BT Practical : 4

Name : Avinash Kakade
Roll No : 25

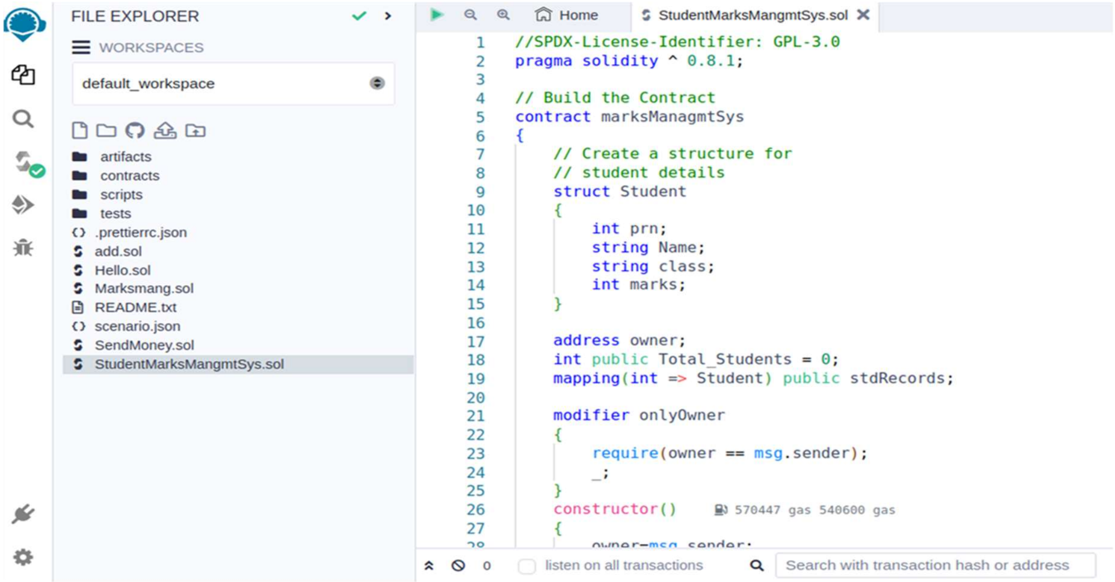
Title : Write a program in solidity to create Student data. Use the following constructs: • Structures • Arrays • Fallback Deploy this as smart contract on Ethereum and Observe the transaction fee and Gas values.

Code :

```
//SPDX-License-Identifier: GPL-3.0
pragma solidity ^ 0.8.1;
// Build the Contract
contract marksManagmtSys
{
// Create a structure for student details
struct Student
{
int prn;
string Name;
string class;
int marks;
}
address owner;
int public Total_Students = 0;
mapping(int => Student) public stdRecords;
modifier onlyOwner
{
require(owner == msg.sender);
 _;
}
constructor()
{
owner=msg.sender;
}
// Create a function to add the new records
//Array

function addNewRecords(int _prn,
string memory _Name,
string memory _class,
int _marks) public onlyOwner
{ // Increase the count by 1

Total_Students = Total_Students + 1;
// Fetch the student details with the help of Total_Students
stdRecords[Total_Students] = Student(_prn, _Name,
_class, _marks);

}

// Create a function to add bonus marks
function bonus_marks(int marks) public onlyOwner
{
stdRecords[Total_Students].marks = stdRecords[Total_Students].marks
+ marks;
} }
```

Output :

