Blockchain Assignment 1
Creating a DApp

Ram Kartikeya Boyini
3340-2126

The Functions:

```
More Explainations about these in the comments in source code<>.  Here are

some points:



function bid() public payable

~Only bids higher than the highest bids are placed, we use 'require' here.

~Return the previous highest bid to the previous highest bidder by adding

his remaining payable balance to it.



function withdraw() public returns (bool)

~Preventing reentrancy just by setting the amount to 0 till withdrawal is

complete.

~Also, if the transaction is unsuccessful, we replace the money back to

the biider's acc in the pending_returns Map.



function auctionEnd()

~Adding a flag that changes to 1 to avoid repeated endAuction Calls

~Also I used require to verify only beneficiary(the initiator) is the one

who successfully executes auctionEnd()
```

The Demo and Gas Price:

To calculate Gas value in ETH > We approximate 20gwei per Gas.
1000000000 gwei = 1 ETH

The compiled output looks like this:

```
PS C:\Users\hp\Documents\BlockchainDapp\hw1-source> truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\Auction.sol
> Compiling .\contracts\Migrations.sol
> Artifacts written to C:\Users\hp\Documents\BlockchainDapp\hw1-source\build\contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang

PS C:\Users\hp\Documents\BlockchainDapp\hw1-source>
```

The truffle migrate command looks like this:

```
Starting migrations...
======================
> Network name:    'ganache'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0xafebfc3137137bb743f132cc1d2af62c1404bf2338b97d4c55772e20c2a14c38
   > Blocks: 0            Seconds: 0
   > contract address:    0x1DA14EE89b41332E24CDD48adF8F4B2A776485 46
   > block number:        1
   > block timestamp:     1644545465
   > account:             0x3030507F48bEC7b190e047F46387F60D24e3c8ec
   > balance:             99.9967165
   > gas used:            164175 (0x2814f)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.0032835 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:           0.0032835 ETH


2_deploy_contracts.js
=====================

   Replacing 'Auction'
   -------------------
   > transaction hash:    0xf2289310bf9cc56f2b3601721bdd8b671b7b6c65ff7dea278dd815d3d4d2f014
   > Blocks: 0            Seconds: 0
   > contract address:    0x9D7bd2F7fD2830A76202B9476dF5de05d93BaCA7
   > block number:        3
   > block timestamp:     1644545468
   > account:             0x3030507F48bEC7b190e047F46387F60D24e3c8ec
   > balance:             99.98790324
   > gas used:            398322 (0x613f2)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00796644 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00796644 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.01124994 ETH


PS C:\Users\hp\Documents\BlockchainDapp\hw1-source>
```

