

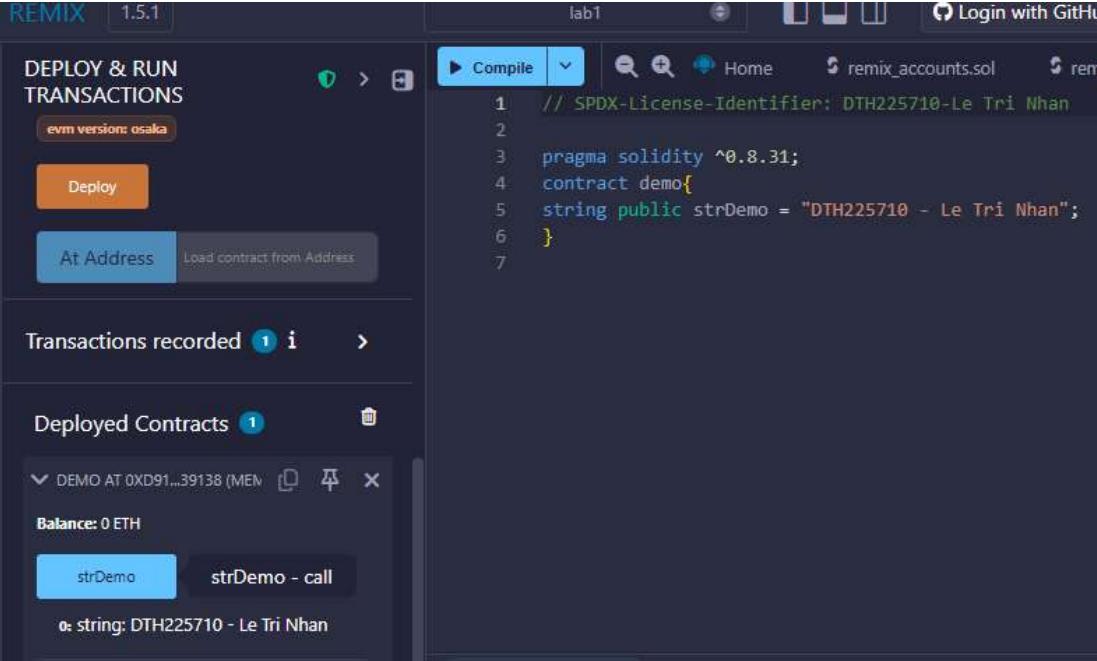
MSSV: DTH225710

Họ và tên: Lê Trí Nhàn

Lớp: DH23TH2

Ngày thực hiện: 22/01/2026

LAB 1:



The screenshot shows the Remix IDE interface. The top navigation bar includes 'REMIX 1.5.1', a profile icon, and 'lab1'. On the right side of the top bar are icons for 'Deploy', 'Compile', 'Run', 'Home', and 'GitHub'. Below the top bar, there's a search bar and a link to 'remix_accounts.sol'. The main workspace displays the following Solidity code:

```
// SPDX-License-Identifier: DTH225710-Le Tri Nhan
pragma solidity ^0.8.31;
contract demo{
    string public strDemo = "DTH225710 - Le Tri Nhan";
}
```

On the left side of the interface, there are sections for 'DEPLOY & RUN TRANSACTIONS' with a 'Deploy' button, and 'Transactions recorded' and 'Deployed Contracts' sections. The 'Deployed Contracts' section shows a deployed contract named 'DEMO AT 0xD91...39138 (MEN)' with a balance of 0 ETH. Below the contract details are buttons for 'strDemo' and 'strDemo - call', with the result '0: string: DTH225710 - Le Tri Nhan' displayed.

LAB 2:

```
// SPDX-License-Identifier: DTH225710-Le Tri Nhan
pragma solidity ^0.8.31;

contract VariableExample{
    uint public statevar = 9;
    function localVariable() public view returns(uint){    infinite gas
        uint localVal = 20;
        return statevar + localval;
    }
    function globalVariable() public view returns (address){    401 gas
    return msg.sender;
}
}
```

LAB 3:

```
// SPDX-License-Identifier: DTH225710-Le Tri Nhan
pragma solidity ^0.8.31;
contract FunctionsExample {
    uint numBor = 99;

    function setNumber(uint _number) public {    22492 gas
        numBor = _number;
    }

    function getNumber() public view returns (uint) {    2431 gas
        return numBor;
    }
}
```

LAB 4:

```
1 // SPDX-License-Identifier: DTH225710-Le Tri Nhan
2
3 pragma solidity ^0.8.31;
4 contract if_else_Example {
5     function checkNumber(uint _number) public pure returns(string memory){
6         if(_number > 10){
7             return "so > 10";
8         }else{
9             return "so <= 10";
10        }
11    }
12 }
```

The image contains two side-by-side screenshots of the Truffle UI interface. Both screenshots show a transaction history for the 'IF_ELS... EXAMPLE AT 0X358...' contract. The left screenshot shows a transaction for the 'checkNumber' function with an input of '10', and the output is 'so <= 10'. The right screenshot shows a transaction for the same function with an input of '11', and the output is 'so > 10'. Both screenshots also show a balance of '0 ETH'.

