Shakeib Shaida shakeib@xord.com https://twitter.com/shakeib98

ZKU Background Assignment

A. Conceptual Knowledge

1. ***What is a smart contract? How are they deployed? You should be able to describe how a smart contract is deployed and the necessary steps.***
    a. Smart contracts are a set of instructions that is not only stored in one computer, but a set of computers. These sets of instructions can be anything. To store how many candies have you bought, how much money you have in your piggy bank and many more things.
    b. In order to write these sets of instructions you need a language. For smart contracts Solidity is used to write the instruction. This piece of instruction is called code.
    c. After writing the code there's an intermediary check, like a security check that all the code is correct and there is no mistake in the set of instructions. This process is called compilation. And the security guard who does the work is known as compiler.
    d. After all the things are done, the compiled code is sent to a wizardly place where once things are stored they can never be deleted. This place is known as blockchain and you can send any instructions over it.
    e. This wizardly place aka blockchain is a set of computers which are running to ensure that your piece of code does not get changed after getting stored.
2. ***What is gas? Why is gas optimization such a big focus when building smart contracts?***
    a. Whenever we want to travel from point A to point B in car, we need gasoline in our car. Remember we discussed smart contracts and how they are deployed. If you want to add data into the blockchain (that spooky network), you have to have some gas for it. Other than storing the code, if you want to add data such as you want to increase the count of candies eaten today by 1, this action also requires gas.
    b. Even in the real world, we have cars that are old and consume more gas and we have new and optimized cars which consume very less gas and give a good average. The same is with the writing of smart contracts. Your set of instructions should be well defined and optimized so that upon executing them, a minimal amount of gas is used. Imagine if you had to pay $1 worth of gas to update your candy count which also cost $1. That is not efficient.
3. ***What is a hash? Why do people use hashing to hide information?***
    a. Hash is a long fixed length random mumble jumble word. Like this "6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b". This is equivalent to "1". Hashing is useful for large pieces of information and data. Because it doesn't matter upon the size of data, the hash word or string will

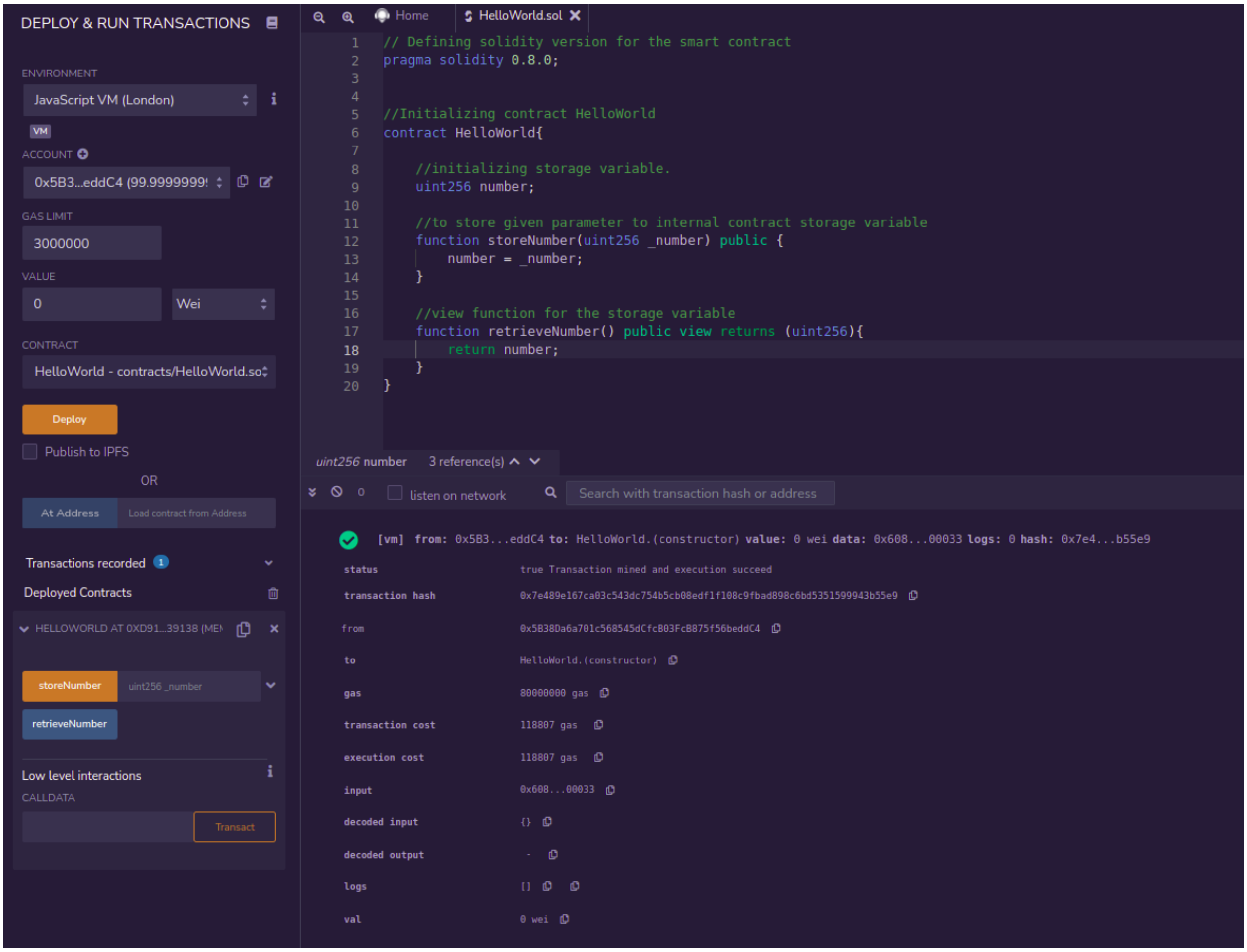
be of the same length. If you change the information even from lower case to upper case, the hash of the data will change.