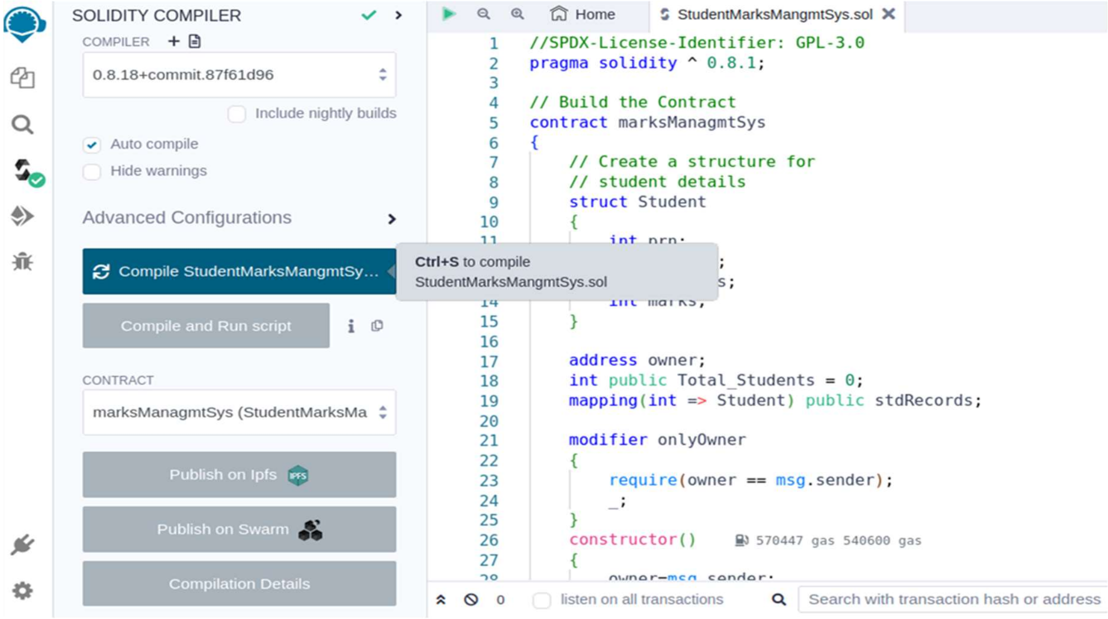
Step1: Open remix IDE and write/develop the code(logic) .



```
//SPDX-License-Identifier: GPL-3.0
pragma solidity ^ 0.8.1;

// Build the Contract
contract marksManagmtSys
{
    // Create a structure for
    // student details
    struct Student
    {
        int prn;
        string Name;
        string class;
        int marks;
    }

    address owner;
    int public Total_Students = 0;
    mapping(int => Student) public stdRecords;

    modifier onlyOwner
    {
        require(owner == msg.sender);
        _;
    }
    constructor()     570447 gas 540600 gas
    {
        owner=msg.sender;
```

Step 2: Compile the code.

Step 3 :


SOLIDITY COMPILER

COMPILER: 0.8.18+commit.87f61d96

☐ Include nightly builds

☑ Auto compile

☐ Hide warnings

Advanced Configurations

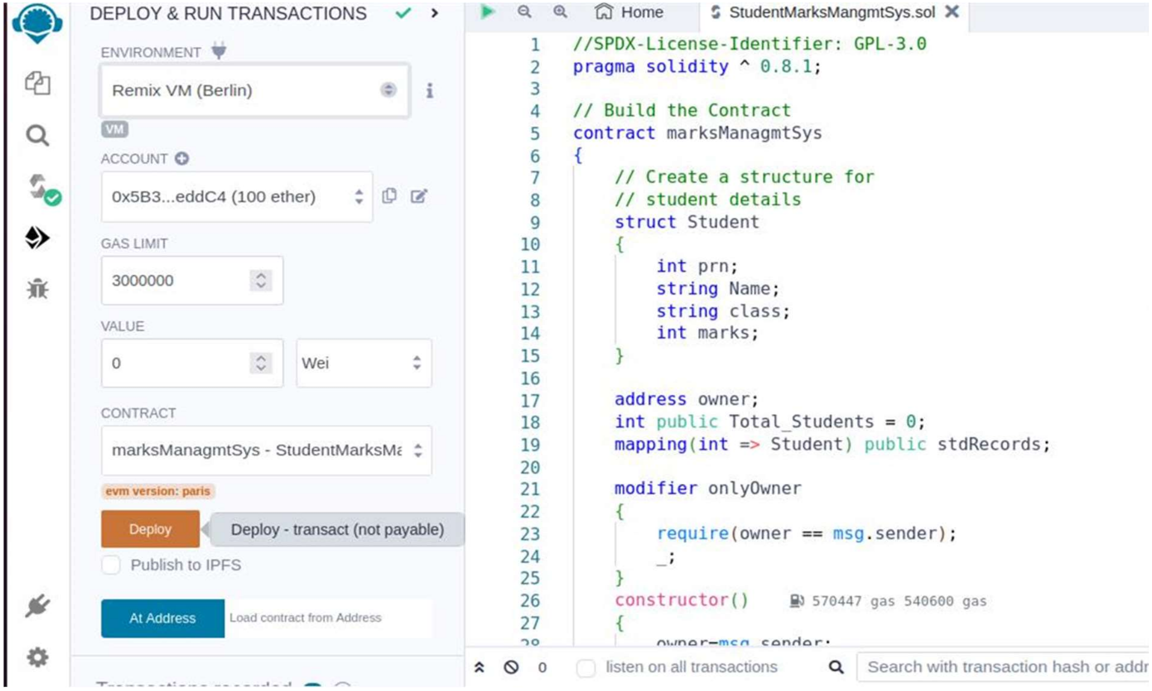
CONTRACT: marksManagmtSys (StudentMarksMa