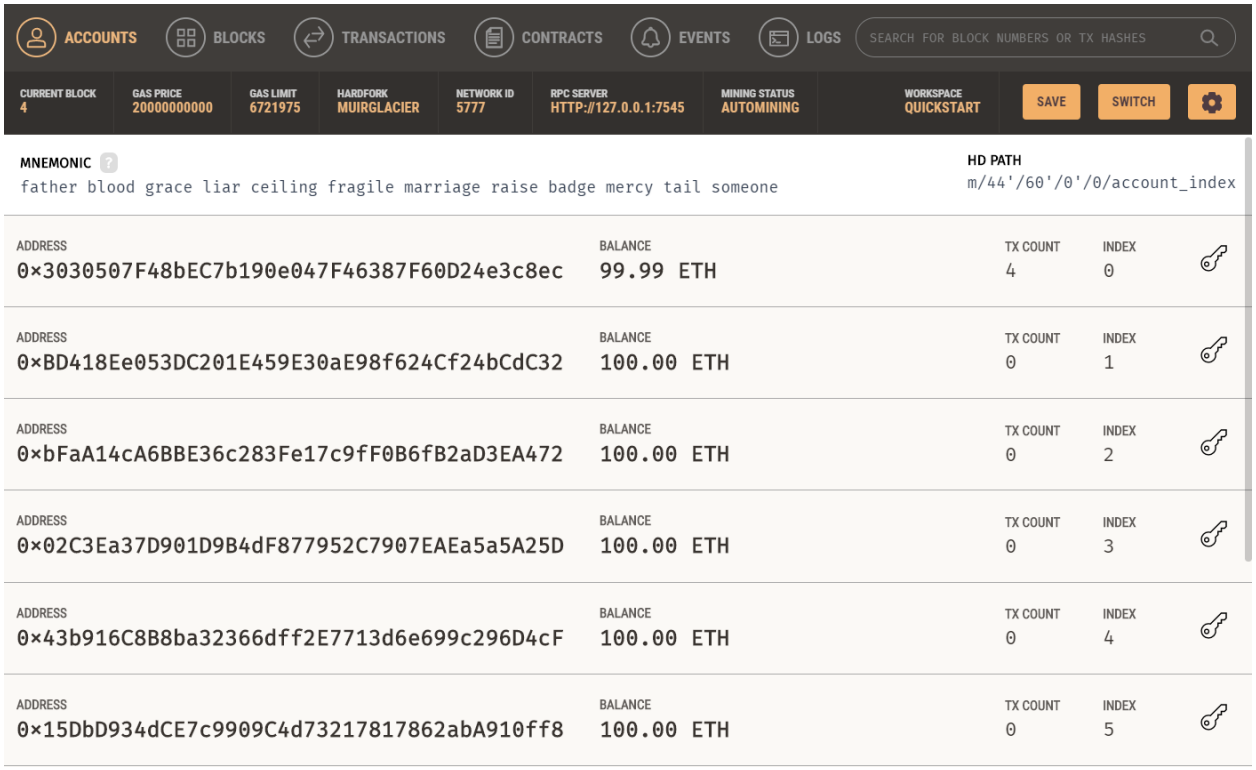
Here we see that for the entire initial deployment of the Auction :

We require a total cost of `0.00796644 ETH`
We also see that around 500,000 gas is used.

The first contract is the beneficiary.

The Ganache console looks like this. The gas cost for deployment has been deducted from the beneficiary.

Note: Ganache takes 10 uses for experiment with 100 ETH balance each.



We enter console by doing the following

```
PS C:\Users\hp\Documents\BlockchainDapp\hw1-source> truffle console
truffle(ganache)>
```

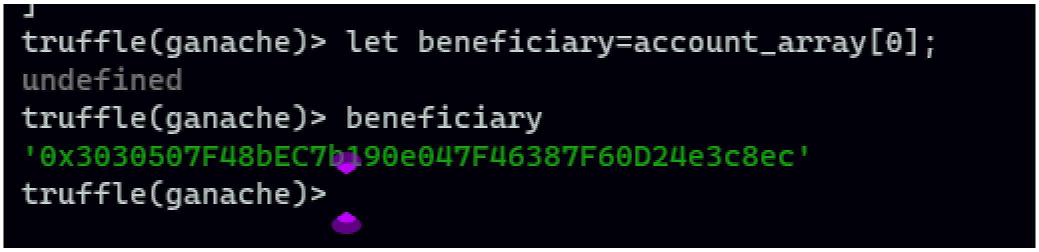
Then we take the instance of the deployed auction:

```
truffle(ganache)> const instance = await Auction.deployed();
undefined
truffle(ganache)> instance
TruffleContract {
  constructor: [Function: TruffleContract] {
    _constructorMethods: {
      configureNetwork: [Function: configureNetwork],
      setProvider: [Function: setProvider],
      new: [Function: new],
      at: [AsyncFunction: at],
      deployed: [AsyncFunction: deployed],
      defaults: [Function: defaults],
      hasNetwork: [Function: hasNetwork],
      isDeployed: [Function: isDeployed],
      detectNetwork: [AsyncFunction: detectNetwork],
      setNetwork: [Function: setNetwork],
      setNetworkType: [Function: setNetworkType],
      setWallet: [Function: setWallet],
      resetAddress: [Function: resetAddress],
      link: [Function: link],
      clone: [Function: clone],
      addProp: [Function: addProp],
      toJSON: [Function: toJSON],
```

