



zku.ONE background assignment

Darshana, vdcrw7@gmail.com, darz#2944

All answers, code snippets and screenshots can be found at
<https://github.com/darshana-v/zku.one>

A. Conceptual Knowledge

1. Smart contract is a code that gets executed automatically when predetermined terms and conditions are met. It is essentially a set of self-executing instructions that let all participants be immediately certain of the outcome, without any 3rd party intermediary's involvement. They can also automate a workflow, triggering the next action when conditions are met.

Prerequisite for smart contract deployment:

- Code compiler and hasher
- Some tokens for gas fees
- Deployment script
- Node access to mainnet

Deployment steps:

- Setup mainnet node access
- Install tools
- Test your smart contract, ideally in testnet
- Connect to the wallet, and some tokens for deployment using tools
- Verify your smart contract using any tracker app

2. Gas refers to the cost necessary to perform a transaction on the network. Miners set the price of gas based on supply and demand for the computational power of the network needed to process smart contracts and other transactions.

It is important to optimize gas because it leads to affordable and efficient transactions which in turn leads to the simplicity of transactions, making it more scalable. Furthermore, by decreasing the number of transactions, the network is less prone to malicious attacks and is hence more secure.

3. Hash refers to a unique fixed length of bits produced by a hash function after a piece of data is submitted through it. Hash functions are mathematical algorithms that convert an input value of any size to an output (hash) of fixed size.

It is used to hide information because it is a one-way process and information cannot be easily retrieved with just the hash digest without heavy brute force to solve for all possibilities(impossible with the computing power we have today).

4. I would ask the color blind person to choose one object that I would identify after swapping (secretly) as many ever times as needed by him to prove that they are of different colors.

B. Solidity basics

1. Deployed "Helloworld" smart contract

The screenshot displays the Remix Ethereum IDE interface. On the left, the 'ENVIRONMENT' panel shows the contract is deployed on the 'JavaScript VM (London)' network. The 'ACCOUNT' is '0x5B3...eddC4' with a balance of '0.9999999'. The 'GAS LIMIT' is set to '3000000'. The 'VALUE' is '0 Wei'. The 'CONTRACT' is 'HelloWorld - contracts/HelloWorld.s'. The 'Deploy' button is visible. Below it, the 'Transactions recorded' section shows a list of transactions, including one for 'storeNumber' and 'retrieveNumber'. The 'Low level interactions' section is also visible.

The central code editor shows the 'HelloWorld' contract code:

```

6  * @title HelloWorld
7  * @dev Store & retrieve value in a variable
8  * @custom:dev-run-script
9  */
10 contract HelloWorld {
11
12     uint256 number;
13
14     /**
15      * @dev Store value in variable
16      * @param num value to storeNumber
17      */
18     function storeNumber(uint256 num) public {
19         number = num;
20     }
21
22     /**
23      * @dev Return value
24      * @return value of 'number'
25      */
26     function retrieveNumber() public view returns (uint256){
27         return number;
28     }
29 }

```

The right sidebar shows the 'ContractDefinition HelloWorld' with 1 reference(s). Below it, the 'Transaction logs' section displays a list of transactions, including a successful deployment transaction with the following details:

- status: true Transaction mined and execution succeed
- transaction hash: 0x72631b07ec9aef109f8b0db7cdf7b9d3c38712c54f615ee4480df83a9d7b
- from: 0x5B3804670115085434cf8875887550b66C4
- to: HelloWorld.(constructor)
- gas: 14529 gas
- transaction cost: 125677 gas
- execution cost: 125677 gas
- input: 0x6080...70833
- decoded input: ()
- decoded output: -
- logs: []

A 'Snipping Tool' overlay is visible on the right side of the interface, indicating a future update.

2. Went through and implemented the Ballot script
3. Updated the Ballot script to revert transactions after the 5-minute deadline
 - Successful transactions after deploying

The screenshot shows the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, showing the deployment of the 'Ballot - contracts/Ballot.sol' contract. The 'ENVIRONMENT' is set to 'JavaScript VM (London)', and the 'ACCOUNT' is '0x5B3...eddC4 (99.99999999)'. The 'GAS LIMIT' is set to '3000000'. The 'CONTRACT' is 'Ballot - contracts/Ballot.sol'. The 'Deploy' button is highlighted. Below the deployment panel, the 'Transactions recorded' section shows a single transaction with the status 'true Transaction mined and execution succeed'. The 'Deployed Contracts' section shows the 'Ballot AT 0x08B...33fAB (MEMORY)'.

The main editor displays the Solidity code for the 'Ballot.sol' contract. The code defines a 'Voter' struct with fields for 'weight', 'voted', 'delegate', and 'vote'. It also defines a 'Proposal' struct with fields for 'name' and 'voteCount'. The contract includes a 'chairperson' address and a 'winnerName()' function.

