Practical 4

Practical - Write a program in solidity to create Student data. Use the following constructs:

Structures
Arrays
Fallback

Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas values.

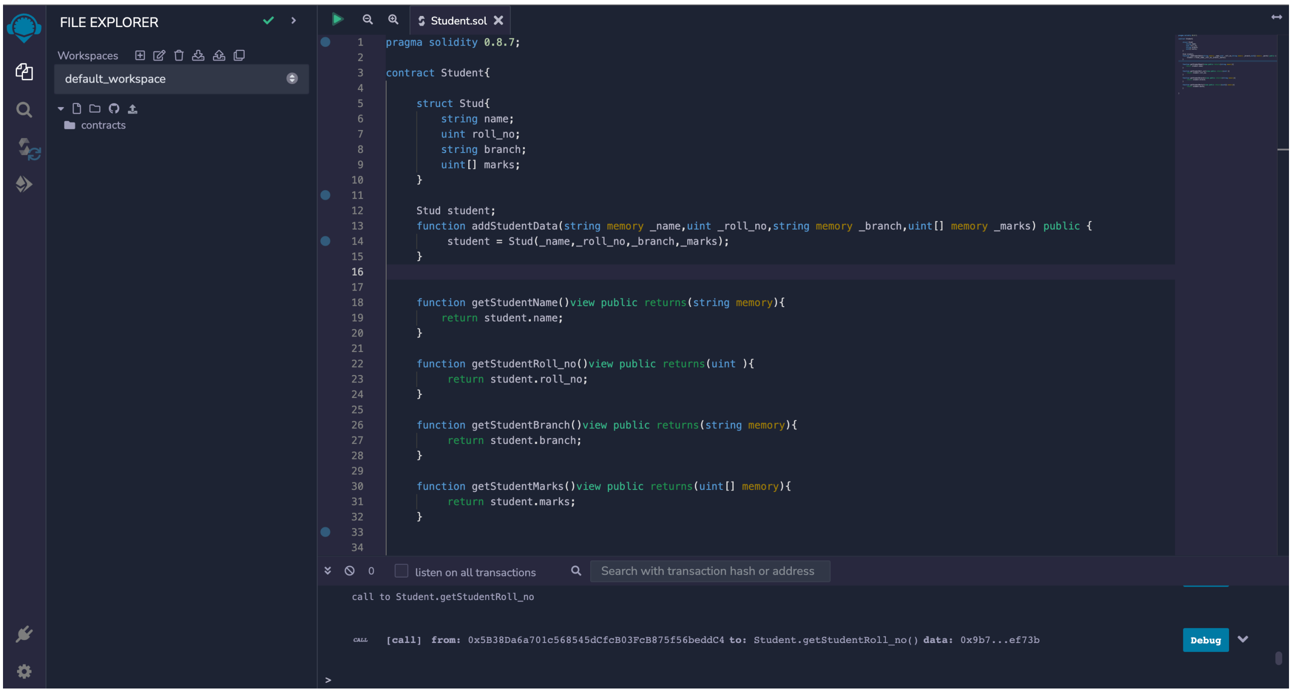
Code -

```
pragma solidity 0.8.7;

contract Student{

    struct Stud{
       string name;
       uint roll_no;
       string branch;
       uint[] marks;
    }

    Stud student;
    function addStudentData(string memory _name,uint _roll_no,string memory
_branch,uint[] memory _marks) public {
        student = Stud(_name,_roll_no,_branch,_marks);
    }


    function getStudentName()view public returns(string memory){
       return student.name;
    }

    function getStudentRoll_no()view public returns(uint ){
        return student.roll_no;
    }

    function getStudentBranch()view public returns(string memory){
        return student.branch;
    }

    function getStudentMarks()view public returns(uint[] memory){
        return student.marks; }
}
```

Execution -

Code -

Writing code about saving student data via contract in solidity.



