Blockchain Lab

Assignment-3

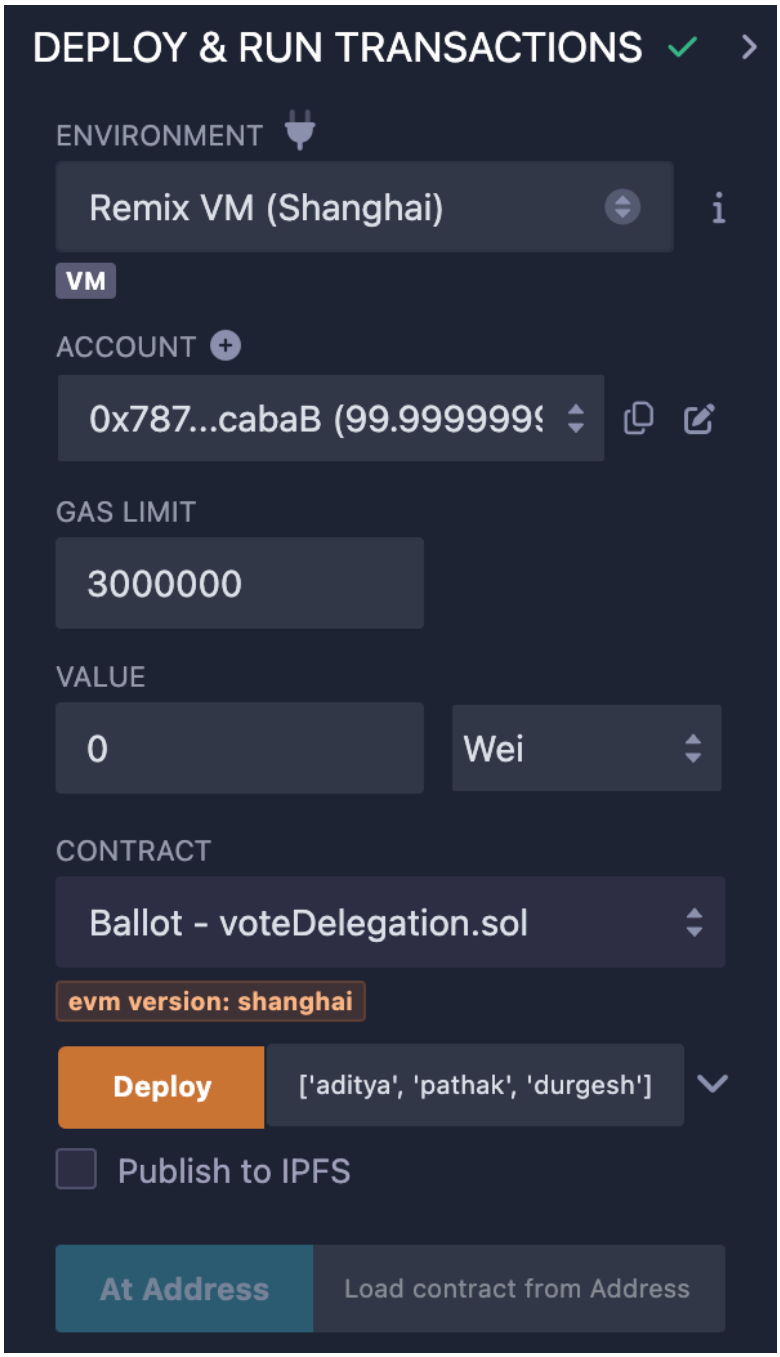
Name: Aditya Pathak
Subgroup: 3NC2
Roll Number: 102115044

