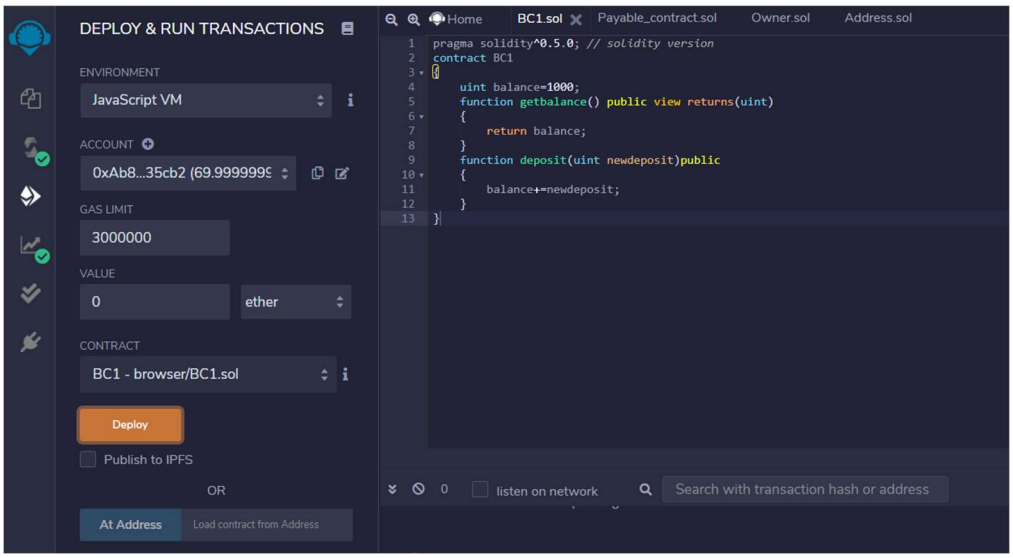
CRYPTO CURRENCIES AND BLOCKCHAIN

LAB_2

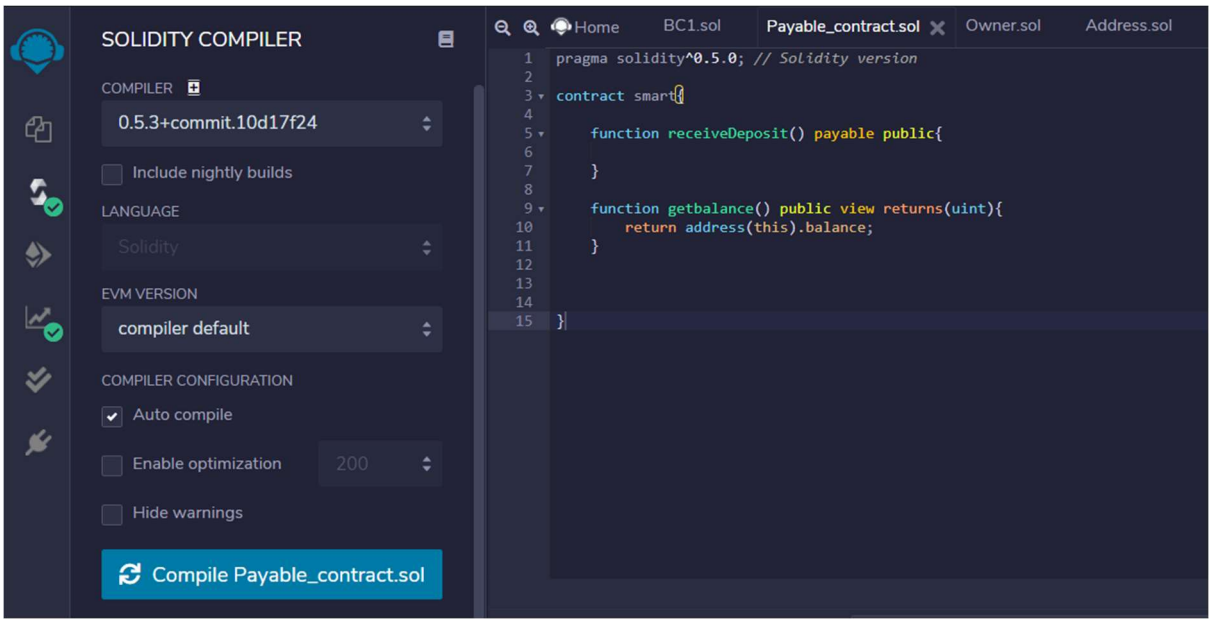
