

تمرین عملی ۳

عرفان نصرتی ۹۷۱۰۲۵۵۸

دکتر مداح علی

1. سوال یک : در سوال یک ما پس از Deploy کردن Greeter.sol ورودی خواسته شده را وارد کردیم.

The screenshot shows the Remix IDE interface. On the left, the 'VOTING AT 0XE3D...B93A6 (BLOCKCHAIN)' and 'GREETER AT 0X9E6...DFA9E (BLOCKCHAIN)' are listed. The 'Low level interactions' section shows a 'Transact' button. The main area displays the transaction details for the Greeter.greet() function call, including the transaction hash, from, to, hash, input, decoded input, decoded output, and logs. The decoded output shows the string 'Hello From 97102558'.

در بالا می بینیم تراکنش Mine شده است و Hello From 97102558 را نمایش داده است.

2. سوال دو : در ابتدا دو mapping داریم که یکی Address را به یک unit256 ؛ Map می کند که

این نشان دهنده مقدار حساب یا دارایی آن ها را نشان می دهد. و دیگری به این صورت است که نشان

می دهد آدرس اول به آدرس دوم اجازه چقدر انتقال را داده است.

حال تک تک توابع را توضیح می دهیم:

```
function balanceOf(address add) public view returns (uint256) {  
    return balances[add];  
}
```

این تابع یک تابع getter است که مقدار حساب Address add را اعلام می‌کند..

```
function transfer(address to, uint256 value) public returns (bool) {
    require(balances[msg.sender] >= value);
    require(to != address(0));
    balances[to] = SafeMath.add(balances[to] , value);
    balances[msg.sender] = SafeMath.sub(balances[msg.sender] , value);
    emit Transfer(msg.sender, to , value );
    return true;
}
```

در تابع بالا که توسط msg.sender صدا می‌شود ما می‌خواهیم پولی را از msg.sender به to بفرستیم.

برای این کار ابتدا باید چک کنیم که آیا مقدار توکنی که در حساب msg.sender بیشتر یا مساوی مقدار انتقال داده شده است یا خیر. در قسمت بعدی ما چک می‌کنیم که آدرسی که به آن پول را می‌فرستیم خالی نباشد. و سپس از حساب msg.sender مقدار Value را کم کرده و به حساب to آن مقدار را اضافه می‌کنیم.

و در آخر تراکنش را روی شبکه emit کرده تا در آنجا ثبت شود. با توجه به ویژگی تابع transfer که اگر نتواند مقدار را انتقال دهد error اگر بتواند مقدار را انتقال دهد مقدار true را بر می‌گردانیم.

```
function allowance(address owner, address spender) public view returns (uint256) {
    return allowed[owner][spender];
}
```

در این تابع مقداری که owner به spender اجازه خرج کردن داده است را برمی‌گردانیم.

```
function transferFrom(address from, address to, uint256 value) public returns (bool) {
    require( balances[from] >= value );
    require( allowance(from,msg.sender) >= value );
    balances[from] = SafeMath.sub(balances[from] , value);
    balances[to] = SafeMath.add(balances[to] , value);
    allowed[from][msg.sender] =SafeMath.sub(allowed[from][msg.sender] , value);
    emit Transfer(from , to , value);
    return true;
}
```

در این تابع transferFrom به اندازه value از آدرس from به آدرس to ارسال می‌کنیم. اول چک می‌کنیم که مقدار توکن from از value بیشتر باشد. سپس می‌بینیم آیا مقداری که msg.sender می‌خواهد به to انتقال دهد از مقداری که from به او اجازه برای انتقال داده است کمتر باشد. سپس پس از چک کردن درست بودن شرط ها مقدار value را از from کم کرده و به to اضافه می‌کنیم و سپس این مقدار را نیز از مقداری که from به msg.sender برای خرج کردن اجازه داده است کم کرده و سپس تراکنش را emit کرده و در صورت موفقیت مقدار true را بر می‌گردانیم.

```
function approve(address spender, uint256 value) public returns (bool) {
    require(balances[msg.sender] >= value);
    require( spender != address(0));
    allowed [msg.sender][spender] = value;
    emit Approval(msg.sender , spender , value);
    return true;
}
```

در این تابع ابتدا چک می‌کنیم مقدار توکن msg.sender بیشتر از value باشید و آدرس spender برابر صفر نباشد. سپس msg.sender یا صدا کننده این تابع به یک آدرس spender اجازه خرج کردن به اندازه value را می‌دهد و سپس value و آدرس spender را در بلاکچین قرار می‌دهیم .

3. سوال سه:

```
using SafeMath for uint256;
using AddressUtils for address;

bytes4 constant ERC721_RECEIVED = 0xf0b9e5ba;

mapping(uint256 => address) internal tokenOwner;
mapping (address => uint256) internal ownedTokensCount;
mapping (uint256 => address) internal tokenApprovals;

// address "A" allows address "B" to operate all A's assets
mapping (address => mapping (address => bool)) internal operatorApprovals;
```

توکن های این سوال با یکدیگر متفاوتند و هر کدام با دیگری فرق می کنند.

