root remains.
* If an odd number of items remains at any level, the last item can be duplicated (common Merkle tree approach).







**Block**

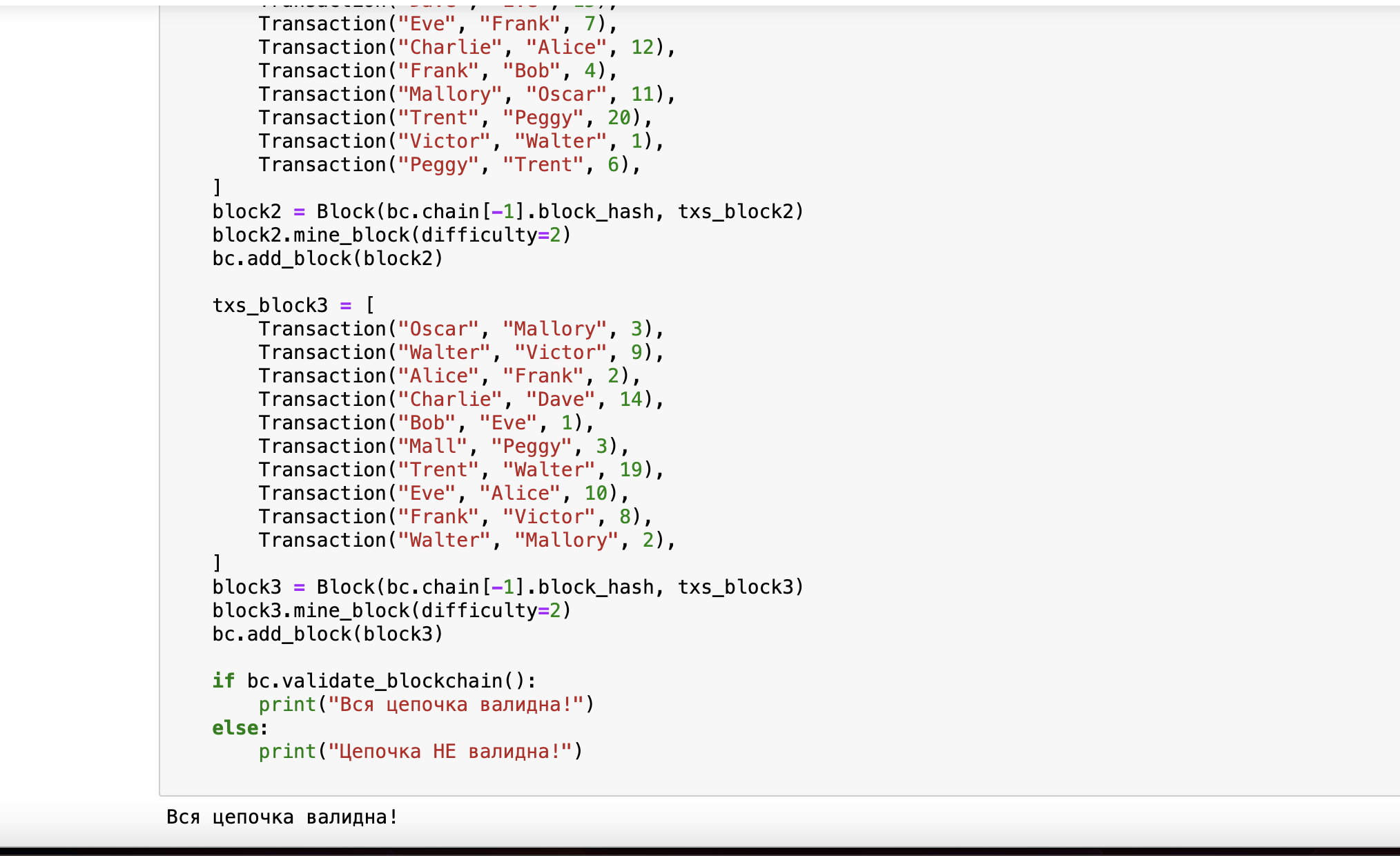
* **Stores previous\_hash, timestamp, merkle\_root, the list of 10 transactions, and the mining variables nonce and block\_hash.**
* **calculate\_hash() returns the hash of concatenating previous\_hash, timestamp, merkle\_root, and nonce.**
* **mine\_block(difficulty=2) increments nonce until the resulting hash starts with two zeros ("00").**

**Blockchain**

* **Holds the entire chain in a list.**
* **Automatically generates a genesis block with '0'\*64 as its previous\_hash.**
* **validate\_blockchain() recalculates each block’s hash and Merkle root, checking consistency with the chain.**

**Demonstration**

* **Creates the blockchain object (bc).**
* **Builds 10 transactions in txs\_block2, creates block2, mines it, and appends to the chain.**
* **Repeats for txs\_block3.**
* **Invokes bc.validate\_blockchain() to verify integrity.**



The project demonstrates how to implement a basic blockchain with manual SHA-256, Merkle tree, block mining, and chain validation.

Each member must be able to explain code lines, especially bitwise operators (>>, <<, &, etc.) and critical functions (validate\_blockchain(), mine\_block()).

Future improvements: digital signatures, dynamic difficulty, storing balances, or introducing a peer-to-peer network layer.

https://github.com/Adikkk17/-.git

### **Final Remarks**

* This solution meets the stated assignment requirements:
  + **Hashing Algorithm**: Implemented from scratch (SHA-256).
  + **Blockchain Implementation**: Each block has previous\_hash, timestamp, merkle\_root, plus transactions.
  + **Merkle Tree and Transactions**: 10 transactions per block, combined via build\_merkle\_root().
  + **Mining and Validation**: mine\_block() with difficulty=2, chain verification in validate\_blockchain().
  + **Report Quality**: Clear instructions, references, and explanation.