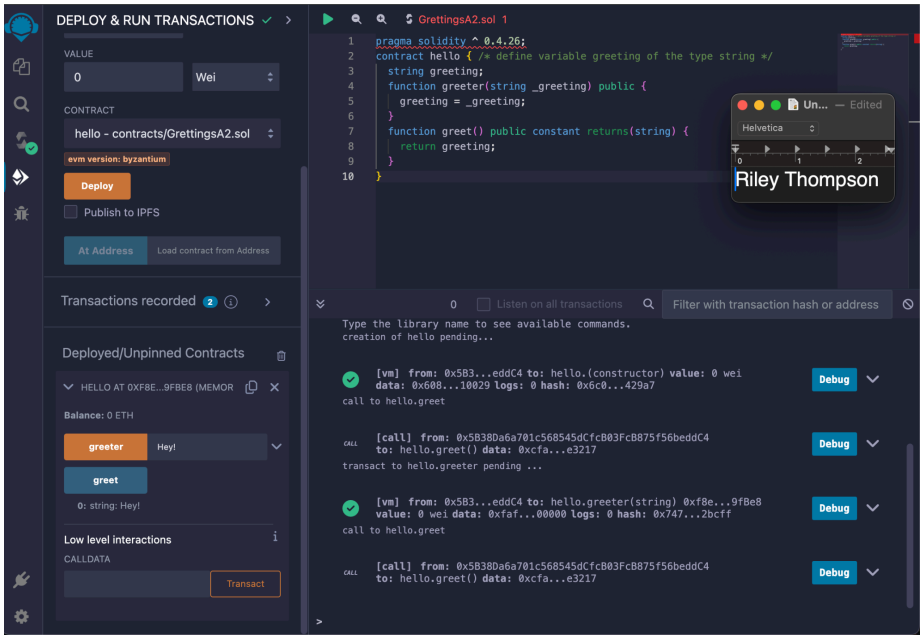
Riley Thompson
826526487

Blockchain Lab 2

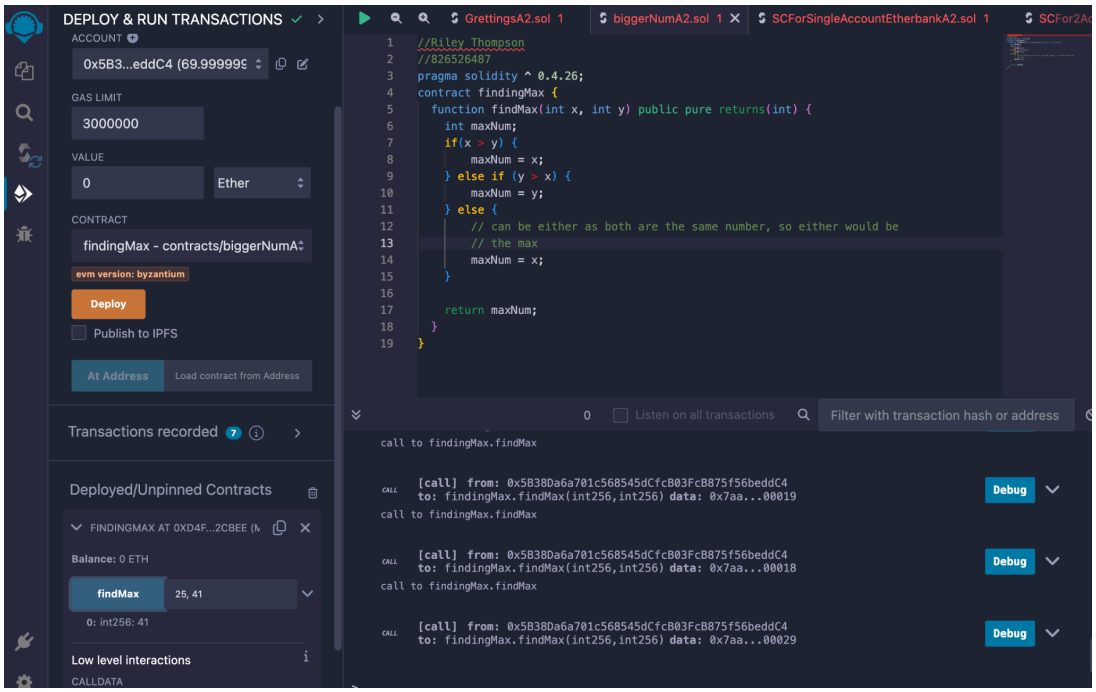
1.

Called the greeter function with parameter hey!, and then when I called greet, it displays hey!