VIBHAV SHARMA

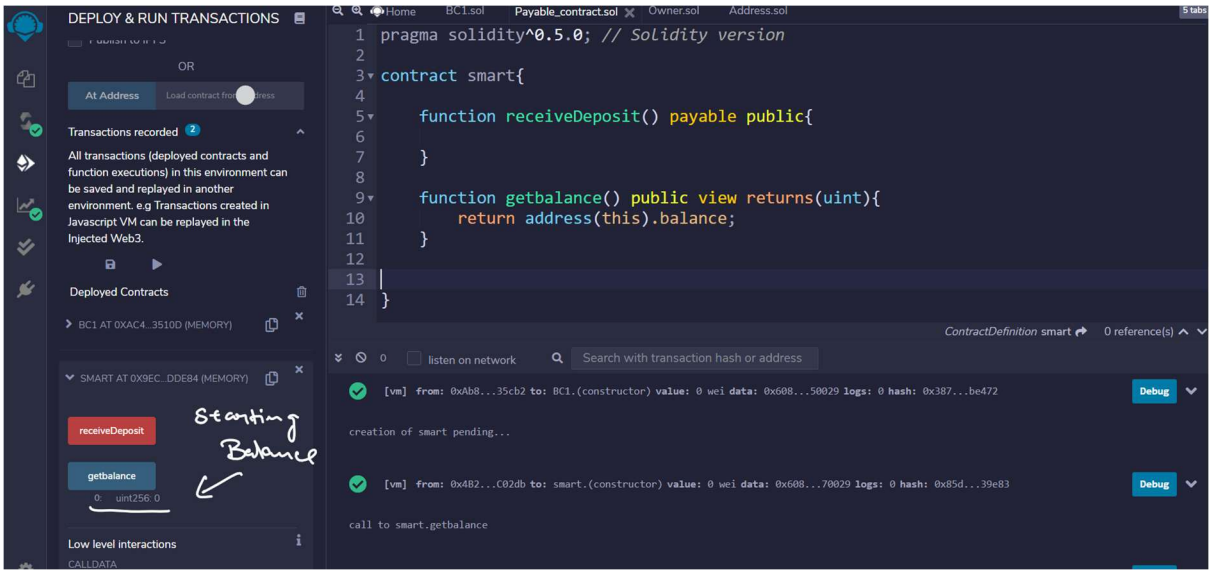
E18CSE206

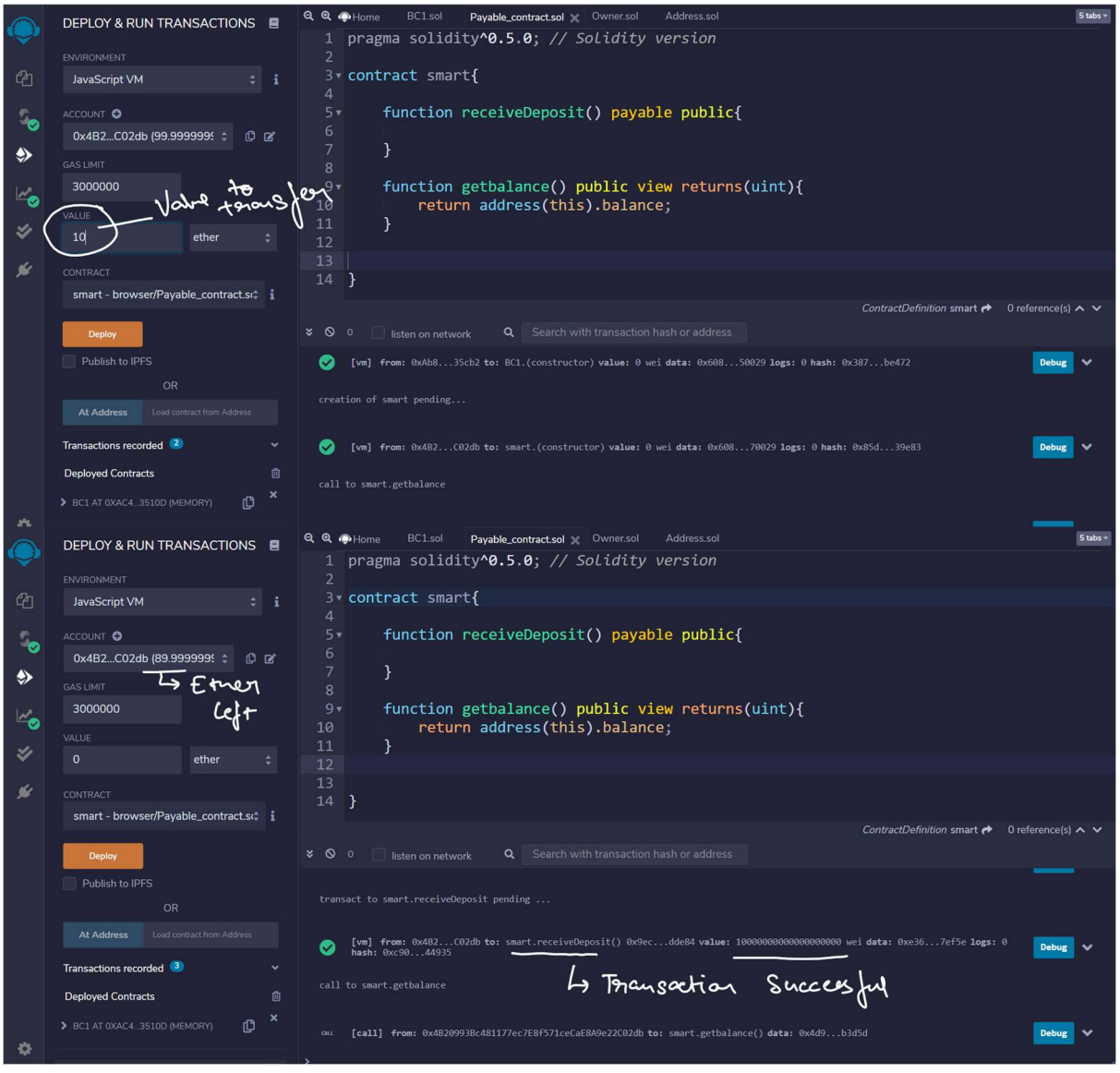
Sample Test :

