Semester End Practical Exam!!

Name: Dhavalkumar Vijaykumar Patel

Class: M.Sc. Cyber Security Sem-3

Enrollment Number: 032200300002034

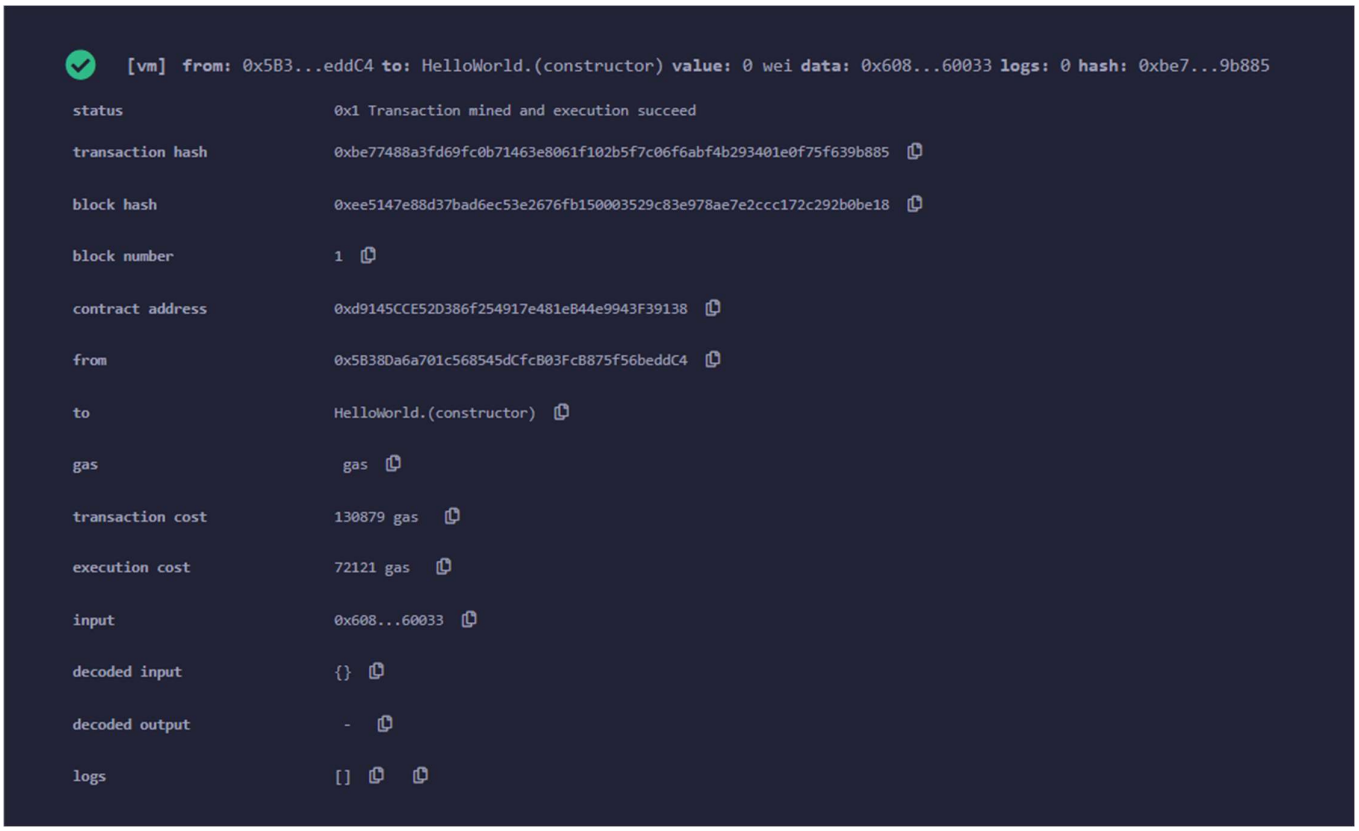
Subject: Blockchain and Cryptocurrencies (CTMSCS SIII L1)

Date: 10 January 2024

Q.1 Create Solidity Program for the Following Task.

a) Basic smart contract that prints “Hello, World!” when executed

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.6.12 <0.9.0;

contract HelloWorld {
  /**
   * @dev Prints Hello World string
   */
  function print() public pure returns (string memory) {
    return "Hello World!";
  }
}
```

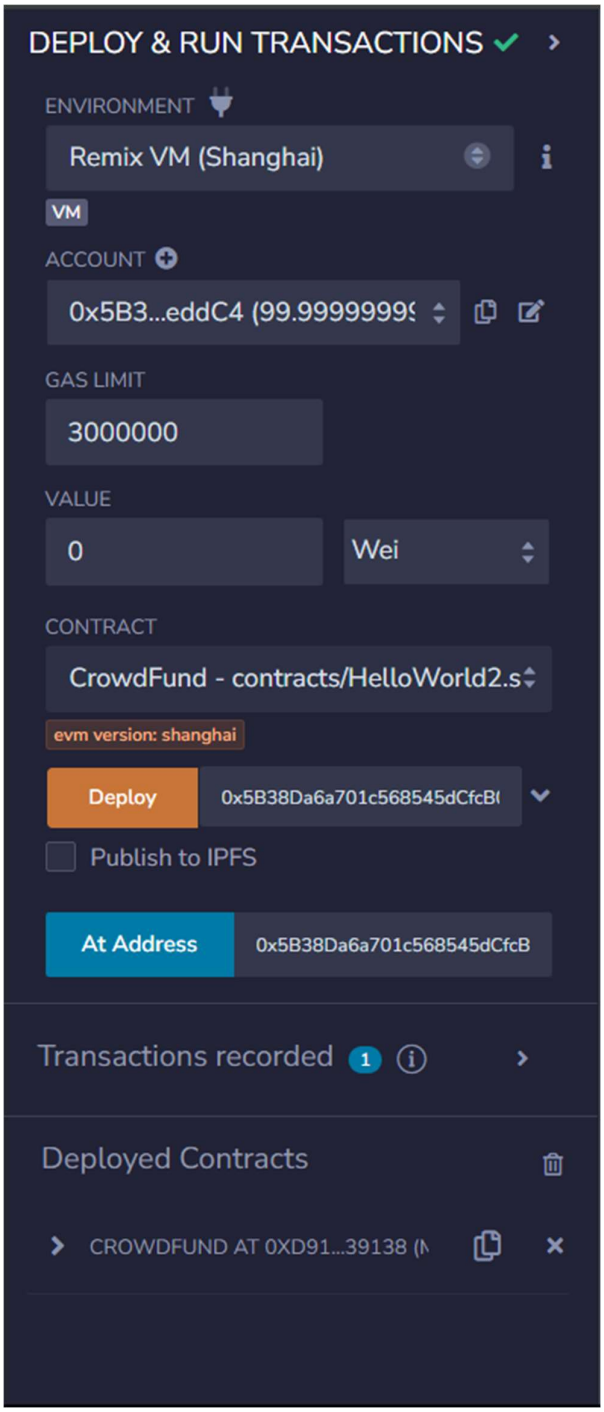


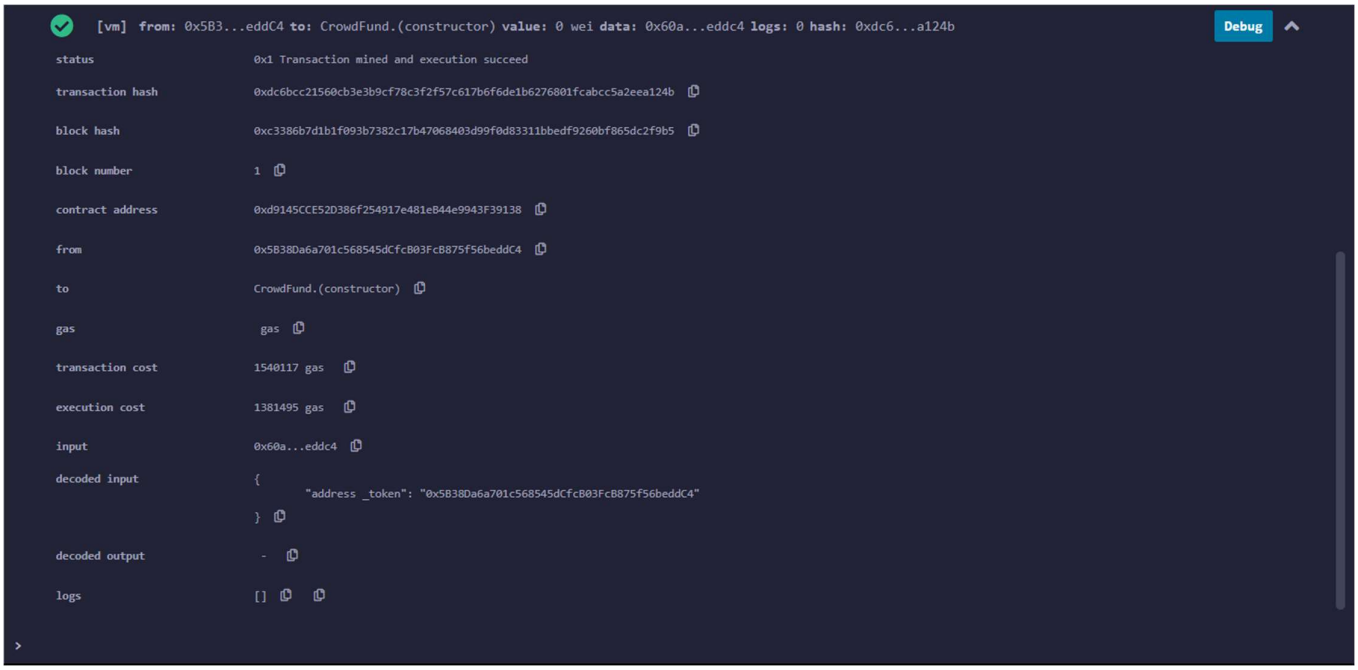
b) Create a simple crowdfunding contract where participants can contribute fund and a project owner can withdraw the fund when goal is reached.

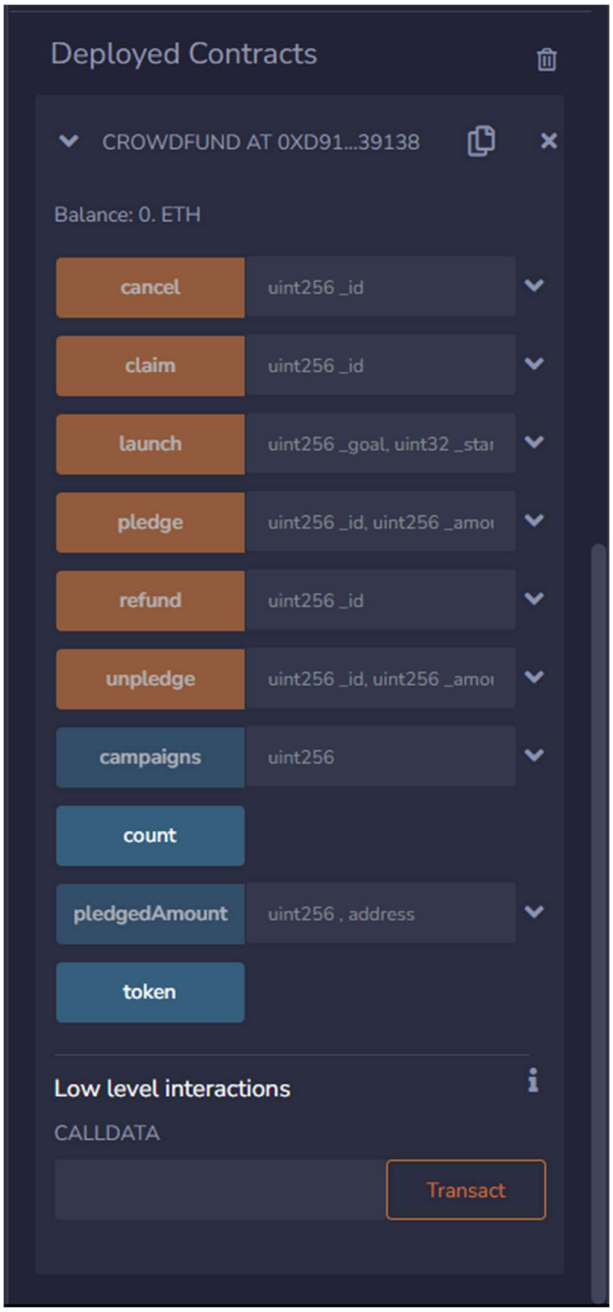
```
// SPDX-License-Identifier: MIT
pragma solidity >=0.6.12 <0.9.0;


