Question 1 - Bitcoin Testnet Transaction

In [ ]:

```
#Import required libraries

from importlib import reload
from helper import run
import ecc
import helper
import script
import tx
```

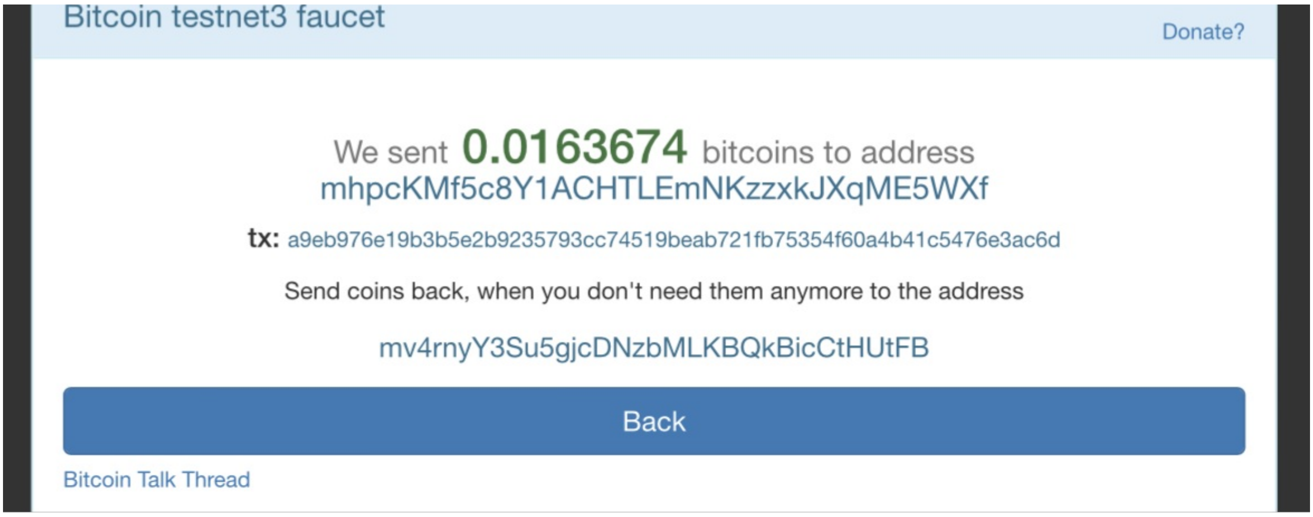
1. Create 4 Bitcoin Testnet addresses. Add below the addresses and the corresponding secrets, namely address1, address2, address3, address4.

In [ ]:

```
from ecc import PrivateKey
from helper import hash256, little_endian_to_int

#Address 1
secret1 = little_endian_to_int(hash256(b'qwerty1'))
private_key1 = PrivateKey(secret1)
address1 = private_key1.point.address(testnet=True)
print("qwerty1 Address: "+ address1)
print("Secret 1: "+ str(secret1))

#Address 2
secret2 = little_endian_to_int(hash256(b'qwerty2'))
private_key2 = PrivateKey(secret2)
address2 = private_key2.point.address(testnet=True)
print("\qwerty2 Address: "+ address2)
print("Secret 2: "+ str(secret2))

#Address 3
secret3 = little_endian_to_int(hash256(b'qwerty3'))
private_key3 = PrivateKey(secret3)
address3 = private_key3.point.address(testnet=True)
print("\nqwerty3 Address: "+ address3)
print("Secret 3: "+ str(secret3))

#Address 4
secret4 = little_endian_to_int(hash256(b'qwerty4'))
private_key4 = PrivateKey(secret4)
address4 = private_key4.point.address(testnet=True)
print("\nqwerty4 Address: "+ address4)
print("Secret 4: "+ str(secret4))
```

```
Address 1: mhpcKMf5c8Y1ACHTLEmNKzzxkJXqME5WXf
Secret 2: 32096358358466310253816131080209211792447520358323468872018607224048626020377

Address 2: mjDPt86pt2S8gEn4TtEyD9fCFexaFT9Ri6
Secret 2: 24943396098195252479067043297771534656501176597458792586405242986383690978899

Address 3: n3NqV5KpWcEMUwi17CxXpHH6zb4YWPA6Sh
Secret 3: 104533069907879726791087557265415846272574193065210720528477029931153684609072

Address 4: mru9wdZVop4EwWzZSQCF69AiSSiRPyMHP4
Secret 4: 40509952545988927686870876049803613874045357454363086651978550990193568726178
```