حال در Mapping اول یک uint256 را به یک آدرس Map می کنیم که انگار یک توکن را به یک آدرس نسبت می‌دهیم. سپس در بعدی یک آدرس را به uint256 نسبت می‌دهیم که تعداد توکن هایی است که یک نفر دارد را می‌دهد. Mapping بعدی آدرس هایی که می‌توانند یک توکن را استفاده کنند آمده است. در آخر این را مشخص می‌کند که یک آدرس آیا اجازه دارد از توکن های یک آدرس را استفاده کند یا خیر.

```

modifier canTransfer(uint256 _tokenId) {
    require(isApprovedOrOwner(msg.sender , _tokenId));
    _;
}

function balanceOf(address _owner) public view returns (uint256) {
    require(_owner != address(0));
    return ownedTokensCount[_owner];
}

```

Modifier اول چک می کند که آیا msg.sender دارنده توکن هست یا خیر.

تابع balanceOf تعداد توکن های آدرس owner که به تابع پاس داده شده است را بر می گرداند.

```

function ownerOf(uint256 _tokenId) public view returns (address) {
    address owner = tokenOwner[_tokenId];
    require (owner != address(0));
    return owner;
}

function isApprovedOrOwner(address _spender, uint256 _tokenId) internal view returns (bool) {
    address owner = tokenOwner[_tokenId];
    bool result = ( owner == _spender || getApproved(_tokenId)==_spender || isApprovedForAll(owner,_spender));
    return result;
}

```

در تابع ownerOf یک توکن به تابع داده می شود و از آنجایی که توکن ها با هم فرق دارند و می توان آدرس آن را پیدا کرد.

در تابع بعدی چک می کنیم آدرسی که به تابع پاس شده است می تواند توکن را خرج کند یا صاحب آن است یا خیر برای این چک می کنیم که آیا صاحب آن است یا می تواند از آن استفاده کند.

```

function transferFrom(address _from, address _to, uint256 _tokenId) public canTransfer(_tokenId) {
    require(isApprovedOrOwner(msg.sender , _tokenId));
    clearApproval(_from , _tokenId);
    removeTokenFrom(_from , _tokenId);
    addTokenTo(_to , _tokenId);
}

```

در تابع transferftom توکن پاس شده به تابع را از فرد from به فرد to می فرستد.

```
function safeTransferFrom(address _from, address _to, uint256 _tokenId, bytes memory data) public {
    // if target address is a contract, make sure it supports ERC721 interface
    if(! AddressUtils.isContract(_to)){
        transferFrom( _from, _to , _tokenId);
    }
    else{
        bytes4 result = onERC721Received ( _from , _tokenId , data);
        require( result == ERC721_RECEIVED );
        transferFrom(_from , _to , _tokenId);
    }
}

function safeTransferFrom(address _from, address _to, uint256 _tokenId) public {
    safeTransferFrom(_from , _to , _tokenId , "");
}
```

در تابع safetransform ابتدا شرطی چک می‌شود تا در تراکنش اشتباهی رخ ندهد. سپس تابع transformfrom صدا می‌شود. این شرایط عبارت است از اینکه تراکنش اگر contract است حتما ERC721 باشد. و اگر نیست شرط مساوی بودن ERC721Recieved با مقدار default آن که در بالا آمده است چک شود.

```
function clearApproval(address _owner, uint256 _tokenId) internal {
    require(ownerOf(_tokenId) == _owner);
    tokenApprovals[_tokenId] == address(0);
}

function removeTokenFrom(address _from, uint256 _tokenId) internal {
    require (ownerOf(_tokenId) == _from);
    ownedTokensCount[_from] = SafeMath.sub(ownedTokensCount[_from],1);
    tokenOwner[_tokenId] = address(0);
}
```

در تابع ClearApproval صاحب توکن افرادی را که برای این توکن تایید کرده بود را پاک می‌کند. در تابع removeTokenFrom صاحب توکن پاس داده شده به تابع پاک می‌شود.

```
function addTokenTo(address _to, uint256 _tokenId) internal {
    require(ownerOf(_tokenId) == address(0));
    tokenOwner[_tokenId] = _to;
    ownedTokensCount[_to] = SafeMath.add(ownedTokensCount[_to] , 1);
}

function isApprovedForAll(address _owner, address _operator) public view returns (bool) {
    return operatorApprovals[_owner][_operator];
}
```

در تابع addTokento یک توکن به یک فرد جدید نسبت داده می‌شود.(صاحب جدید)

در تابع `isApprovedForAll` با توجه به mapping ای که در بالا تعریف شد مشخص می‌شود آدرس دوم جز افراد مورد تایید نفر اول است یا خیر و تابع درست یا غلط را باز می‌گرداند.

```
function getApproved(uint256 _tokenId) public view returns (address) {
    return tokenApprovals[_tokenId];
}

function approve(address _approved, uint256 _tokenId) public {
    address owner = ownerOf(_tokenId);
    require(_approved != owner && ( msg.sender == owner || isApprovedForAll(owner,msg.sender)));
    tokenApprovals[_tokenId] = _approved;
    emit Approval( owner , _approved , _tokenId);
}
```

در تابع `getApproved` آدرس هایی که برای یک توکن تایید شده هستند را بر می‌گرداند.