interface IERC20 {
    function transfer(address, uint) external returns (bool);


    function transferFrom(address, address, uint) external returns (bool);
}


contract CrowdFund {
    event Launch(
        uint id,
        address indexed creator,
        uint goal,
        uint32 startAt,
        uint32 endAt
    );
    event Cancel(uint id);
    event Pledge(uint indexed id, address indexed caller, uint amount);
    event Unpledge(uint indexed id, address indexed caller, uint amount);
    event Claim(uint id);
    event Refund(uint id, address indexed caller, uint amount);


    struct Campaign {
        // Creator of campaign
        address creator;
        // Amount of tokens to raise
        uint goal;
        // Total amount pledged
        uint pledged;
        // Timestamp of start of campaign
        uint32 startAt;
        // Timestamp of end of campaign
        uint32 endAt;
        // True if goal was reached and creator has claimed the tokens.
        bool claimed;
    }


    IERC20 public immutable token;
    // Total count of campaigns created.
    // It is also used to generate id for new campaigns.
    uint public count;
    // Mapping from id to Campaign
    mapping(uint => Campaign) public campaigns;
    // Mapping from campaign id => pledger => amount pledged
    mapping(uint => mapping(address => uint)) public pledgedAmount;
```

```
    constructor(address _token) {
        token = IERC20(_token);
    }


    function launch(uint _goal, uint32 _startAt, uint32 _endAt) external {
        require(_startAt >= block.timestamp, "start at < now");
        require(_endAt >= _startAt, "end at < start at");
        require(_endAt <= block.timestamp + 90 days, "end at > max duration");


        count += 1;
        campaigns[count] = Campaign({
            creator: msg.sender,
            goal: _goal,
            pledged: 0,
            startAt: _startAt,
            endAt: _endAt,
            claimed: false
        });


        emit Launch(count, msg.sender, _goal, _startAt, _endAt);
    }


    function cancel(uint _id) external {
        Campaign memory campaign = campaigns[_id];
        require(campaign.creator == msg.sender, "not creator");
        require(block.timestamp < campaign.startAt, "started");


        delete campaigns[_id];
        emit Cancel(_id);
    }


    function pledge(uint _id, uint _amount) external {
        Campaign storage campaign = campaigns[_id];
        require(block.timestamp >= campaign.startAt, "not started");
        require(block.timestamp <= campaign.endAt, "ended");


        campaign.pledged += _amount;
        pledgedAmount[_id][msg.sender] += _amount;
        token.transferFrom(msg.sender, address(this), _amount);


        emit Pledge(_id, msg.sender, _amount);
    }


    function unpledge(uint _id, uint _amount) external {
        Campaign storage campaign = campaigns[_id];
        require(block.timestamp <= campaign.endAt, "ended");
```

```
        campaign.pledged -= _amount;
        pledgedAmount[_id][msg.sender] -= _amount;
        token.transfer(msg.sender, _amount);

        emit Unpledge(_id, msg.sender, _amount);
    }

    function claim(uint _id) external {
        Campaign storage campaign = campaigns[_id];
        require(campaign.creator == msg.sender, "not creator");
        require(block.timestamp > campaign.endAt, "not ended");
        require(campaign.pledged >= campaign.goal, "pledged < goal");
        require(!campaign.claimed, "claimed");

        campaign.claimed = true;
        token.transfer(campaign.creator, campaign.pledged);

        emit Claim(_id);
    }

    function refund(uint _id) external {
        Campaign memory campaign = campaigns[_id];
        require(block.timestamp > campaign.endAt, "not ended");
        require(campaign.pledged < campaign.goal, "pledged >= goal");

        uint bal = pledgedAmount[_id][msg.sender];
        pledgedAmount[_id][msg.sender] = 0;
        token.transfer(msg.sender, bal);