2. From a BTC testnet faucet send testnet bitcoin to one of the addresses.

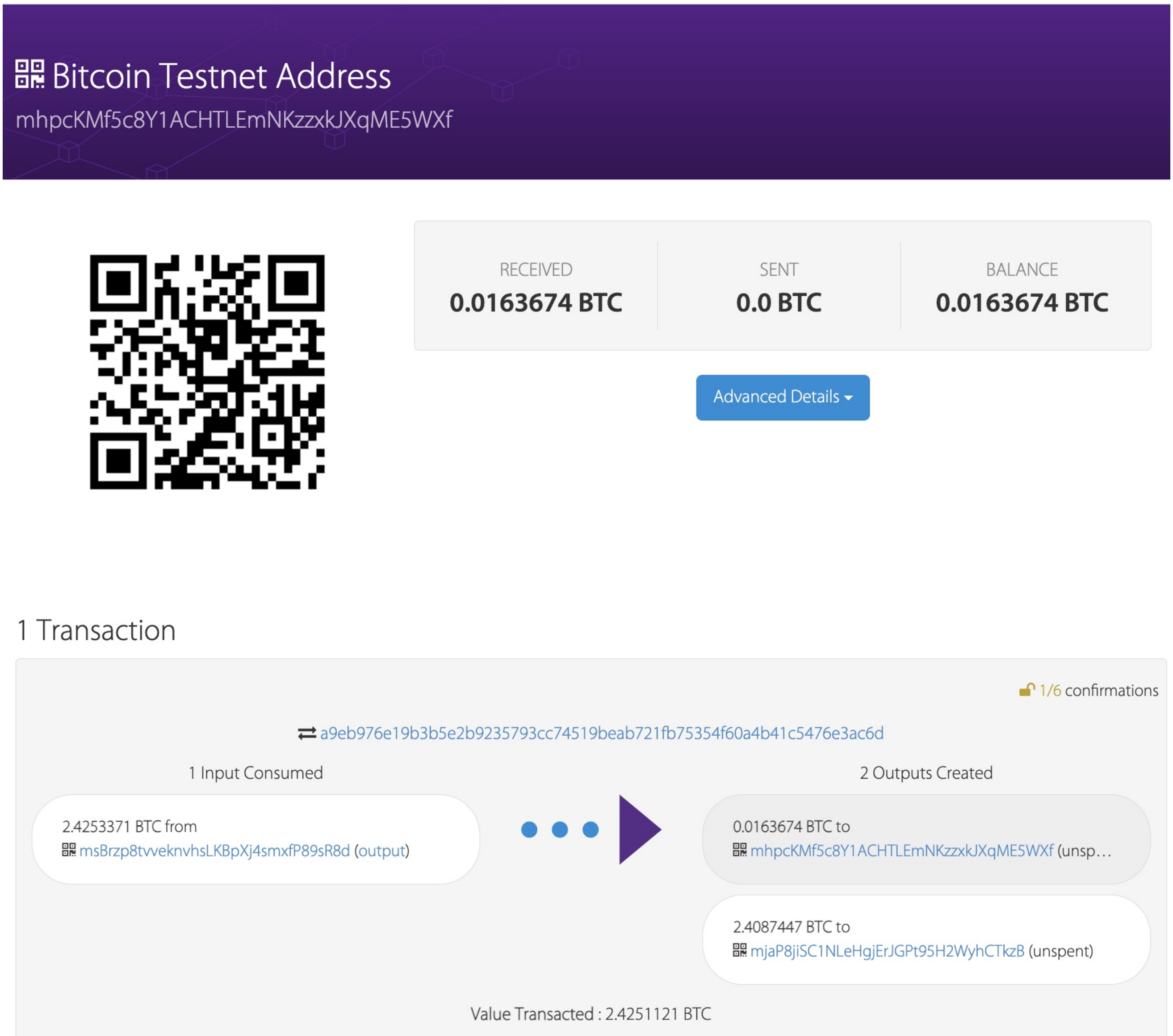


We sent **0.0163674** bitcoins to address
mhpcKMf5c8Y1ACHTLEmNKzzxkJXqME5WXf

tx: a9eb976e19b3b5e2b9235793cc74519beab721fb75354f60a4b41c5476e3ac6d

Send coins back, when you don't need them anymore to the address

mv4rnyY3Su5gjcDNzbMLKBQkBicCtHUtFB



Bitcoin Testnet Address

mhpcKMf5c8Y1ACHTLEmNKzzxkJXqME5WXf

RECEIVED	SENT	BALANCE
0.0163674 BTC	0.0 BTC	0.0163674 BTC

1 Transaction

1/6 confirmations

⇄ a9eb976e19b3b5e2b9235793cc74519beab721fb75354f60a4b41c5476e3ac6d

1 Input Consumed

2.4253371 BTC from
msBrzp8tvveknvhsLKBpXj4smxfP89sR8d (output)

2 Outputs Created

0.0163674 BTC to
mhpcKMf5c8Y1ACHTLEmNKzzxkJXqME5WXf (unsp…

2.4087447 BTC to
mjaP8jiSC1NLeHgjErJGPt95H2WyhCTkzB (unspent)

Value Transacted : 2.4251121 BTC

Bitcoin received: 0.0163674 BTC
Transaction ID: a9eb976e19b3b5e2b9235793cc74519beab721fb75354f60a4b41c5476e3ac6d

3. Create a 1-input 3-outputs transactions transferring 50%, 30%, 15% of the amount respectively into address2, address3, address4 respectively.

In [ ]:

```
from helper import decode_base58, SIGHASH_ALL
from script import p2pkh_script, Script
from tx import TxIn, TxOut, Tx

