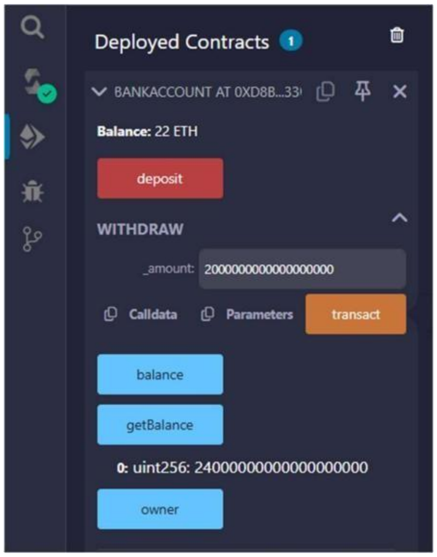
Name: Hanuman Bavane
Roll No: 14108
Class: BE – A – A1

Practical 3

Program:

```
// SPDX-License-Identifier: MIT
// Specifies the license under which the code is released.
pragma solidity ^0.8.18;

/**
* @title BankAccount
* @dev A simple smart contract that allows a user to deposit, withdraw,  * and check
  their balance in Ether.
 */
contract BankAccount {

   // State variable to store the balance of the account in Wei.     //
'public' keyword automatically creates a getter function for it.
   uint256 public balance;

   // The address of the person who deploys the contract.
   address public owner;

   // Event to log deposits for a transparent transaction history.
   event Deposit(address indexed account, uint amount);      // Event
to log withdrawals for a transparent transaction history.
   event Withdraw(address indexed account, uint amount);

   /**
* @dev The constructor is called only once when the contract is deployed.      * It sets the
  deployer of the contract as the 'owner'.
    */
   constructor() {
owner = msg.sender;
   }

   /**
* @dev Allows anyone to deposit Ether into this contract.
* 'payable' is a special keyword that allows a function to receive Ether.
    */
   function deposit() public payable {
      // 'msg.value' holds the amount of Ether sent with the transaction.
      balance += msg.value;
      // Log the deposit event to the blockchain.
emit Deposit(msg.sender, msg.value);
   }

   /**
* @dev Allows only the owner to withdraw a specific amount of Ether.
* @param _amount The amount of Ether (in Wei) to withdraw.
    */
   function withdraw(uint256 _amount) public {
      // 1. Check if the person calling the function is the owner.
      require(msg.sender == owner, "Only the owner can withdraw.");
      // 2. Check if the requested withdrawal amount is available in the balance.
      require(_amount <= balance, "Insufficient balance for withdrawal.");

      // 3. Subtract the amount from the balance *before* sending Ether.
      // This is a crucial security step to prevent re-entrancy attacks.
      balance -= _amount;

      // 4. Transfer the Ether to the owner's address.
      payable(msg.sender).transfer(_amount);

      // 5. Log the withdrawal event.
      emit Withdraw(msg.sender, _amount);
   }

   /**
* @dev A function to view the contract's current balance.
* Note: While the 'balance' variable is public, this function provides a      * clear, explicit way
  for users or other contracts to query it.
* 'view' means it doesn't change the state of the blockchain and doesn't cost gas to call.      *
  @return The current balance in Wei.
    */
   function getBalance() public view returns (uint256) {
return balance;
   }
}
```

Output:



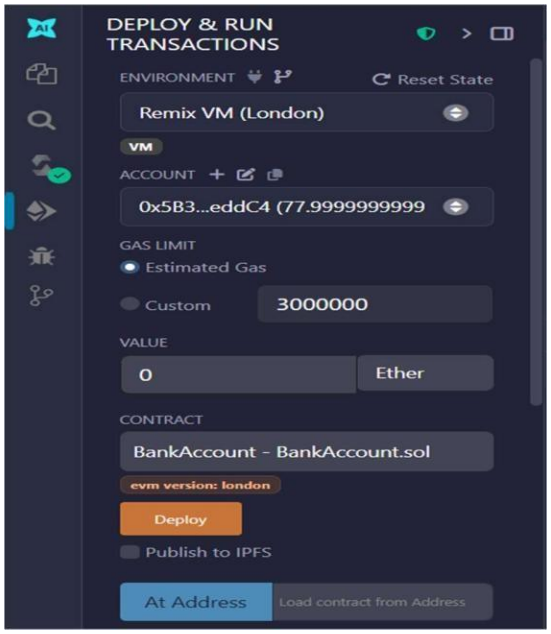
Deployed Contracts 1

BANKACCOUNT AT 0XD8B...33

Balance: 22 ETH

_amount: 2000000000000000000

0: uint256: 24000000000000000000



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Remix VM (London)