        emit Refund(_id, msg.sender, bal);
    }
}
```


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Remix VM (Shanghai)
VM
ACCOUNT
0x5B3...eddC4 (99.9999999
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
CrowdFund - contracts/HelloWorld2.s
evm version: shanghai
Deploy
0x5B38Da6a701c568545dCfcB(
Publish to IPFS
At Address
0x5B38Da6a701c568545dCfcB
Transactions recorded
1
Deployed Contracts
CROWDFUND AT 0XD91...39138 (M


[vm] from: 0x5B3...eddC4 to: CrowdFund.(constructor) value: 0 wei data: 0x60a...eddc4 logs: 0 hash: 0xdc6...a124b
Debug
status 0x1 Transaction mined and execution succeed
transaction hash 0xdc6bcc21560cb3e3b9cf78c3f2f57c617b6f6de1b6276801fcabcc5a2eea124b
block hash 0xc3386b7d1b1f093b7382c17b47068403d99f0d83311bbedf9260bf865dc2f9b5
block number 1
contract address 0xd9145CCE52D386f254917e481eB44e9943F39138
from 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to CrowdFund.(constructor)
gas gas
transaction cost 1540117 gas
execution cost 1381495 gas
input 0x60a...eddc4
decoded input { "address _token": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4" }
decoded output -
logs []



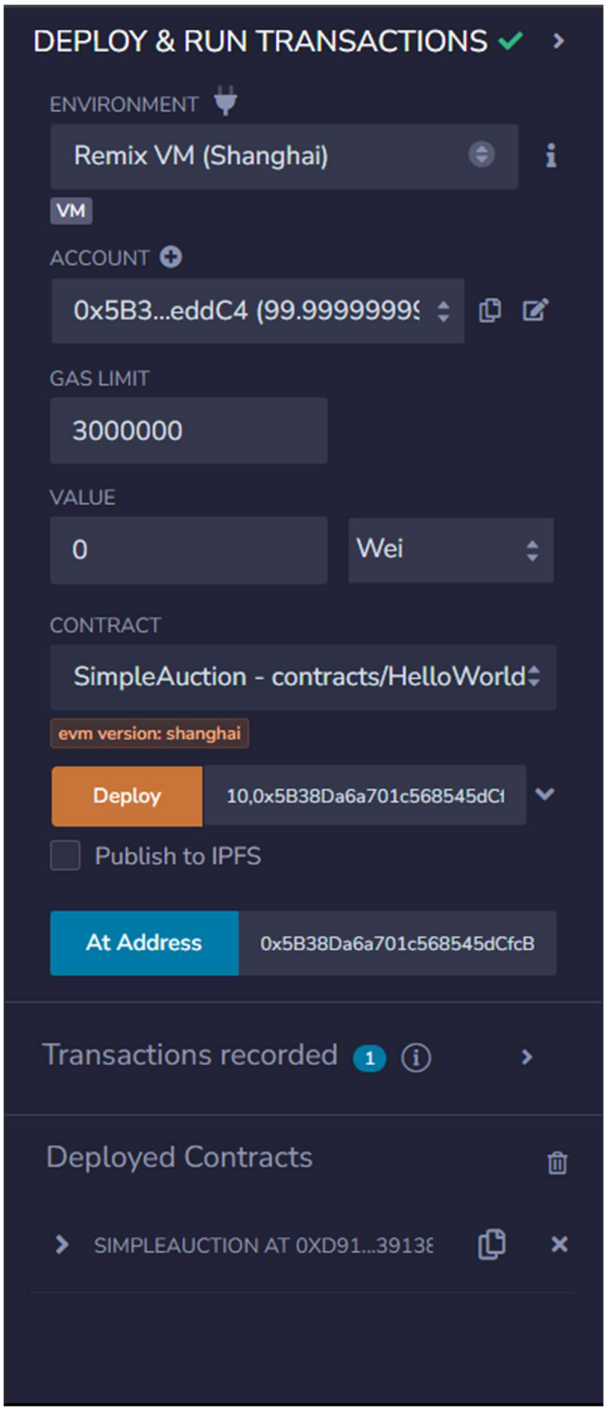
c) Increment a simple voting smart contract web participant can work for different options.

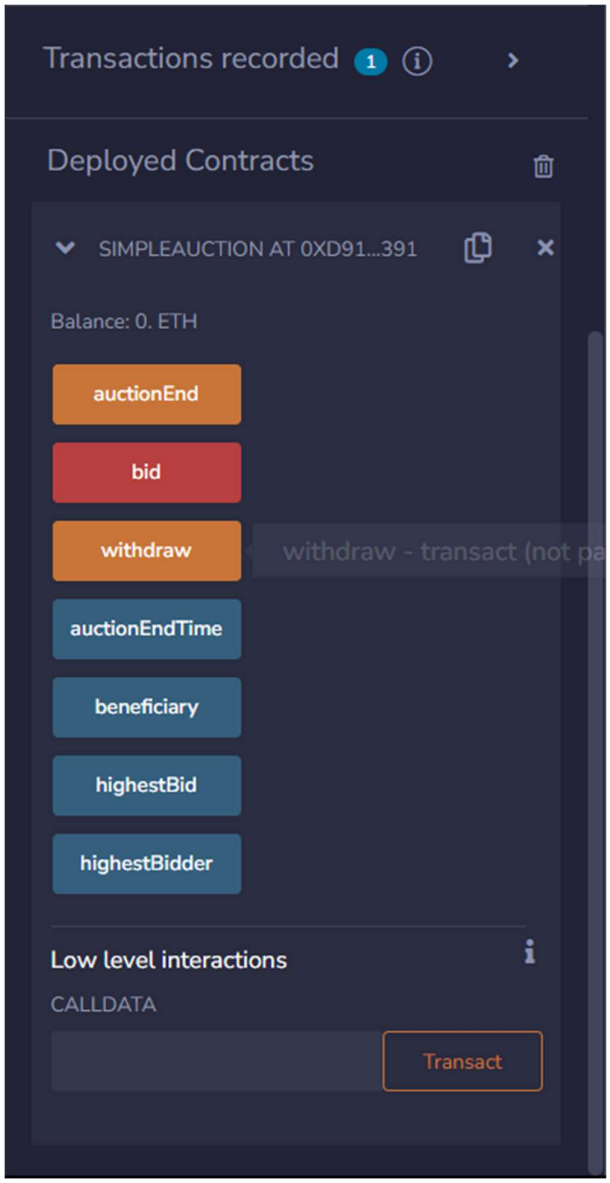
```
// SPDX-License-Identifier: MIT
pragma solidity >=0.6.12 <0.9.0;


contract SimpleAuction {
    // Parameters of the auction. Times are either
    // absolute unix timestamps (seconds since 1970-01-01)
    // or time periods in seconds.
    address payable public beneficiary;
    uint public auctionEndTime;
```

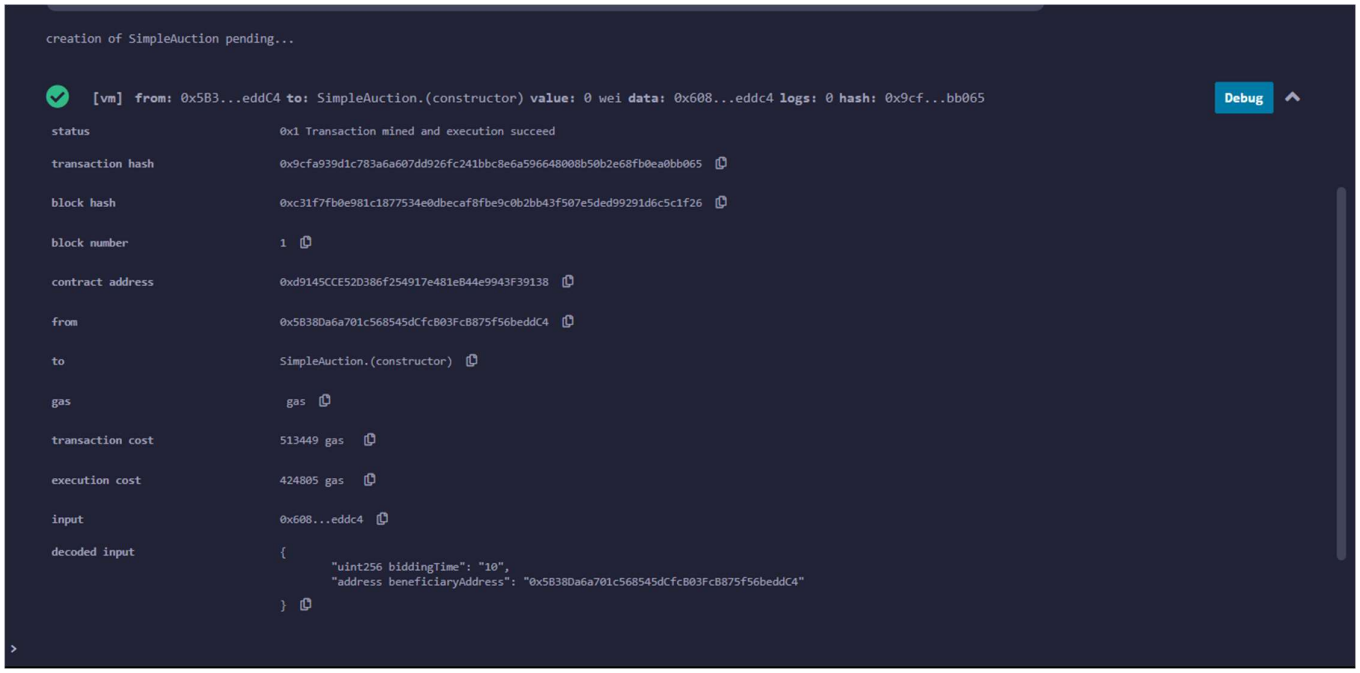
```
    // Current state of the auction.
    address public highestBidder;
    uint public highestBid;


    // Allowed withdrawals of previous bids
    mapping(address => uint) pendingReturns;


    // Set to true at the end, disallows any change.
    // By default initialized to `false`.
    bool ended;


    // Events that will be emitted on changes.
    event HighestBidIncreased(address bidder, uint amount);
    event AuctionEnded(address winner, uint amount);


    // Errors that describe failures.


    // The triple-slash comments are so-called natspec
    // comments. They will be shown when the user
    // is asked to confirm a transaction or
    // when an error is displayed.


    /// The auction has already ended.
    error AuctionAlreadyEnded();
    /// There is already a higher or equal bid.
    error BidNotHighEnough(uint highestBid);
    /// The auction has not ended yet.
    error AuctionNotYetEnded();
    /// The function auctionEnd has already been called.
    error AuctionEndAlreadyCalled();


    /// Create a simple auction with `biddingTime`
    /// seconds bidding time on behalf of the
    /// beneficiary address `beneficiaryAddress`.
    constructor(
        uint biddingTime,
        address payable beneficiaryAddress
    ) {
        beneficiary = beneficiaryAddress;
        auctionEndTime = block.timestamp + biddingTime;
    }


    /// Bid on the auction with the value sent
    /// together with this transaction.
    /// The value will only be refunded if the
    /// auction is not won.
    function bid() external payable {
        // No arguments are necessary, all
```

```
        // information is already part of
        // the transaction. The keyword payable
        // is required for the function to
        // be able to receive Ether.


        // Revert the call if the bidding
        // period is over.
        if (block.timestamp > auctionEndTime)
            revert AuctionAlreadyEnded();


        // If the bid is not higher, send the
        // Ether back (the revert statement
        // will revert all changes in this
        // function execution including
        // it having received the Ether).
        if (msg.value <= highestBid)
            revert BidNotHighEnough(highestBid);


        if (highestBid != 0) {
            // Sending back the Ether by simply using
            // highestBidder.send(highestBid) is a security risk
            // because it could execute an untrusted contract.
            // It is always safer to let the recipients
            // withdraw their Ether themselves.
            pendingReturns[highestBidder] += highestBid;
        }
        highestBidder = msg.sender;
        highestBid = msg.value;
        emit HighestBidIncreased(msg.sender, msg.value);
    }


    /// Withdraw a bid that was overbid.
    function withdraw() external returns (bool) {
        uint amount = pendingReturns[msg.sender];
        if (amount > 0) {
            // It is important to set this to zero because the recipient
            // can call this function again as part of the receiving call
            // before `send` returns.
            pendingReturns[msg.sender] = 0;


            // msg.sender is not of type `address payable` and must be
            // explicitly converted using `payable(msg.sender)` in order
            // use the member function `send()`.
            if (!payable(msg.sender).send(amount)) {
                // No need to call throw here, just reset the amount owing
                pendingReturns[msg.sender] = amount;
                return false;
            }
        }
        return true;
```

```
    }

    /// End the auction and send the highest bid
    /// to the beneficiary.
    function auctionEnd() external {
        // It is a good guideline to structure functions that interact
        // with other contracts (i.e. they call functions or send Ether)
        // into three phases:
        // 1. checking conditions
        // 2. performing actions (potentially changing conditions)
        // 3. interacting with other contracts
        // If these phases are mixed up, the other contract could call
        // back into the current contract and modify the state or cause
        // effects (ether payout) to be performed multiple times.
        // If functions called internally include interaction with external
        // contracts, they also have to be considered interaction with
        // external contracts.