The account array looks like this:

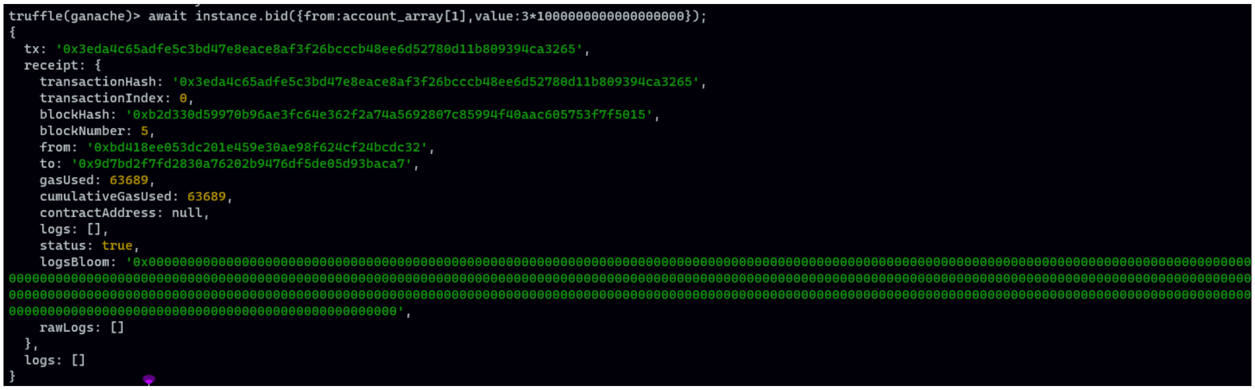
```
truffle(ganache)> let account_array=await web3.eth.getAccounts();
undefined
truffle(ganache)> account_array
[
  '0x3030507F48bEC7b190e047F46387F60D24e3c8ec',
  '0xBD418Ee053DC201E459E30aE98f624Cf24bCdC32',
  '0xbFaA14cA6BBE36c283Fe17c9fF0B6fB2aD3EA472',
  '0x02C3Ea37D901D9B4dF877952C7907EAEa5a5A25D',
  '0x43b916C8B8ba32366dff2E7713d6e699c296D4cF',
  '0x15DbD934dCE7c9909C4d73217817862abA910ff8',
  '0x8f218156121655956Ae69095074396769c893C85',
  '0xBB5d44dB357cbf5C44F3bAbB7d1f29487c1dF836',
  '0x15846667707c19Ba43eddb143d0b82f096Cc84bE',
  '0x96ee3807BFa7030DaD8e512d43dc34626BbfB19b'
]
truffle(ganache)>
```

We set the first account as the beneficiary:



```
]
truffle(ganache)> let beneficiary=account_array[0];
undefined
truffle(ganache)> beneficiary
'0x3030507F48bEC7b190e047F46387F60D24e3c8ec'
truffle(ganache)>
```

When Account 1 places a 3 ether bid :