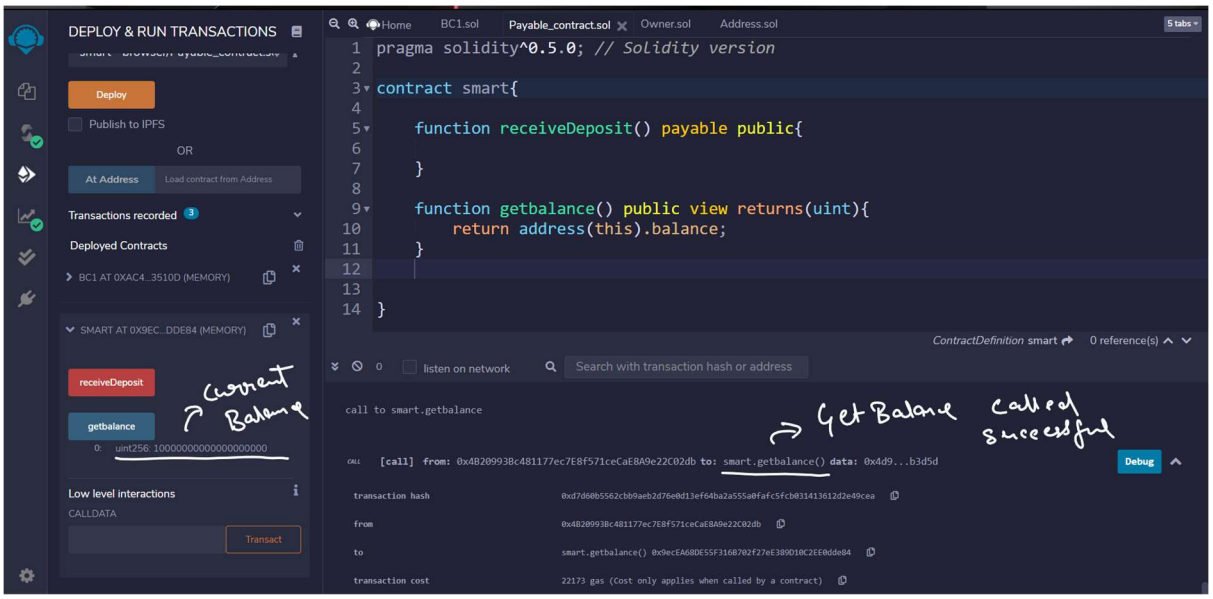


a) Transferring funds from an account to the contract using payable :

