

EX.NO: 1

Basic Programs Using Solidity

Date:27/06/2023

Aim:

To implement the basic programs using solidity smart control in remix IDE

Procedure:

Step 1: Open the Remix IDE

Step 2: Create a new file by using .sol Extension

Step 3: Rename the file and give the basic coding for the basic programs

Step 4: Select the Compiler version using compilation section

Step 5: Also select the Environment to compile the program

Step 6: Compile the program by using Compile option of the file

Step 7: Then Deploy the compiled program for transaction

Step 8: Get the Input of the program by using the get and set method

Step 9: Then transact or call the input

Step 10: Output will be printed then stop the transaction

Programs:

```
1.pragma solidity >= 0.5.1< 0.9.0;
```

```
contract MyContract {
```

```
    Person[] public people;
```

```
    uint256 public peopleCount;
```

```
    struct Person {
```

```
        string _firstName;
```

```
        string _lastName;
```

```
    }
```

```
    function addPerson(string memory _firstName, string memory _lastName) public {
```

```
        people.push(Person(_firstName, _lastName));
```

```
        peopleCount += 1;
```

```
    }
```

```
}
```

Output:

MYCONTRACT AT 0XD91...39138

Balance: 0 ETH

addPerson

_firstName: Alen

_lastName: Stew

Calldata Parameters **transact**

people

: 1

to: MyContract.addPerson(string,string) 0xd9145CCE52D386f254917e481e844e9943F39138

gas: 130728 gas

transaction cost: 113676 gas

execution cost: 91700 gas

input: 0x22f...00000

decoded input: { "string_firstName": "Alen", "string_lastName": "Stew" }

MYCONTRACT AT 0XD91...39138

Balance: 0 ETH

addPerson

_firstName: Alen

_lastName: Stew

Calldata Parameters **transact**

people

: 1

CALL [call] from: 0x5B380a6a701c568545dCfcB03FcB875f56beddC4 to: MyContract.peopleCount() data: 0x267...c68e0

from: 0x5B380a6a701c568545dCfcB03FcB875f56beddC4

to: MyContract.peopleCount() 0xd9145CCE52D386f254917e481e844e9943F39138

execution cost: 2429 gas (Cost only applies when called by a contract)

input: 0x267...c68e0

decoded input: {}

decoded output: { "0": "uint256: 1" }

2. pragma solidity >= 0.5.0 <0.9.0;

```
contract HelloWorld{
    string userInput;

    function set(string memory finalValue) public
    {
        userInput = finalValue;
    }

    function get() public view returns(string memory){
        return userInput;
    }
}
```

Output:

The first screenshot shows the 'Low level interactions' panel on the left and the transaction details on the right. The transaction details include:

- transaction cost: 45126 gas
- execution cost: 23642 gas
- input: 0x4ed...00000
- decoded input: { "string finalValue": "3" }
- decoded output: {}
- logs: []
- val: 0 wei

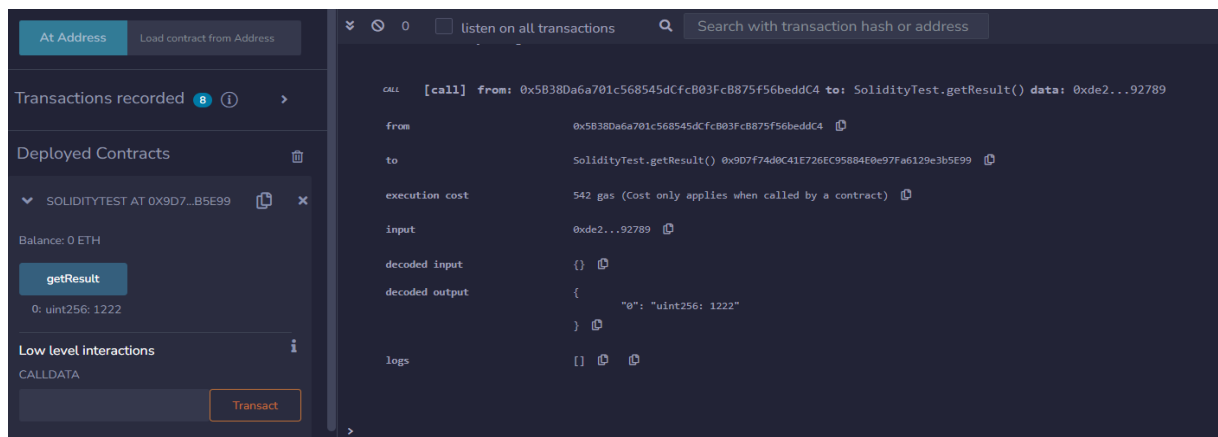
The second screenshot shows the 'Low level interactions' panel on the left and the call details on the right. The call details include:

- val: 0 wei
- call to HelloWorld.get
- CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: HelloWorld.get() data: 0x6d4...ce63c
- from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
- to: HelloWorld.get() 0xf8e81D47203A594245E36C48e151709F0C19f8e8
- execution cost: 3431 gas (Cost only applies when called by a contract)
- input: 0x6d4...ce63c
- decoded input: {}
- decoded output: { "0": "string: 3" }

3. `pragma solidity >=0.5.1 < 0.9.0;`

```
contract SolidityTest {  
    constructor() public {  
    }  
    function getResult() public view returns(uint){  
        uint a = 1000;  
        uint b = 222;  
        uint result = a + b;  
        return result;  
    }  
}
```

Output:



4. pragma solidity >=0.5.0<0.9.1;

```
contract LedgerBalance {  
    mapping(address => uint) public balances;  
    function updateBalance(uint newBalance) public {  
        balances[msg.sender] = newBalance;  
    }  
}  
  
contract Updater {  
    function updateBalance() public returns (uint) {  
        LedgerBalance ledgerBalance = new LedgerBalance();  
        ledgerBalance.updateBalance(10);  
        return ledgerBalance.balances(address(this));  
    }  
}
```

Output:

Transactions recorded **17** ⓘ >

Deployed Contracts ⓘ

▼ LEDGERBALANCE AT 0xD4F...2C ⓘ ✕

Balance: 0 ETH

updateBalance ^

newBalance: 10

Calldata Parameters **transact**

balances ^

: 10

Calldata Parameters **call**

0 ⏸ ☐ listen on all transactions 🔍 Search with transaction hash or address

from	0x58380a6a701c568545dCfc803FcB875f56beddC4 ⓘ
to	LedgerBalance.updateBalance(uint256) 0xD4Fc541236927E2Eaf8F27606b07309C1Fc2cbee ⓘ
gas	50385 gas ⓘ
transaction cost	43813 gas ⓘ
execution cost	22609 gas ⓘ
input	0xba7...0000a ⓘ
decoded input	{ "uint256 newBalance": "10" } ⓘ
decoded output	{ } ⓘ
logs	[] ⓘ ⓘ
val	0 wei ⓘ

Result:

- Thus all the basic programs has been implemented and executed out successfully using remix IDE.