Ctrl+S to compile StudentMarksMangmtSys.sol

```
//SPDX-License-Identifier: GPL-3.0
pragma solidity ^ 0.8.1;

// Build the Contract
contract marksManagmtSys
{
    // Create a structure for
    // student details
    struct Student
    {
        int prn;
                ;
               s;
        int marks;
    }

    address owner;
    int public Total_Students = 0;
    mapping(int => Student) public stdRecords;

    modifier onlyOwner
    {
        require(owner == msg.sender);
        _;
    }
    constructor()      570447 gas 540600 gas
    {
        owner=msg.sender;
```

Compile StudentMarksMangmtSy...
Compile and Run script
Publish on Ipfs
Publish on Swarm
Compilation Details
Home
StudentMarksMangmtSys.sol
0
☐ listen on all transactions
Search with transaction hash or address

Deploy and add entry of one student.


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Remix VM (Berlin)

ACCOUNT: 0x5B3...eddC4 (100 ether)

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: marksManagmtSys - StudentMarksMa

evm version: paris

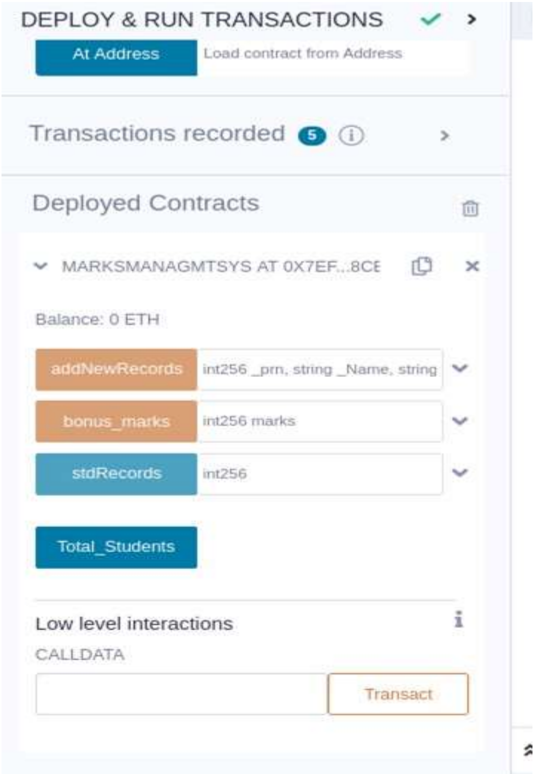
Deploy - transact (not payable)