        // 1. Conditions
        if (block.timestamp < auctionEndTime)
            revert AuctionNotYetEnded();
        if (ended)
            revert AuctionEndAlreadyCalled();

        // 2. Effects
        ended = true;
        emit AuctionEnded(highestBidder, highestBid);

        // 3. Interaction
        beneficiary.transfer(highestBid);
    }
}
```


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Remix VM (Shanghai)
VM
ACCOUNT
0x5B3...eddC4 (99.9999999
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
SimpleAuction - contracts/HelloWorld
evm version: shanghai
Deploy
10,0x5B38Da6a701c568545dC
Publish to IPFS
At Address
0x5B38Da6a701c568545dCfcB
Transactions recorded
1
Deployed Contracts
SIMPLEAUCTION AT 0XD91...39138


Transactions recorded 1
Deployed Contracts
SIMPLEAUCTION AT 0XD91...391
Balance: 0. ETH
auctionEnd
bid
withdraw
withdraw - transact (not pa
auctionEndTime
beneficiary
highestBid
highestBidder
Low level interactions
CALLDATA
Transact


creation of SimpleAuction pending...
[vm] from: 0x5B3...eddC4 to: SimpleAuction.(constructor) value: 0 wei data: 0x608...eddc4 logs: 0 hash: 0x9cf...bb065
Debug
status 0x1 Transaction mined and execution succeed
transaction hash 0x9cfa939d1c783a6a607dd926fc241bbc8e6a596648008b50b2e68fb0ea0bb065
block hash 0xc31f7fb0e981c1877534e0dbecaf8fbe9c0b2bb43f507e5ded99291d6c5c1f26
block number 1
contract address 0xd9145CCE52D386f254917e481eB44e9943F39138
from 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to SimpleAuction.(constructor)
gas gas
transaction cost 513449 gas
execution cost 424805 gas
input 0x608...eddc4
decoded input
{
"uint256 biddingTime": "10",
"address beneficiaryAddress": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4"
}

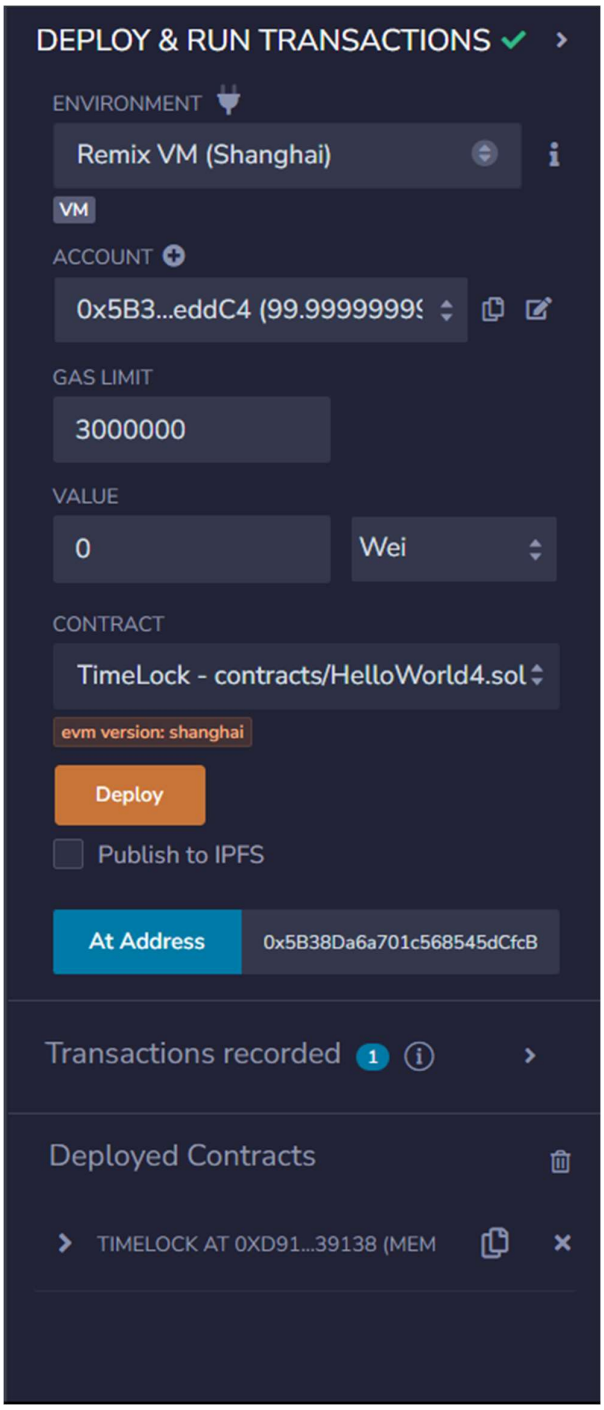
d) Create a time-locked wallet where funds cannot be withdrawn until the specific time.

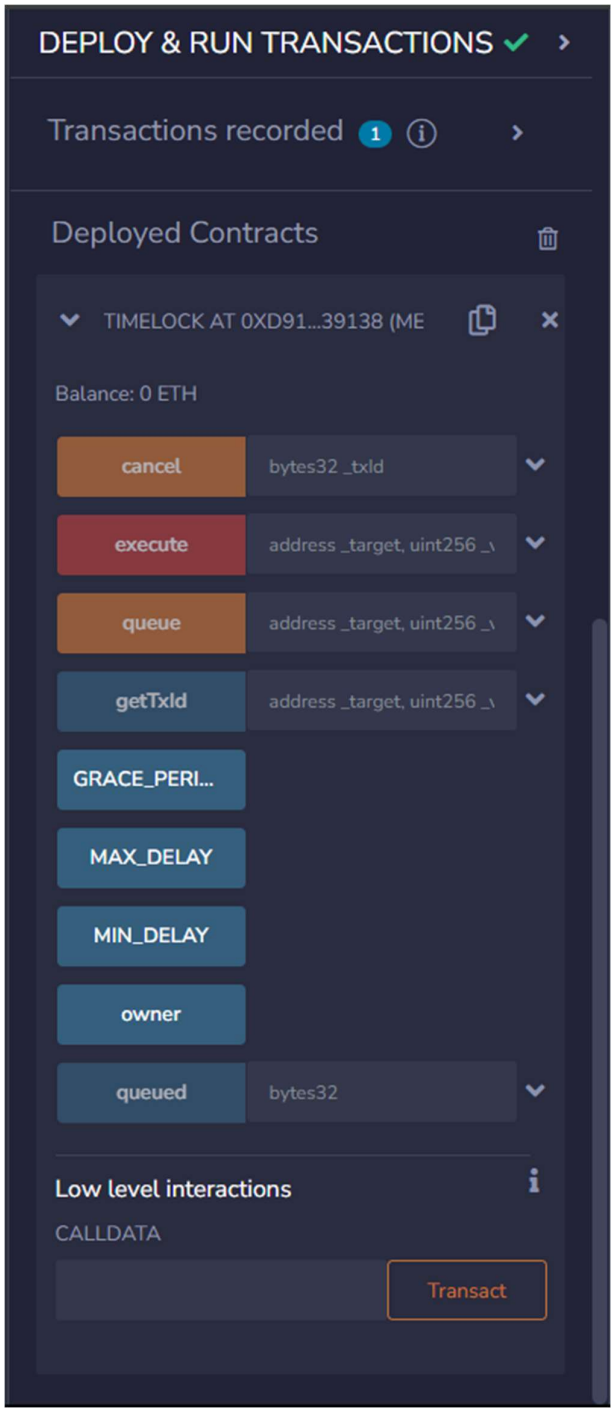
```
// SPDX-License-Identifier: MIT
pragma solidity >=0.6.12 <0.9.0;

contract TimeLock {
    error NotOwnerError();
    error AlreadyQueuedError(bytes32 txId);
    error TimestampNotInRangeError(uint blockTimestamp, uint timestamp);
    error NotQueuedError(bytes32 txId);
    error TimestampNotPassedError(uint blockTimestmap, uint timestamp);
    error TimestampExpiredError(uint blockTimestamp, uint expiresAt);
    error TxFailedError();


    event Queue(
        bytes32 indexed txId,
        address indexed target,
        uint value,
        string func,
        bytes data,
        uint timestamp
    );
    event Execute(
        bytes32 indexed txId,
        address indexed target,
        uint value,
        string func,
        bytes data,
        uint timestamp
    );
    event Cancel(bytes32 indexed txId);


    uint public constant MIN_DELAY = 10; // seconds
    uint public constant MAX_DELAY = 1000; // seconds
    uint public constant GRACE_PERIOD = 1000; // seconds


    address public owner;
    // tx id => queued
    mapping(bytes32 => bool) public queued;