- b. It also provides security to your data. Suppose you don’t want to store the names of the candies in the smart contract, as everyone will know then. You will make hashes of the name and store the information.
- c. Hashing is a very core component of the spooky network aka blockchain.

4. ***How would you prove to a colorblind person that two different colored objects are actually of different colors?***
    - a. Let’s consider two actors, Alice and Bob. Alice wants to prove to Bob that she holds two different colored balls. Bob is color blind. Now this is how the interaction between both of them will go through
        - i. Bob will take both balls and will switch the balls and give them back to Alice.
        - ii. Bob will ask Alice if the balls are switched or not?
        - iii. Alice can see colors, so if the balls are switched she will respond positively.
        - iv. This interaction repeats multiple times so that Bob trusts Alice that she isn’t lying at all.
        - v. If the balls are not switched and Alice lies that they are switched, Bob will ultimately know that Alice is lying.

B. You sure you're solid with Solidity?

1. Github URL (https://github.com/shakeib98/zku-assignment/blob/main/HelloWorld.sol)



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: JavaScript VM (London)

ACCOUNT: 0x5B3...eddC4 (99.9999999!

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: HelloWorld - contracts/HelloWorld.so

Transactions recorded 1

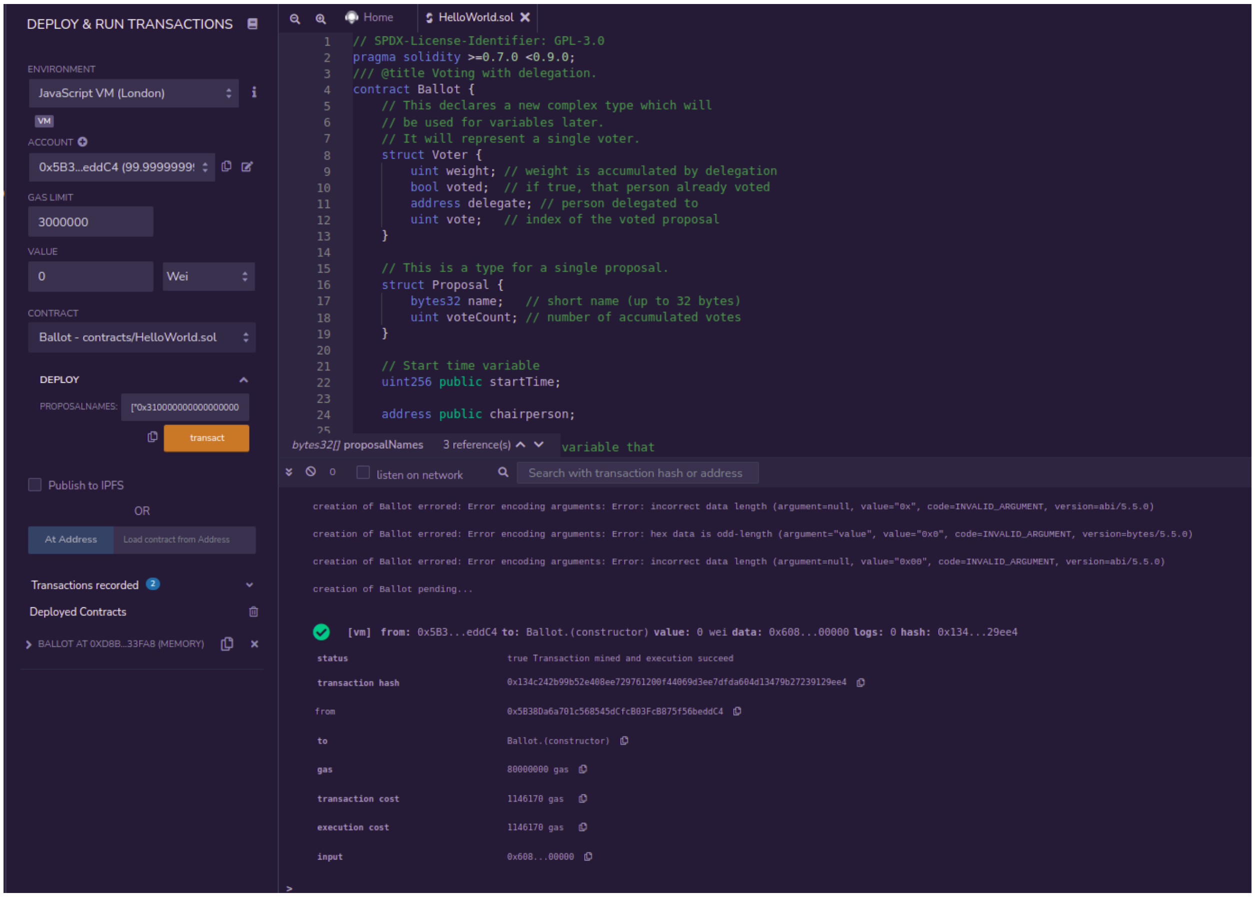
Deployed Contracts

HELLOWORLD AT 0XD91...39138 (MEI

```