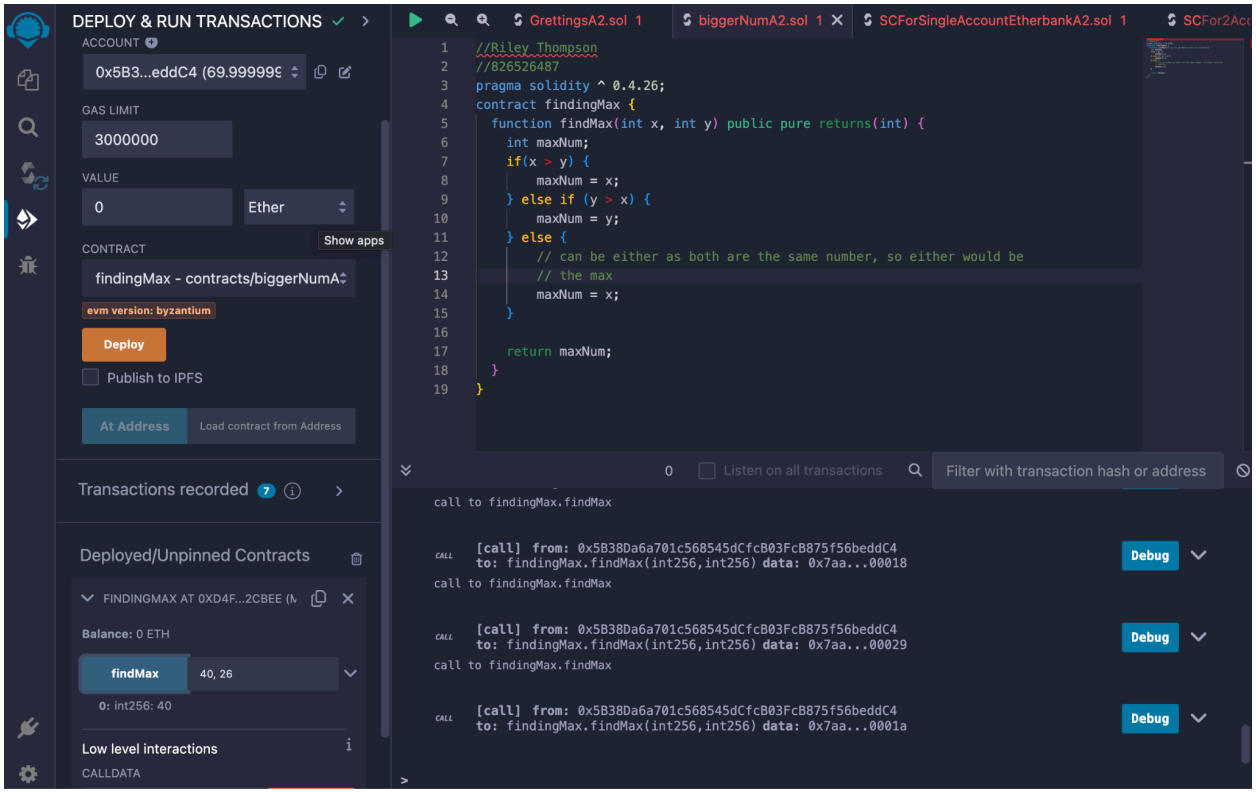


2.

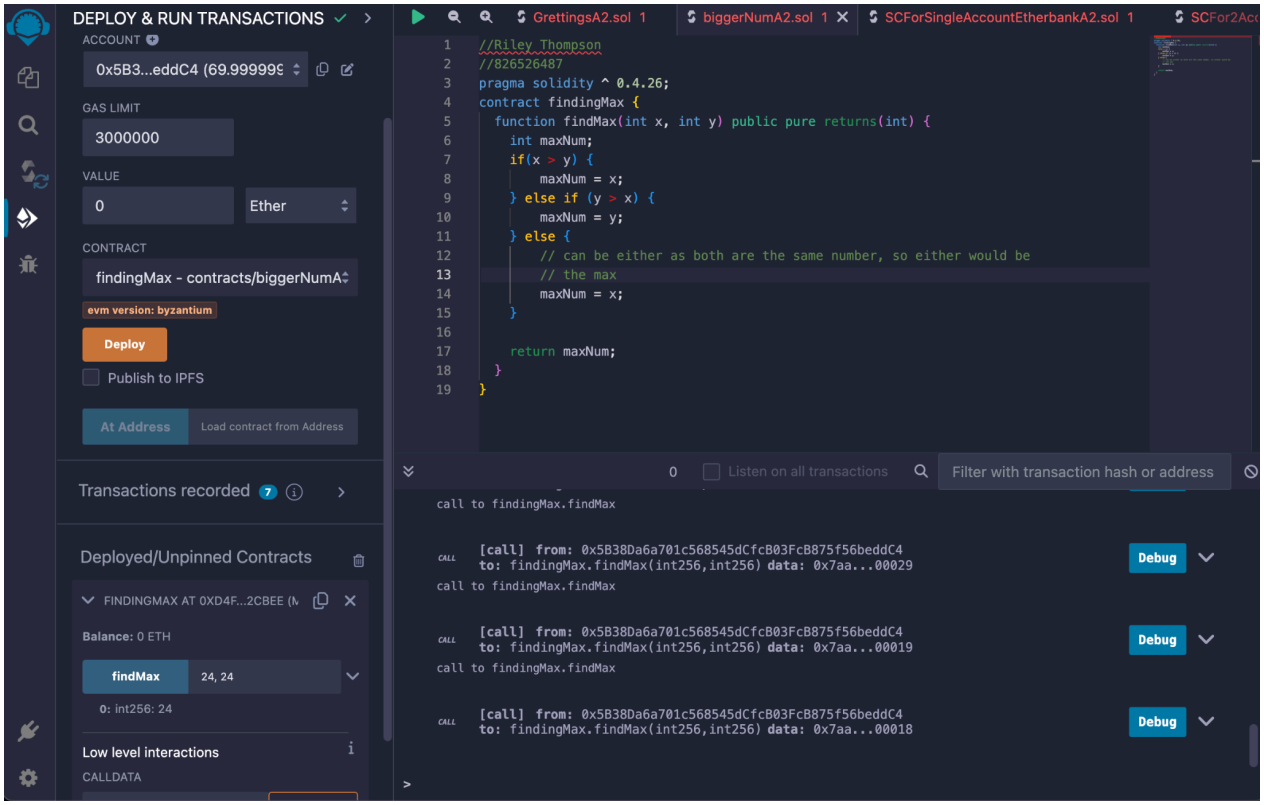
Shows when y is greater than x, returns y