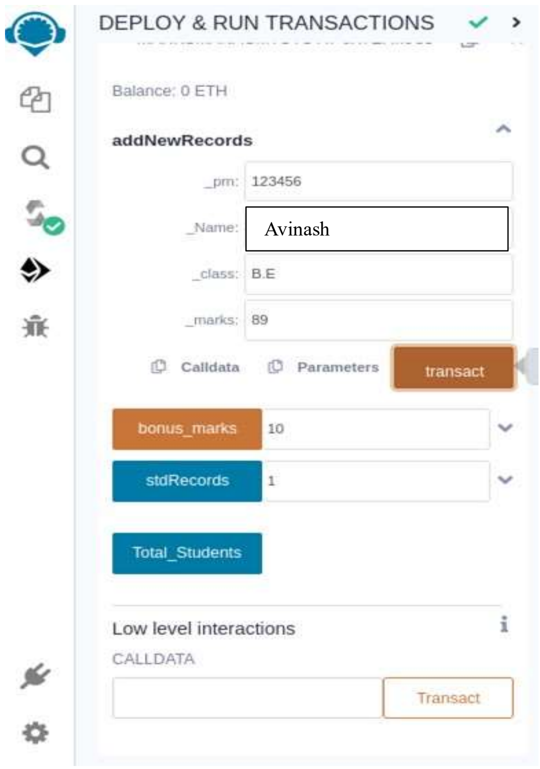
☐ Publish to IPFS

```
//SPDX-License-Identifier: GPL-3.0
pragma solidity ^ 0.8.1;

// Build the Contract
contract marksManagmtSys
{
    // Create a structure for
    // student details
    struct Student
    {
        int prn;
        string Name;
        string class;
        int marks;
    }

    address owner;
    int public Total_Students = 0;
    mapping(int => Student) public stdRecords;

    modifier onlyOwner
    {
        require(owner == msg.sender);
        _;
    }
    constructor()      570447 gas 540600 gas
    {
        owner=msg.sender;
```

VM
Deploy
At Address
Load contract from Address
Home
StudentMarksMangmtSys.sol
0
☐ listen on all transactions
Search with transaction hash or addr

You can see different transactions here .





_prn: 123456

_Name: Avinash

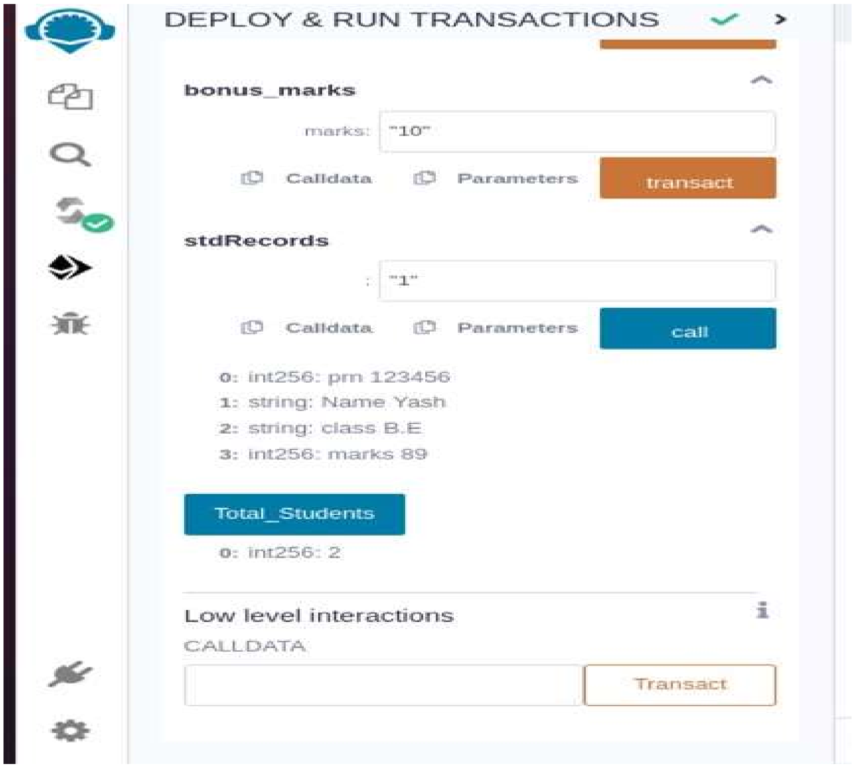
_class: B.E

_marks: 89

bonus_marks 10

stdRecords 1

We can see students entries .