# 1 input
# Define previous transaction when qwerty1 received 0.0163674 BTC (1636740 Satoshis) from BTC testnet3 faucet
prev_tx = bytes.fromhex('a9eb976e19b3b5e2b9235793cc74519beab721fb75354f60a4b41c5476e3ac6d')
prev_index = 0
tx_in1 = TxIn(prev_tx, prev_index)

# 3 outputs
tx_outs = []
# Transfer 0.0081837 BTC (50%) to qwerty2 - Output1
target_amount1 = int(818370)
target_h160_1 = decode_base58('mjDPt86pt2S8gEn4TtEyD9fCFexaFT9Ri6')
target_script1 = p2pkh_script(target_h160_1)
target_output1 = TxOut(amount=target_amount1, script_pubkey=target_script1)

# Transfer 0.00491022 BTC (30%) to qwerty3 - Output2
target_amount2 = int(491022)
target_h160_2 = decode_base58('n3NqV5KpWcEMUwi17CxXpHH6zb4YWPA6Sh')
target_script2 = p2pkh_script(target_h160_2)
target_output2 = TxOut(amount=target_amount2, script_pubkey=target_script2)
```

```python
# Transfer 0.00245511 BTC (15%) to qwerty4 - Output3
target_amount3 = int(245511)
target_h160_3 = decode_base58('mru9wdZVop4EwWzZSQCF69AiSSiRPyMHP4')
target_script3 = p2pkh_script(target_h160_3)
target_output3 = TxOut(amount=target_amount3, script_pubkey=target_script3)

# Define the transaction
tx_obj = Tx(1, [tx_in1], [target_output1, target_output2, target_output3], 0, True)
print(tx_obj)
```