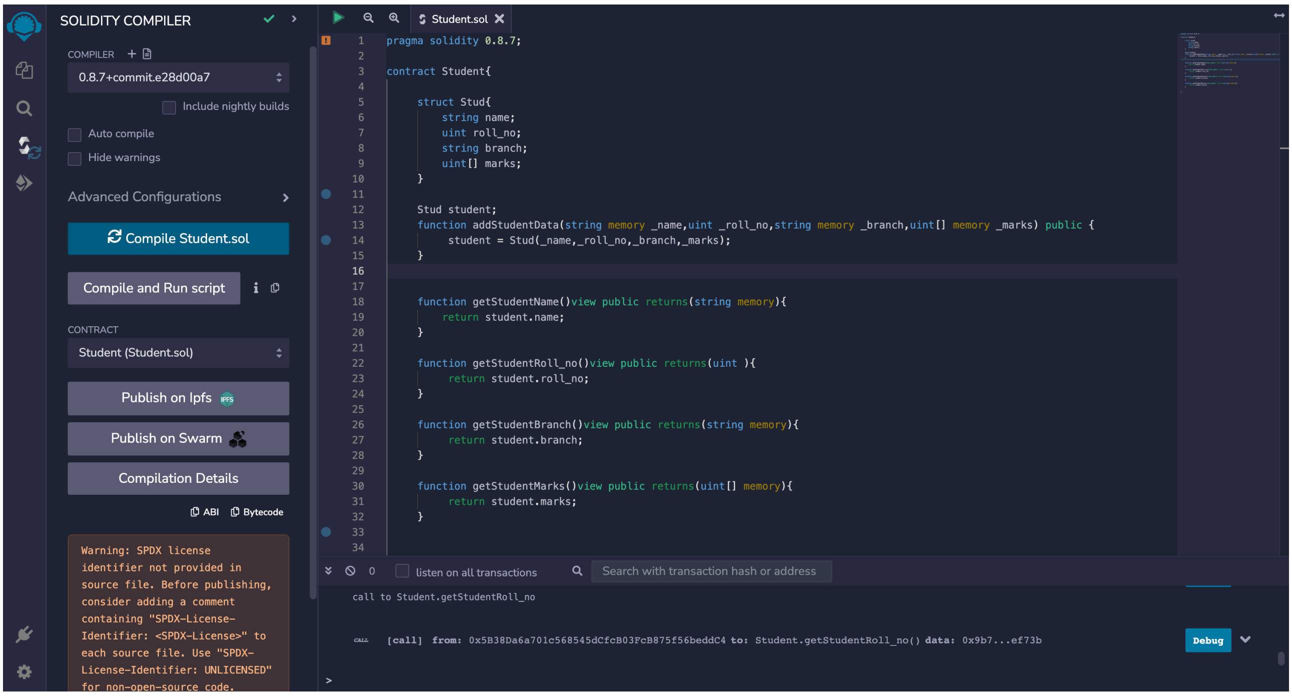
```
pragma solidity 0.8.7;

contract Student{

    struct Stud{
        string name;
        uint roll_no;
        string branch;
        uint[] marks;
    }

    Stud student;
    function addStudentData(string memory _name,uint _roll_no,string memory _branch,uint[] memory _marks) public {
        student = Stud(_name,_roll_no,_branch,_marks);
    }


    function getStudentName()view public returns(string memory){
        return student.name;
    }

    function getStudentRoll_no()view public returns(uint ){
        return student.roll_no;
    }

    function getStudentBranch()view public returns(string memory){
        return student.branch;
    }

    function getStudentMarks()view public returns(uint[] memory){
        return student.marks;
    }

```

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

```
call to Student.getStudentRoll_no

CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Student.getStudentRoll_no() data: 0x9b7...ef73b
```

Compile -

Code get complied that contract with name of Student.