Shows when x is greater than y, returns x

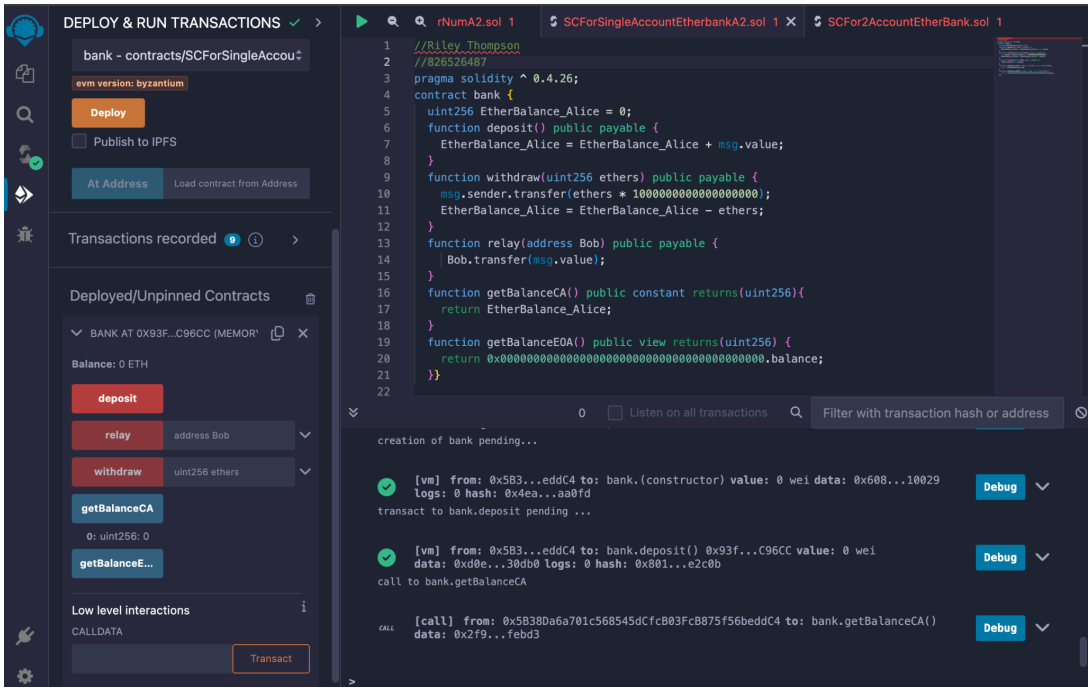


Shows when they are equal and would return x as they are the same, so could be x or y