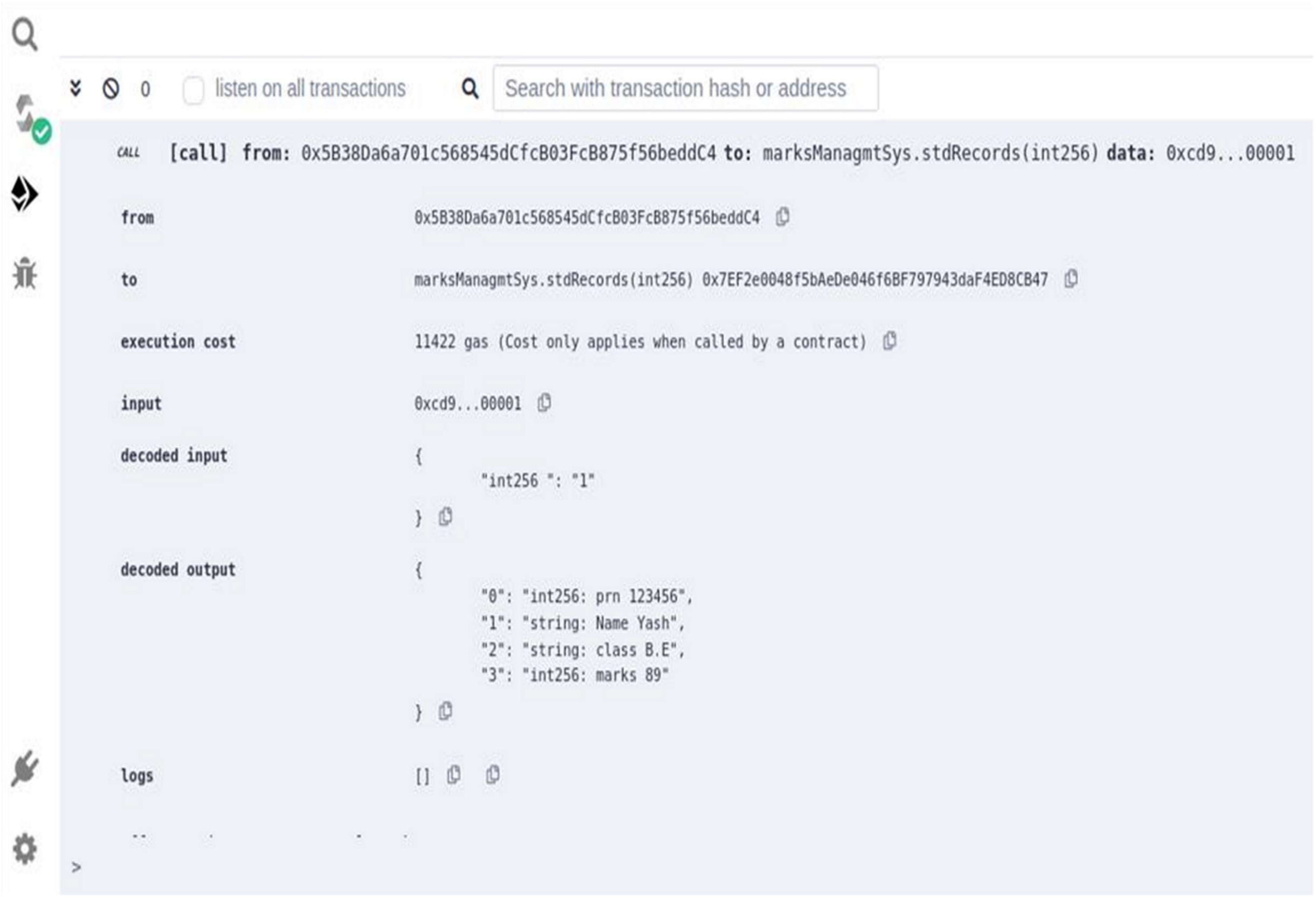
marks: "10"

: "1"

```
0: int256: prn 123456
1: string: Name Yash
2: string: class B.E
3: int256: marks 89
```