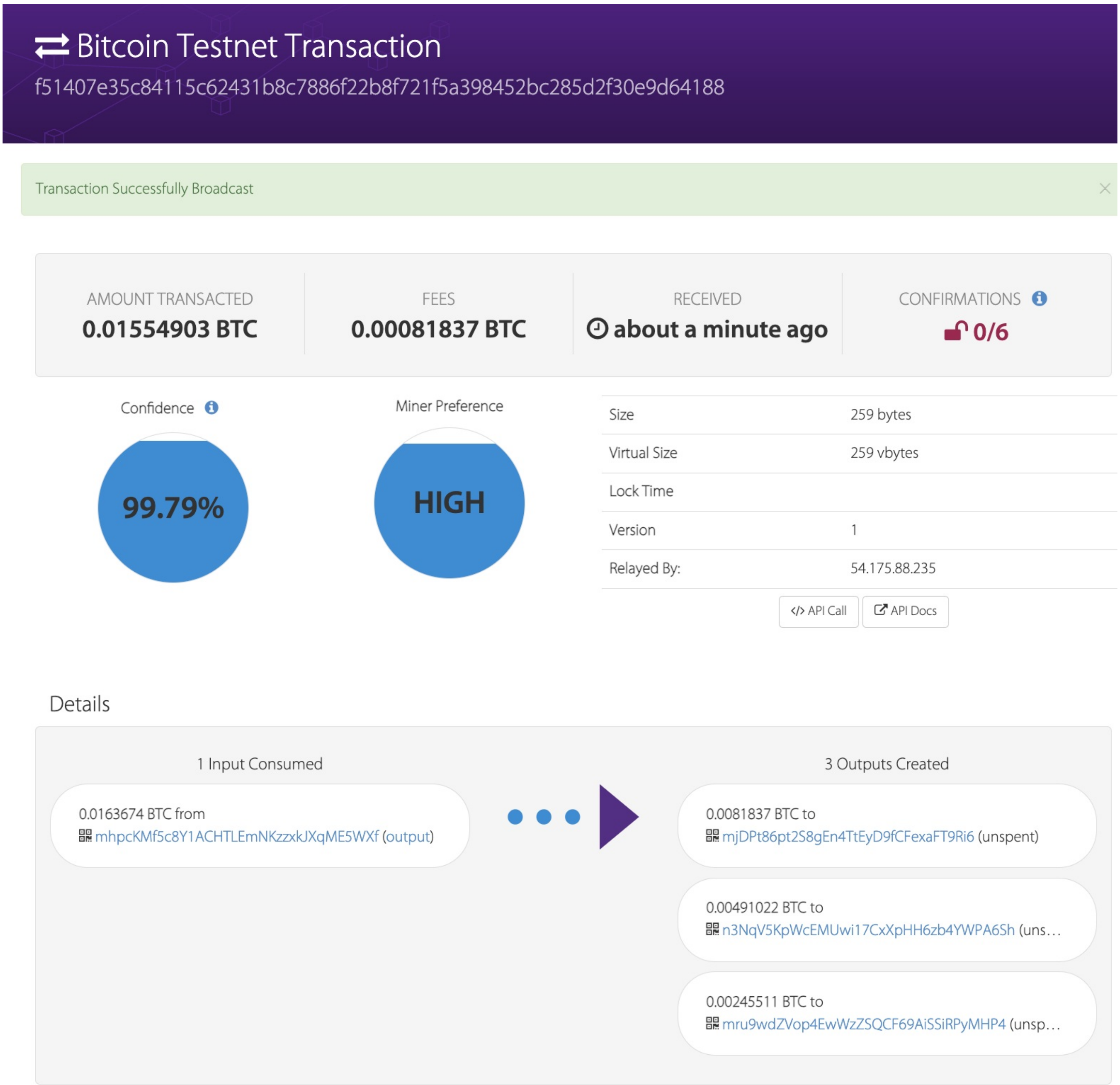
```
tx: 58d996d5391bb26febd1ea805f4e160d0379b62bb87aed7630510541339eec5c
version: 1
tx_ins:
a9eb976e19b3b5e2b9235793cc74519beab721fb75354f60a4b41c5476e3ac6d:0
tx_outs:
818370:OP_DUP OP_HASH160 288e60072517588748cc187f5514ddc9af96023b OP_EQUALVERIFY OP_CHECKSIG
491022:OP_DUP OP_HASH160 efc9c13eefad7b44818cf5dbc78dafde7ca39dfe OP_EQUALVERIFY OP_CHECKSIG
245511:OP_DUP OP_HASH160 7cdc42088d7e55e84c262b73430ae1d540c7009e OP_EQUALVERIFY OP_CHECKSIG
locktime: 0
```