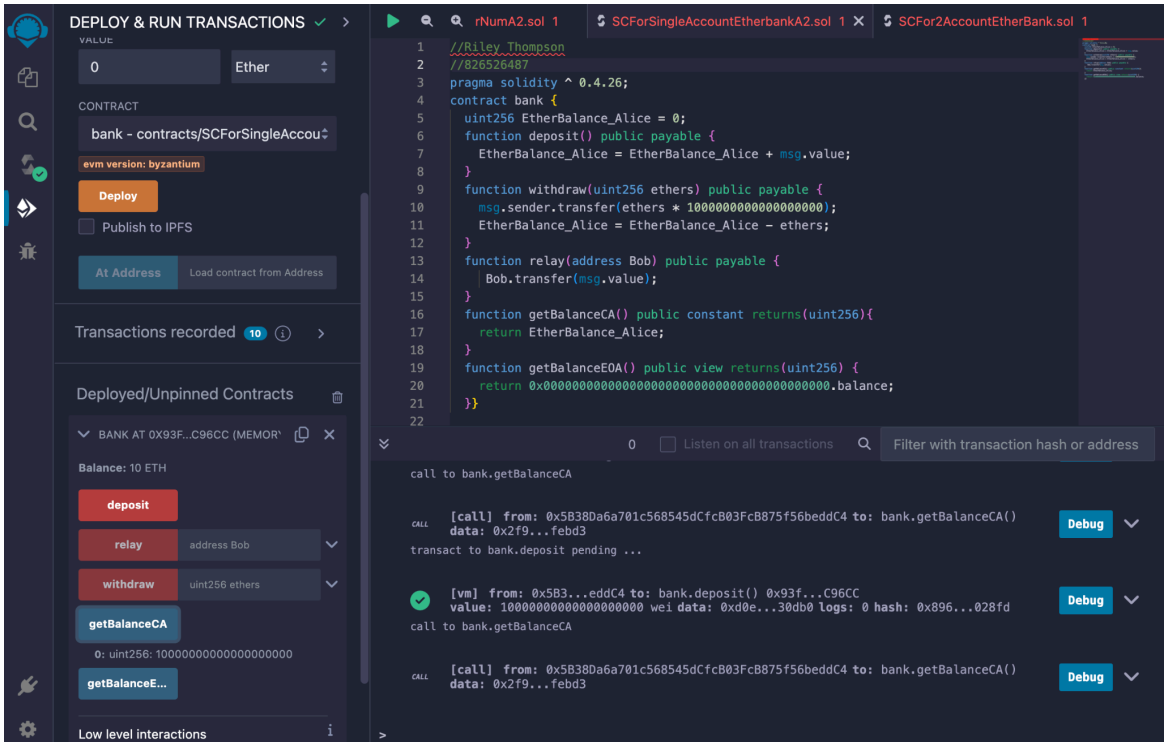


The file that is submitted is biggerNumA2.sol

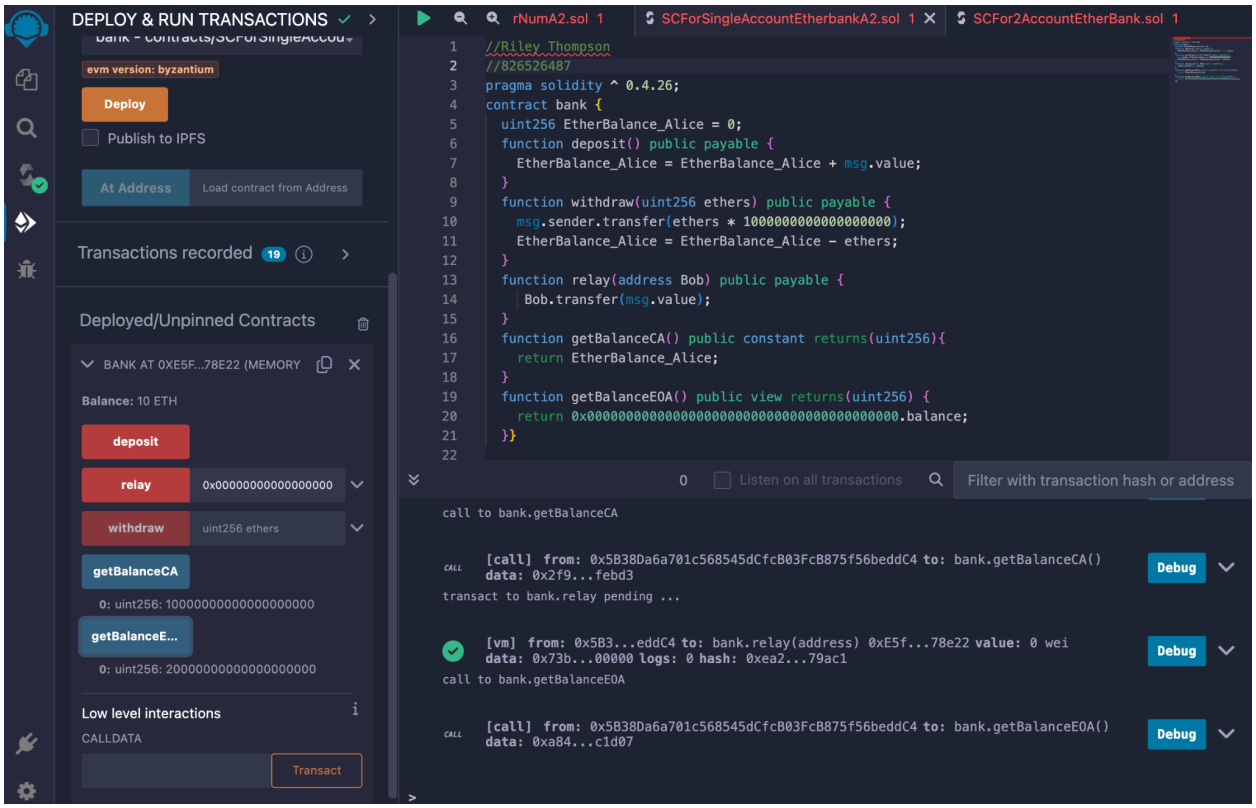
3.
After deployment, click the 'getBalanceCA' button to show the balance(which is 0):



Deposited from Alice Adress with Ether 10, so displays 10000000000000000000 wei which is 10 ether:



Called relay function with argument 0x0000000000000000000000000000000000000000 and getBalanceEOA, returns the balance of that address which is at 20 ether or 2000000000000000000 wei

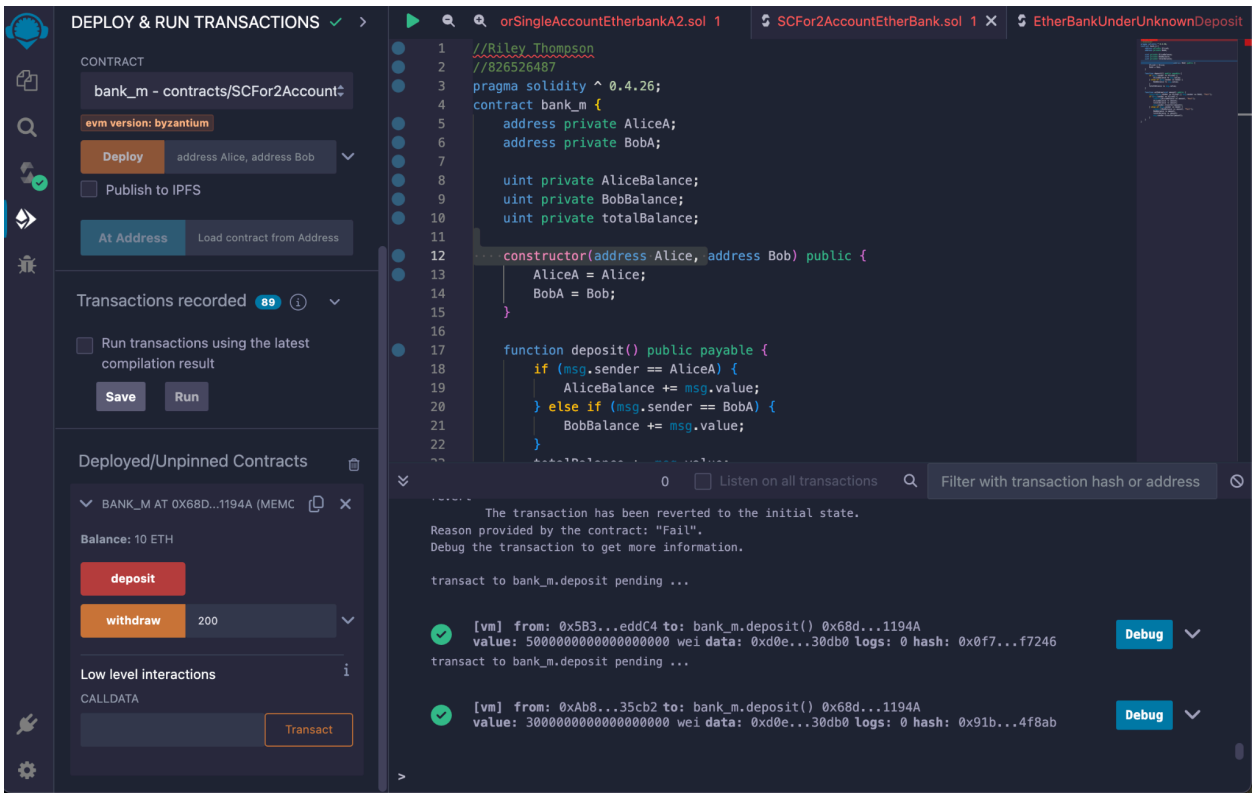


4.