COMPILER
0.8.7+commit.e28d00a7

☐ Include nightly builds

☐ Auto compile

☐ Hide warnings

CONTRACT
Student (Student.sol)

```
Warning: SPDX license
identifier not provided in
source file. Before publishing,
consider adding a comment
containing "SPDX-License-
Identifier: <SPDX-License>" to
each source file. Use "SPDX-
License-Identifier: UNLICENSED"
for non-open-source code.
```

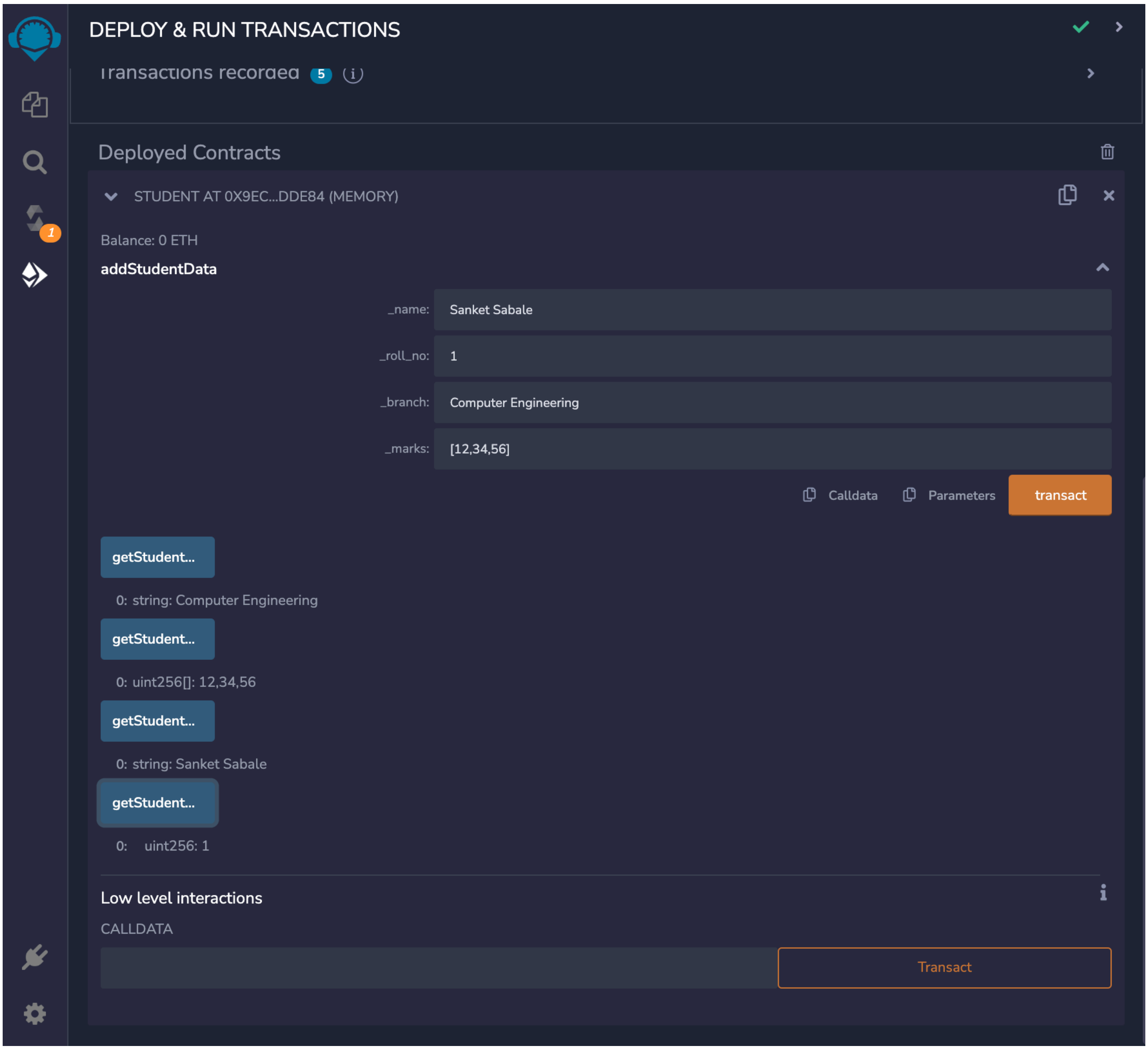
```
pragma solidity 0.8.7;

contract Student{

    struct Stud{
        string name;
        uint roll_no;
        string branch;
        uint[] marks;
    }

    Stud student;
    function addStudentData(string memory _name,uint _roll_no,string memory _branch,uint[] memory _marks) public {
        student = Stud(_name,_roll_no,_branch,_marks);
    }


    function getStudentName()view public returns(string memory){
        return student.name;
    }

    function getStudentRoll_no()view public returns(uint ){
        return student.roll_no;
    }

    function getStudentBranch()view public returns(string memory){
        return student.branch;
    }

    function getStudentMarks()view public returns(uint[] memory){
        return student.marks;
    }