    constructor() {
        owner = msg.sender;
    }


    modifier onlyOwner() {
        if (msg.sender != owner) {
            revert NotOwnerError();
        }
        _;
```

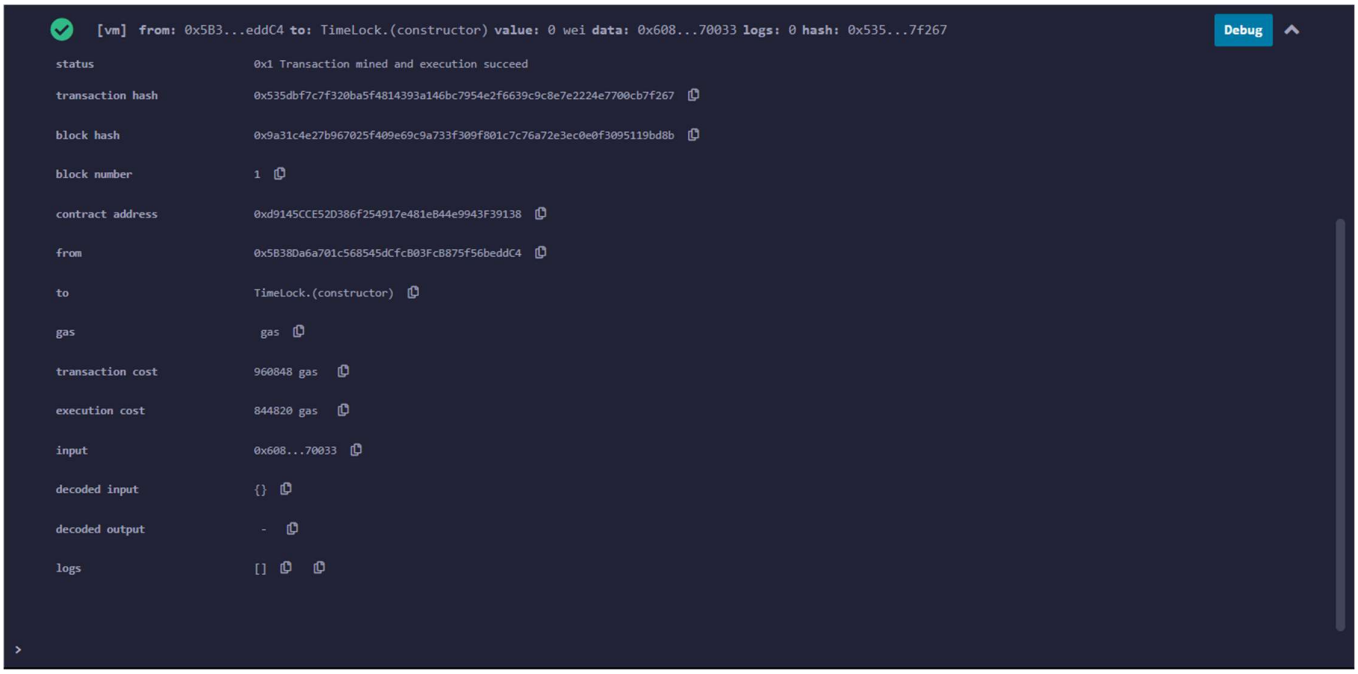
```
    }

    receive() external payable {}

    function getTxId(
        address _target,
        uint _value,
        string calldata _func,
        bytes calldata _data,
        uint _timestamp
    ) public pure returns (bytes32) {
        return keccak256(abi.encode(_target, _value, _func, _data, _timestamp));
    }

    /**
     * @param _target Address of contract or account to call
     * @param _value Amount of ETH to send
     * @param _func Function signature, for example "foo(address,uint256)"
     * @param _data ABI encoded data send.
     * @param _timestamp Timestamp after which the transaction can be executed.
     */
    function queue(
        address _target,
        uint _value,
        string calldata _func,
        bytes calldata _data,
        uint _timestamp
    ) external onlyOwner returns (bytes32 txId) {
        txId = getTxId(_target, _value, _func, _data, _timestamp);
        if (queued[txId]) {
            revert AlreadyQueuedError(txId);
        }
        // ---|------------|---------------|-------
        //  block    block + min     block + max
        if (
            _timestamp < block.timestamp + MIN_DELAY ||
            _timestamp > block.timestamp + MAX_DELAY
        ) {
            revert TimestampNotInRangeError(block.timestamp, _timestamp);
        }

        queued[txId] = true;

        emit Queue(txId, _target, _value, _func, _data, _timestamp);
    }

    function execute(
        address _target,
        uint _value,
        string calldata _func,
```

```
        bytes calldata _data,
        uint _timestamp
    ) external payable onlyOwner returns (bytes memory) {
        bytes32 txId = getTxId(_target, _value, _func, _data, _timestamp);
        if (!queued[txId]) {
            revert NotQueuedError(txId);
        }
        // ----|-------------------|-------
        //  timestamp    timestamp + grace period
        if (block.timestamp < _timestamp) {
            revert TimestampNotPassedError(block.timestamp, _timestamp);
        }
        if (block.timestamp > _timestamp + GRACE_PERIOD) {
            revert TimestampExpiredError(block.timestamp, _timestamp + GRACE_PERIOD);
        }


        queued[txId] = false;


        // prepare data
        bytes memory data;
        if (bytes(_func).length > 0) {
            // data = func selector + _data
            data = abi.encodePacked(bytes4(keccak256(bytes(_func))), _data);
        } else {
            // call fallback with data
            data = _data;
        }


        // call target
        (bool ok, bytes memory res) = _target.call{value: _value}(data);
        if (!ok) {
            revert TxFailedError();
        }


        emit Execute(txId, _target, _value, _func, _data, _timestamp);


        return res;
    }


    function cancel(bytes32 _txId) external onlyOwner {
        if (!queued[_txId]) {
            revert NotQueuedError(_txId);
        }


        queued[_txId] = false;