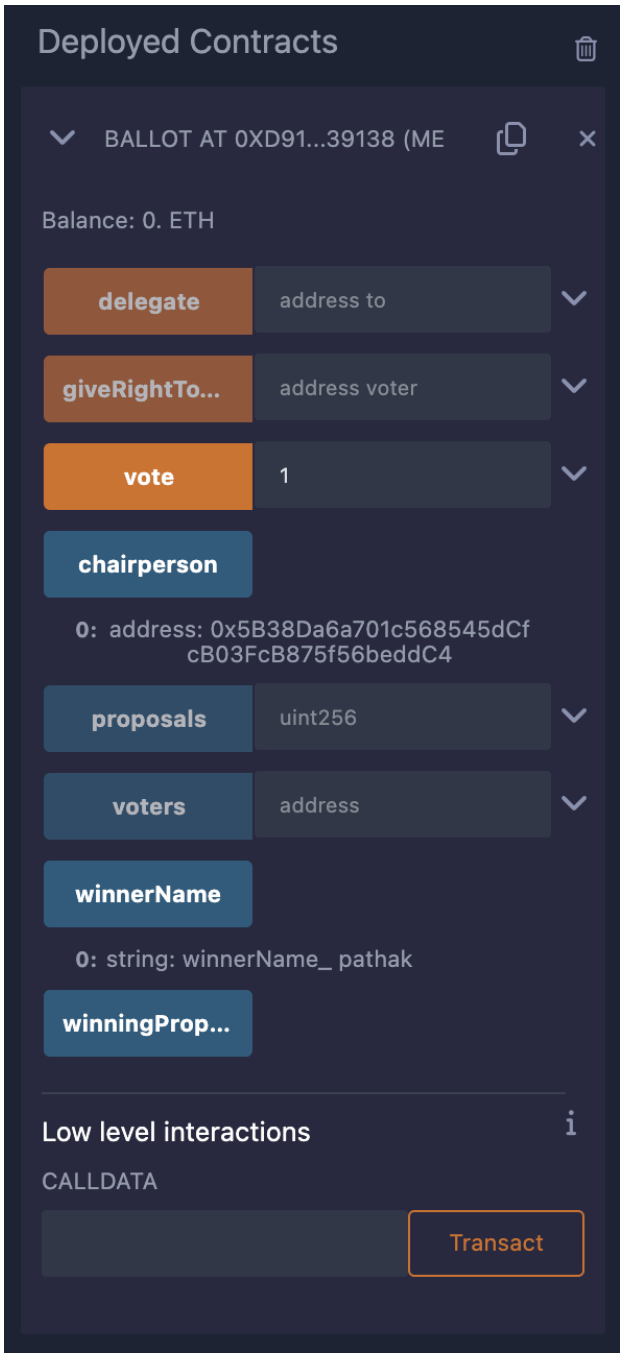
```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

// Aditya Pathak | 102115044 | 3NC2
// Blockchain Assignment 3

contract Ballot {
    struct Voter {
        uint weight;
        bool voted;
        address delegate;
        uint vote;
    }

    struct Proposal {
        string name;
        uint voteCount;
    }

    address public chairperson;
    mapping(address => Voter) public voters;
    Proposal[] public proposals;

    constructor(string[] memory proposalNames) {    infinite gas 895200 gas
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

        for (uint i = 0; i < proposalNames.length; i++) {
            proposals.push(Proposal({
                name: proposalNames[i],
                voteCount: 0
            }));
        }
    }

    function giveRightToVote(address voter) public {    29325 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[voter].voted, "The voter already voted.");
        require(voters[voter].weight == 0);
        voters[voter].weight = 1;
    }

    function delegate(address to) public {    infinite gas
        Voter storage sender = voters[msg.sender];
        require(!sender.voted, "You already voted.");
        require(to != msg.sender, "Self-delegation is disallowed.");

        while (voters[to].delegate != address(0)) {
```


Code (I)