```

```
call to Student.getStudentRoll_no

CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Student.getStudentRoll_no() data: 0x9b7...ef73b
```

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

Deploy -

Before Deploy it gets compiled and then it deployed on Solidity VM Test Network.



DEPLOY & RUN TRANSACTIONS

Transactions recorded 5

Deployed Contracts

STUDENT AT 0X9EC...DDE84 (MEMORY)

Balance: 0 ETH

addStudentData

_name: Sanket Sabale

_roll_no: 1

_branch: Computer Engineering

_marks: [12,34,56]

getStudent...

0: string: Computer Engineering

getStudent...

0: uint256[]: 12,34,56

getStudent...

0: string: Sanket Sabale

getStudent...

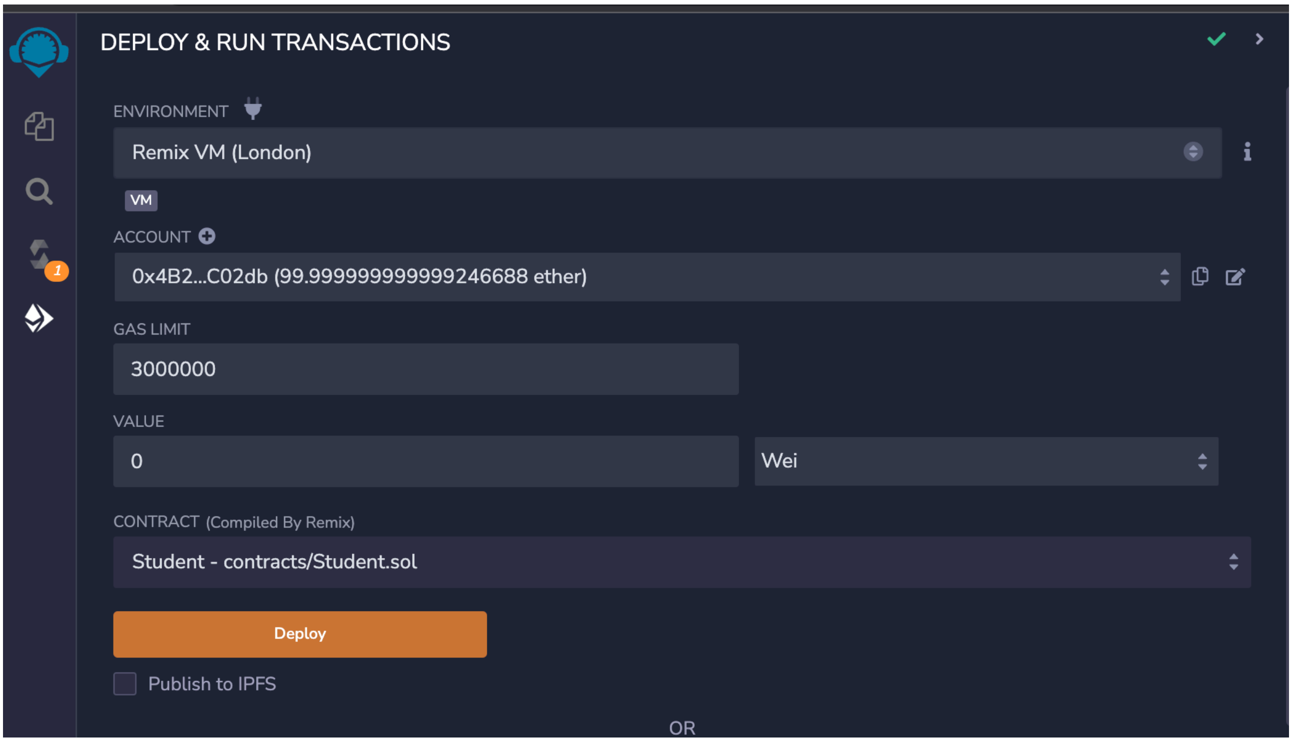
0: uint256: 1

Low level interactions

CALLDATA

Gas fee -

Every single Transaction need some gas we can set limit as well we can see by 100 ether its minus some gas fee and remaining are there.