LAB 5:

The image shows the Truffle UI interface. On the left, there is a sidebar titled 'DEPLOY & RUN TRANSACTIONS' listing several deployed contracts: 'DEMO AT 0XD91...', 'VARIABLEEXAMPLE AT 0XD8B...', 'FUNCTIONSEXAMPLE AT 0XF8E...', 'IF_ELSE_EXAMPLE AT 0XA0...', 'IF_ELSE_EXAMPLE AT 0X358...', and 'FOR_LOOP_EXAMPLES AT 0X91...'. Below this, it says 'Balance: 0 ETH'. On the right, the 'Compiled' tab is selected, showing the Solidity source code for the 'FOR_LOOP_EXAMPLES' contract:

```
1 // SPDX-License-Identifier: DTH225710-Le Tri Nhan
2
3 pragma solidity ^0.8.31;
4 contract for_loop_Examples{
5     function sum(uint _num) public pure returns (uint){
6         uint total = 0;
7         for(uint i=0;i<=_num;i++){
8             total+=i;
9         }
10        return total;
11    }
12 }
```

Below the code, there is an 'Explain contract' button and a search/filter bar at the bottom.

LAB 6:

```
1 // SPDX-License-Identifier: DTH225710-Le Tri Nhan
2
3 pragma solidity ^0.8.31;
4 contract array_Example{
5     uint[] public arrayNumber;
6
7     function addNumber(uint _num) public {    46864 gas
8         arrayNumber.push(_num);
9     }
10
11    function getNumber(uint index) public view returns(uint){
12        return arrayNumber[index];
13    }
14
15    function popNumber() public returns (string memory) {    46864 gas
16        if (arrayNumber.length == 0) {
17            return "mang rong!!";
18        } else {
19            arrayNumber.pop();
20            return "da xoa phan tu cuoi";
21        }
22    }
23 }
```



Xóa mảng

```
decoded output      {
                      "0": "string: da xoa phan tu cuoi"
                    } i0
decoded input       {}
decoded output      {
                      "0": "string: mang rong!!"
                    }
```

LAB 7:

```
1 // SPDX-License-Identifier: DTH225710-Le Tri Nhan
2
3 pragma solidity ^0.8.31;
4 contract StructExample{
5     struct Student {
6         string name;
7         uint age;
8     }
9
10    Student[] public students;
11
12    function addStudent(string memory _name, uint _age) public {    ⚡ infinite gas
13        students.push(Student(_name, _age));
14    }
15
16    function getStudent(uint index) public view returns(string memory,uint){    ⚡ in
17        Student memory s = students[index];
18        return (s.name, s.age);
19    }
20 }
```

The image displays two side-by-side screenshots of a blockchain interface, likely from a Metamask extension or similar tool. Both screenshots show the same contract at address 0xBE4...0.

Left Screenshot (Contract State):

- Balance:** 0 ETH
- addStudent:** minh thu, 18
- getStudent:** 1
 - 0: string: minh thu
 - 1: uint256: 18
- students:** 1
 - 0: string: name minh thu
 - 1: uint256: age 18

Right Screenshot (Contract State after Interaction):

- Balance:** 0 ETH
- addStudent:** minh thu, 18
- getStudent:** 0
 - 0: string: tri nhan
 - 1: uint256: 22
- students:** 0
 - 0: string: name tri nhan
 - 1: uint256: age 22

LAB 8:

```
1 // SPDX-License-Identifier: DTH225710-Le Tri Nhan
2
3 pragma solidity ^0.8.31;
4 contract Mapping_Examples{
5     mapping(address=>uint) public balances;
6
7     function setBalance(uint _value) public{    22623 ↗
8         balances[msg.sender] = _value;
9     }
10
11    function getBalance() public view returns(uint){
12        return balances[msg.sender];
13    }
14 }
```

The image displays two side-by-side screenshots of a blockchain interface, likely from a browser-based wallet or developer tool. Both screens show the same contract at address 0x39.

