RIGA TECHNICAL UNIVERSITY

Faculty of Computer Science, Information Technology and Energy

Institute of Applied Computer Systems

Alina Verkholomova

Academic Master Study Programme “Business Informatics”

Assignment for blockchain development

Assignment on Introduction to Blockchain Technology

Scientific adviser Arnis Staško

RIGA 2024

TASK DESCRIPTION

1) Code and test a blockchain:
    a) Write a code
    b) Add advanced blockchain features
    c) Create a blockchain with a minimum of 10 blocks. Visualize the result.
    d) Try to make an unauthorized change
        i) Change a block in the middle of the blockchain
        ii) Change the last block of the blockchain
        iii) Add another block at the end of the blockchain
        iv) Is it possible to detect a fake? How?
2) Upload the code you have created, tests, samples and description summarized in a document.

SUMMARY

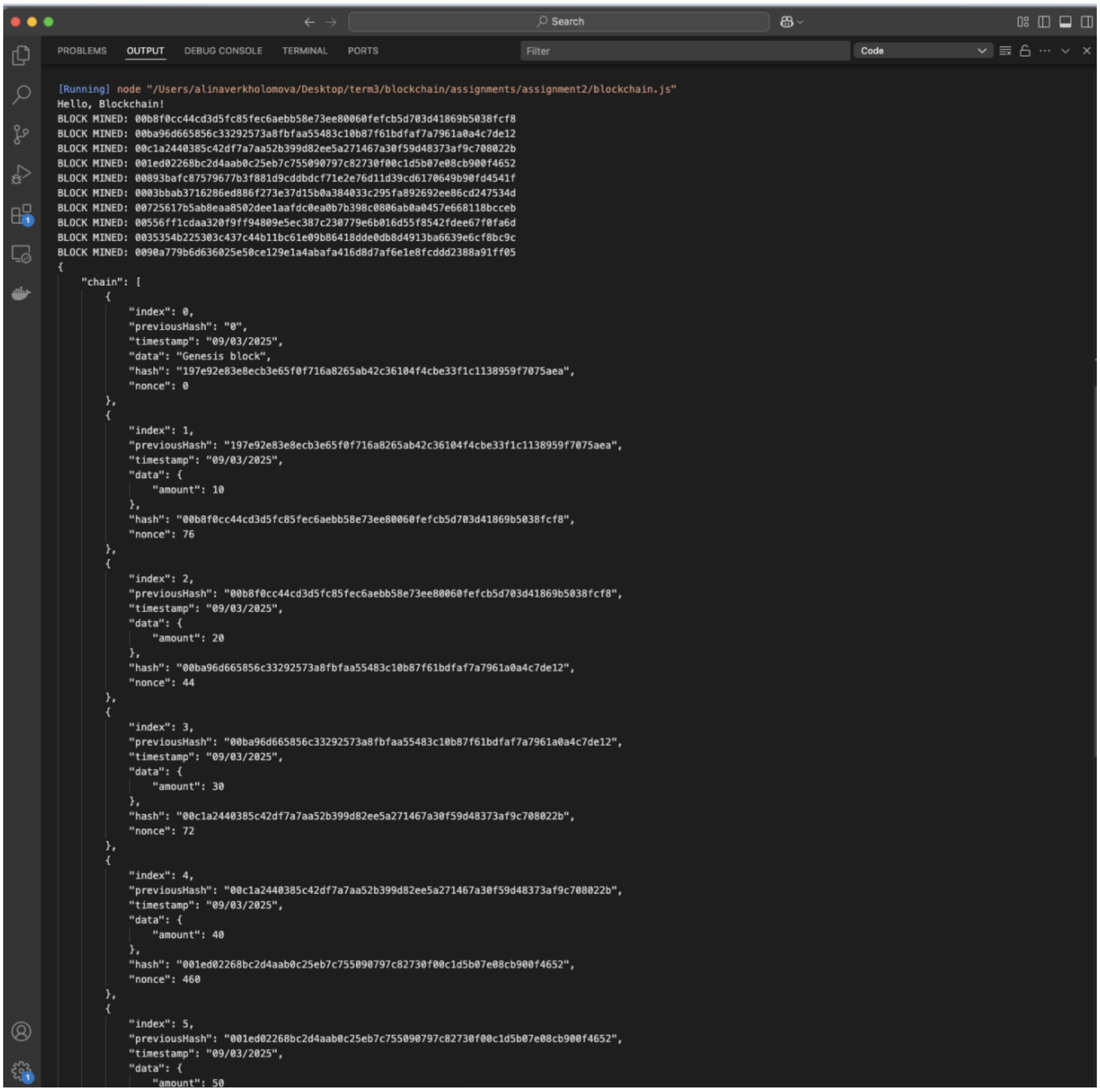
A code was developed to build a blockchain chain with 10 blocks and advanced features (proof-of-work).

```
const SHA256 = require("crypto-js/sha256");

class Block {
  constructor(index, timestamp, data, previousHash = '') {

    this.index = index;
    this.previousHash = previousHash;
    this.timestamp = timestamp;
    this.data = data;
    this.hash = this.calculateHash();
    this.nonce = 0;

  }

  calculateHash() {
    return SHA256(this.index + this.previousHash + this.timestamp + this.nonce +