// Defining solidity version for the smart contract
pragma solidity 0.8.0;


//Initializing contract HelloWorld
contract HelloWorld{

    //initializing storage variable.
    uint256 number;

    //to store given parameter to internal contract storage variable
    function storeNumber(uint256 _number) public {
        number = _number;
    }

    //view function for the storage variable
    function retrieveNumber() public view returns (uint256){
        return number;
    }
}
```

```
[vm] from: 0x5B3...eddC4 to: HelloWorld.(constructor) value: 0 wei data: 0x608...00033 logs: 0 hash: 0x7e4...b55e9
status              true Transaction mined and execution succeed
transaction hash    0x7e489e167ca03c543dc754b5cb08edf1f108c9fbad898c6bd5351599943b55e9
from                0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                  HelloWorld.(constructor)
gas                 80000000 gas
transaction cost    118807 gas
execution cost      118807 gas
input               0x608...00033
decoded input       {}
decoded output       -
logs                []
val                 0 wei
```

2. Github URL (https://github.com/shakeib98/zku-assignment/blob/main/BallotPaper.sol)



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: JavaScript VM (London)

VM

ACCOUNT: 0x5B3...eddC4 (99.9999999!

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: Ballot - contracts/HelloWorld.sol

DEPLOY

PROPOSALNAMES: ["0x310000000000000000

transact

□ Publish to IPFS

OR

At Address | Load contract from Address

Transactions recorded 2

Deployed Contracts

BALLOT AT 0XD8B...33FA8 (MEMORY)