Left Screenshot (Initial State):

- Balance:** 0 ETH
- setBalance:** 225710
- balances:** address
- getBalance:** getBalance - call
- Output:** 0: uint256: 225710

Right Screenshot (After Set Balance):

- Balance:** 0 ETH
- setBalance:** 123456
- balances:** address
- getBalance:** getBalance - call
- Output:** 0: uint256: 123456

LAB 9:

The screenshot shows a Solidity IDE interface with the following components:

- TRANSACTIONS** sidebar:
 - FUNCTIONEXAMPLE AT 0XF8F
 - IF_ELSE_EXAMPLE AT 0XDA0...4
 - IF_ELSE_EXAMPLE AT 0X358...D
 - FOR_LOOP_EXAMPLES AT 0X9C
 - ARRAY_EXAMPLE AT 0XD2A...F1
 - STRUCTEXAMPLE AT 0XB4E...0F
 - MAPPING_EXAMPLES AT 0X39F
 - EVENTEXAMPLE AT 0X9DA...D1** (selected)
- Balance: 0 ETH** status bar.
- Contract Interface**:
 - Method: **setValue** (orange button) with value **123456**.
 - Method: **value** (blue button).
 - Output: **0: uint256: 123456**
- Logs** section:
 - decoded input: {} (with "uint256 _value": "123456" highlighted).
 - decoded output: {} (empty).
 - logs: [{} (with "from": "0x9dA7c849c200e671315E77CB689811b05EDefE6", "topic": "0xc3bb07d672a05c6f9c6b4e49e138c0a87683d02b54f1536ff2c05f23acd8fd3a", "event": "EventChanged", "args": {"0": "123456"} highlighted).
- Bottom Bar**: Explain contract, Listen on all transactions, Filter with transaction.

LAB 10:

The screenshot shows the Truffle UI interface. On the left, a sidebar lists various transaction examples. The 'PAYABLEEXAMPLE AT 0X3CA...' item is expanded, revealing the contract code. On the right, the contract code for `PayableExample` is displayed:

```
// SPDX-License-Identifier: DTH225710-Le Tri Nhan
pragma solidity ^0.8.31;
contract PayableExample{
    event Receive(address sender, uint amount);
    receive() external payable {
        emit Receive(msg.sender, msg.value);
    }
    function getBalance() public view returns(uint){
        return address(this).balance;
    }
}
```

Below the code, a button labeled `getBalance` is shown, with the result `0: uint256: 0`. At the bottom, there's an 'Explain contract' section with a search bar and a filter button.

LAB 11:

The screenshot shows the Truffle UI interface. On the left, a sidebar lists deployed contracts. The 'MODIFIEREXAMPLE AT 0XD91...' item is expanded, revealing the contract code. On the right, the contract code for `ModifierExample` is displayed:

```
// SPDX-License-Identifier: DTH225710-Le Tri Nhan
pragma solidity ^0.8.31;
contract ModifierExample{
    address public owner;
    constructor(){
        owner = msg.sender;
    }
    modifier onlyOwner(){
        require(owner == msg.sender,"Khong phai chu so huu");
    }
    function changerOwner(address _newOwner) public onlyOwner{
        owner = _newOwner;
    }
}
```

Below the code, a button labeled `changerOwner` is shown, with the result `0: address: 0x5B38Da6a701c568545 dCfcB03FcB875f56beddC4`. At the bottom, there's an 'Explain contract' section with a search bar and a filter button.

- transact to ModifierExample.changerOwner errored: Error encoding arguments: TypeError: invalid address (argument="address", value="An Giang", code=INVALID_ARGUMENT, version=6.14.0) (argument="", value="An Giang", code=INVALID_ARGUMENT, version=6.14.0)
- transact to ModifierExample.changerOwner errored: Error encoding arguments: TypeError: invalid address (argument="address", value="Long Xuyêñ", code=INVALID_ARGUMENT, version=6.14.0) (argument="", value="Long Xuyêñ", code=INVALID_ARGUMENT, version=6.14.0)

LAB 12:

The screenshot shows a blockchain interface with two main sections: a transaction view on the left and a contract view on the right.

