

ASSIGNMENT NO.3

Name: Phule Prathamesh Dnyandev

Roll No.: COBB059

Input:

//SPDX-License-Identifier: MIT

pragma solidity ^0.6;

contract banking{

mapping(address=>uint) public user_account;

mapping(address=>bool) public user_exists;

function create_account() public payable returns(string memory){

require(user_exists[msg.sender]==false,'Account already created');

if(msg.value==0){

user_account[msg.sender]=0;

user_exists[msg.sender]=true;

return "Account created";

}

require(user_exists[msg.sender]==false,"Account already created");

user_account[msg.sender]=msg.value;

user_exists[msg.sender]=true;

return "Account created";

}

function deposit() public payable returns(string memory){

require(user_exists[msg.sender]==true,"Account not created");

require(msg.value>0,"Value for deposit is Zero");

user_account[msg.sender]=user_account[msg.sender]+msg.value;

return "Deposited Successfully";

}

function withdraw(uint amount) public payable returns(string memory){

require(user_account[msg.sender]>amount,"Insufficient Balance");

require(user_exists[msg.sender]==true,"Account not created");

require(amount>0,"Amount should be more than zero");

user_account[msg.sender]=user_account[msg.sender]-amount;

msg.sender.transfer(amount);

return "Withdrawl Successful";

}

function transfer(address payable userAddress, uint amount) public returns(string memory){

require(user_account[msg.sender]>amount,"Insufficient balance in Bank account");

require(user_exists[msg.sender]==true,"Account is not created");

require(user_exists[userAddress]==true,"Transfer account does not exist");

require(amount>0,"Amount should be more than zero");

user_account[msg.sender]=user_account[msg.sender]-amount;

user_account[userAddress]=user_account[userAddress]+amount;

return "Transfer Successful";

}

function send_amt(address payable toAddress, uint256 amount) public payable returns(string memory){

```

    require(user_account[msg.sender]>amount,"Insufficeint balance in Bank account");
    require(user_exists[msg.sender]==true,"Account is not created");
    require(amount>0,"Amount should be more than zero");
    user_account[msg.sender]=user_account[msg.sender]-amount;
    toAddress.transfer(amount);
    return "Transfer Success";
}

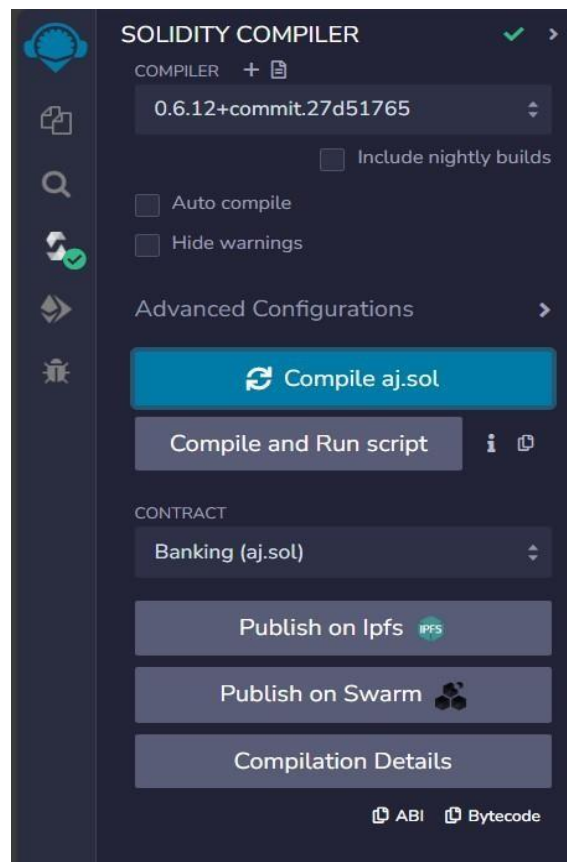
function user_balance() public view returns(uint){
    return user_account[msg.sender];
}

function account_exist() public view returns(bool){
    return user_exists[msg.sender];
}
}

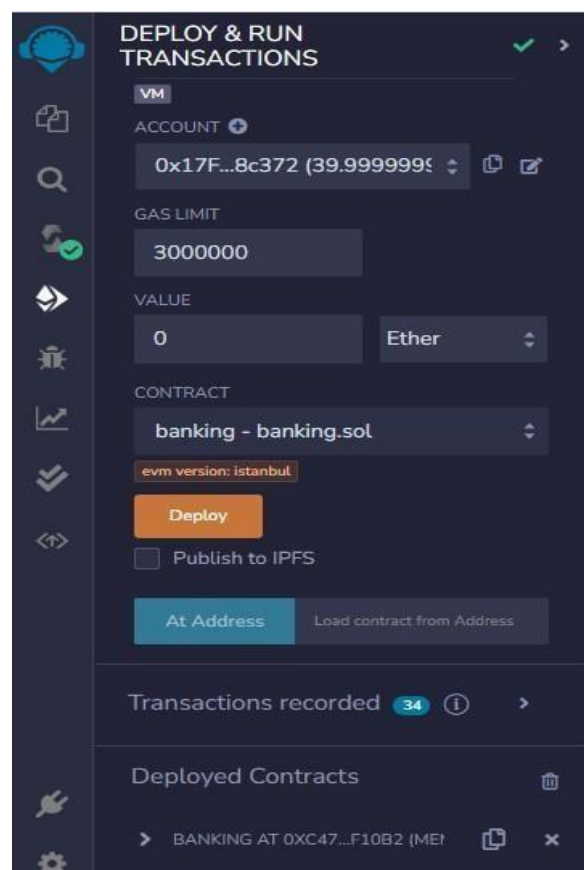
```

Output:

Compile SOL code –



Deploy contract –



Create Account –

The screenshot displays a web interface for a banking contract. On the left, a sidebar shows 'Deployed Contracts' with a balance of 0 ETH. The 'create_account' button is highlighted. The main area shows the transaction details for 'create_account'.