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

Remix VM (London)

VM

ACCOUNT

0x4B2...C02db (99.999999999999246688 ether)

GAS LIMIT

3000000

VALUE

0 Wei

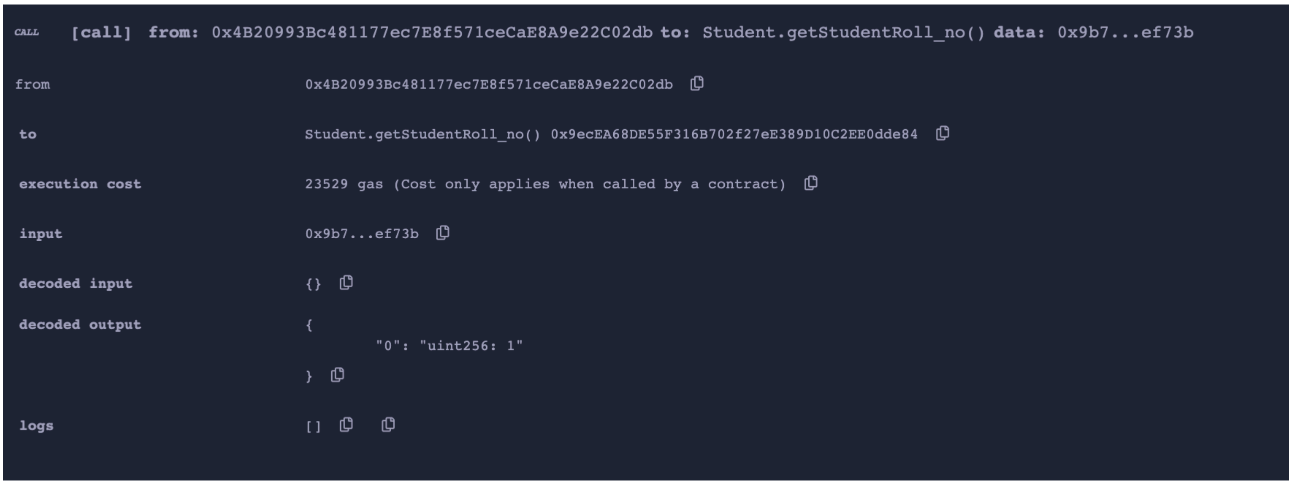
CONTRACT (Compiled By Remix)

Student - contracts/Student.sol

Deploy

□ Publish to IPFS

OR



```
CALL  [call] from: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db to: Student.getStudentRoll_no() data: 0x9b7...ef73b

from                  0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db

to                    Student.getStudentRoll_no() 0x9ecEA68DE55F316B702f27eE389D10C2EE0dde84

execution cost        23529 gas (Cost only applies when called by a contract)

input                 0x9b7...ef73b

decoded input         {}

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

logs                  []
```