Transaction View (Left):

- To:** "0x14A919590E83B987aF5f7A3273I" (Address A)
- Amount:** "10" (highlighted with a green circle)
- Call Data:** (empty)
- Parameters:** (empty)
- Buttons:** "transact" (orange), "call" (blue)
- BALANCE OF:** "0xAb8483F64d9C6d1EcF9b849Ae6" (Address A)
- Call Data:** (empty)
- Parameters:** (empty)
- Value:** "0: uint256: 90" (highlighted with a red circle)
- Decimals:** (button)
- 0:** "uint8: 22" (button)
- Name:** (button)
- 0:** "string: DTH225710" (button)
- Symbol:** (button)
- 0:** "string: Le_Tri_Nhan" (button)
- Total Supply:** (button)
- 0:** "uint256: 100" (highlighted with a red circle)

Contract View (Right):

```

8   uint public totalSupply;
9
10  mapping (address=>uint) public balanceOf;
11
12  constructor(uint initialSupply){    infinite gas
13      totalSupply = initialSupply;
14      balanceOf[msg.sender] = initialSupply;
15  }
16
17  function transfer(address to, uint amount) public {
18      require(balanceOf[msg.sender] >= amount, "khong co tien");
19      balanceOf[msg.sender] -= amount;
20  }

```

Explain contract:

- 0: Listen on all transactions
- block hash: 0xd43dcadfc2c42f00c27a79d7d903797cc894c738acac819
- block number: 32
- from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
- to: SimpleERC20.transfer(address,uint256)
- transaction cost: 50387 gas

LAB 13:

DEPLOY & RUN TRANSACTIONS

- > SIMPLEERC20 AT 0X7EF..8CB47
- > SIMPLEERC20 AT 0X7EF..8CB47
- > SIMPLEERC20 AT 0XA13..EAD95
- > SIMPLEERC20 AT 0XA83..5B10A
- > SIMPLEERC20 AT 0XBBA..FOCE8
- > SIMPLEERC20 AT 0XB34..24E5F
- > SIMPLEERC20 AT 0X14A..94CD0

DATALOCATIONEXAMPLE AT 0)

Balance: 0 ETH

addNumber 15

getTempArray

o: uint256[]: 3,6,9,12,15

number 36

Contract Code (Compiled)

```

3 pragma solidity ^0.8.31;
4 contract DatalocationExample{
5     uint[] public number;
6
7     function addNumber(uint _number)public{ 46842 gas
8         number.push(_number);
9     }
10
11    function getTempArray()public view returns(uint[] memory){ infinite
12        uint[] memory tmp;
13        tmp = number;
14        return tmp;
15    }

```

Explain contract

0 Listen on all transactions Filter with transaction hash or address

[vm] from: 0xAB8...35cb2 to: DatalocationExample.addNumber(uint256) 0x424...15e81 value: 0 wei data: 0xfc...00009 logs: 0 hash: 0x8e6...f7499 transact to DatalocationExample.addNumber pending ...

[vm] from: 0xAB8...35cb2 to: DatalocationExample.getTempArray() 0x424...15e81 value: 0 wei data: 0xfc...0000c logs: 0 hash: 0x77f...0114a call to DatalocationExample.getTempArray

DATALOCATIONEXAMPLE AT 0)

Balance: 0 ETH

addNumber 15

getTempArray

o: uint256[]: 3,6,9,12,15

number 36

Thêm 15 vào chuỗi

LAB 14:

The screenshot shows the Truffle UI interface. On the left, a sidebar lists various transactions and deployed contracts. In the main area, the `ErrorExample` contract is selected. The balance is shown as 0 ETH. A button labeled `giaoDich` has a value of 99, with a yellow circle highlighting it. Below this, a blue button labeled `value` is present. The output section shows two transaction logs:

- [vm] from: 0xAb8...35cb2 to: ErrorExample.(constructor) value: 0 wei data: 0x608...f0033 logs: 0 hash: 0xa7d...296f1 transact to ErrorExample.giaoDich pending ...
- [vm] from: 0xAb8...35cb2 to: ErrorExample.giaoDich(uint256) 0x24f...b27f4 value: 0 wei data: 0x112...00063 logs: 0 hash: 0xba2...47091 call to ErrorExample.value

Below this, another section for `DATLOCATIONEXAMPLE` shows a balance of 0 ETH and a `giaoDich` value of 0, with a yellow arrow pointing to it.

LAB 15:

The screenshot shows the Truffle UI interface. On the left, a sidebar lists various transactions and deployed contracts. In the main area, the `Parent` and `Child` contracts are shown. The `Parent` contract has a `greetInParent()` function returning "I am Parent !!!". The `Child` contract overrides this with its own `greetChil()` function, also returning "I am Parent !!!". The balance is shown as 0 ETH. A blue button labeled `greetChil` is present. The output section shows a transaction log:

- [call] from: 0xAb8483F64d1EcF9b849Ae677dD3315835cb2 to: ErrorExample.value() data: 0x3fa...4f245 creation of Child pending...
- [vm] from: 0xAb8...35cb2 to: Child.(constructor) value: 0 wei data: 0x608...f0033 logs: 0 hash: 0x331...97212 call to Child.greetInParent