`Approve` در این تابع اگر صدا کننده ی تابع صاحب توکن باشد یا اجازه خرج کردن آن را داشته باشید و خود صاحب توکن آدرس پاس داده شده به توکن نباشد آدرس پاس داده شده به آدرس های اجازه دار آن توکن اضافه می‌شود.

```
function setApprovalForAll(address _operator, bool _approved) public {
    require( _operator != msg.sender);
    operatorApprovals [msg.sender][_operator] = _approved;
    emit ApprovalForAll(msg.sender , _operator , _approved);
}

function onERC721Received(address, uint256, bytes memory) public returns (bytes4) {
    return ERC721_RECEIVED;
}
```

در تابع اول که یک تابع `setter` است یک آدرس و یک مقدار `Boolean` می‌گیرید و برحسب `true` یا `false` بودن مقدار آن آدرس را به آدرس های مورد تایید اضافه می‌کند یا خیر.

4. سوال ۴ :

در این سوال ابتدا یک شخص contract را صدا می‌زند و مقدار value reserved و مقدار بلاک هایی را که می‌خواهد در هر فاز صبر کند را می‌فرستد. سپس هر موقع خواست activateAuction() را می‌زند و به با صدا کردن و با صدا کردن آن contract از فاز pending خارج می‌شود و وارد فاز commitment می‌شود و در این مرحله متقاضیان باید یک فایل proposal را به همراه مقدار مورد نظر خود که همراه با یک nonce هش گرفته اند تا مقدار آن لو نرود را به تابع bid می‌فرستد. سپس admin ۳ نفر از بهترین proposal ها را انتخاب می‌کند و برای مرحله Opening آنها را می‌فرستد بعد از آن آن سه نفر به اندازه تعداد بلاک هایی که در ابتدا مشخص شد فرصت دارند مقدار value و nonce و را بفرستند تا ثابت کنند این مقدار را در مرحله commitment ارسال کرده اند. سپس پس از طی شدن زمان تعیین شده admin تابع finalize را صدا می‌زند و مقدار کمترین value انتخاب و پول به حساب او واریز می‌شود و باقی مانده آن به حساب خودش برمی‌گردد. توابع را در متن بالا در هنگام شرح سناریو مزاییده توضیح داده ام.

```
modifier duringCommitment {
    require(phase == Phase.Commitment);
    require( block.number <= startPhaseBlock + commitment_len);
    _;
}

/// @dev This modifier allow to invoke the function only during the Opening phase.
modifier duringOpening {
    require(phase == Phase.Opening);
    require(block.number <= startPhaseBlock + opening_len );
    _;
}
```

در modifier های بالا چک می‌کنیم آیا در فاز Commitment و Opening است.

```
function getReserveFund() public view returns (uint256) {
    return ReserveFund;
}

function getFile( address add ) public view returns ( bytes32 fileAdd) {
    require(add != address(0));
    return bids[add].FileAddress;
}
```

دو تابع بالا دو تابع getter است و مقدار ReserveFund و fileAdd را بر می‌گرداند تا در تابع اول کسی که

در مزایده شرکت می‌کند بداند حداکثر پول درخواستی چقدر است و در تابع آخر باید admin ؛ Proposal شرکت کننده ها را ببیند تا بتواند ۳ تا از بهترین ها را انتخاب کند.

```
/// @notice This function will activate the auction.
function activateAuction() public onlyAdmin {
    require(phase == Phase.Pending);
    phase = Phase.Commitment;
    startPhaseBlock = block.number;
    description.startBlock = startPhaseBlock;
    emit auctionStarted();
}

///@notice This function allow people to make bid.
///@notice Note that a bid will be taken into account if the value sent is >= the minimum deposit.
///@dev This function can be invoked only during the commitment phase.
///@param _bidHash this is the hash of the tuple <value,nonce>. See `GenerateBid` contract for more info.
function bid(bytes32 _bidHash, bytes32 _FileAddress) public duringCommitment payable {
    require(_bidHash != 0 && _FileAddress != 0);
    require(bids[msg.sender].hash == 0 && bids[msg.sender].FileAddress == 0);
    bids[msg.sender].hash = _bidHash;
    bids[msg.sender].FileAddress = _FileAddress;
}
```

در تابع activateAuction ؛ admin می‌تواند فاز را از pending به Commitment تغییر دهد. و باید برای شمردن تعداد بلاک ها بلاک start را ست کنیم. و بلاک استارت در Auction را هم ست می‌کنیم. در تابع bid همانطور که در بالا گفته شده hash و proposal را شرکت کننده می‌فرستد تا بعدا بتواند با ارائه مقدار value و nonce مقداری که در این مرحله Commit کرده است را اثبات کند.

```
///@notice This function allow people to open their bid.
function open( uint value, bytes32 _nonce) public duringOpening {

    // Control the correctness of the bid
    require(chooses[msg.sender]);
    require(bids[msg.sender].value <= ReserveFund);
    require(bids[msg.sender].hash == sha256(abi.encodePacked(value, _nonce)));
    // Update the bid status
    bids[msg.sender].value = value;
    bids[msg.sender].nonce = _nonce;
    if (firstOpen)
    {
        lowestBid = bids[msg.sender].value;
        lowestBidder = msg.sender;
    }
    else
    {
        if (bids[msg.sender].value < lowestBid)
        {
            lowestBid = bids[msg.sender].value;
            lowestBidder = msg.sender;
        }
    }
}
```