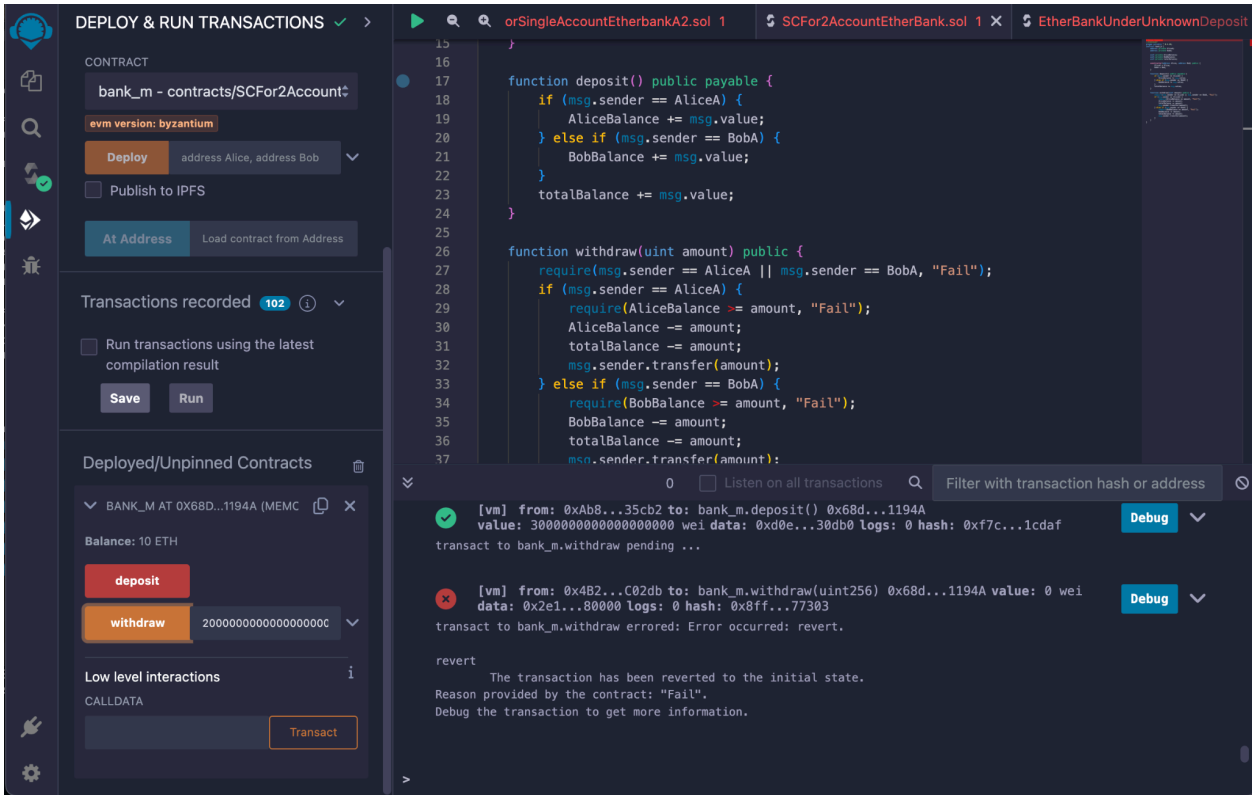
I put public at the withdraw function as it gave warning otherwise and didn't like having it have warning and said just defaults to public anyways, so I put public on withdraw function:

```
contracts/SCFor2AccountEtherBank.sol:26:5: Warning: No visibility specified.
Defaulting to "public".
function withdraw(uint amount)
{
^ (Relevant source part starts here and spans across multiple lines).
```

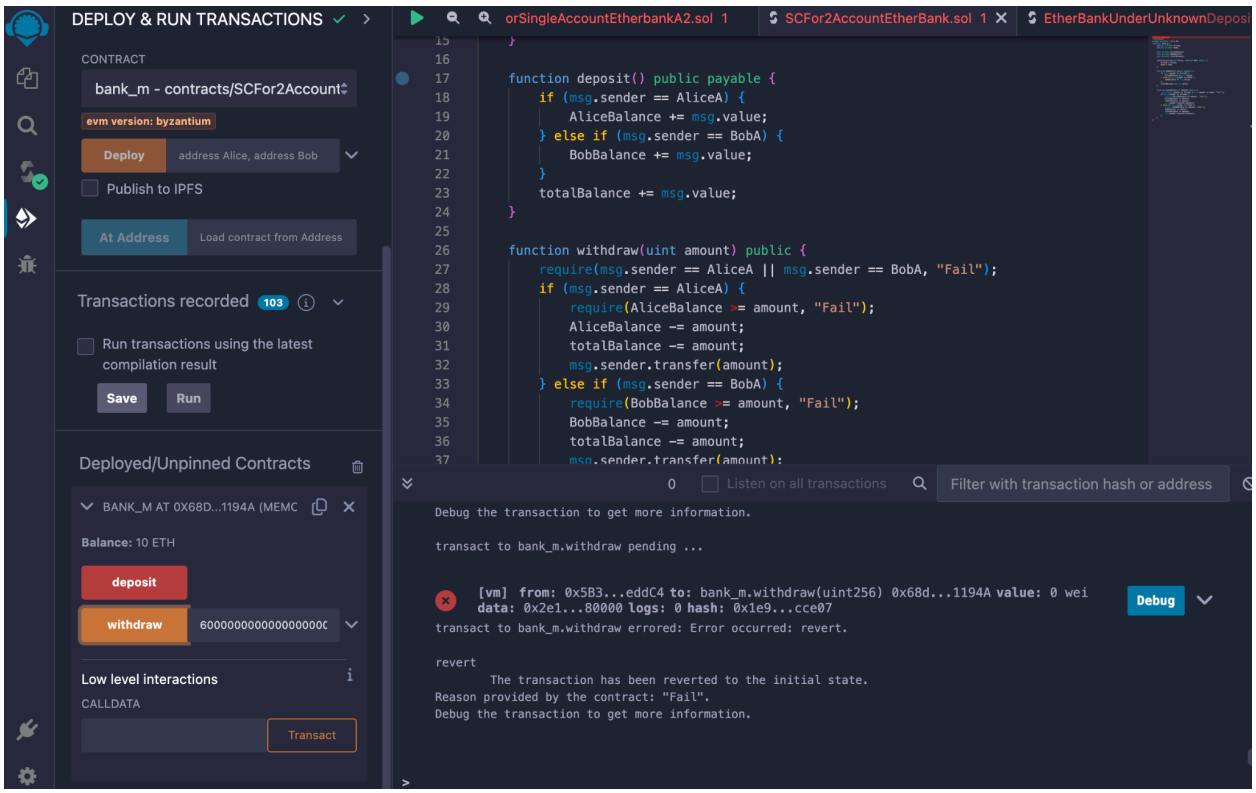
I began with depositing 5 and 3 from Alice(5 ether), then Bob(3 ether):



