

Take Home Exercise for Smart Contracts Developer

Instructions

1. Please do not devote more than 3 hours total to this take home exercise.
2. Attempt **both** exercises described in this document
3. Please submit your code as a single, flattened file. Attached the file to an email sent to your recruiter,
4. Include any requested written explanations in a document as an email attachment (.DOCX format) or shared Google Doc (with appropriate access permissions).

Exercise A: OTC Swap

Write a smart contract for conducting atomic over-the-counter (OTC) swaps of ERC20 tokens. Specifically, the contract you write should support the use case where Alice and Bob agree to an exchange – Alice will trade her M of tokenX for Bob's N of tokenY, to be completed in a single "atomic" transaction.

Additionally:

- Swaps should be able to specify a counterparty – if Alice and Bob agree to a swap, Eve (a third party) should not be able to replace either Alice or Bob in the swap, unless they have explicitly agreed to make the same swap with Eve
- Swaps should expire, i.e. they should only be executable within a specified timeframe

Provide a brief explanation of some of the decisions you made in the process.

Write a few (four or less!) test cases for the contract. **YOU DO NOT HAVE TO WRITE THE TESTS THEMSELVES** – merely explain how you would test the contract and why you have chosen these tests.

Requirements:

1. Write the Solidity code for the smart contract.
2. Write a few (FOUR or less) test cases for the contract.
 - a. **YOU DO NOT HAVE TO WRITE THE TESTS THEMSELVES** – merely explain how you would test the contract and why you have chosen these tests.
3. Make sure your tests cover key functional aspects of the smart contract (apply judgment).

Deliverables:

1. Solidity code for the smart contract. Submit a single, flattened file.
2. A document outlining
 - a. Test cases (Optional: Code for a few tests)
 - b. Explanation of design choices/decisions, and any assumptions you made.
3. [Optional] A README.md file explaining how to run the tests (if you are including code for these tests).

Note:

This is a take-home exercise and you have 2 hours to complete it. You may use any resources you need, except for direct help from another person. Please submit your code and documentation via email to the recruiter once you have completed the exercise.

Exercise B: Multisignature Wallet

Please write a multisignature smart contract wallet. The wallet should require **m-of-n signatures** for any action, with the threshold (m), and list of signatories (members of m) adjustable through multisig actions.

The wallet should support queuing an arbitrary number of actions (**either calls and/or delegatecalls**) as part of a single transaction – that is, signatories should be able to sign on a *set* of actions with a single signature.

Provide a brief explanation of some of the decisions you made in the process.

Write a few (four or less!) test cases for the contract. **YOU DO NOT HAVE TO WRITE THE TESTS THEMSELVES** – merely explain how you would test the contract and why you have chosen these tests.

Extra Credit: Support deploying contracts using `CREATE2` from the wallet

Requirements:

1. Write the Solidity code for the smart contract.
2. Write a few (FOUR or less) test cases for the contract.
 - a. **YOU DO NOT HAVE TO WRITE THE TESTS THEMSELVES** – merely explain how you would test the contract and why you have chosen these tests.
3. Make sure your tests cover key functional aspects of the smart contract (apply judgment).

Deliverables:

1. Solidity code for the smart contract. Submit a single, flattened file.
2. A document outlining
 - a. Test cases (Optional: Code for a few tests)

- b. Explanation of design choices/decisions, and any assumptions you made.
3. [Optional] A README.md file explaining how to run the tests (if you are including code for these tests).

Note:

This is a take-home exercise and you have 2 hours to complete it. You may use any resources you need, except for direct help from another person. Please submit your code and documentation via email to the recruiter once you have completed the exercise.