در تابع Open مقدار و nonce را به تابع پاس می‌کنیم تا بتوانیم راستی آزمایی کنیم که مقدار درستی فرستاده است یا خیر. سپس اگر اولین نفری بود که داشت این تابع را صدا می‌زد مقدار lowestbid و lowestbidder رو ست می‌کنیم زیرا قبل آن مقداری ندارد سپس از آن به بعد چک می‌شود آیا این مقدار کمتر از مقدار تابع قبلی است یا خیر.

```
///@notice This function finalize and close the contract.
function finalize() public onlyAdmin {
    require(phase == Phase.Opening);
    require(block.number >= startPhaseBlock + opening_len);
    if (firstOpen)
    {
        description.admin.transfer(ReserveFund);
        description.winnerAddress = address(0);
        description.winnerBid = 0 ;
    }
    else
    {
        lowestBidder.transfer(lowestBid);
        description.admin.transfer(ReserveFund - lowestBid);
        description.winnerAddress = lowestBidder;
        description.winnerBid = lowestBid ;
    }

    emit auctionFinished(lowestBidder , lowestBid);
}
```

تابع finalize پس آخرین تابع است که صدا می‌شود اگر firstOpen true باشد می‌دانیم هیچ کسی توابع را یا به درستی پر نکرده یا proposal داده نشده است. اما اگر نبود پس حداقل یکی پر شده است. و با توجه به تابع های قبل کمترین مقداری است که پیشنهاد شده. سپس مقدار درخواستی را به او و باقی مانده را به خودمان می‌فرستیم. سپس event تمام شدن مزایده را broadcast می‌کنیم.

5. سوال ۵: در زیر سوال را حل کرده و مرحله به مرحله اسکرین شات می‌گذاریم و توضیح می‌دهیم. در

اول همه تراکنش‌ها را می‌گذاریم.

x79848d69b541D265DC6441A05d4C85bF2a0293660

بالا آدرس تراکنش contract است.

Transactions

Internal Txns

Contract

Events

Latest 13 from a total of 13 transactions

	Txn Hash	Block	Age	From	To	Value	Txn Fee
	0x9904db10041087483...	9286976	1 min ago	0x7c34ef063ac5b8974...	0x79848d69b541d265d...	0 Ether	0.000183571824
	0xbb016bfd0b6ac360e...	9286955	3 mins ago	0x68f73b71f53926ab01...	0x79848d69b541d265d...	0 Ether	0.000451768763
	0x367ca85a798b6b3a1...	9286950	4 mins ago	0x6b17f737d25289e9e...	0x79848d69b541d265d...	0 Ether	0.000451768763
	0x5e7e543918c503f5d...	9286948	4 mins ago	0xe35fbc0f113b614a84...	0x79848d69b541d265d...	0 Ether	0.000616195799
	0x8a9ac6631b9fd0bcf1...	9286942	5 mins ago	0x7c34ef063ac5b8974...	0x79848d69b541d265d...	0 Ether	0.000133388692
	0xdcef8de9e3ed05363...	9286942	5 mins ago	0x7c34ef063ac5b8974...	0x79848d69b541d265d...	0 Ether	0.000550973075
	0x90020f90288677d3b...	9286910	8 mins ago	0xcd77e432104f1c486...	0x79848d69b541d265d...	0 Ether	0.000367516349
	0x707169fe1126fb1a58...	9286908	9 mins ago	0xee80756dc3cd33febf...	0x79848d69b541d265d...	0 Ether	0.000367516349
	0x9d1e84e97a8e5410e...	9286901	10 mins ago	0x68f73b71f53926ab01...	0x79848d69b541d265d...	0 Ether	0.000367450578
	0x5a9c90744a18ae8ce...	9286896	10 mins ago	0x6b17f737d25289e9e...	0x79848d69b541d265d...	0 Ether	0.000367450578
	0x2ef22b0eeacd292e4...	9286891	11 mins ago	0xe35fbc0f113b614a84...	0x79848d69b541d265d...	0 Ether	0.000367516349
	0x13a7c6e7b60fd7d58...	9286887	11 mins ago	0x7c34ef063ac5b8974...	0x79848d69b541d265d...	0 Ether	0.000468573206
	0x33736a04f5bae4257...	9286875	13 mins ago	0x7c34ef063ac5b8974...	Contract Creation	0 Ether	0.005851399171

Download CSV Export

همه تراکنش‌ها از contract تا finalize در بالا آمده است. ادمین contract را شروع می‌کند و

سپس ۵ نفر تابع bid را صدا می‌کنند. سپس پس از گذشتن زمان ست شده ادمین تابع

startOpening را صدا می‌کند تا وارد مرحله Opening شویم سپس پس از آن ۳ نفری که توسط

