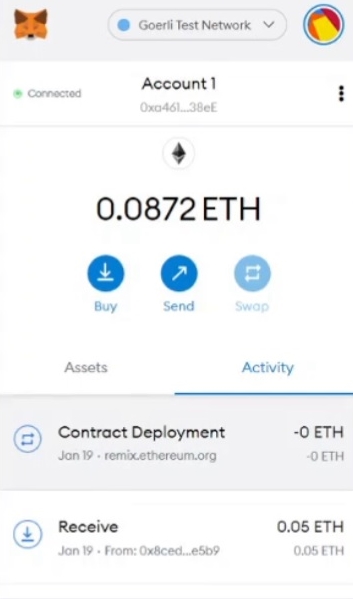
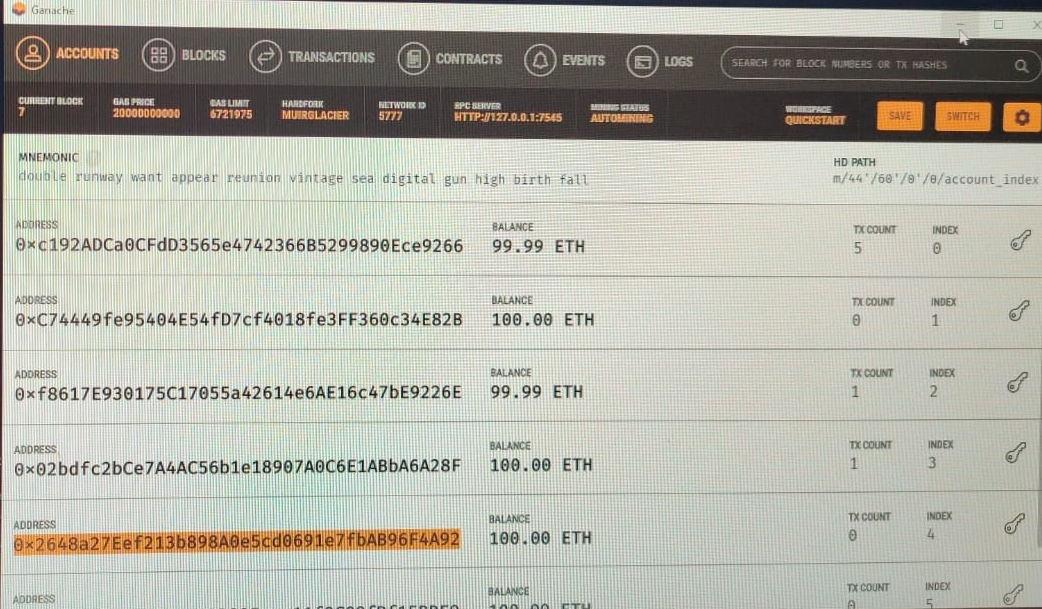
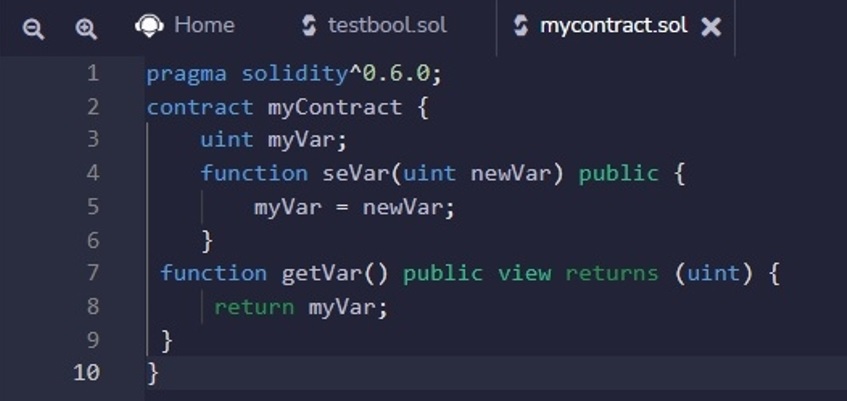
**Assignment-1**