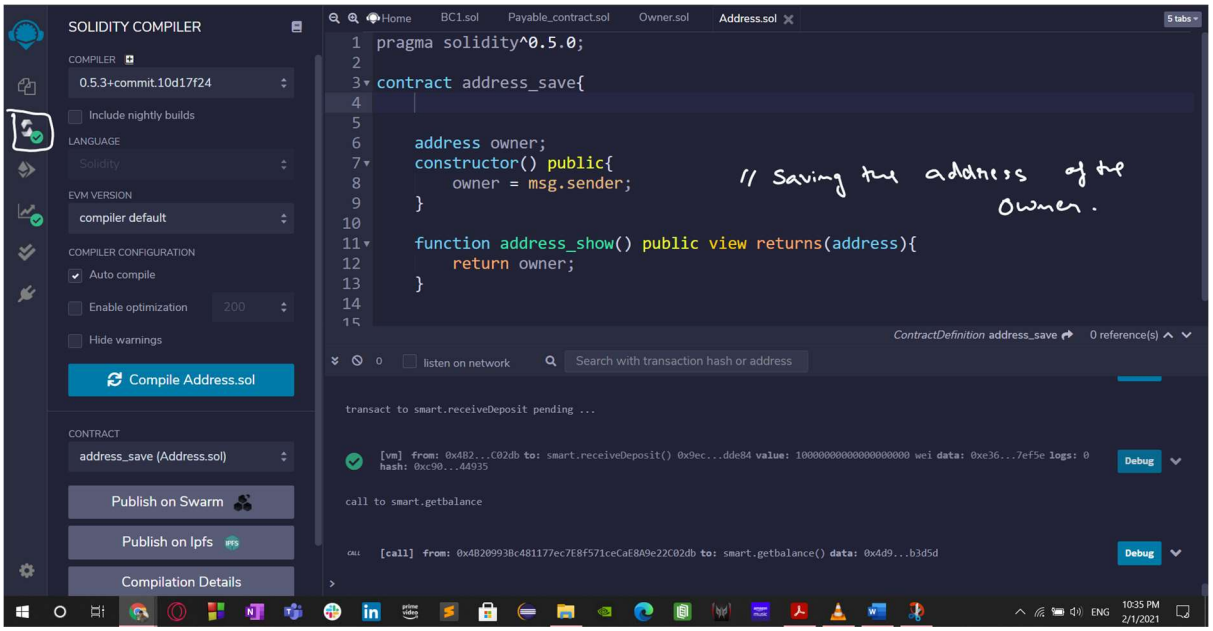


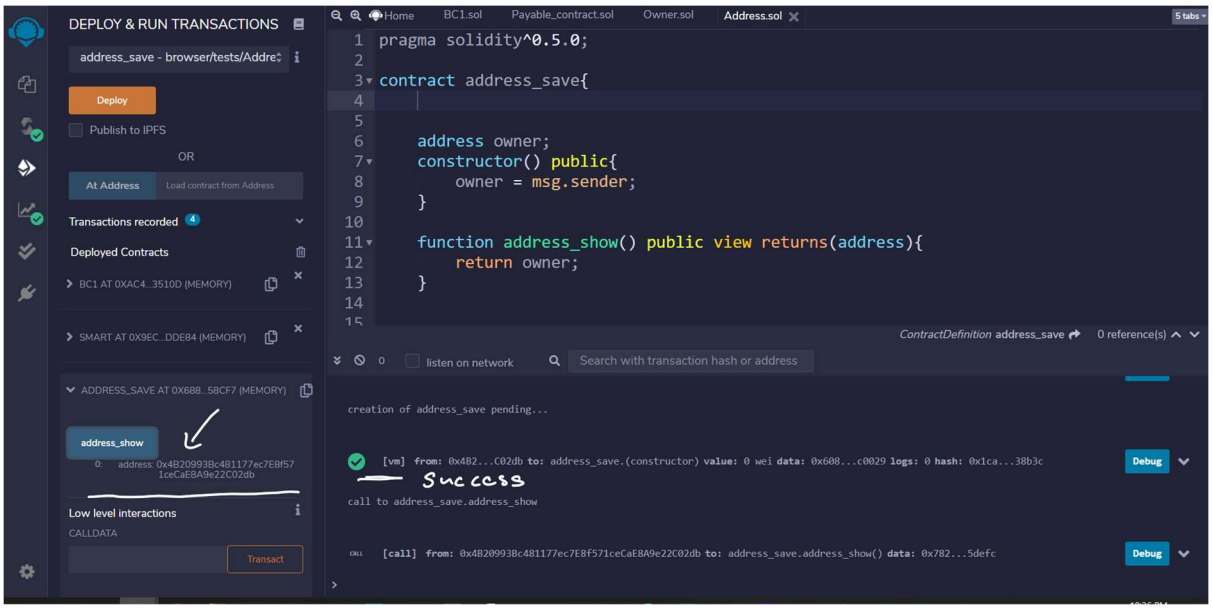

DEPLOY & RUN TRANSACTIONS
OR
At Address
Load contract from Address
Transactions recorded 2
All transactions (deployed contracts and function executions) in this environment can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.
Deployed Contracts
BC1 AT 0XAC4...3510D (MEMORY)
SMART AT 0X9EC...DDE84 (MEMORY)
receiveDeposit
getbalance
0: uint256: 0
Starting Balance
Low level interactions
CALLDATA
Home
BC1.sol
Payable_contract.sol
Owner.sol
Address.sol
5 tabs
pragma solidity^0.5.0; // Solidity version
contract smart{
function receiveDeposit() payable public{
}
function getbalance() public view returns(uint){
return address(this).balance;
}
}
ContractDefinition smart
0 reference(s)
listen on network
Search with transaction hash or address
[vm] from: 0xAb8...35cb2 to: BC1.(constructor) value: 0 wei data: 0x608...50029 logs: 0 hash: 0x387...be472
Debug
creation of smart pending...
[vm] from: 0x4B2...C02db to: smart.(constructor) value: 0 wei data: 0x608...70029 logs: 0 hash: 0x85d...39e83
Debug
call to smart.getbalance


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
JavaScript VM
ACCOUNT
0x4B2...C02db (99.999999
GAS LIMIT
3000000
VALUE
10
ether
Value to transfer
CONTRACT
smart - browser/Payable_contract.sc
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 2