ادمین انتخاب شده اند با توجه به زمان ست شده وقت دارند مقدار nonce و value خود را بفرستند

✓

[block:9286887 txIndex:7] from: 0x7C3...fe7f4 to: CustomAuction.activateAuction() 0x798...29366 value: 0 wei data: 0x605...fed87 logs: 1 hash: 0x13a...36c8d

status

true Transaction mined and execution succeed

transaction hash

0x13a7c6e7b60fd7d58d941a0f404bf93aadb4ad8c6a4bee4792bb71f071b36c8d

from

0x7C34Ef063ac58897429297555141CDd5d95fe7f4

to

CustomAuction.activateAuction() 0x79848d69b541D265DC6441A05d4C85bF2a029366

gas

85492 gas

transaction cost

85492 gas

hash

0x13a7c6e7b60fd7d58d941a0f404bf93aadb4ad8c6a4bee4792bb71f071b36c8d

input

0x605...fed87

decoded input

{}

decoded output

-

logs

[{ "from": "0x79848d69b541D265DC6441A05d4C85bF2a029366", "topic": "0xee2679bc2382e067cc3dfbda872852b5b1702b359c1fd6ce272c32dae5bacabf", "event": "auctionStarted", "args": {} }]

OverviewLogs (1)State

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0x13a7c6e7b60fd7d58d941a0f404bf93aadb4ad8c6a4bee4792bb71f071b36c8d

Status:

Success

Block:

9286887200 Block Confirmations

Timestamp:

25 mins ago (Dec-18-2020 05:15:14 PM +UTC)

From:

0x7c34ef063ac5b097429297555141cdd5d95fe7f4

To:

Contract 0x79848d69b541d265dc6441a05d4c85bf2a029366

Value:

0 Ether (\$0.00)

Transaction Fee:

0.00046857320633 Ether (\$0.000000)

Gas Price:

0.000000005480901211 Ether (5.480901211 Gwei)

Gas Limit:

85,492

Gas Used by Transaction:

85,492 (100%)

Nonce

157

Input Data:

Function: activateAuction() ***

MethodID: 0x605fed87

View Input As

Click to see Less

activateAuction توسط ادمین.(اکانت اول)

<div> ✓ [block:9286891 txIndex:1] from: 0xe35...3B29a to: CustomAuction.bid(bytes32,bytes32) 0x798...29366 value: 0 wei data: 0x434...0594a logs: 0 hash: 0x2ef...47104 </div>	
status	true Transaction mined and execution succeed
transaction hash	0x2ef22b0eeacd292e45b7cb002dbe586ab2bb3457ed92c0d808e406950c447104 🔗
from	0xe35fbc0f113b614a844506bbd0bb5ED87793B29a 🔗
to	CustomAuction.bid(bytes32,bytes32) 0x79848d69b541D265DC6441A05d4C85bF2a029366 🔗
gas	67054 gas 🔗
transaction cost	67054 gas 🔗
hash	0x2ef22b0eeacd292e45b7cb002dbe586ab2bb3457ed92c0d808e406950c447104 🔗
input	0x434...0594a 🔗
decoded input	{ "bytes32 _bidHash": "0xc19d0da3e95f91536a8c927277f2ba72b8ee8f720660ec6f563506762b916fb0", "bytes32 _fileAddress": "0xcba7e01117af3a75f8644f02c0d1a9895d434ea2a80743798f4bb4287b40594a" } 🔗
decoded output	- 🔗
logs	🔗 🔗 🔗
value	0 wei 🔗

Overview State ⓘ	
[This is a Ropsten Testnet transaction only]	
① Transaction Hash:	0x2ef22b0eeacd292e45b7cb002dbe586ab2bb3457ed92c0d808e406950c447104 🔗
② Status:	<div> ✓ Success </div>
③ Block:	<div> 9286891 222 Block Confirmations </div>
④ Timestamp:	⌚ 27 mins ago (Dec-18-2020 05:15:25 PM +UTC)
⑤ From:	0xe35fbc0f113b614a844506bbd0bb5ed87793b29a 🔗
⑥ To:	Contract 0x79848d69b541d265dc6441a05d4c85bf2a029366 ✓ 🔗
⑦ Value:	<div> 0 Ether (\$0.00) </div>
⑧ Transaction Fee:	0.0003675163498 Ether (\$0.000000)
⑨ Gas Price:	0.000000005480901211 Ether (5.480901211 Gwei)
⑩ Gas Limit:	67,054
⑪ Gas Used by Transaction:	67,054 (100%)
⑫ Nonce Position	<div> 2 1 </div>
⑬ Input Data:	<div> Function: bid(bytes32 _bidHash, bytes32 _fileAddress) *** MethodID: 0x434f967c [0]: c19d0da3e95f91536a8c927277f2ba72b8ee8f720660ec6f563506762b916fb0 [1]: cba7e01117af3a75f8644f02c0d1a9895d434ea2a80743798f4bb4287b40594a </div> <div> View Input As ↕ Decode Input Data </div>
Click to see Less ↑	

درخواست توسط اکانت دوم.