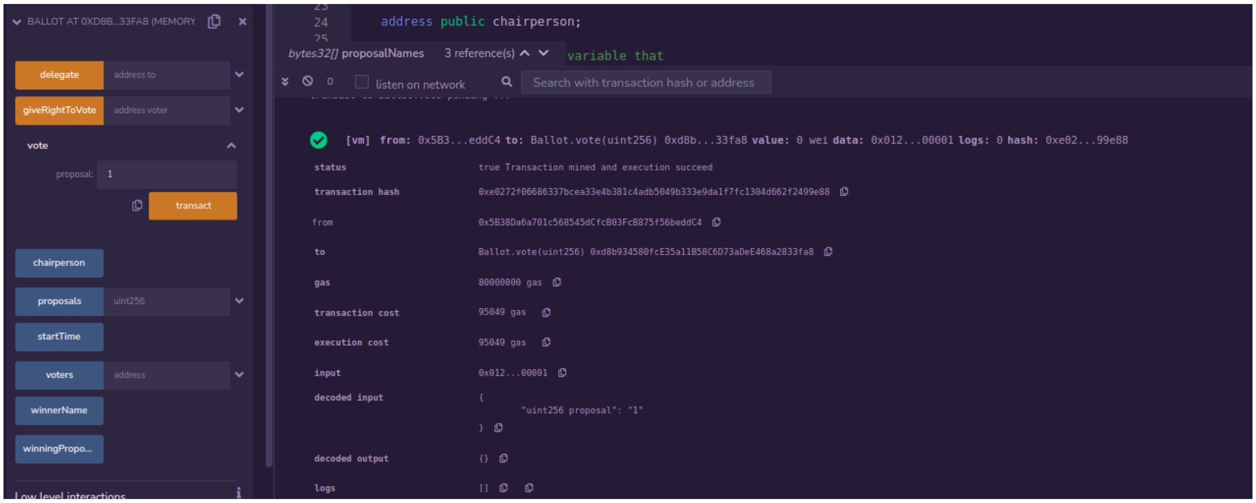
Home | HelloWorld.sol

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;
/// @title Voting with delegation.
contract Ballot {
    // This declares a new complex type which will
    // be used for variables later.
    // It will represent a single voter.
    struct Voter {
        uint weight; // weight is accumulated by delegation
        bool voted;  // if true, that person already voted
        address delegate; // person delegated to
        uint vote;   // index of the voted proposal
    }

    // This is a type for a single proposal.
    struct Proposal {
        bytes32 name;   // short name (up to 32 bytes)
        uint voteCount; // number of accumulated votes
    }

    // Start time variable
    uint256 public startTime;

    address public chairperson;
```

bytes32[] proposalNames 3 reference(s) variable that

□ listen on network | Search with transaction hash or address

```
creation of Ballot errored: Error encoding arguments: Error: incorrect data length (argument=null, value="0x", code=INVALID_ARGUMENT, version=abi/5.5.0)

creation of Ballot errored: Error encoding arguments: Error: hex data is odd-length (argument="value", value="0x0", code=INVALID_ARGUMENT, version=bytes/5.5.0)

creation of Ballot errored: Error encoding arguments: Error: incorrect data length (argument=null, value="0x00", code=INVALID_ARGUMENT, version=abi/5.5.0)

creation of Ballot pending...

✓ [vm] from: 0x5B3...eddC4 to: Ballot.(constructor) value: 0 wei data: 0x608...00000 logs: 0 hash: 0x134...29ee4
status                  true Transaction mined and execution succeed
transaction hash        0x134c242b99b52e408ee729761200f44069d3ee7dfda604d13479b27239129ee4
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      Ballot.(constructor)
gas                     80000000 gas
transaction cost        1146170 gas
execution cost          1146170 gas
input                   0x608...00000
>
```



BALLOT AT 0XD8B...33FA8 (MEMORY)

delegate | address to

giveRightToVote | address voter

vote

proposal: 1

transact

chairperson

proposals | uint256

startTime

voters | address

winnerName

winningPropo...

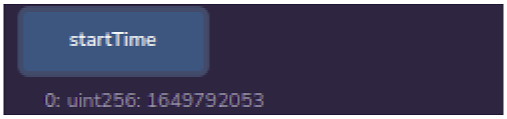
Low level interactions

```
    address public chairperson;
```

bytes32[] proposalNames 3 reference(s) variable that

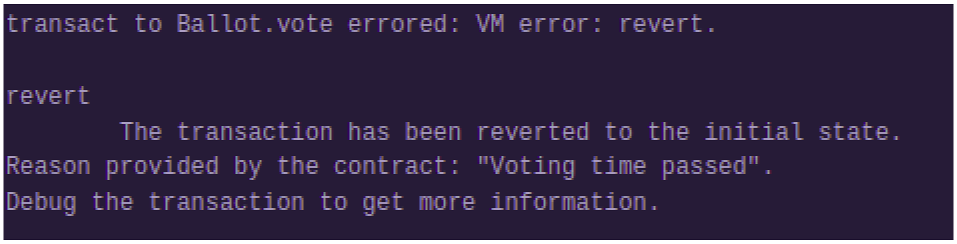
□ listen on network | Search with transaction hash or address

```
✓ [vm] from: 0x5B3...eddC4 to: Ballot.vote(uint256) 0xd8b...33fa8 value: 0 wei data: 0x012...00001 logs: 0 hash: 0xe02...99e88
status                  true Transaction mined and execution succeed
transaction hash        0xe0272f06686337bcea33e4b381c4adb5049b333e9da1f7fc1304d662f2499e88
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      Ballot.vote(uint256) 0xd8b934580fcE35a11B58C6D73aDeE468a2833fa8
gas                     80000000 gas
transaction cost        95049 gas
execution cost          95049 gas
input                   0x012...00001
decoded input           {
                                "uint256 proposal": "1"
                        }
decoded output          {}
logs                    []
```



startTime

0: uint256: 1649792053



```
transact to Ballot.vote errored: VM error: revert.

revert
	The transaction has been reverted to the initial state.
Reason provided by the contract: "Voting time passed".
Debug the transaction to get more information.
```