The console shows the following transaction details:

- Transaction hash: 0x6b2a52b2e9114b63657903a116f174d4e93a758326b3a8098434a69
- From: 0x5B30b6a701c569545dcf803fcb875f56eddC4
- To: Ballot (constructor)
- Gas: 139863 gas
- Transaction cost: 113911 gas
- Execution cost: 113911 gas
- Input: 0x608...0000

- Successful voting after deploying

The screenshot shows the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, showing the deployment of the 'Ballot - contracts/Ballot.sol' contract. The 'ENVIRONMENT' is set to 'JavaScript VM (London)', and the 'ACCOUNT' is '0x5B3...eddC4 (99.99999999)'. The 'GAS LIMIT' is set to '3000000'. The 'CONTRACT' is 'Ballot - contracts/Ballot.sol'. The 'Deploy' button is highlighted. Below the deployment panel, the 'Transactions recorded' section shows a single transaction with the status 'true Transaction mined and execution succeed'. The 'Deployed Contracts' section shows the 'Ballot AT 0x08B...33fAB (MEMORY)'.

The main editor displays the Solidity code for the 'Ballot.sol' contract. The code defines a 'Voter' struct with fields for 'weight', 'voted', 'delegate', and 'vote'. It also defines a 'Proposal' struct with fields for 'name' and 'voteCount'. The contract includes a 'chairperson' address and a 'winnerName()' function.

The console shows the following transaction details:

- Transaction hash: 0x38b8f791b583f98911e31284c3b4f3a382a563777894881e1436a7e92a3
- From: 0x48483f649c641ef9b849a4677d03315635c62
- To: Ballot.delegate(address) 0x08b83f649c641ef9b849a4677d03315635c62
- Gas: 62727 gas
- Transaction cost: 54545 gas
- Execution cost: 54545 gas
- Input: 0x5c1...eddC4

- Unsuccessful voting after the 5-minute deadline

Ubuntu-1.0.0 (Snapshot_Setup) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Google Chrome May 4 17:17

zku.one/background... Background Assignment x Remix - Ethereum IDE x contract deployment - E x 163593542-51e3f780-0dc x eyyubmermer/Backgrou x MetaMask

remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.7+commit.e28d00a7.js

DEPLOY & RUN TRANSACTIONS

CONTRACT: Ballot - contracts/Ballot.sol

Deploy [0x6699f000]

OR

At Address Load contract from Address

Transactions recorded 1

Deployed Contracts

BALLOT AT 0x088...39f8B (MEMORY)

delegate

to: 0x8330c4fa701c568545dc

transact

giveRightToVote

voter: 0x58380c4fa701c568545dc

transact

vote

proposal: 0x4b4e3f4d0c6d1ef9b8

transact

chairperson

0: address: 0x58380c4fa701c568545dc

endTime

```
148 }
149 }
150 }
151 }
152 // Calls winningProposal() function to get the index
153 // of the winner contained in the proposals array and then
154 // returns the name of the winner
155 function winnerName() external view
156 {
157     returns (bytes32 winnerName_)
158 }
159 winnerName_ = proposals[winningProposal()].name;
160 }
```

ContractDefinition: Ballot 1 reference(s)

Search with transaction hash or address

[vm] from: 0x482...C02db to: Ballot.vote(uint256) 0x0db...33fa8 value: 0 wei data: 0x012...35cb2 logs: 0 hash: 0xbde...7fada

status: false Transaction mined but execution failed

transaction hash: 0xbde0e2c0b9e347aa50196c846cf0bf3c89d2ba30e0f823416f66a07fada

from: 0x4820993b4d1177e7e8f571c4c6a9a02c02db

to: Ballot.vote(uint256) 0x0db934580fc35a11858c0073a0e46a2033fa8

gas: 3000000 gas

transaction cost: 23958 gas

execution cost: 23958 gas

Debug

- Reverting transaction if the voter has already voted

Ubuntu-1.0.0 (Snapshot_Setup) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Google Chrome May 4 17:17

zku.one/background... Background Assignment x Remix - Ethereum IDE x contract deployment - E x 163593542-51e3f780-0dc x eyyubmermer/Backgrou x MetaMask

remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.7+commit.e28d00a7.js

DEPLOY & RUN TRANSACTIONS

CONTRACT: Ballot - contracts/Ballot.sol

Deploy [0x6699f000]

OR

At Address Load contract from Address

Transactions recorded 1

Deployed Contracts

BALLOT AT 0x088...39f8B (MEMORY)

delegate

to: 0x8330c4fa701c568545dc

transact

giveRightToVote

voter: 0x58380c4fa701c568545dc

transact

vote

proposal: 0x4b4e3f4d0c6d1ef9b8

transact

chairperson

0: address: 0x58380c4fa701c568545dc

endTime

```
148 }
149 }
150 }
151 }
152 // Calls winningProposal() function to get the index
153 // of the winner contained in the proposals array and then
154 // returns the name of the winner
155 function winnerName() external view
156 {
157     returns (bytes32 winnerName_)
158 }
159 winnerName_ = proposals[winningProposal()].name;
160 }
```

ContractDefinition: Ballot 1 reference(s)

Search with transaction hash or address

[vm] from: 0x482...C02db to: Ballot.vote(uint256) 0x0db...33fa8 value: 0 wei data: 0x012...35cb2 logs: 0 hash: 0xbde...7fada

status: false Transaction mined but execution failed

transaction hash: 0xbde0e2c0b9e347aa50196c846cf0bf3c89d2ba30e0f823416f66a07fada

from: 0x4820993b4d1177e7e8f571c4c6a9a02c02db

to: Ballot.vote(uint256) 0x0db934580fc35a11858c0073a0e46a2033fa8

gas: 3000000 gas

transaction cost: 23958 gas

execution cost: 23958 gas

Debug