

Programming Test

Please list the approximate amount of time spent completing each exercise. Be sure to state all your assumptions made when solving each exercise. Code samples will be tested and evaluated as if they were going to enter production. The quality of your responses will determine whether the interview process continues, so don't take them lightly.

Keep in mind there may be many ways to solve these problems.

1) You have been given 2 unusually durable PS5's. You are in an office building that is 100 stories high. Using the fewest possible number of drops from windows in your office building, determine the highest floor you can drop a PS5 from and have it survive. For example, they might be able to take the drop from the 30th floor, but not the 31st. You can break both PS5s in your search. State the worst case number of drops needed and explain how you arrived at that answer.

2) What tools do you use on a day to day basis? Explain why they are important.

3) Imagine that you are interviewing candidates for this position. Write an example question to be used during the interview process. Explain how this question tests the candidate's knowledge, and how this knowledge is representative of the type of problems you would expect to tackle when working as a front-end developer.

4) In Javascript, **Without using built-in functions or imported libraries/modules**, write a function with the following signature that, given a matrix of integers, returns a string with the entries of that matrix appended in clockwise order. For instance, the 3x4 matrix below:

Hint: Keep in mind there could be many different dimensions of a matrix passed in!

```
const Matrix = [2, 3, 4, 8,  
                5, 7, 9, 12,  
                1, 0, 6, 10]
```

would make the string "2, 3, 4, 8, 12, 10, 6, 0, 1, 5, 7, 9".

```
function BuildStringFromMatrix(inMatrixElements, NumRows, NumColumns)  
{  
    // Your code goes here  
}
```

For all the following questions you may work on a private GitHub repo and then share it with us.

Feel free to email with questions or problems as necessary

5) Smart Contract Challenge

Setup a project and create a contract

Summary

ETHPool provides a service where people can deposit ETH and they will receive weekly rewards. Users must be able to take out their deposits along with their portion of rewards at any time. New rewards are deposited manually into the pool by the ETHPool team each week using a contract function.

Requirements

- Only the team can deposit rewards.
- Deposited rewards go to the pool of users, not to individual users.
- Users should be able to withdraw their deposits along with their share of rewards considering the time when they deposited.

5a)

Let say we have user ***A*** and ***B*** and team ***T***.

A deposits 100, and ***B*** deposits 300 for a total of 400 in the pool. Now ***A*** has 25% of the pool and ***B*** has 75%. When ***T*** deposits 200 rewards, ***A*** should be able to withdraw 150 and ***B*** 450 once the period ends.

Given the above specification and scenario, which of the following is true? Explain your answer.

A) *A* deposits then *T* deposits then *B* deposits then *A* withdraws and finally *B* withdraws.

B) *A* should get their deposit + all the rewards.

C) *B* should only get their deposit because rewards were sent to the pool before they participated.

D) *A* and *B* should get their deposit + the corresponding rewards based on the time they have been in the pool.

Goal

Design and code a contract for ETHPool, take all the assumptions you need to move forward. Think about the most gas-efficient implementation you can.

You can use any development tools you prefer: Hardhat, Truffle, Brownie, Solidity, Vyper.

Useful resources:

- Solidity Docs: <https://docs.soliditylang.org/en/v0.8.7>
- Educational Resource: <https://github.com/scaffold-eth/scaffold-eth>

6) Deploy your contract from question 6

Deploy the contract to any Ethereum testnet of your preference. Keep record of the deployed address.

Bonus:

- Add Radspec
- Verify the contract in Etherscan

7) Interact with the contract from question 7

Create a script (or a Hardhat task) to query the total amount of ETH held in the contract and any other thing you find interesting.

You can use any library you prefer: Ethers.js, Web3.js, Web3.py, eth-brownie

8) Create a subgraph

Create a subgraph that index users of the contract from the first challenge into entities as you see convenient. For example, it would be great to query all user deposits to the pool. You can be creative and add other information that you find relevant.

<https://thegraph.com/docs/quick-start>