JSON.stringify(this.data)).toString();
  }

  mineBlock(difficulty) {
    while (this.hash.substring(0, difficulty) !== Array(difficulty + 1).join("0")){
      this.nonce++;
      this.hash = this.calculateHash();
    }

    console.log("BLOCK MINED: " + this.hash);
  }

}

class Blockchain {
  constructor() {
    this.chain = [this.createGenesisBlock()];
    this.difficulty = 2;
    }

    createGenesisBlock() {
        return new Block(0, "09/03/2025", "Genesis block", "0");
    }

    getLatestBlock(){
        return this.chain[this.chain.length - 1];
    }

    addBlock(newBlock) {
        newBlock.previousHash = this.getLatestBlock().hash;
        newBlock.mineBlock(this.difficulty);
        this.chain.push(newBlock);
    }

    isChainValid() {
        for (let i = 1; i < this.chain.length; i++) {
            const currentBlock = this.chain[i];
            const previousBlock = this.chain[i - 1];

            if (currentBlock.hash !== currentBlock.calculateHash()) {
                return false;
            }

            if (currentBlock.previousHash !== previousBlock.hash) {
                return false;
            }
        }
        return true;
    }

}

console.log("Hello, Blockchain!");
let myBlockchain = new Blockchain();

for (let i = 1; i <= 10; i++) {
    myBlockchain.addBlock(new Block(i, "09/03/2025", { amount: i * 10 }));
}
```

