

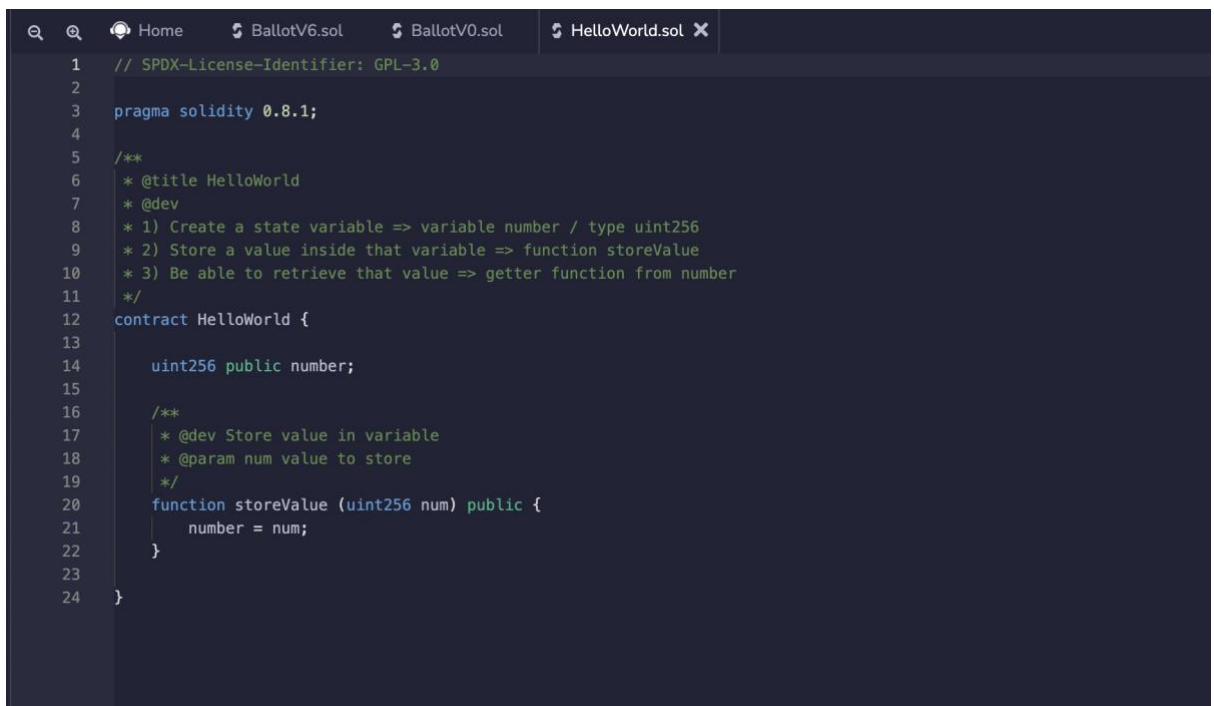
ZKU.One Background Assignment

mail: enricobottazzi@icloud.com

discord Username: enrico.eth#8998

Github repo: <https://github.com/enricobottazzi/zku.One>

1. Program a super simple “Hello World” smart contract: store an unsigned integer and then retrieve it. Please clearly comment your code. Once completed, deploy the smart contract on [Remix](#). Include the .sol file and a screenshot of the Remix UI once deployed in your final submission pdf (more info about submission formatting below).
- HelloWorld.sol screenshot



```

1  // SPDX-License-Identifier: GPL-3.0
2
3  pragma solidity 0.8.1;
4
5  /**
6   * @title HelloWorld
7   * @dev
8   * 1) Create a state variable => variable number / type uint256
9   * 2) Store a value inside that variable => function storeValue
10  * 3) Be able to retrieve that value => getter function from number
11  */
12  contract HelloWorld {
13
14      uint256 public number;
15
16      /**
17       * @dev Store value in variable
18       * @param num value to store
19       */
20      function storeValue (uint256 num) public {
21          number = num;
22      }
23
24  }
    
```

- HelloWorld.sol file: <https://github.com/enricobottazzi/zku.One/blob/master/contracts/HelloWorld.sol>
2. On the documentation page, the “Ballot” contract demonstrates a lot of features on Solidity. Read through the script and try to understand what each line of code is doing, then implement the Possible Improvements by reducing the number of transactions in the “giveRightToVote” function while maintaining the same functionality of the program.
 3. Deploy your script on Remix and compare the difference in gas fees between the original script and the improved script when giving 10 voters the right to vote. Once completed, submit (via pdf or Github) (1) your improved version of the contract (.sol

file) with comments describing the changes you made, and (2) screenshots (before and after) of the gas fees for the transaction(s) to give 10 voters the right to vote. All code to be submitted either in pdf (only code snippets where ever required) or via Github links (copy the repo/pull-request links and the commit in the pdf).

Goal: giving 10 voters the right to vote. List of addresses:

"0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB", "0x617F2E2fD72FD9D5503197092aC168c91465E7f2",
 "0x17F6AD8Ef982297579C203069C1DbfFE4348c372", "0x5c6B0f7Bf3E7ce046039Bd8FABdF3f9F5021678",
 "0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7", "0x1aE0EA34a72D944a8C7603FfB3eC30a6669E454C",
 "0x0A098Eda01Ce92ff4A4CCb7A4fFb5A43EBC70DC", "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
 "0x14723A09ACff6D2A60DcdF7aA4Aff308FDDC160C", "0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB"

- BallotV0 (*before improvement* version) =>

<https://github.com/enricobottazzi/zku.One/blob/master/contracts/BallotV0.sol>

The screenshot displays a web interface for deploying and running transactions. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel shows a 'Deploy' button and a list of deployed contracts. The main panel shows the 'BallotV0.sol' contract code with a function 'giveRightToVote'. Below the code, a transaction execution summary is displayed, showing the gas cost of 48657 gas, which is highlighted with a green box. The transaction details include the status, transaction hash, from address, to address, gas, transaction cost, execution cost, hash, input, and decoded input.

Gas cost to execute *giveRightToVote* to a single address = 48657 gas

Gas cost to execute *giveRightToVote* to 10 addresses = 486570 gas

- BallotV6 (after gas improvement version):
<https://github.com/enricobottazzi/zku.One/blob/master/contracts/BallotV6.sol>

```
// ADDED, take array of address as function input instead of a single address
// by doing so the 'msg.sender == chairperson' requirement has to be called only once rather than 10 times
function giveRightToVote(address [] calldata _votersAddresses) external {

    require(
        msg.sender == chairperson,
        "Only chairperson can give right to vote."
    );

    for (uint i = 0; i < _votersAddresses.length; i++) {

        // ADDED assign a reference to _voter
        // By doing so the function doesn't have to always get data from a state variable, resulting in gas savings
        Voter storage _voter = voters[_votersAddresses[i]];

        require(
            !_voter.voted,
            "The voter already voted."
        );

        require(_voter.weight == 0);
        _voter.weight=1;
    }
}
```

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, showing the contract 'Ballot - contracts/BallotV6.sol' and the 'Deploy' button. The main editor displays the Solidity code for the 'giveRightToVote' function. The right sidebar shows the transaction details for the deployment, including the gas cost of 277197 gas, which is highlighted with a green box.

Gas cost to execute *giveRightToVote* to 10 addresses = 277197 gas (BallotV6)
 Gas cost to execute *giveRightToVote* to 10 addresses = 486570 gas (BallotV0)
Improvement between V0 and V6 = 209373 gas