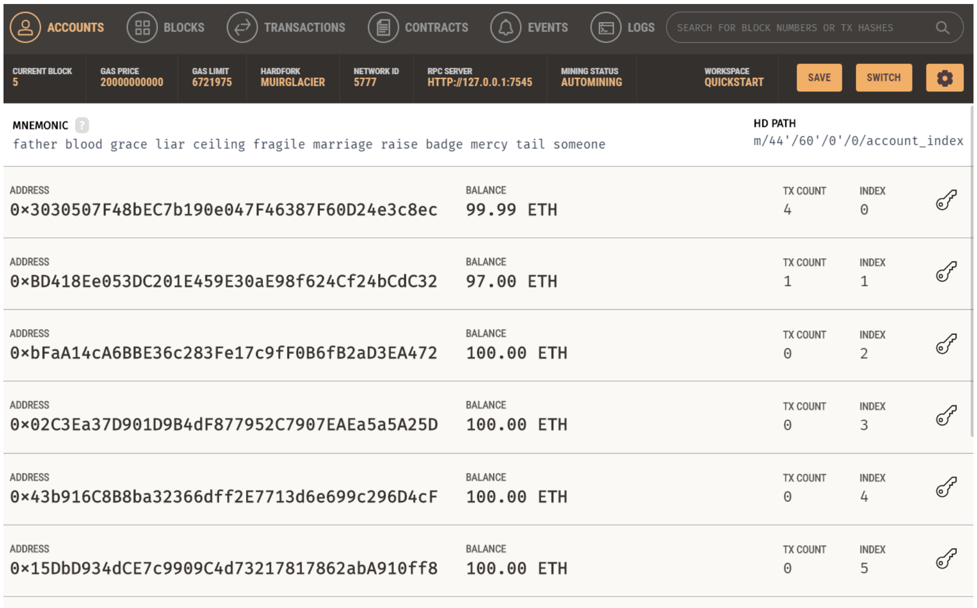
```
truffle(ganache)> await instance.bid({from:account_array[1],value:3*1000000000000000000});
{
  tx: '0x3eda4c65adfe5c3bd47e8eace8af3f26bcccb48ee6d52780d11b809394ca3265',
  receipt: {
    transactionHash: '0x3eda4c65adfe5c3bd47e8eace8af3f26bcccb48ee6d52780d11b809394ca3265',
    transactionIndex: 0,
    blockHash: '0xb2d330d59970b96ae3fc64e362f2a74a5692807c85994f40aac605753f7f5015',
    blockNumber: 5,
    from: '0xbd418ee053dc201e459e30ae98f624cf24bcdc32',
    to: '0x9d7bd2f7fd2830a76202b9476df5de05d93baca7',
    gasUsed: 63689,
    cumulativeGasUsed: 63689,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000',
    rawLogs: []
  },
  logs: []
}
```

The Gas used is **63389**

The Gas cost put to 20gwei per gas could be

`0.00126778 ETH`

for placing bid.

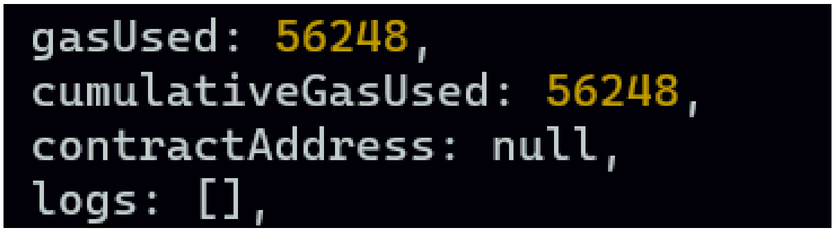


CURRENT BLOCK 5 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | HARDFORK MUIRGLACIER | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING | WORKSPACE QUICKSTART

MNEMONIC
father blood grace liar ceiling fragile marriage raise badge mercy tail someone

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×3030507F48bEC7b190e047F46387F60D24e3c8ec	99.99 ETH	4	0
0×BD418Ee053DC201E459E30aE98f624Cf24bCdC32	97.00 ETH	1	1
0×bFaA14cA6BBE36c283Fe17c9fF0B6fB2aD3EA472	100.00 ETH	0	2
0×02C3Ea37D901D9B4dF877952C7907EAEa5a5A25D	100.00 ETH	0	3
0×43b916C8B8ba32366dff2E7713d6e699c296D4cF	100.00 ETH	0	4
0×15DbD934dCE7c9909C4d73217817862abA910ff8	100.00 ETH	0	5



```
gasUsed: 56248,
cumulativeGasUsed: 56248,
contractAddress: null,
logs: [],
```

When Account 3 places a 5 ether bid:
Gas Used is 56248
Cost is 20gwei

`0.00112496 ETH` for placing a bid again.