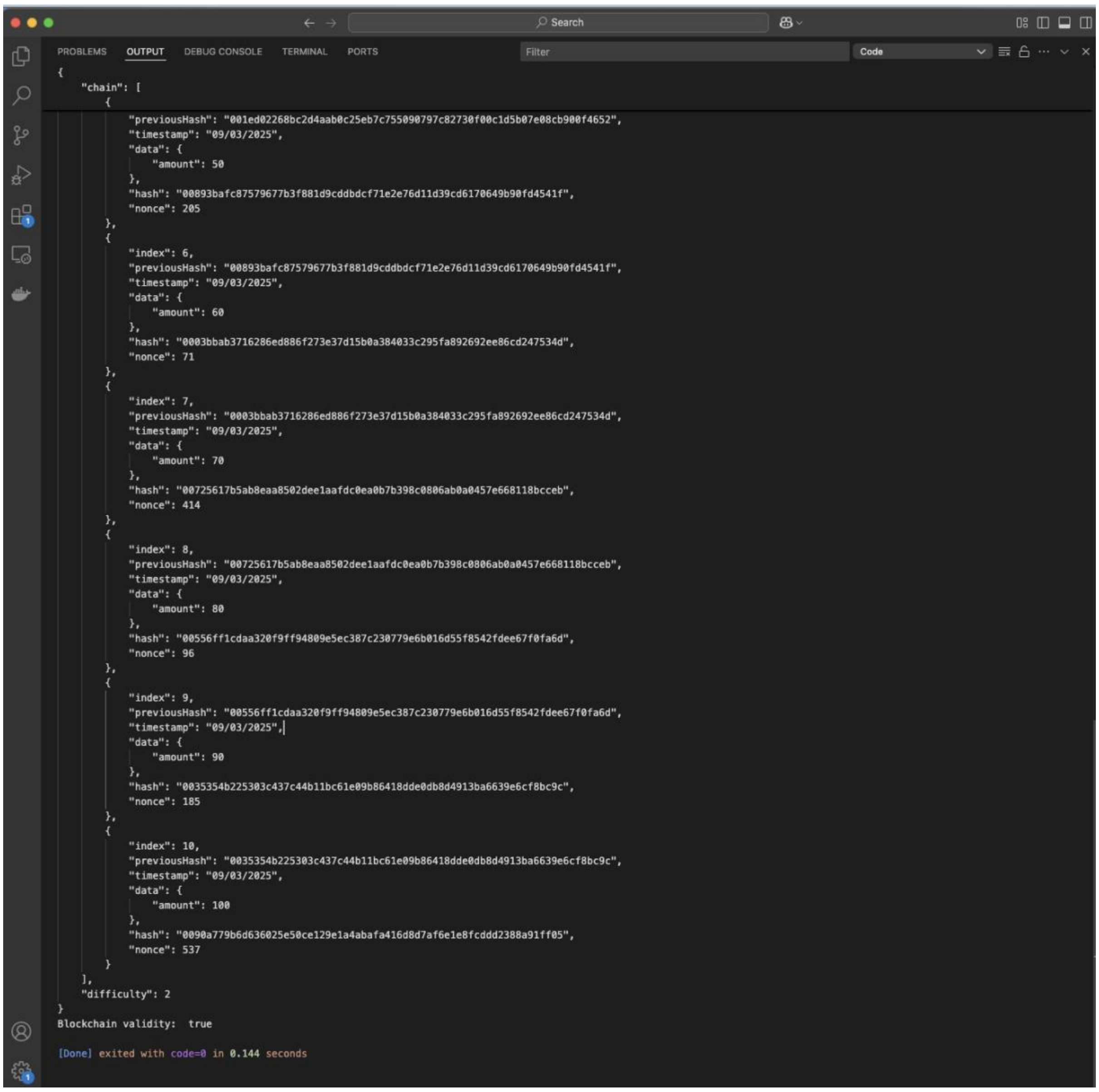
```
console.log(JSON.stringify(myBlockchain, null, 4));
console.log("Blockchain validity: ", myBlockchain.isChainValid());
```