        emit Cancel(_txId);
    }
```

```
}
```


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Remix VM (Shanghai)
VM
ACCOUNT
0x5B3...eddC4 (99.9999999
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
TimeLock - contracts/HelloWorld4.sol
evm version: shanghai
Deploy
Publish to IPFS
At Address
0x5B38Da6a701c568545dCfcB
Transactions recorded 1
Deployed Contracts
TIMELOCK AT 0XD91...39138 (MEM


DEPLOY & RUN TRANSACTIONS
Transactions recorded 1
Deployed Contracts
TIMELOCK AT 0XD91...39138 (ME
Balance: 0 ETH
cancel
bytes32 _txId
execute
address _target, uint256 _
queue
address _target, uint256 _
getTxId
address _target, uint256 _
GRACE_PERI...
MAX_DELAY
MIN_DELAY
owner
queued
bytes32
Low level interactions
CALLDATA
Transact

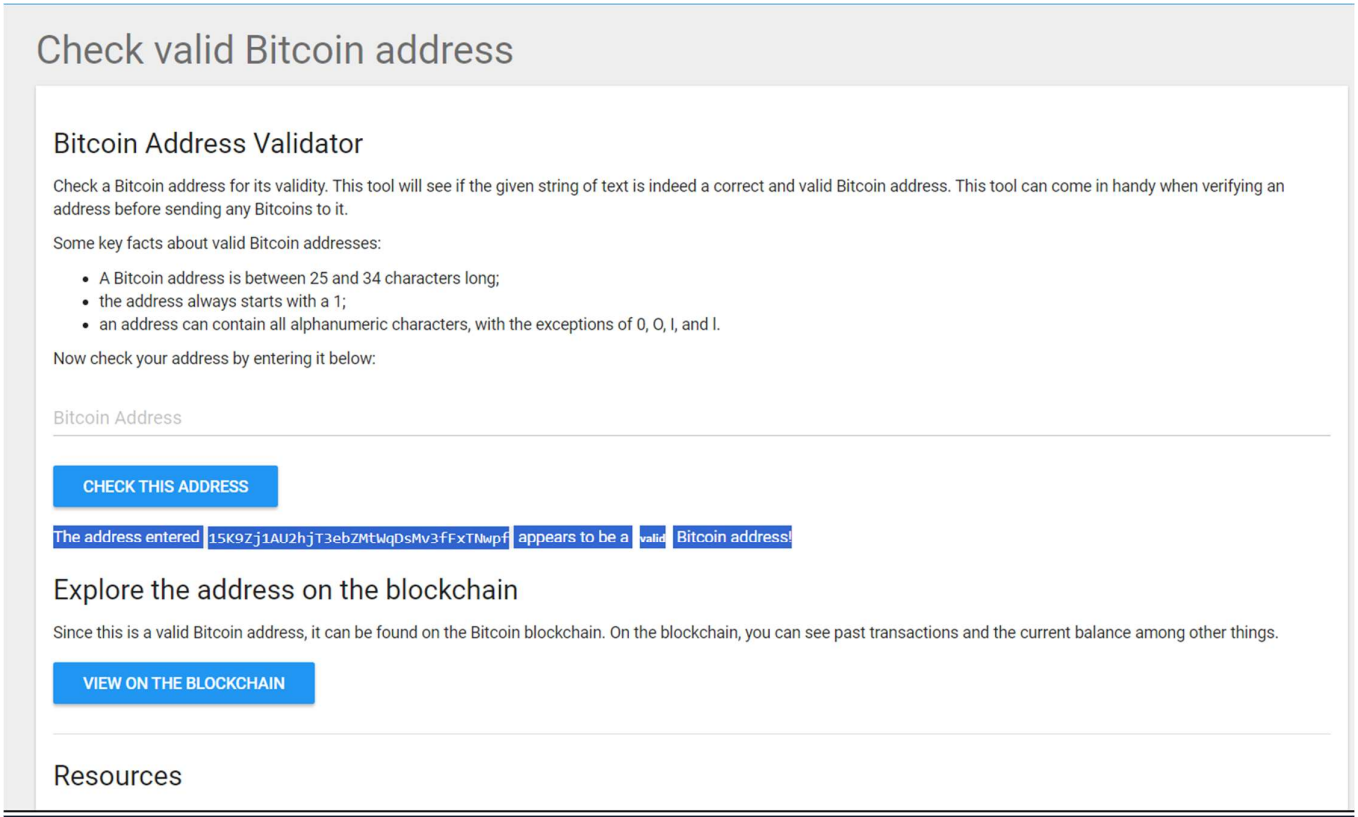


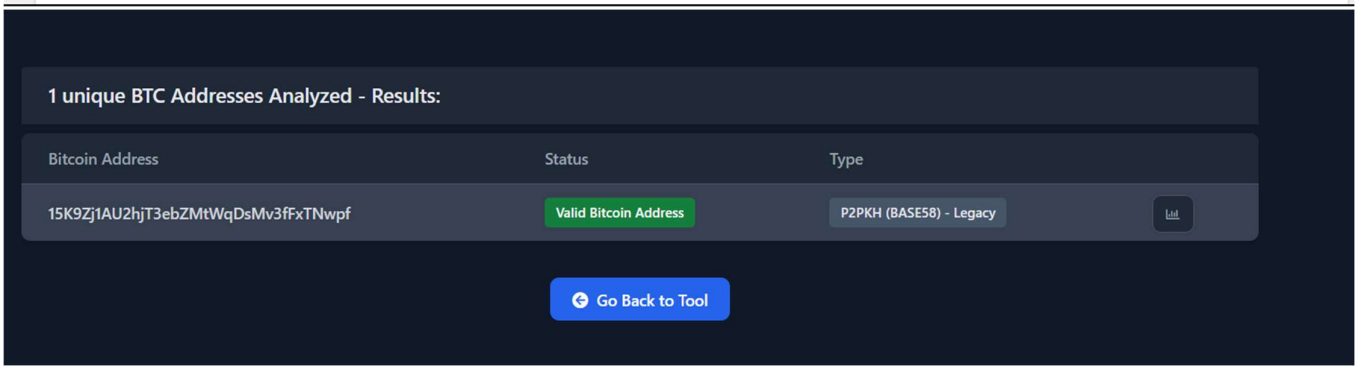
Q.2 As an investigator, you came across a wallet address. Perform a following action for the observed wallet address.

Wallet address: 15K9Zj1AU2hjT3ebZMtWqDsMv3fFxTNwpf

a) Validate the address.

Go to https://thomas.vanhoutte.be/tools/validate-bitcoin-address.php or https://awebanalysis.com/en/bitcoin-address-validate/ and enter the wallet address



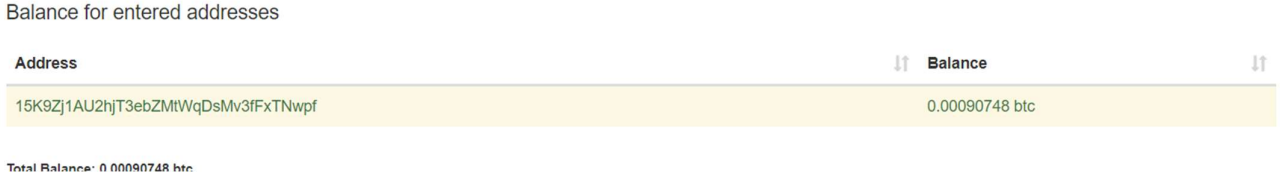


“This is valid Wallet Address

b) Confirm the address balance.

Go to any of this to Confirm

- **https://www.homebitcoin.com/easybalance/**
- **https://www.cointracker.io/wallet/bitcoin**
- **https://bitref.com/**
- **https://www.blockonomics.co/**



Bitcoin Balance

Enter public address to check Bitcoin wallet balance

15K9Zj1AU2hjT3ebZMtWqDsMv3fFxTNwpf

BTC

BTC wallet balance

0.00090748 BTC $41.73

Jan 10, 2024 11:41 AM

Add wallet to CoinTracker

BitRef

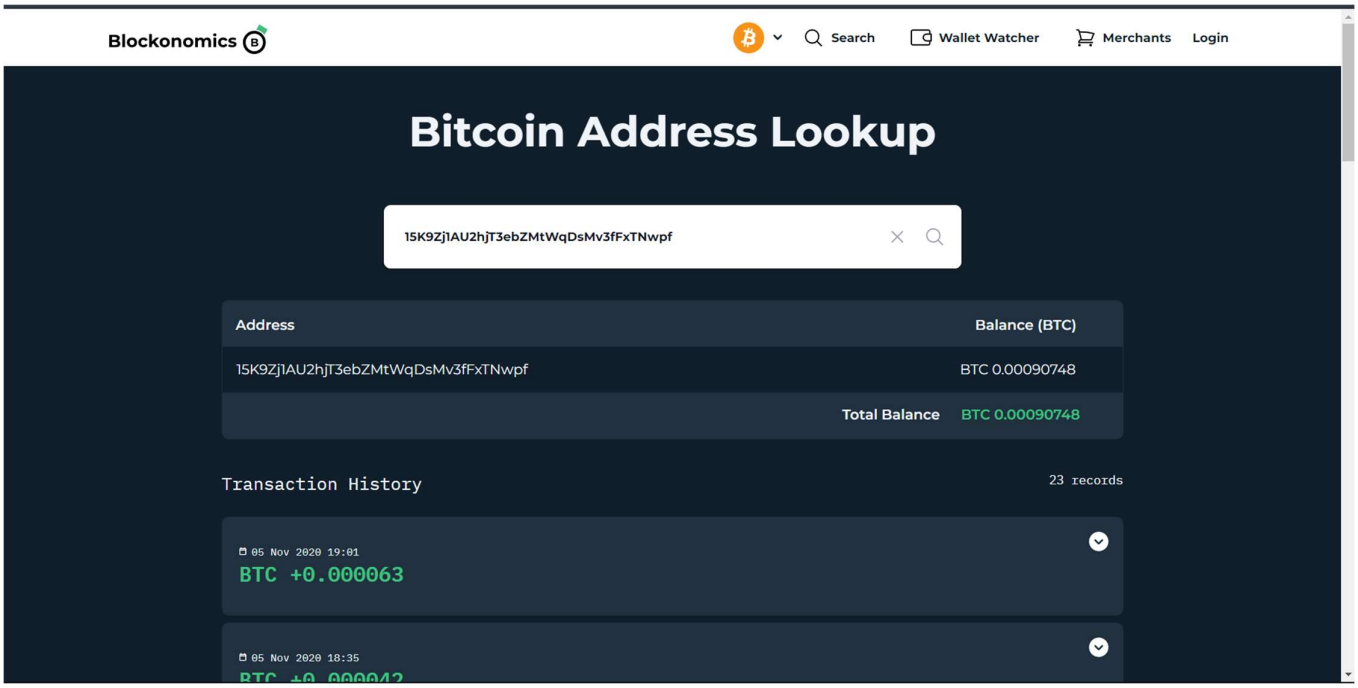
Enter the Bitcoin address here | Check

15K9Zj1AU2hjT3ebZMtWqDsMv3fFxTNwpf

Total Received:	0.09036343
Total Sent:	0.08945595
Final Balance:	0.00090748

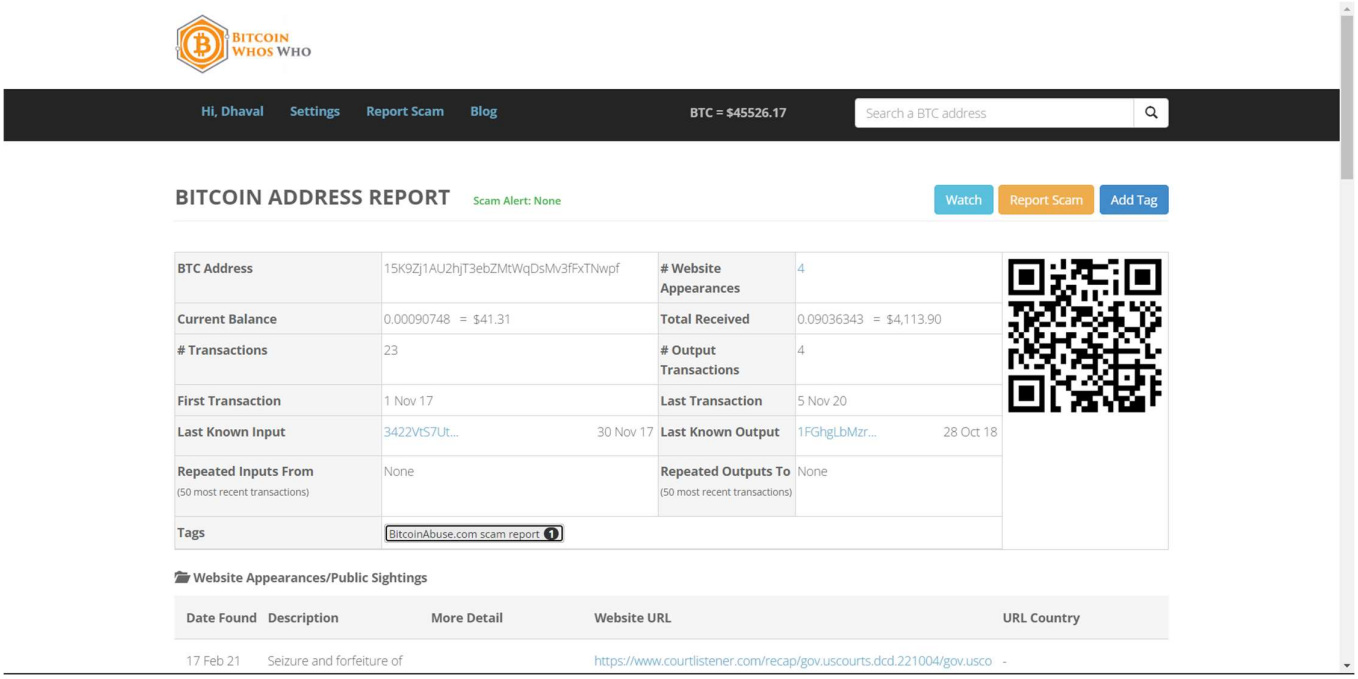
Total transactions: **23**. Most recent:

	Date ▼	Amount	USD
✓	2020-11-05 19:01:21	0.00006300	$2.89
✓	2020-11-05 18:35:02	0.00004200	$1.93
✓	2020-10-28 11:29:48	0.00001000	$0.46



c) Prepare report on Bitcoin address with should include SPAM/SCAM alert, Current balance, I/O transactions. IP address, Web Appearance, etc.

Go to https://bitcoinwhoswho.com



BITCOIN WHOS WHO

Hi, Dhaval | Settings | Report Scam | Blog | BTC = $45526.17 | Search a BTC address

BITCOIN ADDRESS REPORT

Scam Alert: None

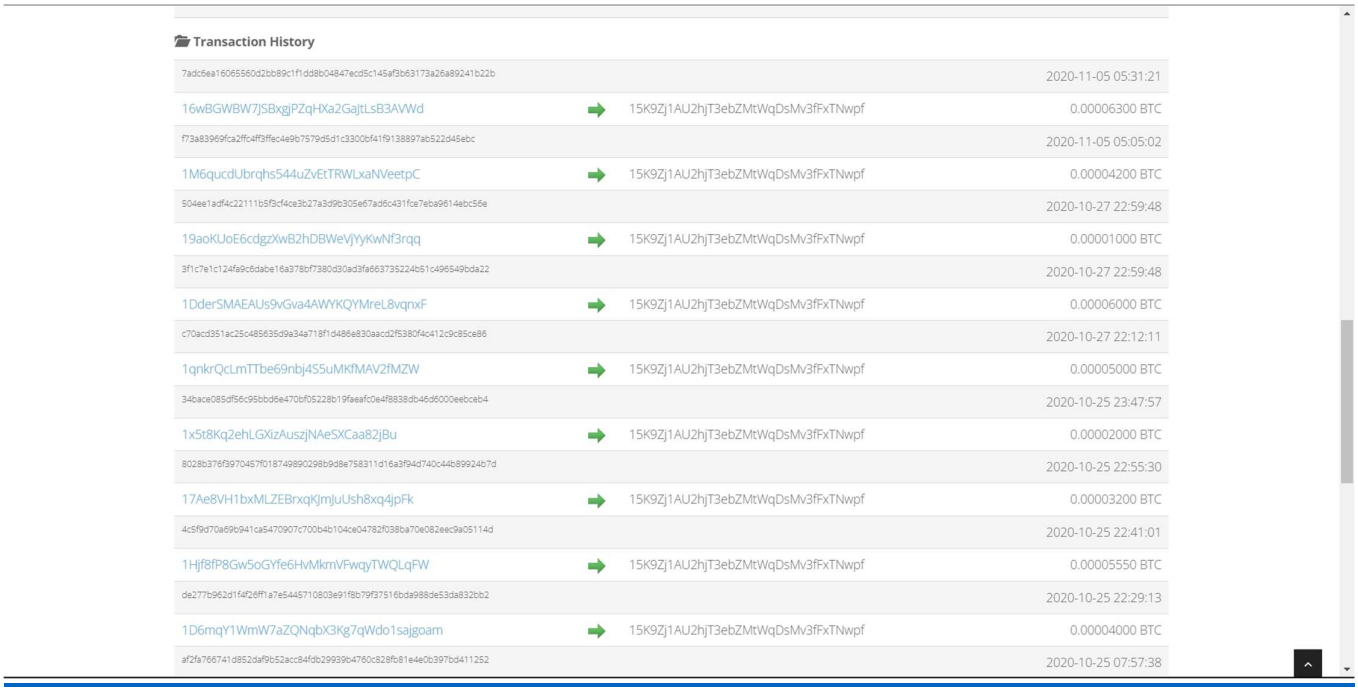
Report Scam | Add Tag

Couldn't read BTC address data from blockchain. Please refresh the page or try again later. You can still report a scam using the button above.

BTC Address	15K9Zj1AU2hjT3ebZMtWqDsMv3fFxTNwpf	**# Website Appearances**	4
Current Balance	= $0	**Total Received**	= $0
# Transactions		**# Output Transactions**	
First Transaction		**Last Transaction**	
Last Known Input	3422VtS7Ut... 30 Nov 17	**Last Known Output**	1FGhgLbMzr... 28 Oct 18
Repeated Inputs From (50 most recent transactions)	None	**Repeated Outputs To** (50 most recent transactions)	None
Tags	BitcoinAbuse.com scam report 1		

Website Appearances/Public Sightings

Date Found	Description	More Detail	Website URL	URL Country
17 Feb 21	Seizure and forfeiture of foreign assets of designated foreign terrorist organizations based in Syria that are linked to al-Qaeda, including the Al-Nusrah Front (â€œANFâ€) and Hayâ€™at Tahrir al-Sham (â€œHTSâ€)		https://www.courtlistener.com/recap/gov.uscourts.dcd.221004/gov.uscourts.dcd.221004.1.0.pdf	-
19 Aug 20	The Taint is Growing: US Prosecutors Attempt to Seize Bitcoin in 150 Addresses Allegedly Tied to Al Qaeda - CoinDesk : Monero	59 votes, 15 comments. 177k members in the Monero community. This is the official subreddit of Monero (XMR), a secure, private, untraceable currency â€¦	https://www.reddit.com/r/Monero/comments/i92eba/the_taint_is_growing_us_prosecutors_attempt_to/	Loading...