For ending auction:

```
truffle(ganache)>  await instance.auctionEnd({from:accounts[0]});
{
  tx: '0x70da8bcdcf52d9750076e001e909c4cb9e2ccb2bd56121f7a976eac97e58
  receipt: {
    transactionHash: '0x70da8bcdcf52d9750076e001e909c4cb9e2ccb2bd5612
    transactionIndex: 0,
    blockHash: '0x07923685801830a2cf3e9819f990f2923bc761202dddc565cf8
    blockNumber: 7,
    from: '0x692281d93e83bcc6bfa87c7d897e56f2edba65c9',
    to: '0x66d46b16e549814d7cd59e44cb8c9f0598105aeb',
    gasUsed: 52956,
    cumulativeGasUsed: 52956,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x00000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000',
    rawLogs: []
  },
  logs: []
}
```

Gas used is 52956
Cost is 20gwei per gas
Therefore-> `0.00105912 ETH` for ending auction.
PS: I deployed the blockchain again as there were power issues with my system;

After ending the auction this is the status ;

CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE		
7	20000000000	6721975	MUIRGLACIER	5777	HTTP://127.0.0.1:7545	AUTOMINING	QUICKSTART	SAVE	SWITCH

MNEMONIC
garden flock boat eternal paddle wage color document aerobic clerk stove aspect

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0x692281d93E83bcc6bFA87C7d897e56f2EDbA65C9	104.99 ETH	5	0
0xD25C7cad964f6E2B6AD5D7acb61f1EBD36d14172	97.00 ETH	1	1
0xA655556eBcef1115EE47a28508469544fcBed277	100.00 ETH	0	2
0xd6AE77d2E5Da0C2000Df959b1eC0Db7ba59D9a1a	95.00 ETH	1	3
0xcdB3613C2196dEEFb2a44cbd25D702f90e8C7d16	100.00 ETH	0	4
0x36470Bf06C96e94F920319202a070277c4A7eEEC	100.00 ETH	0	5
0x69d89E27eA2C2104a3d533F0E91eF739d95f00B0	100.00 ETH	0	6

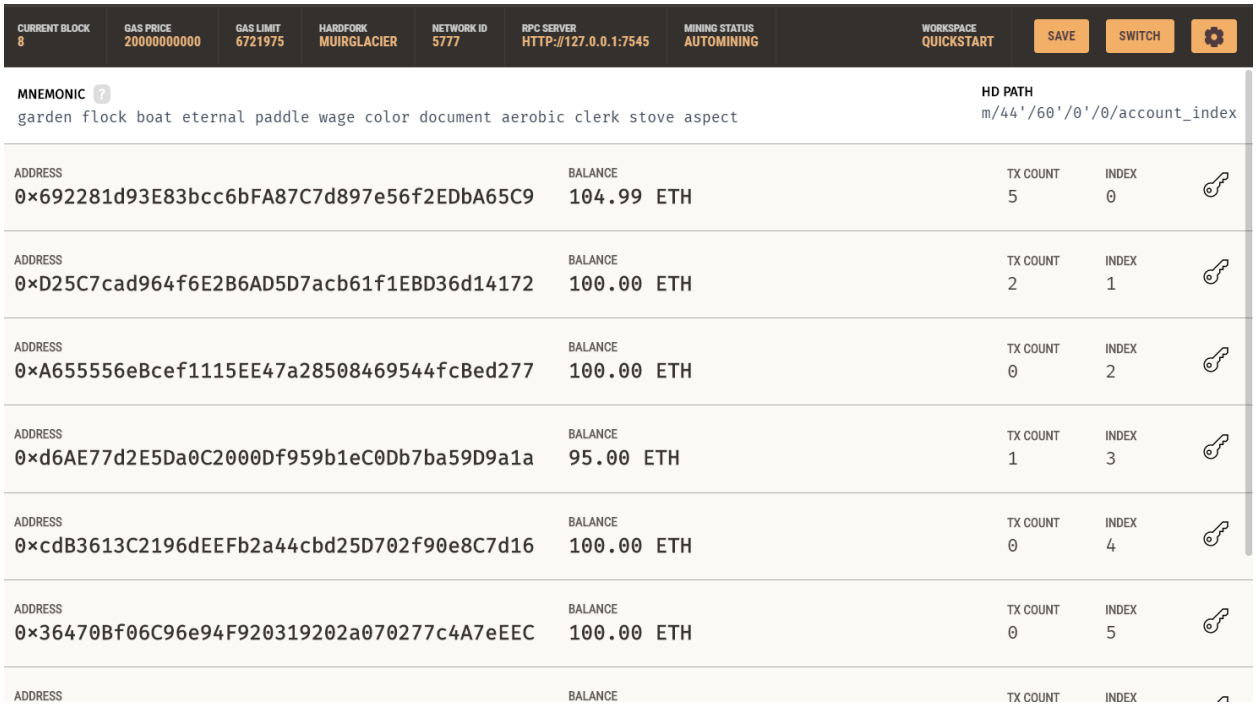
We see that highest bid is deposited to the beneficiary account!