In this code, blocks are created and added to `myBlokchain` by using `for` loop. Then, `myBlockhain` is visualized by using the `console.log` output command and `JSON.stringify`. Afterward, `myBlockchain` is validated.

The output:



```
[Running] node "/Users/alinaverkholomova/Desktop/term3/blockchain/assignments/assignment2/blockchain.js"
Hello, Blockchain!
BLOCK MINED: 00b8f0cc44cd3d5fc85fec6aebb58e73ee80060fefcb5d703d41869b5038fcf8
BLOCK MINED: 00ba96d665856c33292573a8fbfaa55483c10b87f61bdfaf7a7961a0a4c7de12
BLOCK MINED: 00c1a2440385c42df7a7aa52b399d82ee5a271467a30f59d48373af9c708022b
BLOCK MINED: 001ed02268bc2d4aab0c25eb7c755090797c82730f00c1d5b07e08cb900f4652
BLOCK MINED: 00893bafc87579677b3f881d9cddbdcf71e2e76d11d39cd6170649b90fd4541f
BLOCK MINED: 0003bbab3716286ed886f273e37d15b0a384033c295fa892692ee86cd247534d
BLOCK MINED: 00725617b5ab8eaa8502dee1aafdc0ea0b7b398c0806ab0a0457e668118bcceb
BLOCK MINED: 00556ff1cdaa320f9ff94809e5ec387c230779e6b016d55f8542fdee67f0fa6d
BLOCK MINED: 0035354b225303c437c44b11bc61e09b86418dde0db8d4913ba6639e6cf8bc9c
BLOCK MINED: 0090a779b6d636025e50ce129e1a4abafa416d8d7af6e1e8fcddd2388a91ff05
{
    "chain": [
        {
            "index": 0,
            "previousHash": "0",
            "timestamp": "09/03/2025",
            "data": "Genesis block",
            "hash": "197e92e83e8ecb3e65f0f716a8265ab42c36104f4cbe33f1c1138959f7075aea",
            "nonce": 0
        },
        {
            "index": 1,
            "previousHash": "197e92e83e8ecb3e65f0f716a8265ab42c36104f4cbe33f1c1138959f7075aea",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 10
            },
            "hash": "00b8f0cc44cd3d5fc85fec6aebb58e73ee80060fefcb5d703d41869b5038fcf8",
            "nonce": 76
        },
        {
            "index": 2,
            "previousHash": "00b8f0cc44cd3d5fc85fec6aebb58e73ee80060fefcb5d703d41869b5038fcf8",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 20
            },
            "hash": "00ba96d665856c33292573a8fbfaa55483c10b87f61bdfaf7a7961a0a4c7de12",
            "nonce": 44
        },
        {
            "index": 3,
            "previousHash": "00ba96d665856c33292573a8fbfaa55483c10b87f61bdfaf7a7961a0a4c7de12",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 30
            },
            "hash": "00c1a2440385c42df7a7aa52b399d82ee5a271467a30f59d48373af9c708022b",
            "nonce": 72
        },
        {
            "index": 4,
            "previousHash": "00c1a2440385c42df7a7aa52b399d82ee5a271467a30f59d48373af9c708022b",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 40
            },
            "hash": "001ed02268bc2d4aab0c25eb7c755090797c82730f00c1d5b07e08cb900f4652",
            "nonce": 460
        },
        {
            "index": 5,
            "previousHash": "001ed02268bc2d4aab0c25eb7c755090797c82730f00c1d5b07e08cb900f4652",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 50
```

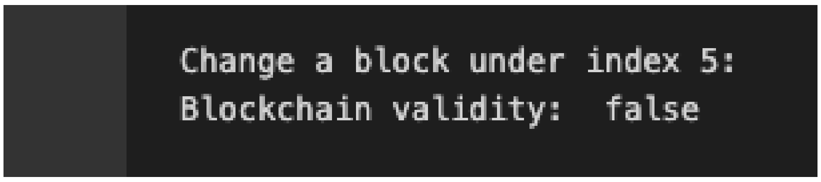


```
{
    "chain": [
        {
            "previousHash": "001ed02268bc2d4aab0c25eb7c755090797c82730f00c1d5b07e08cb900f4652",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 50
            },
            "hash": "00893bafc87579677b3f881d9cddbdcf71e2e76d11d39cd6170649b90fd4541f",
            "nonce": 205
        },
        {
            "index": 6,
            "previousHash": "00893bafc87579677b3f881d9cddbdcf71e2e76d11d39cd6170649b90fd4541f",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 60
            },
            "hash": "0003bbab3716286ed886f273e37d15b0a384033c295fa892692ee86cd247534d",
            "nonce": 71
        },
        {
            "index": 7,
            "previousHash": "0003bbab3716286ed886f273e37d15b0a384033c295fa892692ee86cd247534d",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 70
            },
            "hash": "00725617b5ab8eaa8502dee1aafdc0ea0b7b398c0806ab0a0457e668118bcceb",
            "nonce": 414
        },
        {
            "index": 8,
            "previousHash": "00725617b5ab8eaa8502dee1aafdc0ea0b7b398c0806ab0a0457e668118bcceb",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 80
            },
            "hash": "00556ff1cdaa320f9ff94809e5ec387c230779e6b016d55f8542fdee67f0fa6d",
            "nonce": 96
        },
        {
            "index": 9,
            "previousHash": "00556ff1cdaa320f9ff94809e5ec387c230779e6b016d55f8542fdee67f0fa6d",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 90
            },
            "hash": "0035354b225303c437c44b11bc61e09b86418dde0db8d4913ba6639e6cf8bc9c",
            "nonce": 185
        },
        {
            "index": 10,
            "previousHash": "0035354b225303c437c44b11bc61e09b86418dde0db8d4913ba6639e6cf8bc9c",
            "timestamp": "09/03/2025",
            "data": {
                "amount": 100
            },
            "hash": "0090a779b6d636025e50ce129e1a4abafa416d8d7af6e1e8fcddd2388a91ff05",
            "nonce": 537
        }
    ],
    "difficulty": 2
}
Blockchain validity:  true

[Done] exited with code=0 in 0.144 seconds
```