[block:9286896 txIndex:5] from: 0x6B1...f7dFD to: CustomAuction.bid(bytes32,bytes32) 0x798...29366 value: 0 wei data: 0x434...89405 logs: 0 hash: 0x5a9...31040

status	true Transaction mined and execution succeed
transaction hash	0x5a9c90744a18ae8ce2276fb87e77c9ed820874957ad4966288f8643e50231040
from	0x6B17F737d25289E9ea66db6e993413e16A7f7dFD
to	CustomAuction.bid(bytes32,bytes32) 0x79848d69b541D265DC6441A05d4C85bf2a029366
gas	67042 gas
transaction cost	67042 gas
hash	0x5a9c90744a18ae8ce2276fb87e77c9ed820874957ad4966288f8643e50231040
input	0x434...89405
decoded input	{ "bytes32 _bidHash": "0xb71e58973c8aa85293adc4a2eacc32139eede7b3a6bdd65f0da0b2579c4d049", "bytes32 _FileAddress": "0xe3402d53f0c4d738a62fff28dbe058181d30001c4296bfb6d449a4345a889405" }
decoded output	-
logs	
value	0 wei

OverviewState

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0x5a9c90744a18ae8ce2276fb87e77c9ed820874957ad4966288f8643e50231040

Status:

Success

Block:

9286896231 Block Confirmations

Timestamp:

28 mins ago (Dec-18-2020 05:16:13 PM +UTC)

From:

0x6b17f737d25289e9ea66db6e993413e16a7f7dfd

To:

Contract 0x79848d69b541d265dc6441a05d4c85bf2a029366

Value:

0 Ether (\$0.00)

Transaction Fee:

0.00036745057898 Ether (\$0.000000)

Gas Price:

0.000000005480901211 Ether (5.480901211 Gwei)

Gas Limit:

67,042

Gas Used by Transaction:

67,042 (100%)

Nonce

Position

45

Input Data:

Function: bid(bytes32 _bidHash, bytes32 _FileAddress) ***

MethodID: 0x434f967c

[0]: b71e58973c8aa85293adc4a2eacc32139eede7b3a6bdd65f0da0b2579c4d049


[1]: e3402d53f0c4d738a62fff28dbe058181d30001c4296bfb6d449a4345a889405

View Input As

Decode Input Data














Click to see Less

درخواست توسط اکانت سوم.

<div>  [block:9286901 txIndex:3] from: 0x68F...8249c to: CustomAuction.bid(bytes32,bytes32) 0x798...29366 value: 0 wei data: 0x434...afa88 logs: 0 hash: 0x9d1...d19ca </div>	
status	true Transaction mined and execution succeed
transaction hash	0x9d1e84e97a8e5410e1d40711131cb10886740b3493a570261eb60dca1bdd19ca 🔗
from	0x68F73b71F53926ab01aebE46Cf30A5283D58249c 🔗
to	CustomAuction.bid(bytes32,bytes32) 0x79848d69b541D265DC6441A05d4C85bF2a029366 🔗
gas	67042 gas 🔗
transaction cost	67042 gas 🔗
hash	0x9d1e84e97a8e5410e1d40711131cb10886740b3493a570261eb60dca1bdd19ca 🔗
input	0x434...afa88 🔗
decoded input	{ "bytes32 _bidHash": "0xa55800a33ed062a072e2e0839b9f538e8dd21df64445daace4c6636d2c50ff22", "bytes32 _FileAddress": "0xcc9e7c9ce89b92d13be809d00908679c35e0c13209706917a5a4c7bdb7cfa88" } 🔗
decoded output	- 🔗
logs	[] 🔗 🔗
value	0 wei 🔗

Overview State ⓘ	
[This is a Ropsten Testnet transaction only]	
⑦ Transaction Hash:	0x9d1e84e97a8e5410e1d40711131cb10886740b3493a570261eb60dca1bdd19ca 🔗
⑦ Status:	Success
⑦ Block:	9286901 238 Block Confirmations
⑦ Timestamp:	⌚ 28 mins ago (Dec-18-2020 05:16:56 PM +UTC)
⑦ From:	0x68f73b71f53926ab01aebE46cf30a5283d58249c 🔗
⑦ To:	Contract 0x79848d69b541d265dc6441a05d4c85bf2a029366 🟢 🔗
⑦ Value:	0 Ether (\$0.00)
⑦ Transaction Fee:	0.00036745057898 Ether (\$0.000000)
⑦ Gas Price:	0.000000005480901211 Ether (5.480901211 Gwei)
⑦ Gas Limit:	67,042
⑦ Gas Used by Transaction:	67,042 (100%)
⑦ Nonce Position	2 3
⑦ Input Data:	<div>Function: bid(bytes32 _bidHash, bytes32 _FileAddress) ***</div> <div>MethodID: 0x434f967c</div> <div>[0]: a55800a33ed062a072e2e0839b9f538e8dd21df64445daace4c6636d2c50ff22</div> <div>[1]: cc9e7c9ce89b92d13be809d00908679c35e0c13209706917a5a4c7bdb7cfa88</div>

درخواست توسط اکانت چهارم.