Deployed Contracts
BC1 AT 0XAC4...3510D (MEMORY)
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
JavaScript VM
ACCOUNT
0x4B2...C02db (89.999999
Ether left
GAS LIMIT
3000000
VALUE
0
ether
CONTRACT
smart - browser/Payable_contract.sc
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 3
Deployed Contracts
BC1 AT 0XAC4...3510D (MEMORY)
Home
BC1.sol
Payable_contract.sol
Owner.sol
Address.sol
5 tabs
pragma solidity^0.5.0; // Solidity version
contract smart{
function receiveDeposit() payable public{
}
function getbalance() public view returns(uint){
return address(this).balance;
}
}
ContractDefinition smart
0 reference(s)
listen on network
Search with transaction hash or address
[vm] from: 0xAb8...35cb2 to: BC1.(constructor) value: 0 wei data: 0x608...50029 logs: 0 hash: 0x387...be472
Debug
creation of smart pending...
[vm] from: 0x4B2...C02db to: smart.(constructor) value: 0 wei data: 0x608...70029 logs: 0 hash: 0x85d...39e83
Debug
call to smart.getbalance
transact to smart.receiveDeposit pending ...
[vm] from: 0x4B2...C02db to: smart.receiveDeposit() 0x9ec...dde84 value: 10000000000000000000 wei data: 0xe36...7ef5e logs: 0 hash: 0xc90...44935
Debug
Transaction Successful
call to smart.getbalance
[call] from: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db to: smart.getbalance() data: 0x4d9...b3d5d
Debug