1. Create multiple accounts in Metamask and perform the balance transfer between the various accounts.

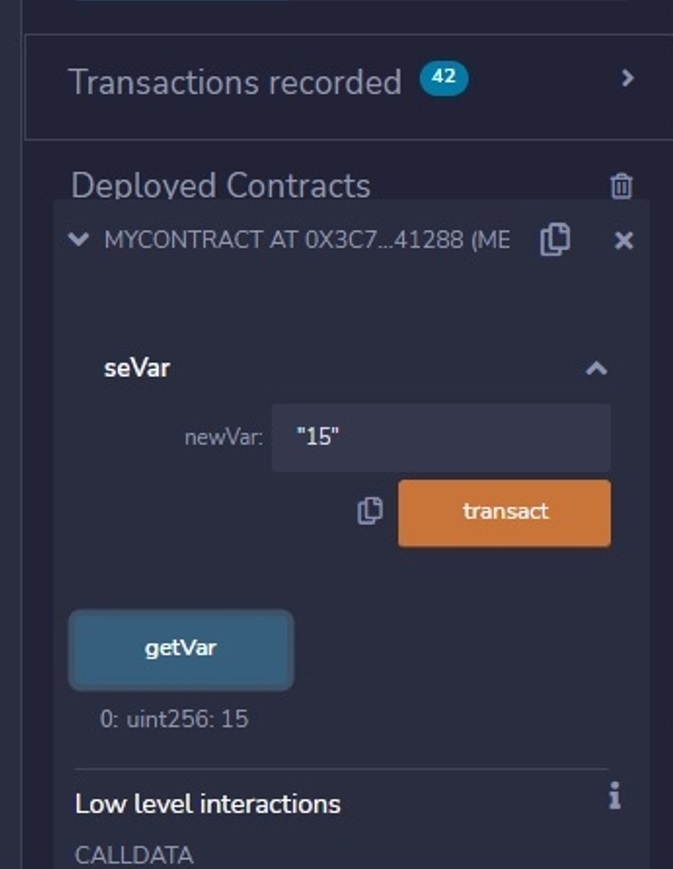




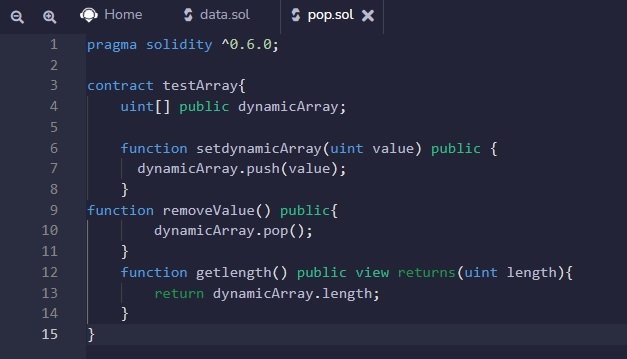
2. Write a solidity program to set variables and get variables.



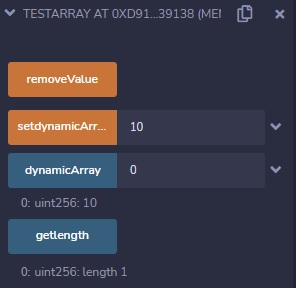
OUTPUT:



 3. Write a solidity program to perform push and pop operations on dynamic array.



OUTPUT:



4. Write a solidity program to set address with a mapping variable

pragma solidity ^0.6.0;

contract testMapping{

    mapping (uint => bool) public myMapping;

    mapping (address => bool) public myAddress;

     function setValue(uint index) public{

        myMapping[index] = true;

    }

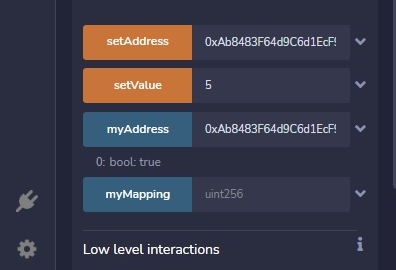
    function setAddress(address add) public {

        myAddress[msg.sender]=true;

    }

}

OUTPUT:



5. Write a solidity program to get the factorial of a number

pragma solidity ^0.6.0;

contract Factorial {

  function fact(uint x) public view returns (uint y) {

    if (x == 0) {

      return 1;

    }

    else {

     return x= x\*fact(x-1);

    }

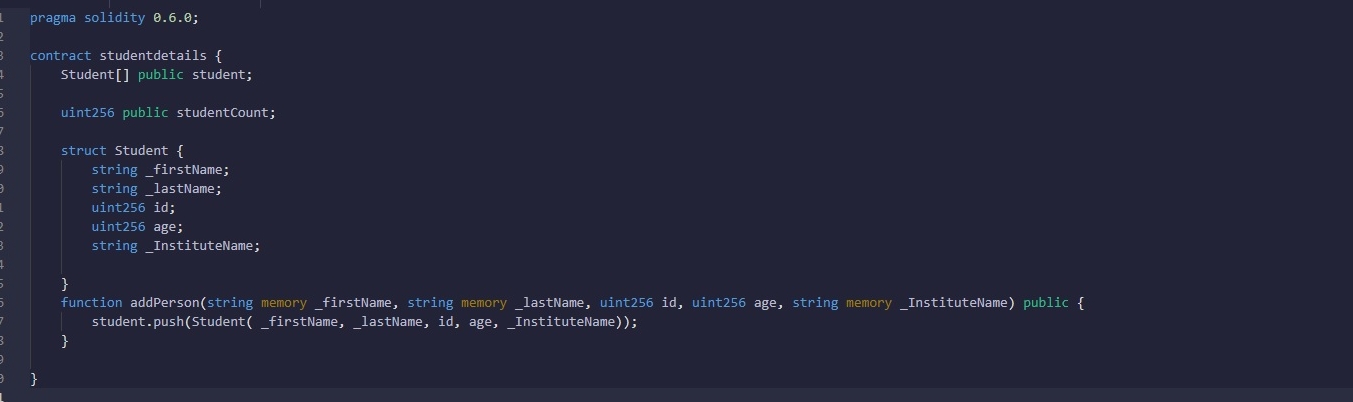
  }

}

OUTPUT:

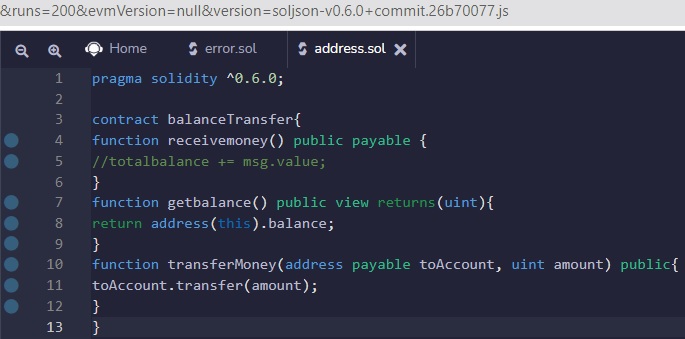


6. Write a solidity program to store information of a student(Name, Roll.No, Institute, Age) using structure

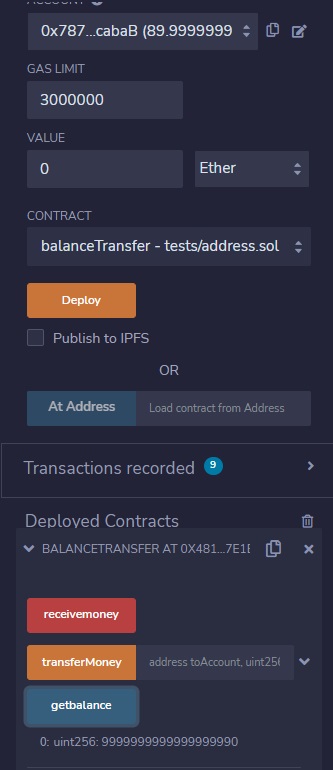


OUTPUT:

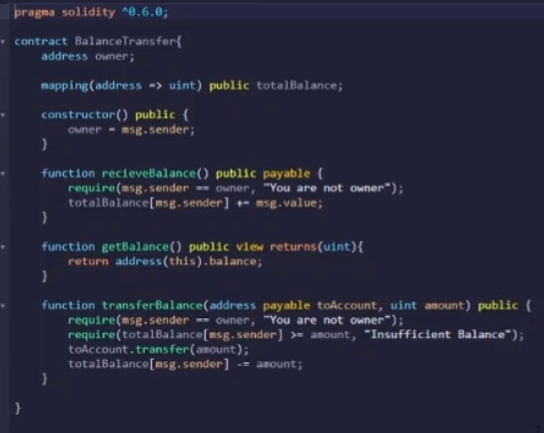


7. Write a smart contract using a solidity program to perform balance transfer from contract to other accounts.

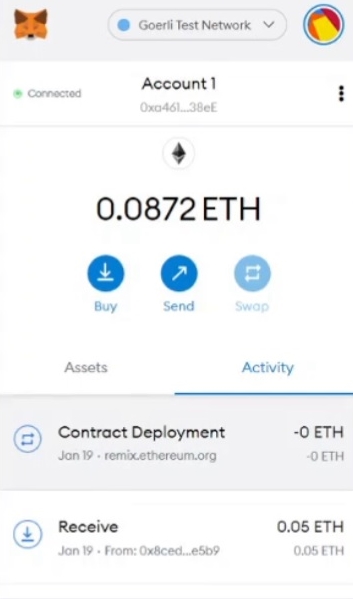
Output:



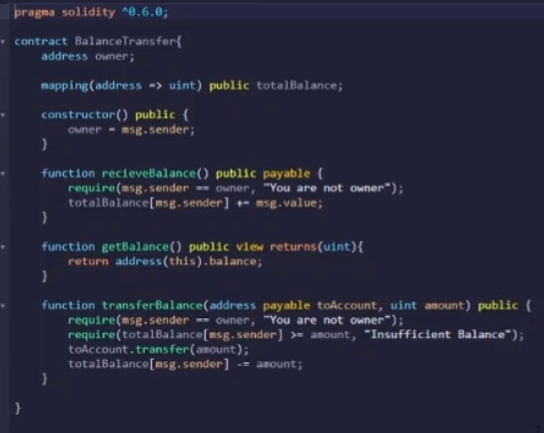
8. Write a smart contract using a solidity program to perform balance transfer with mapping and make sure only the owner can transfer the balance from contract to other contract.