<div>  [block:9286908 txIndex:3] from: 0xee8...Bd820 to: CustomAuction.bid(bytes32,bytes32) 0x798...29366 value: 0 wei data: 0x434...72004 logs: 0 hash: 0x707...bef35 </div>	
status	true Transaction mined and execution succeed
transaction hash	0x707169fe1126fb1a58d7ab088ac33d4b9557bf86f199f1796053b7d000cbef35 
from	0xee80756dc3cd33FEF85e61FE22c487f2461bd820 
to	CustomAuction.bid(bytes32,bytes32) 0x79848d69b541d265dc6441A05d4C85bf2a029366 
gas	67054 gas 
transaction cost	67054 gas 
hash	0x707169fe1126fb1a58d7ab088ac33d4b9557bf86f199f1796053b7d000cbef35 
input	0x434...72004 
decoded input	{ "bytes32 _bidHash": "0x40657b1c4892c856561471098e7daeb436404655cc351749a0eab67fa36a0cbc", "bytes32 _fileAddress": "0xebac41e3154043de2412f26686fbb404c8d7b8136c9365843f1aeb31af172004" } 
decoded output	- 
logs	[]  
value	0 wei 

Overview

State

[This is a Ropsten Testnet transaction only]

⑦ Transaction Hash:

0x707169fe1126fb1a58d7ab088ac33d4b9557bf86f199f1796053b7d000cbef35

⑦ Status:

Success

⑦ Block:

9286908

248 Block Confirmations

⑦ Timestamp:

29 mins ago (Dec-18-2020 05:17:30 PM +UTC)

⑦ From:

0xee80756dc3cd33fe5e61fe22c4b7f2461bd820

⑦ To:

Contract 0x79848d69b541d265dc6441a05d4c85bf2a029366

⑦ Value:

0 Ether (\$0.00)

⑦ Transaction Fee:

0.0003675163498 Ether (\$0.000000)

⑦ Gas Price:

0.000000005480901211 Ether (5.480901211 Gwei)

⑦ Gas Limit:

67,054

⑦ Gas Used by Transaction:

67,054 (100%)

⑦ Nonce

Position

3

⑦ Input Data:

Function: bid(bytes32 _bidHash, bytes32 _fileAddress) ***

MethodID: 0x434f967c

[0]: 40657b1c4892c856561471098e7daeb436404655cc351749a0eab67fa36a0cbc

[1]: ebac41e3154043de2412f26686fbb404c8d7b8136c9365843f1aeb31af172004

View Input As

Decode Input Data

Click to see Less

درخواست توسط اکانت پنجم.

[block:9286910 txIndex:6] from: 0xCd7...35C9E to: CustomAuction.bid(bytes32,bytes32) 0x798...29366 value: 0 wei data: 0x434...afa8 logs: 0 hash: 0x900...0651d

status

true Transaction mined and execution succeed

transaction hash

0x90020f90288677d3b457f36cb0e20d63b8549b5ed26ff95ff24c2c6dca30651d

from

0xCd77e432104f1c486ac5536db09873e6fc135c9e

to

CustomAuction.bid(bytes32,bytes32) 0x79848d69b541d265dc6441a05d4c85bf2a029366

gas

67054 gas

transaction cost

67054 gas

hash

0x90020f90288677d3b457f36cb0e20d63b8549b5ed26ff95ff24c2c6dca30651d

input

0x434...afa8

decoded input

{ "bytes32 _bidHash": "0x5433420c0380fbdecef38847bfef075e4e5e2db21b21df97cda8fb7596b7b189", "bytes32 _FileAddress": "0xcc9e7c9ce89b92d13be809d00908679c35e0c13209706917a5a4c7bdb7cfa88" }

decoded output

-

logs

value

0 wei

OverviewState

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0x90020f90288677d3b457f36cb0e20d63b8549b5ed26ff95ff24c2c6dca30651d

Status:

Success

Block:

9286910253 Block Confirmations

Timestamp:

29 mins ago (Dec-18-2020 05:18:14 PM +UTC)

From:

0xcd77e432104f1c486ac5536db09873e6fc135c9e

To:

Contract 0x79848d69b541d265dc6441a05d4c85bf2a029366

Value:

0 Ether (\$0.00)

Transaction Fee:

0.0003675163498 Ether (\$0.000000)

Gas Price:

0.000000005480901211 Ether (5.480901211 Gwei)

Gas Limit:

67,054

Gas Used by Transaction:

67,054 (100%)

Nonce

Position06

Input Data:

Function: bid(bytes32 _bidHash, bytes32 _FileAddress) ***

MethodID: 0x434f967c

[0]: 5433420c0380fbdecef38847bfef075e4e5e2db21b21df97cda8fb7596b7b189

[1]: cc9e7c9ce89b92d13be809d00908679c35e0c13209706917a5a4c7bdb7cfa88

View Input As

Decode Input Data

Click to see Less

درخواست توسط اکانت پنجم.

```

data: 0xeb5...8249c logs: 1 hash: 0xdce...29748

status      true Transaction mined and execution succeed

transaction hash  0xdcef6de9e3ed0536327d9030549dff544acbd3482e445cdc953a8f3764229748

from      0x7C34Ef063ac58897429297555141CDd5d95fe7f4

to      CustomAuction.startOpening(address,address,address) 0x79848d69b541d265DC6441A05d4C85bF2a029366

gas      100526 gas

transaction cost  100526 gas

hash      0xdcef6de9e3ed0536327d9030549dff544acbd3482e445cdc953a8f3764229748

input      0xeb5...8249c

decoded input  { "address add1": "0xe35Fbc0f113b614A844506bbd0b5ED07793b29a", "address add2":
"0x6817f737d25289E9ea66db6e993413e16A7f7dFD", "address add3": "0x68F73b71F53926ab01aebE46cF30A5283D58249c" }

decoded output  -

logs      [ { "from": "0x79848d69b541d265DC6441A05d4C85bF2a029366", "topic":
"0xe884e87cca623860f1ac029c5df020e7b7ebad72eab657701673da4cf3e90461", "event": "openingStarted", "args": { } } ]

value      0 wei

```

Overview
Logs (1)
State

[This is a Ropsten Testnet transaction only]

Transaction Hash:
0xdcef6de9e3ed0536327d9030549dff544acbd3482e445cdc953a8f3764229748

Status:
Success

Block:
9286942
232 Block Confirmations

Timestamp:
27 mins ago (Dec-18-2020 05:21:46 PM +UTC)

From:
0x7c34ef063ac5b897429297555141cdd5d95fe7f4

To:
Contract 0x79848d69b541d265dc6441a05d4c85bf2a029366

Value:
0 Ether (\$0.00)

Transaction Fee:
0.00055097307513 Ether (\$0.000000)

Gas Price:
0.000000005480901211 Ether (5.480901211 Gwei)

Gas Limit:
100,526

Gas Used by Transaction:
100,526 (100%)

Nonce
Position
16
10

Input Data:

Function: startOpening(address add1, address add2, address add3) ***

MethodID: 0xeb579e12
[0]: 000000000000000000000000e35fbc0f113b614a844506bbd0bb5edb7793b29a
[1]: 0000000000000000000000006b17f737d25289e9ea66db6e993413e16a7f7dfd
[2]: 00000000000000000000000068f73b71f53926ab01aebE46cf30a5283d58249c

View Input As
Decode Input Data

Click to see Less

startOpening ادمين (اكانت اول)

[illegible]

Opening توسط اکانت دوم.

[illegible]

Opening توسط اکانت سوم

[illegible]

Opening توسط اکانت چهارم.

```
✓ [block:9286976 txIndex:10] from: 0x7C3...fe7f4 to: CustomAuction.finalize() 0x798...29366 value: 0 wei data: 0x4bb...278f3 logs: 1
hash: 0x990...c4622

status true Transaction mined and execution succeed

transaction hash 0x9904db10041087483cf322bfc01678f8eba73179608814fd04387504be0c4622

from 0x7C34Ef063ac58897429297555141CdD5d95fe7f4

to CustomAuction.finalize() 0x79848d69b541D265DC6441A05d4C85bf2a029366

gas 33493 gas

transaction cost 33493 gas

hash 0x9904db10041087483cf322bfc01678f8eba73179608814fd04387504be0c4622

input 0x4bb...278f3

decoded input {}

decoded output -

logs [ { "from": "0x79848d69b541D265DC6441A05d4C85bf2a029366", "topic":
"0xd0cf8e3f18a7294910551dae6f3e1e6dc1ce599a026f41196f54bc6aafbb4", "event": "auctionFinished", "args": { "0":
"0x68F73b71F53926ab01aebE46CF30A5283D58249c", "1": "5000000000000000", "winnerAddress":
"0x68F73b71F53926ab01aebE46CF30A5283D58249c", "winnerBid": "5000000000000000" } } ]

value 0 wei
```

Transaction Details

OverviewInternal TxnsLogs (1)State

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0x9904db10041087483cf322bfc01678f8eba73179608814fd04387504be0c4622

Status:

Success

Block:

9286976232 Block Confirmations

Timestamp:

28 mins ago (Dec-18-2020 05:26:04 PM +UTC)

From:

0x7c34ef063ac5b897429297555141cdd5d95fe7f4

To:

Contract 0x79848d69b541d265dc6441a05d4c85bf2a029366

Value:

0 Ether (\$0.00)

Transaction Fee:

0.00018357182426 Ether (\$0.000000)

Gas Price:

0.000000005480901211 Ether (5.480901211 Gwei)

Gas Limit:

33,493

Gas Used by Transaction:

33,493 (100%)

Nonce

1810

Input Data:

Function: finalize() ***
MethodID: 0x4bb278f3

This website uses cookies to improve your experience and has an updated Privacy Policy. [Got it](#)

تمام شدن و انتخاب شدن اکانت ۴.(کمترین)

6. سوال ٦:

Results

Candidate	Votes
5	5
7	2
9	2
12	1

Candidate ID

Wallet Address

0x9049e5d2a96e3261b9bce1ccf5297b56478c5e8a ▼

Vote

Close