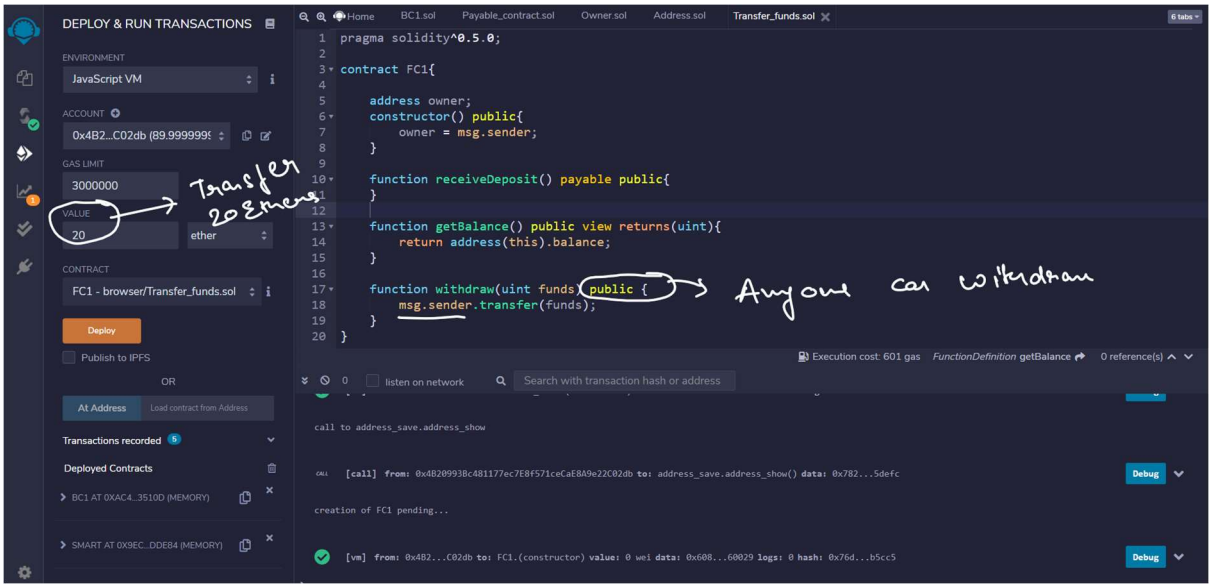


b) Saving the address of the contract creator :