Also, note that money hasn't been deposited back to the bidders. We need to call the withdraw method.

When the Account 1 bidder withdraws money:

```
truffle(ganache)>  await instance.withdraw({from:accounts[1]});
{
  tx: '0xb2b38831f05640fd634aa679ff1f728f2f9f39b3ea5f1f96fb58f423100d6476',
  receipt: {
    transactionHash: '0xb2b38831f05640fd634aa679ff1f728f2f9f39b3ea5f1f96fb58f423100d6476',
    transactionIndex: 0,
    blockHash: '0x48469b37e24c4960e437a7880ba4e6d910d7d97b9e2867fbe526c025e9bd9857',
    blockNumber: 8,
    from: '0xd25c7cad964f6e2b6ad5d7acb61f1ebd36d14172',
    to: '0x66d46b16e549814d7cd59e44cb8c9f0598105aeb',
    gasUsed: 19857,
    cumulativeGasUsed: 19857,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

Gas used is 19857
Therefore cost for withdrawing the contract's value into the account is `0.00039714ETH`


CURRENT BLOCK 8
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE QUICKSTART
SAVE
SWITCH
MNEMONIC
garden flock boat eternal paddle wage color document aerobic clerk stove aspect
HD PATH
m/44'/60'/0'/0/account_index
ADDRESS 0x692281d93E83bcc6bFA87C7d897e56f2EDbA65C9
BALANCE 104.99 ETH
TX COUNT 5
INDEX 0
ADDRESS 0xD25C7cad964f6E2B6AD5D7acb61f1EBD36d14172
BALANCE 100.00 ETH
TX COUNT 2
INDEX 1
ADDRESS 0xA655556eBcef1115EE47a28508469544fcBed277
BALANCE 100.00 ETH
TX COUNT 0
INDEX 2
ADDRESS 0xd6AE77d2E5Da0C2000Df959b1eC0Db7ba59D9a1a
BALANCE 95.00 ETH
TX COUNT 1
INDEX 3
ADDRESS 0xcdB3613C2196dEEFb2a44cbd25D702f90e8C7d16
BALANCE 100.00 ETH
TX COUNT 0
INDEX 4
ADDRESS 0x36470Bf06C96e94F920319202a070277c4A7eEEC
BALANCE 100.00 ETH
TX COUNT 0
INDEX 5
ADDRESS
BALANCE
TX COUNT
INDEX

****