27 May 19	Bitcoin Abuse Database: 15K9Zj1AU2hjT3ebZMtWqDsMv3fFxTNwpf		https://www.bitcoinabuse.com/reports/15K9Zj1AU2hjT3ebZMtWqDsMv3fFxTNwpf	Loading...
20 Jun 18	Tracing a Jihadi cell, kidnappers and a scammer to one address using the blockchain	These finds were made by fusing OSINT (open-source intelligence) methods with cryptocurrency and blockchain knowledge. All of which I have been able to self-learn online	https://medium.com/@benjaminstrick/tracing-syrian-cell-kidnappers-scammers-finances-through-blockchain-e9c52fb6127d	United States



d) In the prepared report, also include identified transactions, identified blocks, Mind blocks and exchange name related detail.

- GO to **https://hashxp.org/**

Address analysis

Some information derived from latest transactions on the blockchain where this address has been involved.

Received TX	19 (19 outputs)
Sent TX	4 (4 inputs)
Transactions	23
UTXO:s	15
Last Sent	2018-10-28 17:08
Last Received	2020-11-05 13:31
Consumed Rate	98%
Received	9 036 343sat
Sent	8 945 595sat
Balance	90 748sat
Address Type	P2PKH
Multi-Sig	
Bitcoin Abuse Database	1 reports

Bitcoin Abuse Database Entries

Bitcoin Abuse Database Entries

Date	Abuser	Description
2019-05-27 18:16	other:Terrorism Funding:Tracing a Jihadi cell, kidnappers and a scammer to one address using the blockchain	These finds were made by fusing OSINT (open-source intelligence) methods with cryptocurrency and blockchain knowledge. All of which I have been able to self-learn online. https://medium.com/@benjaminstrick/tracing-syrian-cell-kidnappers-scammers-finances-through-blockchain-e9c52fb6127d https://www.memri.org/reports/imminent-release-telegrams-cryptocurrency-isiss-encryption-app-choice-%E2%80%93-international

Transactions

The latest transactions on the bitcoin blockchain where this address has been sender or receiver.

Block	Sent	TX	Received	Net
655535		1 2	6 300sat $0.89	6 300sat $0.89
655530		2 2	4 200sat $0.59	4 200sat $0.59
654530		1 2	1 000sat $0.14	1 000sat $0.14
654530		1 2	6 000sat $0.82	6 000sat $0.82
654524		1 2	5 000sat $0.68	5 000sat $0.68
654332		1 2	2 000sat $0.26	2 000sat $0.26

21 Most common senders

The addresses on the left side of the transactions, where the current address is on the right as a receiver. listed together with a count of how many times they have occured in the latest transactions.

- 1M6qucdUbrqhs544uZvEtTRWLxaNVeetpC 2
- 16wBGWBW7JSBxgjPZqHXa2GaJtLsB3AVWd 1
- 19aoKUoE6cdgzXwB2hDBWeVjYyKwNf3rqq 1
- 1DderSMAEAUs9vGva4AWYKQYMreL8vqnxF 1
- 1qnkrQcLmTTbe69nbj4S5uMKfMAV2fMZW 1
- 1x5t8Kq2ehLGXizAuszjNAeSXCaa82jBu 1
- 17Ae8VH1bxMLZEBrxqKJmJuUsh8xq4jpFk 1
- 1Hjf8fP8Gw5oGYfe6HvMkmVFwqyTWQLqFW 1
- 1D6mqY1WmW7aZQNqbX3Kg7qWdo1sajgoam 1
- 16eBuMBv92erEsGTXzokVyBZrt7PfqvMeP 1

4 Most common receivers

The addresses on the right side of the transactions, where the current address is on the left as a sender, listed together with a count of how many times they have occured in the latest transactions.

- 1FGhgLbMzrUV5mgwX9nkEeqHbKbUK29nbQ 1
- 19ckaZLPDzn7s2EcFtsjLobu71eAqWFDD8 1
- 1H9zEVipYuFkB1ZXfNqFF1JtJKpn6vfhdk 1
- 16YHAdaD589hV4nLDVUTEFcgaDpvTncQg2 1

1 Most common sending siblings

The addresses this address is listed together with as a sender, together with a count of how many times they have occured in the latest transactions.

1BQAPykulZibWGAgd8QePpW1vAKHowqLez 1

16 Most common receiving siblings

The addresses this address is listed together with as a receiver, together with a count of how many times these combinations have occured in the latest transactions.

1C87H84PsYMfM3Bn9pyr9Dw2ATEkbQHUro 1

16wBGWBW7JSBxgjPZqHXa2GaJtLsB3AVWd 1

113LbBc4gA922h8Kx8wDkUWtuGrpfsU7rB 1

19aoKUoE6cdgzXwB2hDBWeVjYyKwNf3rqq 1

12YD3LFRLRcFWbjRuBN4mGx6JDDDBVRssj 1

1DderSMAEAUs9vGva4AWYKQYMreL8vqnxF 1

1x5t8Kq2ehLGXizAuszjNAeSXCaa82jBu 1

17Ae8VH1bxMLZEBrxqKJmJuUsh8xq4jpFk 1

1Hjf8fP8Gw5oGYfe6HvMkmVFwqyTWQLqFW 1

16r9ZrPhSpYxAmhaiev7ry2nB2RyqFyQrz 1

Output, input and witness scripts

Bitcoin transactions illustrating the scripts used by this address

Output