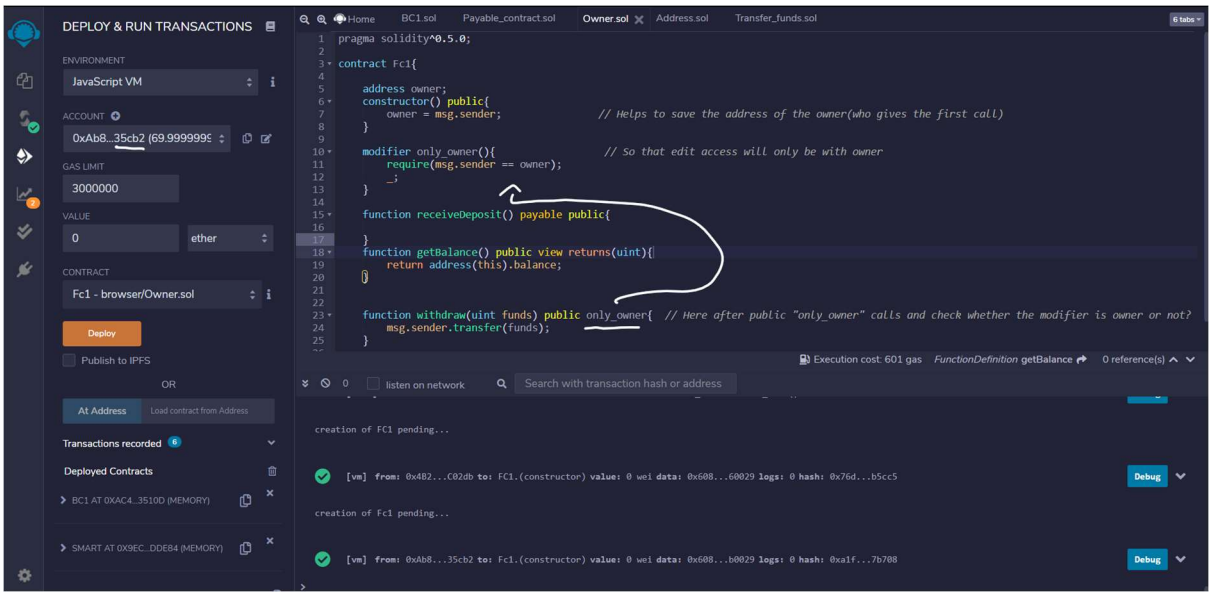


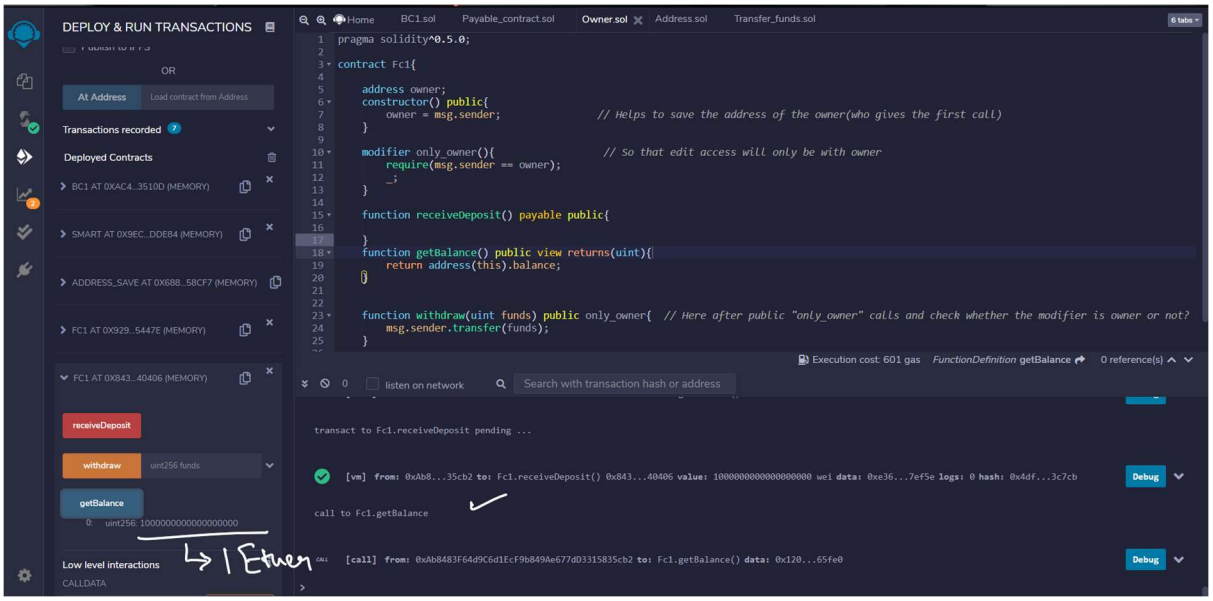


c) Withdrawing funds from the contract to an account :