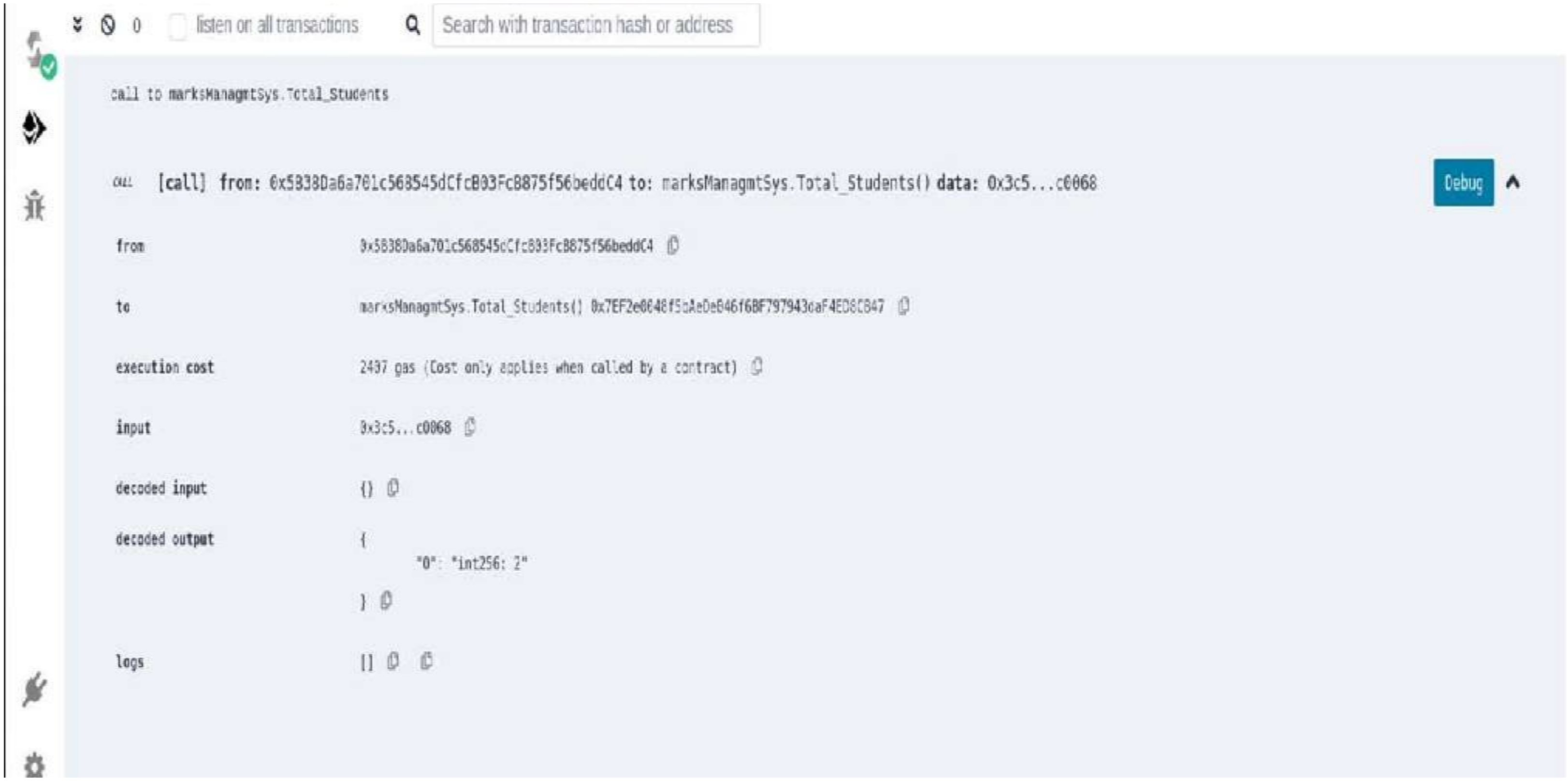
```
0: int256: 2
```

Step 4 : Check the gas limits and no of students in array .



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

```
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: marksManagmtSys.stdRecords(int256) data: 0xcd9...00001

from              0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

to                marksManagmtSys.stdRecords(int256) 0x7EF2e0048f5bAeDe046f6BF797943daF4ED8CB47

execution cost    11422 gas (Cost only applies when called by a contract)

input             0xcd9...00001

decoded input     {
                          "int256 ": "1"
                  }

decoded output    {
                          "0": "int256: prn 123456",
                          "1": "string: Name Yash",
                          "2": "string: class B.E",
                          "3": "int256: marks 89"
                  }

logs              []
```



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

```
call to marksManagmtSys.Total_Students

[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: marksManagmtSys.Total_Students() data: 0x3c5...c0068

from              0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                marksManagmtSys.Total_Students() 0x7EF2e0048f5bAeDe046f6BF797943daF4ED8CB47
execution cost    2437 gas (Cost only applies when called by a contract)
input             0x3c5...c0068
decoded input     {}
decoded output    {
                          "0": "int256: 2"
                  }
logs              []
```