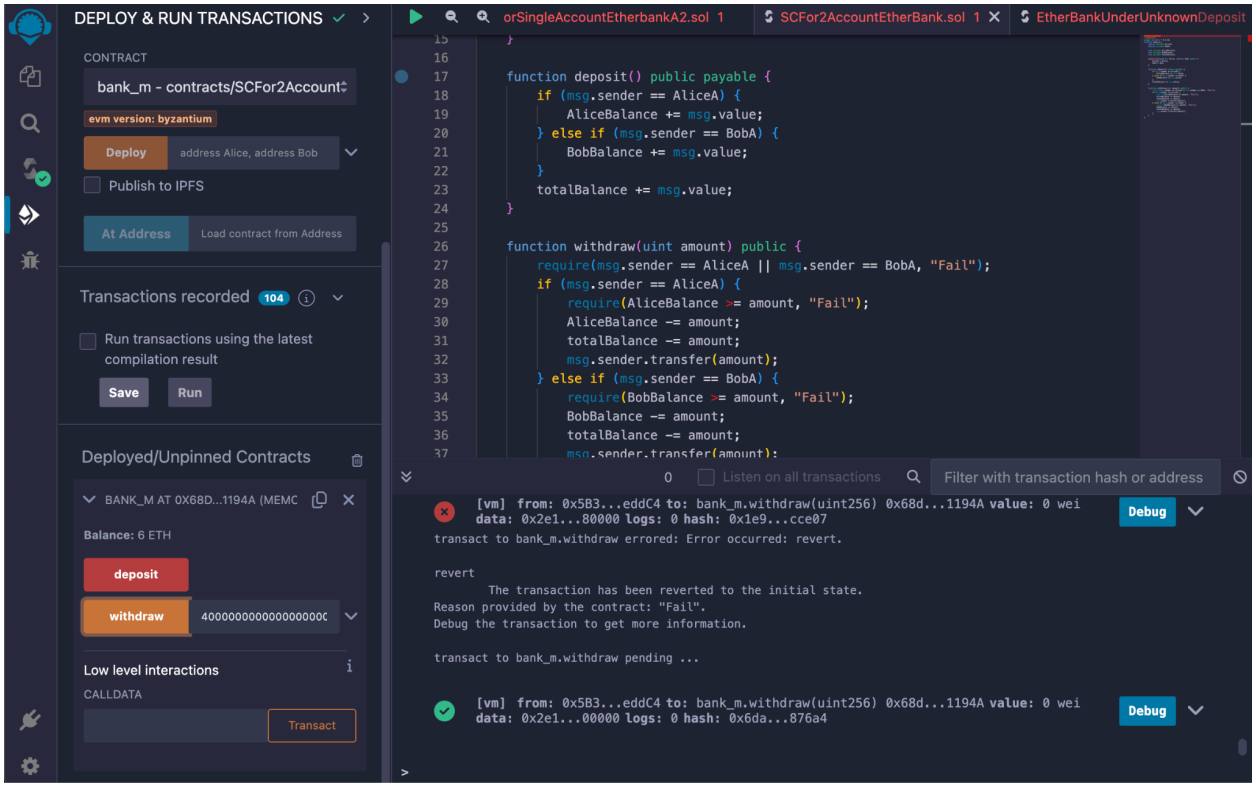
Charlie’s address fails to get money from bank(correct) - only Alice and Bob should:



Cannot withdraw more than amount in own address amount(success) - Alice trying to withdraw 6 ether:

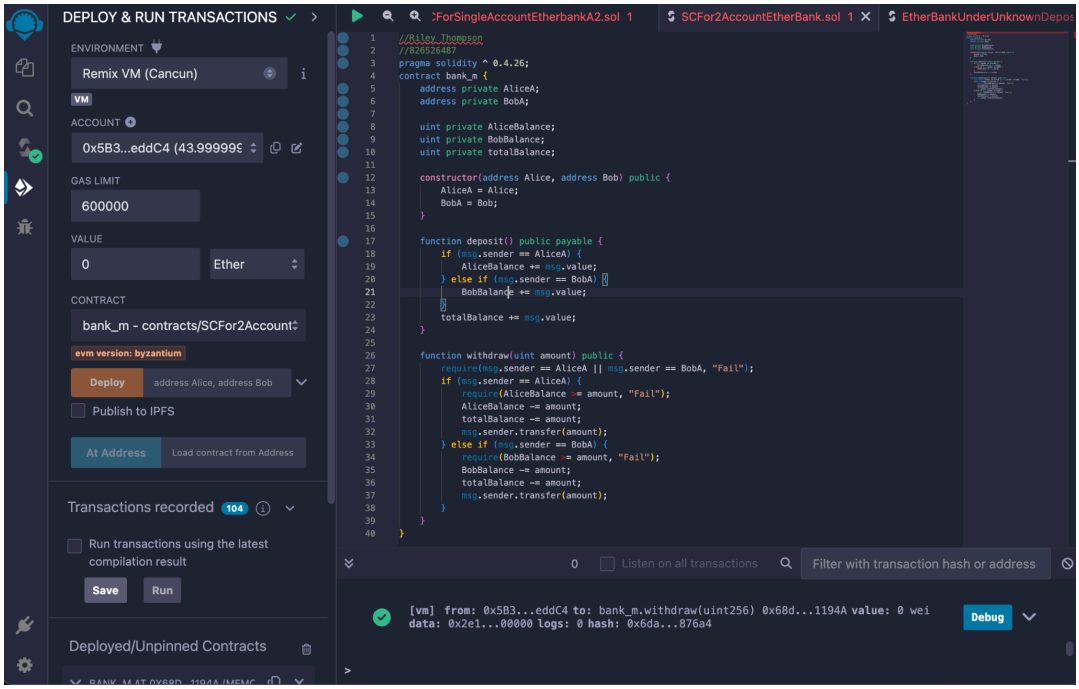


Able to withdraw what own address amount has(success):



Other test cases work, tested(success):

Here is my code in a picture:



The file that is submitted is SCFor2AccountEtherBank.sol

5.
I put public at the withdraw function as it gave warning otherwise and didn't like having it have warning and said just defaults to public anyways, so I put public on withdraw function:

```
contracts/SCFor2AccountEtherBank.sol:26:5: Warning: No visibility specified.
Defaulting to "public".
function withdraw(uint amount)
{
^ (Relevant source part starts here and spans across multiple lines).
```

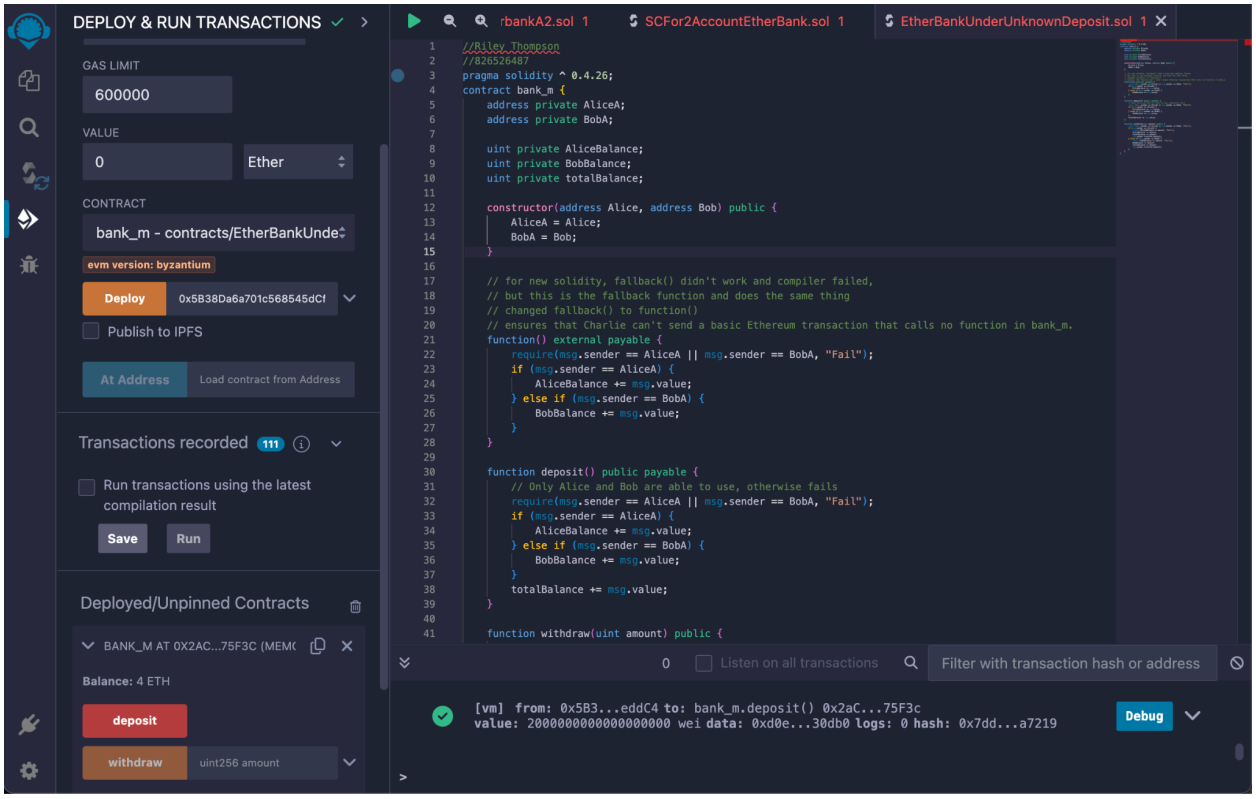
1) Charlie sends a basic Ethereum transaction that calls no function in bank_m.
-Added:

```
// for new solidity, fallback() didn't work and compiler failed,
// but this is the fallback function and does the same thing
// changed fallback() to function()
// ensures that Charlie can't send a basic Ethereum transaction that calls no function in bank_m.
function() external payable {
    require(msg.sender == AliceA || msg.sender == BobA, "Fail");
    if (msg.sender == AliceA) {
        AliceBalance += msg.value;
    } else if (msg.sender == BobA) {
        BobBalance += msg.value;
    }
}
```