Transaction details:

- transaction hash: 0x997c7a7018b70b002751175f4427ccdb0b7c70fa30ea2e8ed96b03afc7f01
- block hash: 0x508c46b2537196b40f500193996b606acd9b50c20043cb74f5252bbdc68955
- block number: 35
- from: 0x17f6AD8EF982297579C203069C1DbFFE4348c372
- to: banking.create_account() 0xC47e70C386290E4BE561284255e27c50f8CF10b2
- gas: 53562 gas
- transaction cost: 46575 gas
- execution cost: 25511 gas
- input: 0x509...f8633
- decoded input: {}
- decoded output: {"0": "string: Account created"}
- logs: []
- val: 0 wei

Deposit Amount –

The screenshot displays a web interface for a banking contract. On the left, a sidebar shows 'Deployed Contracts' with a balance of 4 ETH. The 'deposit' button is highlighted. The main area shows the transaction details for 'deposit'.

Transaction details:

- transaction hash: 0x025338b6aff635bc40f2f2f20a8db6cadfd7517659e0b0f9bfbcb05f63e85
- block hash: 0xc4c5f785d8daecc3358fe321b25a5117a3af74f684d3053abd3e206138480fd
- block number: 37
- from: 0x17f6AD8EF982297579C203069C1DbFFE4348c372
- to: banking.deposit() 0xC47e70C386290E4BE561284255e27c50f8CF10b2
- gas: 53243 gas
- transaction cost: 46298 gas
- execution cost: 25234 gas
- input: 0xd0e...30db0
- decoded input: {}
- decoded output: {"0": "string: Deposited Successfully"}
- logs: []
- val: 0 wei

Send Amount –

The screenshot displays a web interface for a banking contract. On the left, a sidebar shows 'Deployed Contracts' with a balance of 3.99999999999999995 ETH. The 'send_amt' button is highlighted. The main area shows the transaction details for 'send_amt'.

Transaction details:

- transaction hash: 0x17f6AD8EF982297579C203069C1DbFFE4348c372
- block hash: 0xc4c5f785d8daecc3358fe321b25a5117a3af74f684d3053abd3e206138480fd
- block number: 38
- from: 0x17f6AD8EF982297579C203069C1DbFFE4348c372
- to: banking.send_amt(address,uint256) 0xC47e70C386290E4BE561284255e27c50f8CF10b2
- gas: 42480 gas
- transaction cost: 36869 gas
- execution cost: 15309 gas
- input: 0x00a...00005
- decoded input: {"address toAddress": "0x17f6AD8EF982297579C203069C1DbFFE4348c372", "uint256 amount": "5"}
- decoded output: {"0": "string: Transfer Success"}
- logs: []
- val: 0 wei

Check Account Exists –

The screenshot shows a web interface with a sidebar on the left and a main content area on the right. The sidebar contains several buttons: 'account_exist' (selected), 'user_account', 'user_balance', and 'user_exists'. Below these is a section titled 'Low level interactions' with a 'CALLDATA' input field and a 'Transact' button. The main content area displays transaction details for the 'account_exist' function. It includes the 'to' address, 'execution cost' (2494 gas), 'input' (0xcde...6e57b), 'decoded input' ({}), 'decoded output' ({"0": "bool: true"}), and 'logs' ([]).

Field	Value
to	banking.account_exist() 0xc47e78c386298e48e561284255e27c50f8cf10b2
execution cost	2494 gas (Cost only applies when called by a contract)
input	0xcde...6e57b
decoded input	{}
decoded output	{ "0": "bool: true" }
logs	[]

Check User Balance –

The screenshot shows the same web interface as before, but with the 'user_balance' button selected in the sidebar. The 'Low level interactions' section now shows a 'CALLDATA' input field with the value '0: uint256: 3999999999999999995' and a 'Transact' button. The main content area displays transaction details for the 'user_balance' function. It includes the 'decoded input' ({}), 'decoded output' ({"0": "uint256: 3999999999999999995"}), and 'logs' ([]).

Field	Value
decoded input	{}
decoded output	{ "0": "uint256: 3999999999999999995" }
logs	[]

Withdraw Amount –

The screenshot shows the same web interface, but with the 'transfer' button selected in the sidebar. The 'Low level interactions' section now shows a 'CALLDATA' input field with the value 'amount: "50"' and a 'Transact' button. The main content area displays transaction details for the 'transfer' function. It includes the 'execution cost' (15250 gas), 'input' (0x2e1...00032), 'decoded input' ({"uint256 amount": "50"}), 'decoded output' ({"0": "string: Withdrawal Successful"}), and 'logs' ([]).

Field	Value
execution cost	15250 gas
input	0x2e1...00032
decoded input	{ "uint256 amount": "50" }
decoded output	{ "0": "string: Withdrawal Successful" }
logs	[]