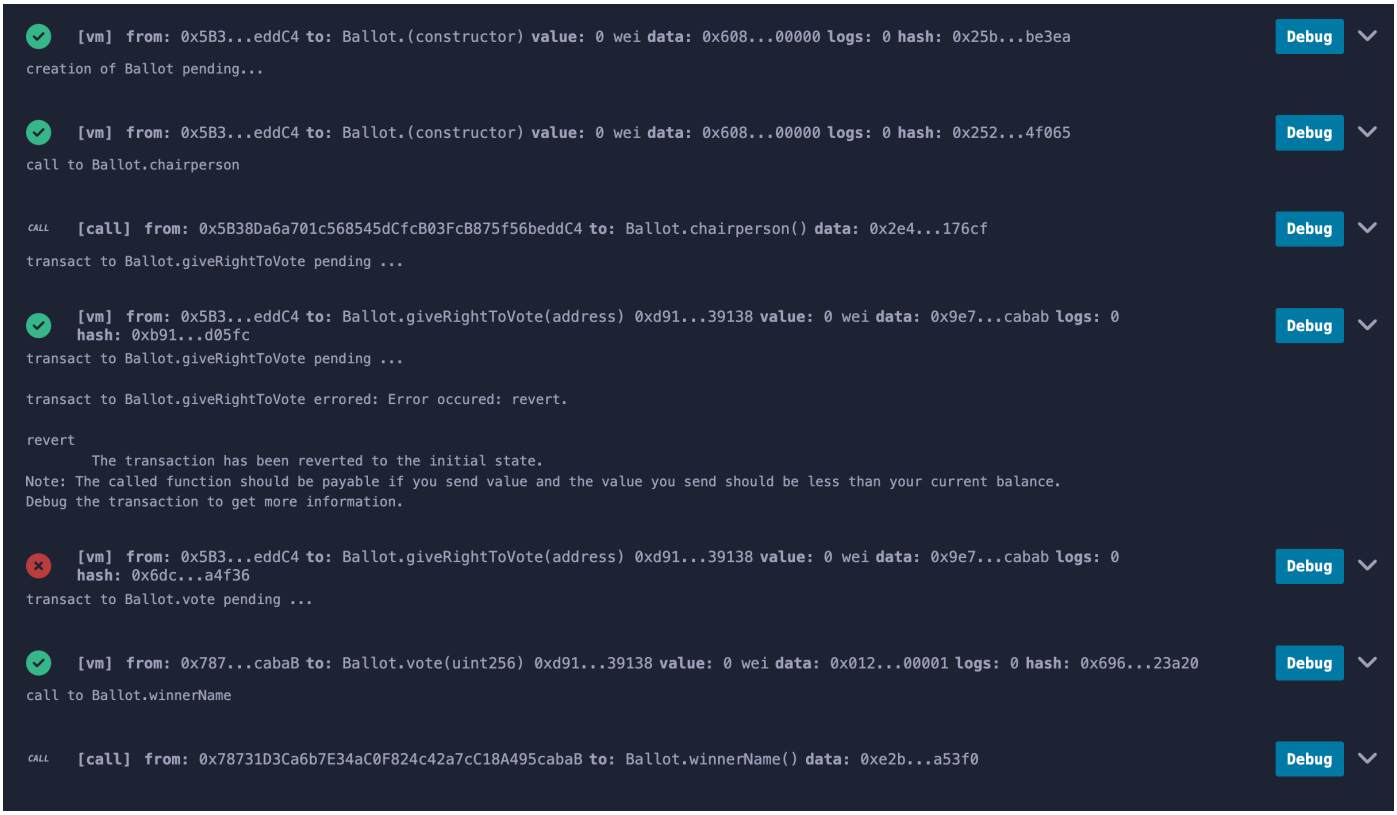
```
47
48          while (voters[to].delegate != address(0)) {
49              to = voters[to].delegate;
50              require(to != msg.sender, "Found loop in delegation.");
51          }
52          sender.voted = true;
53          sender.delegate = to;
54          Voter storage delegate_ = voters[to];
55          if (delegate_.voted) {
56              proposals[delegate_.vote].voteCount += sender.weight;
57          } else {
58              delegate_.weight += sender.weight;
59          }
60      }
61
62      function vote(uint proposal) public {    infinite gas
63          Voter storage sender = voters[msg.sender];
64          require(sender.weight != 0, "Has no right to vote");
65          require(!sender.voted, "Already voted.");
66          sender.voted = true;
67          sender.vote = proposal;
68          proposals[proposal].voteCount += sender.weight;
69      }
70
71      function winningProposal() public view    infinite gas
72              returns (uint winningProposal_)
73      {
74          uint winningVoteCount = 0;
75          for (uint p = 0; p < proposals.length; p++) {
76              if (proposals[p].voteCount > winningVoteCount) {
77                  winningVoteCount = proposals[p].voteCount;
78                  winningProposal_ = p;
79              }
80          }
81      }
82
83      function winnerName() public view    infinite gas
84              returns (string memory winnerName_)
85      {
86          winnerName_ = proposals[winningProposal()].name;
87      }
88  }
```

Code (II)


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Remix VM (Shanghai)
VM
ACCOUNT
0x787...cabaB (99.999999
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
Ballot - voteDelegation.sol
evm version: shanghai
Deploy
['aditya', 'pathak', 'durgesh']
Publish to IPFS
At Address
Load contract from Address


Deployed Contracts
BALLOT AT 0XD91...39138 (ME
Balance: 0. ETH
delegate
address to
giveRightTo...
address voter
vote
1
chairperson
0: address: 0x5B38Da6a701c568545dCf
cB03FcB875f56beddC4
proposals
uint256
voters
address
winnerName
0: string: winnerName_ pathak
winningProp...
Low level interactions
CALLDATA
Transact

Deployment



Transactions on Terminal