Change a block in the middle of the blockchain:

```
console.log("\nChange a block under index 5: ");
myBlockchain.chain[4].data = { amount: 1000 };
console.log("Blockchain validity: ", myBlockchain.isChainValid());
```

The output:



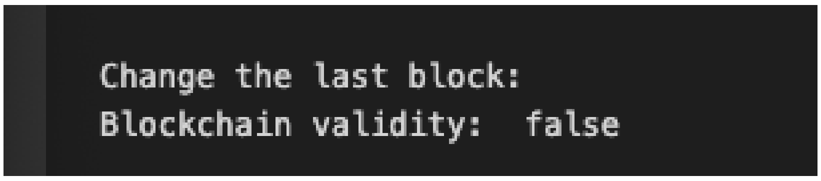
```
Change a block under index 5:
Blockchain validity:  false
```

The validity of `myBlockchain` failed.

Change the last block of the blockchain:

```
console.log("\nChange the last block: ");
myBlockchain.chain[10].data = { amount: 9999 };
console.log("Blockchain validity: ", myBlockchain.isChainValid());
```

The output:



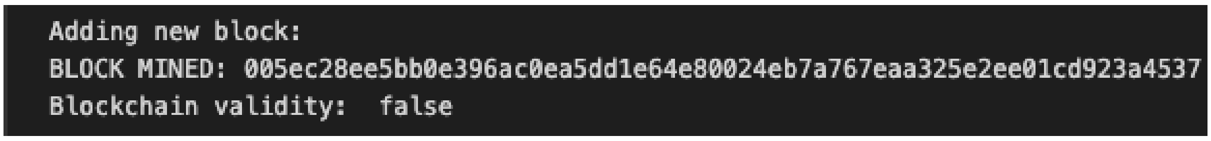
```
Change the last block:
Blockchain validity:  false
```

The validity of `myBlockchain` failed.

Add another block at the end of the blockchain:

```
console.log("\nAdding new block: ");
        myBlockchain.addBlock(new Block(11, "09/03/2025", { amount: 111 }));
        console.log("Blockchain validity: ", myBlockchain.isChainValid());
```

The output:



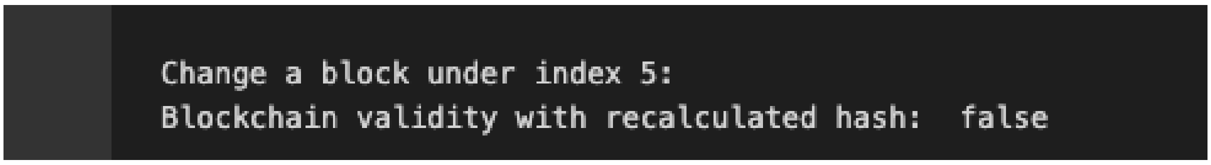
```
Adding new block:
BLOCK MINED: 005ec28ee5bb0e396ac0ea5dd1e64e80024eb7a767eaa325e2ee01cd923a4537
Blockchain validity:  false
```

The validity of `myBlockchain` also failed.

Since I modified some blocks' data, it failed validation. The blockchain detected the fake with the function `isChainValid()`. There is an option to try to recalculate hash after changing the block's data:

```
myBlockchain.chain[4].data = { amount: 1000 };
myBlockchain.chain[4].hash = myBlockchain.chain[4].calculateHash();
```

However, the `previousHash` of the block with index six would be outdated. And the validation of the `myBlockchain` would fail again:



```
Change a block under index 5:
Blockchain validity with recalculated hash:  false
```

Therefore, the hash of all blocks should be recalculated to cover the modification, but it is a time- and energy-consuming option, which makes blockchain a reliable and secure technology.

Link to the GitHub repository:

https://github.com/AlinaVerkholomova/blockhain_dev_assignment