4. Sign the transaction and submit it into the Bitcoin testnet via https://live.blockcypher.com/btc-testnet/pushtx/

In [ ]:

```python
#Sign the transaction
from ecc import PrivateKey
from helper import SIGHASH_ALL
z = tx_obj.sig_hash(0)

# use qwerty1's secret
private_key = PrivateKey(secret=32096358358466310253816131080209211792447520358323468872018607224048626020377)
der = private_key.sign(z).der()
sig = der + SIGHASH_ALL.to_bytes(1, 'big')
sec = private_key.point.sec()
script_sig = Script([sig, sec])
tx_obj.tx_ins[0].script_sig = script_sig
print(tx_obj.serialize().hex())
```

```
01000000016dace376541cb4a4604f3575fb21b7ea9b5174cc935723b9e2b5b3196e97eba9000000006a47304402204c84af3c4a8494d1ea
0cfa709e597904cca453090df85ec3e8fc05edfc521e69022066687ea58d78f83158416f10802478eb90990372a1cfa5e571e4adc177371e
88012102311697f6be5b47230288155e83166105dcd9c98e92dd900c31ce0c193743a37dffffffff03c27c0c00000000001976a914288e60
072517588748cc187f5514ddc9af96023b88ac0e7e0700000000001976a914efc9c13eefad7b44818cf5dbc78dafde7ca39dfe88ac07bf03
00000000001976a9147cdc42088d7e55e84c262b73430ae1d540c7009e88ac00000000
```



⇄ Bitcoin Testnet Transaction

f51407e35c84115c62431b8c7886f22b8f721f5a398452bc285d2f30e9d64188

Transaction Successfully Broadcast ×

AMOUNT TRANSACTED	FEES	RECEIVED	CONFIRMATIONS
0.01554903 BTC	0.00081837 BTC	about a minute ago	0/6

Confidence: 99.79%

Miner Preference: HIGH

Size	259 bytes
Virtual Size	259 vbytes
Lock Time	
Version	1
Relayed By:	54.175.88.235

