

Dear potential candidate,

This assessment is an opportunity for us to check your abilities to build secure and efficient smart contract architectures using the most recent patterns in Solidity.

Assessment

We want you to write the smart contracts of an ETH splitter, a simple DAPP that splits ETH transfers equally between a set of defined recipients.

1. Write the ETHSplitter Contract, the contract must be owned by a single address that can add and remove recipients dynamically. Upon receiving ETH the contract must split it equally and transfer the resulting amount to the recipient
2. Write a ETHSplitterFactory that automates the creation of an ETHSplitter for a user. Make sure a user can create only one ETHSplitter.
3. Write a ETHAndTokenSplitter contract that extends the ETHSplitter contract but can also receive and distribute ERC-677 tokens.
4. Write a ETHSplitterFactoryV2 (and additional contracts if needed) that improves ETHSplitterFactory by minimizing the gas spent for each new deployment of a splitter?

Imagine that you have to write the smart contract of a lottery that awards 100 ETH to the winner. The smart contract can have multiple methods but must satisfy the following requirements.

- The contract must expose a *buyTicket* method that buys a ticket for the caller.
- The contract must expose a *draw* method to select the winner. Method can only be called by the owner of the lottery.
- The lottery is fair and cannot be abused by any stakeholder

Assuming you cannot use an external API or Oracle, how would you generate a random number when the *draw* method is called by the owner? Explain the sequence of events that must happen on-chain, their timing, and the parameters that must be passed to the methods. Explain why your strategy is secure and generates a good random number.

When Finished.

Please try to submit your response on the day of receiving the test.

For submission, please publish your code on Github and send us a link to the repository. Or send youngsung.c@memusic.io a ZIP file