ACCOUNT: 0x5B3...eddC4 (77.9999999999

GAS LIMIT

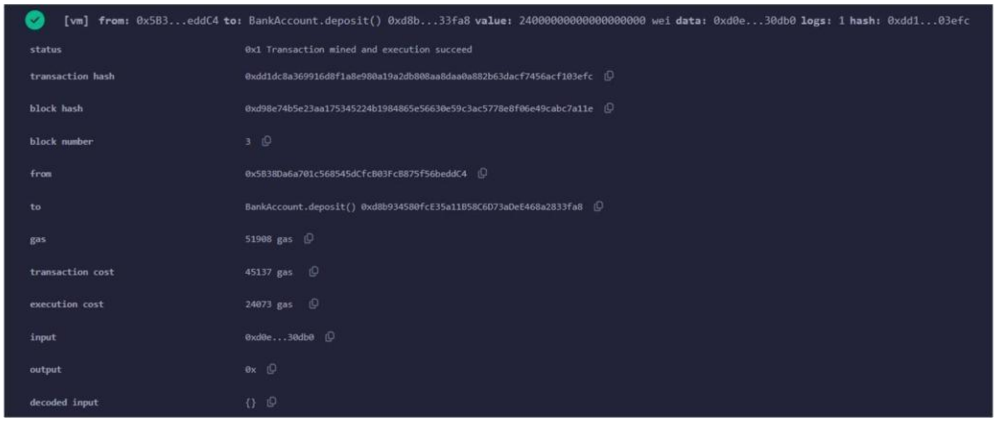
● Estimated Gas

○ Custom 3000000

VALUE: 0 Ether

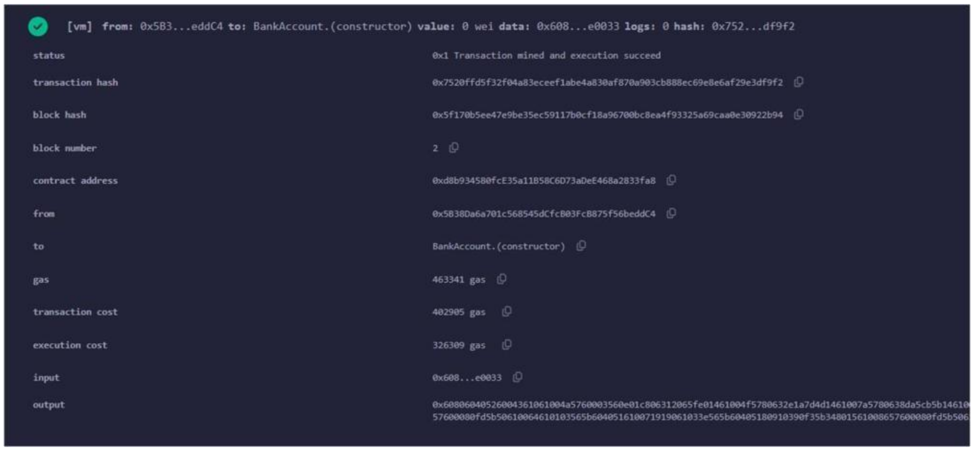
CONTRACT: BankAccount - BankAccount.sol

evm version: london



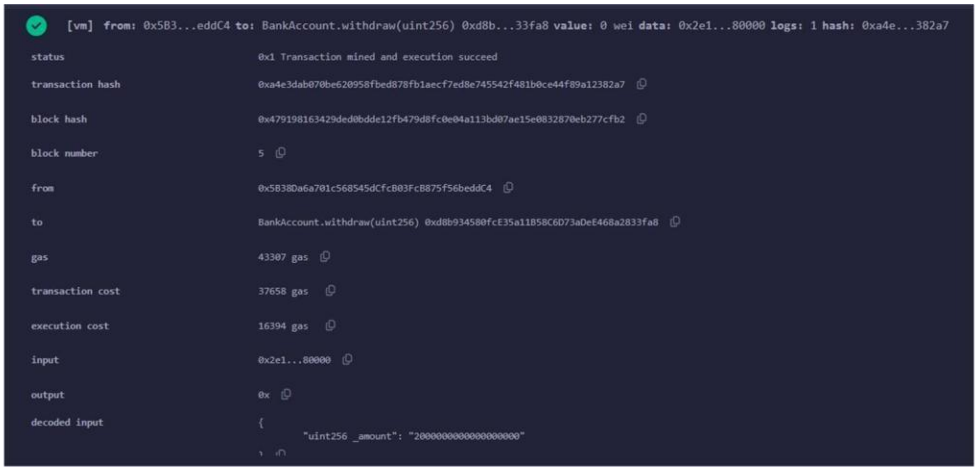
```
[vm] from: 0x5B3...eddC4 to: BankAccount.deposit() 0xd8b...33fa8 value: 24000000000000000000 wei data: 0xd0e...30db0 logs: 1 hash: 0xdd1...03efc
status                  0x1 Transaction mined and execution succeed
transaction hash        0xdd1dc8a369916d8f1a8e980a19a2db808aa8daa0a882b63dacf7456acf103efc
block hash              0xd98e74b5e23aa175345224b1984865e56630e59c3ac5778e8f06e49cabc7a11e
block number            3
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      BankAccount.deposit() 0xd8b934580fcE35a11B58C6D73aDeE468a2833fa8
gas                     51908 gas
transaction cost        45137 gas
execution cost          24073 gas
input                   0xd0e...30db0
output                  0x
decoded input           {}
```

Deposit



```
[vm] from: 0x5B3...eddC4 to: BankAccount.(constructor) value: 0 wei data: 0x608...e0033 logs: 0 hash: 0x752...df9f2
status                  0x1 Transaction mined and execution succeed
transaction hash        0x7520ffd5f32f04a83eceef1abe4a830af870a903cb888ec69e8e6af29e3df9f2
block hash              0x5f170b5ee47e9be35ec59117b0cf18a96700bc8ea4f93325a69caa0e30922b94
block number            2
contract address        0xd8b934580fcE35a11B58C6D73aDeE468a2833fa8
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      BankAccount.(constructor)
gas                     463341 gas
transaction cost        402905 gas
execution cost          326309 gas
input                   0x608...e0033
output                  0x608060405260043610610004a5760003560e01c806312065fe01461004f5780632e1a7d4d1461007a5780638da5cb5b14610
                        57600080fd5b506100646101035656b60405161007191906103e565b60405180910390f35b348015610086576000080fd5b506
```

Check Balance



```
[vm] from: 0x5B3...eddC4 to: BankAccount.withdraw(uint256) 0xd8b...33fa8 value: 0 wei data: 0x2e1...80000 logs: 1 hash: 0xa4e...382a7
status                  0x1 Transaction mined and execution succeed
transaction hash        0xa4e3dab070be620958fbed878fb1aecf7ed8e745542f481b0ce44f89a12382a7
block hash              0x479198163429ded0bdde12fb479d8fc0e04a113bd07ae15e0832870eb277cfb2
block number            5
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      BankAccount.withdraw(uint256) 0xd8b934580fcE35a11B58C6D73aDeE468a2833fa8
gas                     43307 gas
transaction cost        37658 gas
execution cost          16394 gas
input                   0x2e1...80000
output                  0x
decoded input           {
                                "uint256 _amount": "2000000000000000000"
                        }
```

Withdraw