Details

1 Input Consumed

0.0163674 BTC from
mhpcKMf5c8Y1ACHTLEmNKzzxkJXqME5WXf (output)

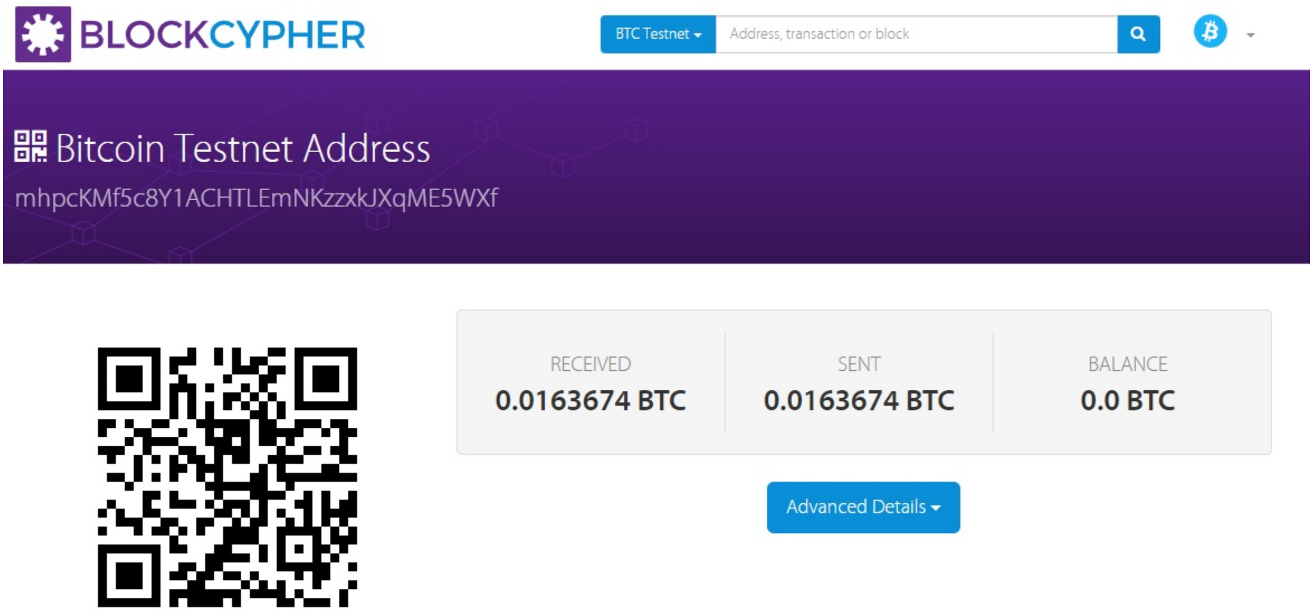
3 Outputs Created

0.0081837 BTC to
mjDPt86pt2S8gEn4TtEyD9fCFexaFT9Ri6 (unspent)

0.00491022 BTC to
n3NqV5KpWcEMUwi17CxXpHH6zb4YWPA6Sh (uns…

0.00245511 BTC to
mru9wdZVop4EwWzZSQCF69AiSSiRPyMHP4 (unsp…

URL: https://live.blockcypher.com/btc-testnet/tx/f51407e35c84115c62431b8c7886f22b8f721f5a398452bc285d2f30e9d64188/



Bitcoin Testnet Address

mhpcKMf5c8Y1ACHTLEmNKzzxkJXqME5WXf

RECEIVED	SENT	BALANCE
0.0163674 BTC	0.0163674 BTC	0.0 BTC

Balance of Address 1

URL: https://live.blockcypher.com/btc-testnet/address/mhpcKMf5c8Y1ACHTLEmNKzzxkJXqME5WXf/

5. How much fees are transferred to the miner and how are they calculated?

The transaction fee is calculated from difference between the total inputs and total outputs. In this scenario there is 1 input and 3 outputs.

Input 1: 0.0163674 BTC Output 1: 0.0081837 BTC Output 2: 0.00491022 BTC Output 3: 0.00245511 BTC

Transaction fee = 0.0163674 - (0.0081837 + 0.00491022 + 0.00245511) = 0.00081837 BTC

Question 2 - Smart contract for cash in DAML

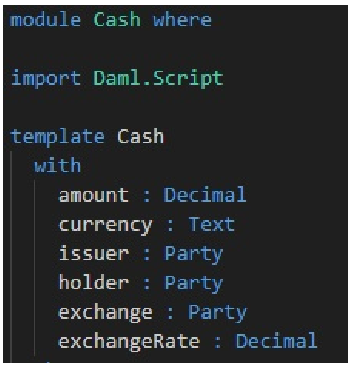
Contract template

1. Start by creating a new project and .daml file.

Using the following command in terminal

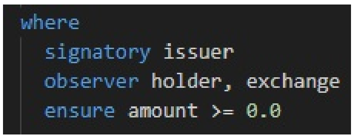
daml new Cash

2. Create a contract template called Cash with parameters



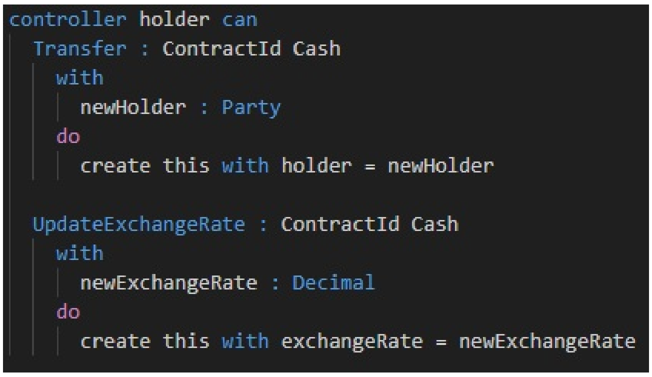
```
module Cash where

import Daml.Script

template Cash
  with
    amount : Decimal
    currency : Text
    issuer : Party
    holder : Party
    exchange : Party
    exchangeRate : Decimal
```

