

Name: Oluwatobi Akintunlese
Discord Name: Tobi Akin#0304

Background Assignment Solution

Link to github repo solution:

<https://github.com/oluwatobi1/ZKU-solidity-et.al/tree/main/background%20Assignment>

Q 1: Hello World snippet

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel displays the 'HelloWorld - contracts/HelloWorld.sol' contract. It has a 'Deploy' button and a 'Publish to IPFS' checkbox. Below, it shows 'Transactions recorded' (6) and 'Deployed Contracts' (1). The 'HELLOWORLD AT 0XD7A...F771B (M)' contract is listed with an 'update' button (value 45) and a 'retrieve' button. The 'Low level interactions' section shows a 'CALLDATA' field and a 'Transact' button. The main editor shows the 'HelloWorld.sol' code:

```
1 // SPDX-License-Identifier: GPL-3.0
2 pragma solidity ^0.7.1;
3
4 contract HelloWorld {
5     // initialize variable helloworldNumber
6     // variable is set to 0 by default
7     uint256 helloworldNumber = 7650;
8
9     // this function return the latest value of helloworldNumber
10    // it reads of the value of network
11    function retrieve() public view returns(uint256){
12        return helloworldNumber;
13    }
14
15    // Extra feature:this function can be used to change the value of helloworldNumber
16    function update(uint256 _helloworldNumber) public {
17        helloworldNumber = _helloworldNumber;
18    }
19 }
```

The bottom panel shows a transaction log with a call from 0x5838Da6a701c568545dCfcB03FcB875f56beddC4 to HelloWorld.retrieve() with data 0x2e6...4cecl and a 'Debug' button.

Q2 & Q 3

Before improvement

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel displays the 'Ballot - contracts/Ballot.sol' contract. It has a 'Deploy' button and a 'Publish to IPFS' checkbox. Below, it shows 'Transactions recorded' (12) and 'Deployed Contracts' (1). The 'BALLOT AT 0XD2A...FD005 (MEMOR)' contract is listed with buttons for 'delegate', 'giveRightToVote', 'vote', 'chairperson', 'proposals', and 'voters'. The main editor shows the 'Ballot.sol' code:

```
51 function giveRightToVote(address voter) external {
52     // If the first argument of 'require' evaluates
53     // to 'false', execution terminates and all
54     // changes to the state and to Ether balances
55     // are reverted.
56     // This used to consume all gas in old EVM versions, but
57     // not anymore.
58     // It is often a good idea to use 'require' to check if
59     // functions are called correctly.
60     // As a second argument, you can also provide an
61     // explanation about what went wrong.
62     require(
63         msg.sender == chairperson,
64         "Only chairperson can give right to vote."
65     );
66     require(
67         !voters[voter].voted,
68         "voter already voted."
69     );
70 }
```

The bottom panel shows a transaction log with a call from 0x5838Da6a701c568545dCfcB03FcB875f56beddC4 to Ballot.giveRightToVote(address) 0xd2a5bC10698F095501Fe6cb468a1789A08fd005 with gas 80000000, transaction cost 48436 gas, and execution cost 48436 gas. The hash is 0xb52370fd846e4e0c20c5129f0b35bf930862910aa7c7103be8ecb04d9643d8c.

After improvement

The screenshot displays the Remix IDE interface with the following components:

- Left Panel (Deploy & Run Transactions):**
 - At Address:** A button to load a contract from a specific address.
 - Transactions recorded:** 10 transactions are listed.
 - Deployed Contracts:** A list of deployed contracts, including 'BALLOT AT 0x358...D5EE3 (MEMORY)'.
 - Contract Functions:** A list of functions available for the selected contract, including 'delegate', 'giveRightToVote', 'vote', 'chairperson', 'proposals', 'voters', 'winnerName', and 'winningPropo...'.
- Center Panel (Code Editor):** Displays the Solidity code for 'Ballot.sol'. The code includes a loop for voting, a function to give the right to vote, and a function to delegate a vote.
- Right Panel (Transaction Details):** Shows the details of a transaction, including the 'to' address, gas used, transaction cost, execution cost, and the transaction hash.

```
36 // this makes it possible for multiple addresses to be given a right
37 // to vote in one transaction
38 for (uint i; i < voters.length; i++) {
39     require(
40         msg.sender == chairperson,
41         "Only chairperson can give right to vote."
42     );
43     require(
44         !voters[voter[i]].voted,
45         "The voter already voted."
46     );
47     require(voters[voter[i]].weight == 0);
48     voters[voter[i]].weight = 1;
49 }
50
51
52 /// Delegate your vote to the voter 'to'.
53 function delegate(address to) external {
```

Transaction Details:

- to:** Ballot.giveRightToVote(address[]) 0x358AA13c52544ECCEf688AD0f801012ADAD5eE3
- gas:** 80000000 gas
- transaction cost:** 277225 gas
- execution cost:** 277225 gas
- hash:** 0x8c59782b38eb4453d235d7222f6462be4bc85620b25b8def86175a6aec8a2ec5