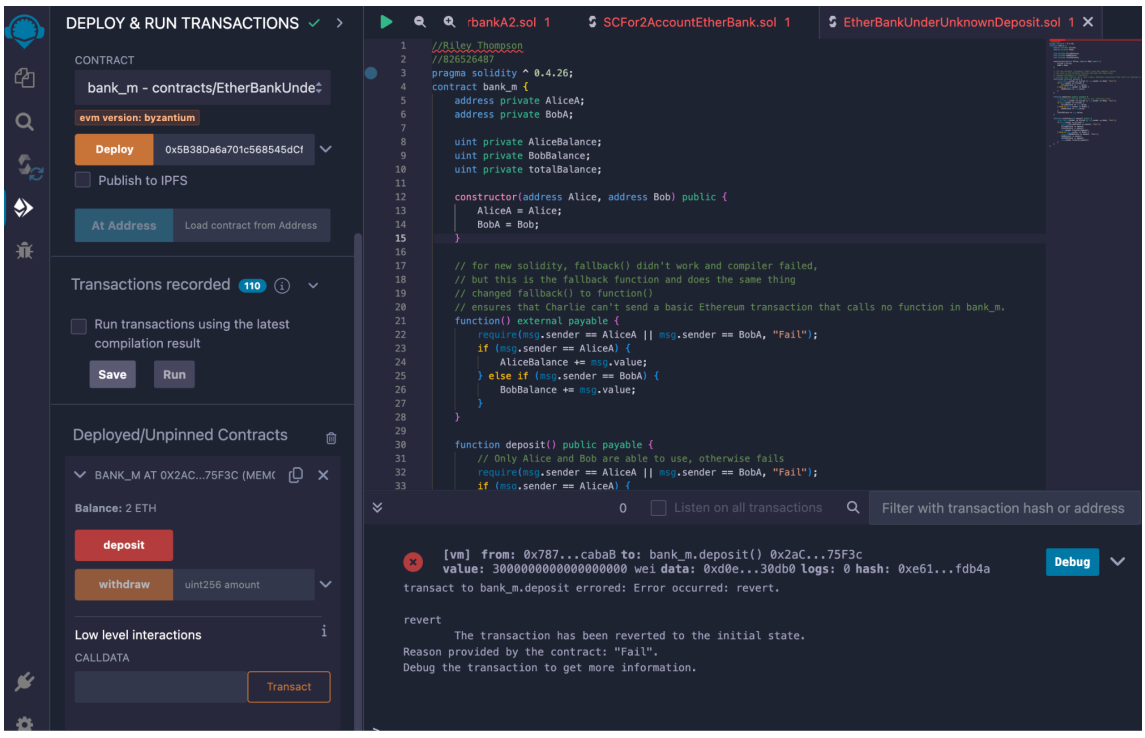
2) Charlie sends a transaction to externally call the deposit() function in bank_m explicitly.
-Added this in deposit() function:

```
    // Only Alice and Bob are able to use, otherwise fails
    require(msg.sender == AliceA || msg.sender == BobA, "Fail");
```

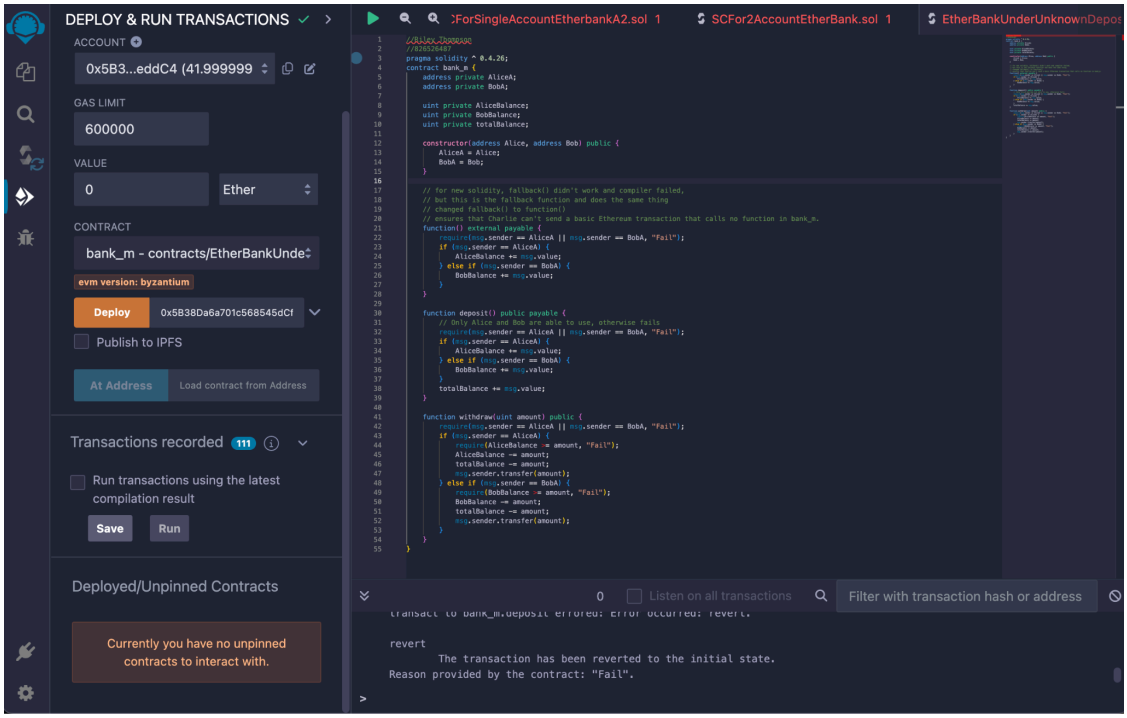
Here is testing the code. Nothing changed aside from adding things that restrict Charlie address from sending Ether, so deposit and withdrawal work as above in problem 4, and now going to show that cannot deposit with Allice or Bob:



But can't with Charlie address:



Here is my code in a picture:



The file that is submitted is EtherBankUnderUnknownDeposit.sol