3. Then, define the roles of the parties. What type of party should the issuer be? And the exchange?
4. Add a condition to ensure the amount of cash is larger than zero



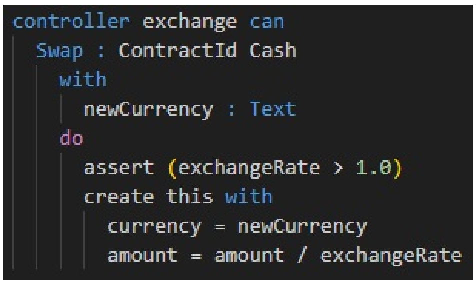
```
  where
    signatory issuer
    observer holder, exchange
    ensure amount >= 0.0
```

5. Add a function Transfer which transfer the cash to new holder, where the controller is the holder
6. Add a function UpdateExchangeRate which sets a new Exchange rate, where the controller is the holder



```
controller holder can
  Transfer : ContractId Cash
    with
      newHolder : Party
    do
      create this with holder = newHolder

  UpdateExchangeRate : ContractId Cash
    with
      newExchangeRate : Decimal
    do
      create this with exchangeRate = newExchangeRate
```

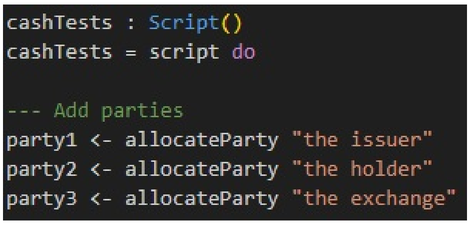
7. Add a function Swap which converts the currency (currency = newCurrency), updates the amount with the specified exchange rate (amount = amount/exchangeRate), and asserts the exchange rate is larger than 1.0 (exchangeRate > 1.0)



```
controller exchange can
  Swap : ContractId Cash
    with
      newCurrency : Text
    do
      assert (exchangeRate > 1.0)
      create this with
        currency = newCurrency
        amount = amount / exchangeRate
```

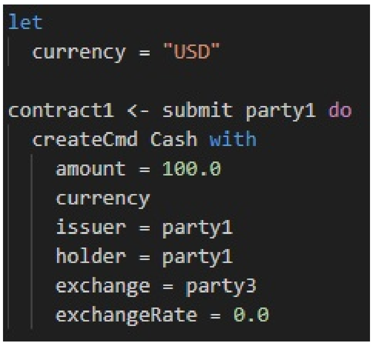
Scenario testing

1. Create three parties: “Party_1” (the issuer), “Party_2” (the holder), “Party_3” (the exchange)



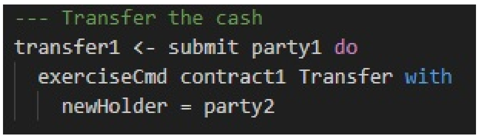
```
cashTests : Script()
cashTests = script do

--- Add parties
party1 <- allocateParty "the issuer"
party2 <- allocateParty "the holder"
party3 <- allocateParty "the exchange"
```

2. Let the issuer “Party 1” issue a new contract where the issuer “Party 1” wishes to transfer 100 USD to “Party 2”. At this stage set the issuer = the holder (2 points), and the exchangeRate = 0.0



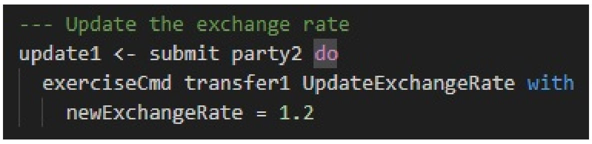
```
let
  currency = "USD"

contract1 <- submit party1 do
  createCmd Cash with
    amount = 100.0
    currency
    issuer = party1
    holder = party1
    exchange = party3
    exchangeRate = 0.0
```