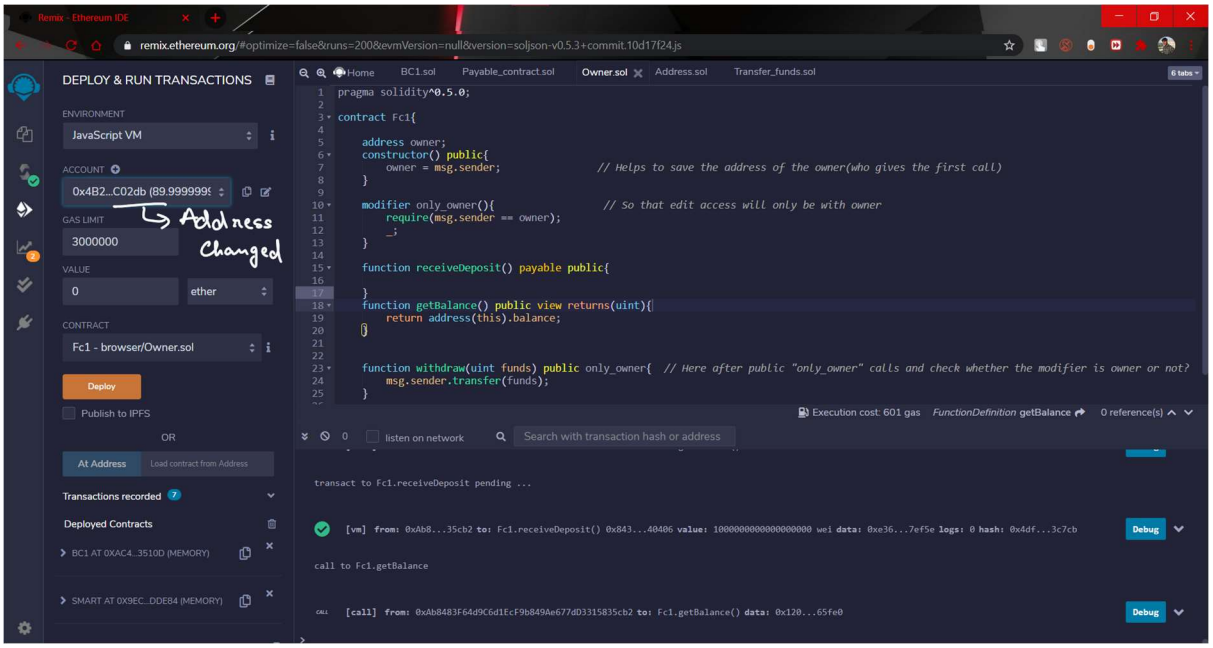


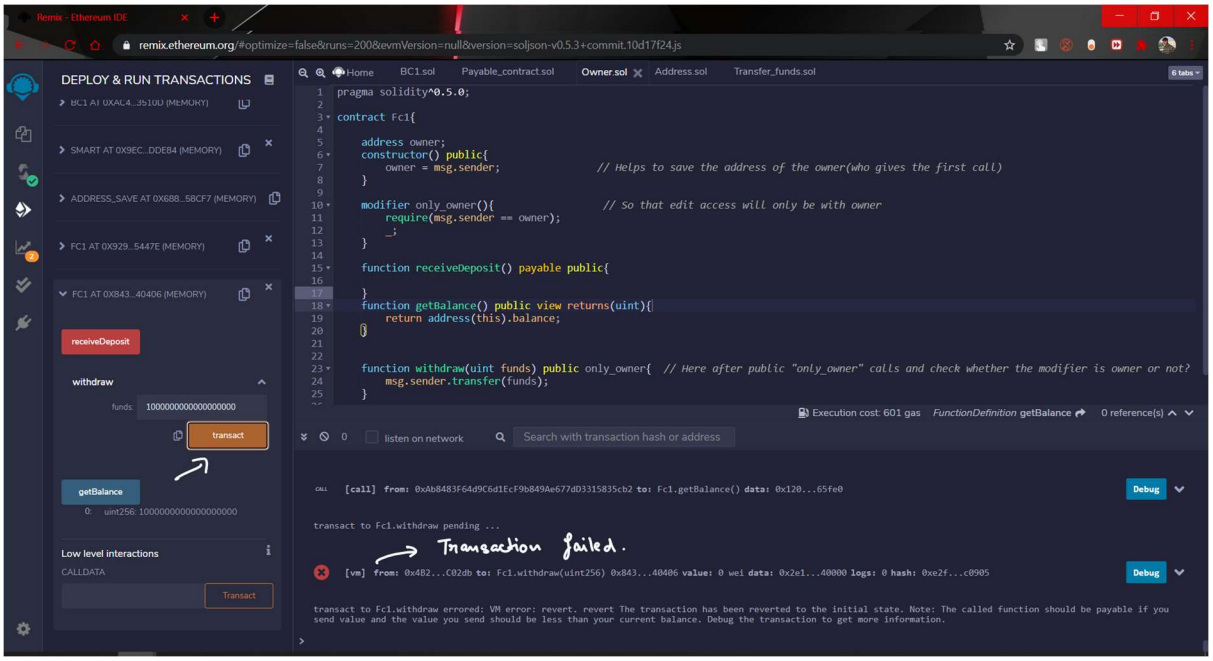
d) Limiting the users' access to functions (Transfer some funds from the contract's balance to the owner). Only the contract's creator is permitted to withdraw.





Let’s try to withdraw this ether from another address:





Only owner can withdraw.