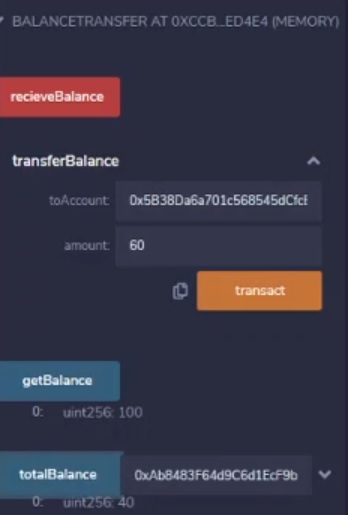
 Output:



9. Write a solidity program to perform the exception handling and describe the details with screenshots.

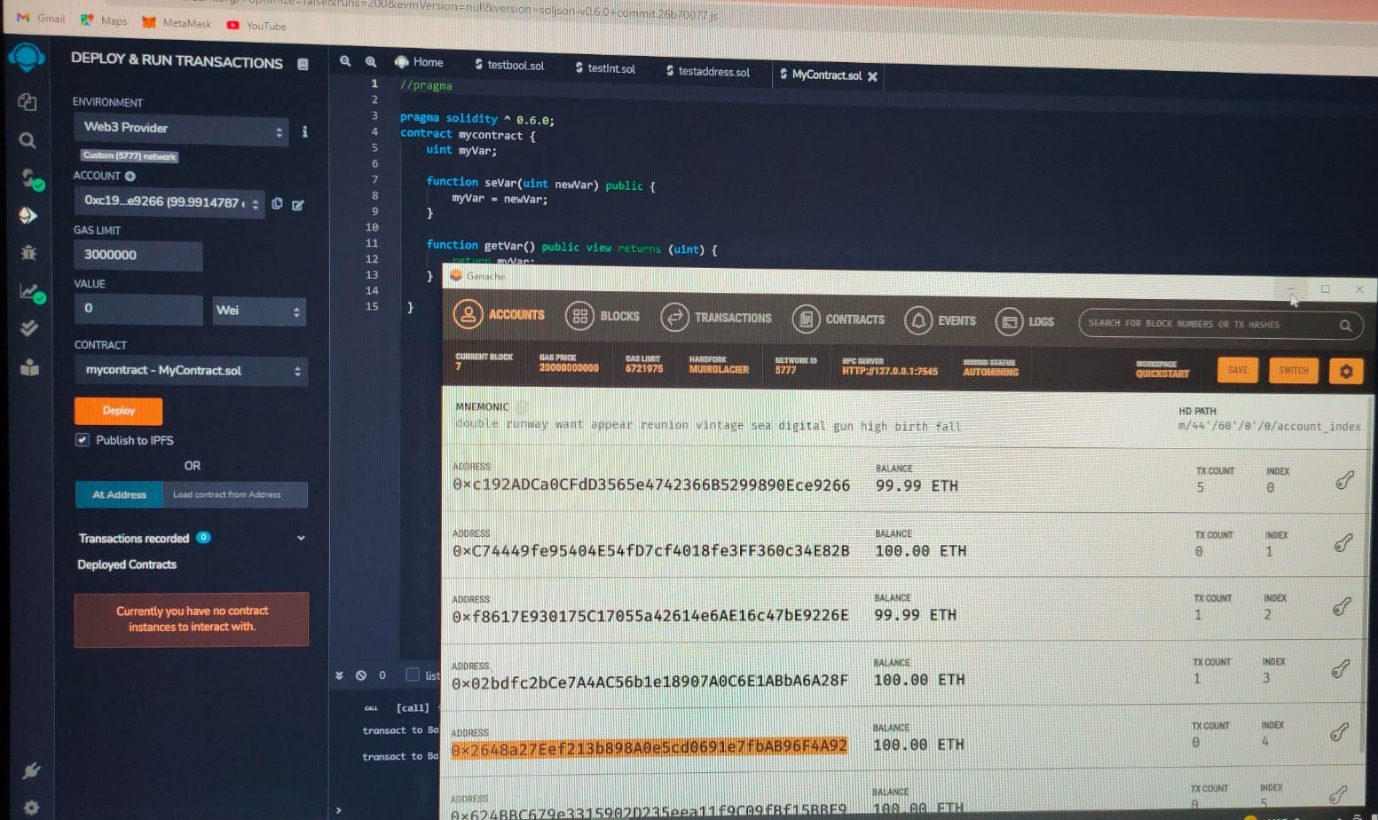


Output:



10. Connect the following tools with the remix environment and perform balance transfer between the accounts with smart contract and share the screen shots.

1. **Ganache:**



**b) Metamask:**