```
0000: 19 .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..   script len: 25
0000: .. 76 .. .. .. .. .. .. .. .. .. .. .. .. .. ..   OP_DUP
0000: .. .. a9 .. .. .. .. .. .. .. .. .. .. .. .. ..   OP_HASH160
0000: .. .. .. 14 2f 4f 23 b5 9c 18 86 6c 9d e8 1b b1   OP_PUSHBYTES_20: 2f4f23b59c18866c9de81bb1dc7d144725a3dcda
0010: dc 7d 14 47 25 a3 dc da .. .. .. .. .. .. .. ..
0010: .. .. .. .. .. .. .. .. 88 .. .. .. .. .. .. ..   OP_EQUALVERIFY
0010: .. .. .. .. .. .. .. .. .. ac .. .. .. .. .. ..   OP_CHECKSIG
```

Input

```
0000: 6b .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..   script len: 107
0000: .. 48 30 45 02 21 00 aa bf cc 0e 4b fc cf 8e 54   OP_PUSHBYTES_72: 3045022100aabfcc0e4bfccf8e546421c5054364fa3ce9bbe1e1f5568513c378f81451e554022
0010: 64 21 c5 05 43 64 fa 3c e9 bb e1 e1 f5 56 85 13   07853984c7cecf5af812dabc63df99bb4c4419d1c188708711b5c6428d1b43fc601
0020: c3 78 f8 14 51 e5 54 02 20 78 53 98 4c 7c ec f5
0030: af 81 2d ab c6 3d f9 9b b4 c4 41 9d 1c 18 87 08
0040: 71 1b 5c 64 28 d1 b4 3f c6 01 .. .. .. .. .. ..
0040: .. .. .. .. .. .. .. .. .. 21 03 06 81 90 49   OP_PUSHBYTES_33: 0306819049e31a96f4bd8612f33261f2fff3d4c3a63df84a1cb5b82e1acb54b052
```


Blockchain Info

Latest block #825101

Latest addresses and transactions investigated:

400b8325337e1d89df00a57f58f47b0cbc5a908178ef1ecf5ceb4552a55680bf

b8e100c944f5be5e38ace8af2c7bb72c8bbd06ba3214e4440dd8ac7e1f577a82

3PXPor9HStdA2XgUidKNDjDjaycjtWZSoP

3FcSkPoJiQbrCySeD386Zy7NcexjhEmfYj

17ucy1K9ZUAaoY6JVtM932W9jUp5LXfyHa

787205

1bdcbe9a42e6a5a1d6587ff07f80221728a05588ab8c5f0503996e96ea7fbce0

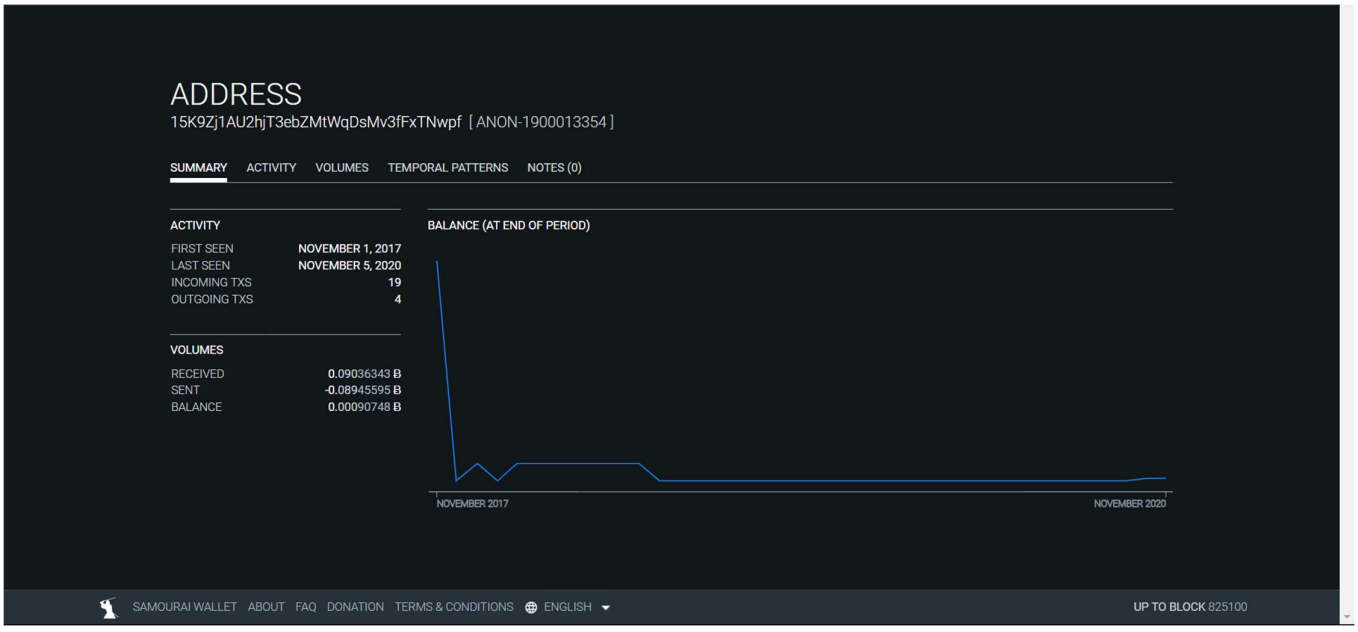
bc1qw56sv938zfrdzdf9k3mdvr5rcq0e4splkhht9e

198r6QVm5C8D4Jz6BqnHnknySuJCxm6doe

630295

Go to Wallet explorer

Check on otx.me



ADDRESS

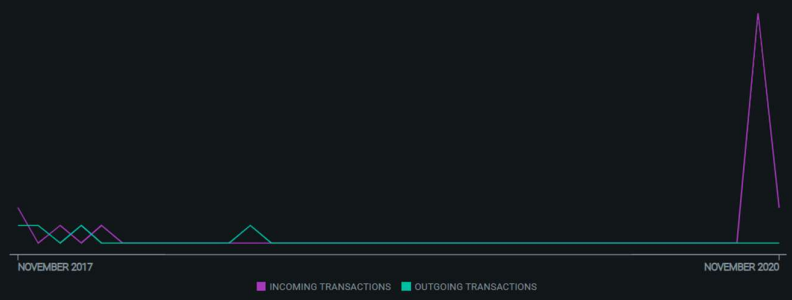
15K9Zj1AU2hjT3ebZMtWqDsMv3fFxTNwpf [ANON-1900013354]

SUMMARY | **ACTIVITY** | VOLUMES | TEMPORAL PATTERNS | NOTES (0)

ACTIVITY

FIRST SEEN	**NOVEMBER 1, 2017**
LAST SEEN	**NOVEMBER 5, 2020**
INCOMING TXS	**19**
OUTGOING TXS	**4**

TRANSACTIONS (DURING PERIOD)



SAMOURAI WALLET ABOUT FAQ DONATION TERMS & CONDITIONS ENGLISH

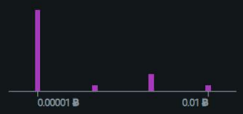
UP TO BLOCK 825100

ADDRESS

15K9Zj1AU2hjT3ebZMtWqDsMv3fFxTNwpf [ANON-1900013354]

SUMMARY | ACTIVITY | **VOLUMES** | TEMPORAL PATTERNS | NOTES (0)

VOLUMES RECEIVED



VOLUMES SENT

VOLUMES (DURING PERIOD)

SAMOURAI WALLET ABOUT FAQ DONATION TERMS & CONDITIONS ENGLISH

UP TO BLOCK 825100

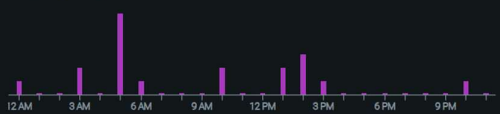
ADDRESS

15K9Zj1AU2hjT3ebZMtWqDsMv3fFxTNwpf [ANON-1900013354]

SUMMARY | ACTIVITY | VOLUMES | **TEMPORAL PATTERNS** | NOTES (0)

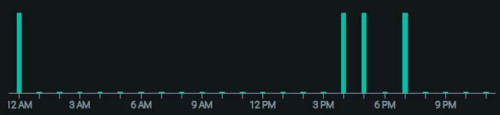
INCOMING TXS - DAY OF THE WEEK

INCOMING TXS - HOUR



OUTGOING TXS - DAY OF THE WEEK

OUTGOING TXS - HOUR



SAMOURAI WALLET ABOUT FAQ DONATION TERMS & CONDITIONS ENGLISH

UP TO BLOCK 825100