3. Let the holder (=issuer) transfer the cash to “Party 2”



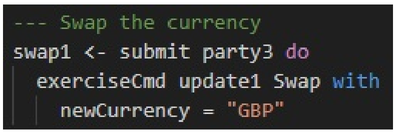
```
--- Transfer the cash
transfer1 <- submit party1 do
  exerciseCmd contract1 Transfer with
    newHolder = party2
```

4. Let the new holder “Party 2” update the contract with exchangeRate = 1.2



```
--- Update the exchange rate
update1 <- submit party2 do
  exerciseCmd transfer1 UpdateExchangeRate with
    newExchangeRate = 1.2
```

5. Let the exchange “Party 3” swap USD to GBP .



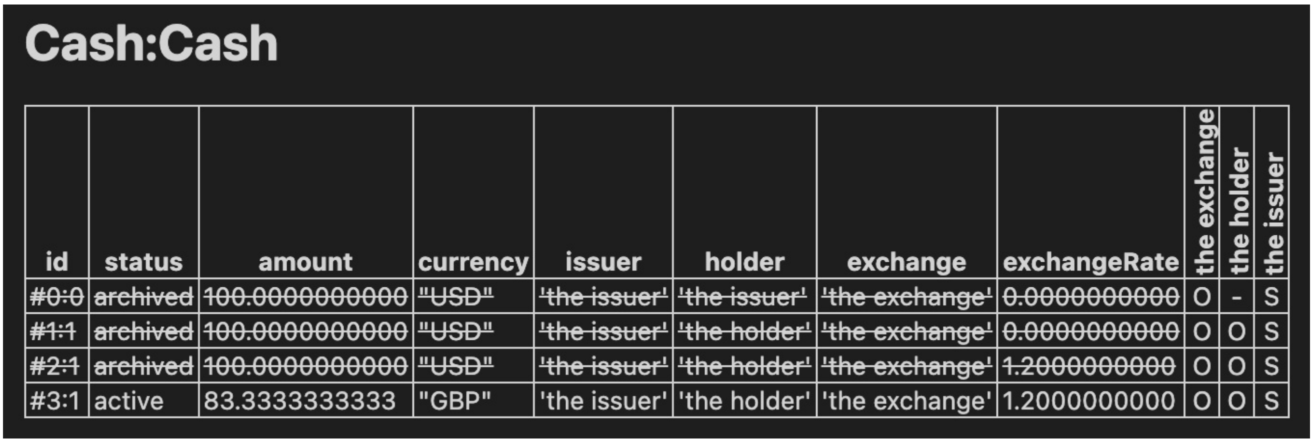
```
--- Swap the currency
swap1 <- submit party3 do
  exerciseCmd update1 Swap with
    newCurrency = "GBP"
```

6. Try to let the holder do the swap. What will happen? Explain why this would happen.

"Script execution failed, displaying state before failing transaction"

Holder doesn't have the authority to use the swap function, swap is only available to exchange party

Ledger State

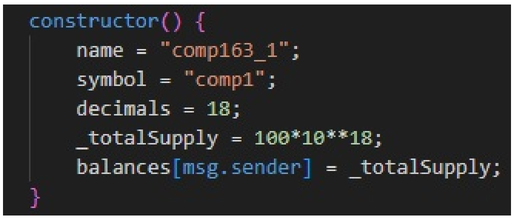


Cash:Cash

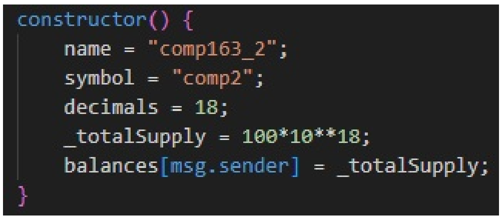
id	status	amount	currency	issuer	holder	exchange	exchangeRate	the exchange	the holder	the issuer
~~#0:0~~	~~archived~~	~~100.0000000000~~	~~"USD"~~	~~'the issuer'~~	~~'the issuer'~~	~~'the exchange'~~	~~0.0000000000~~	O	-	S
~~#1:1~~	~~archived~~	~~100.0000000000~~	~~"USD"~~	~~'the issuer'~~	~~'the holder'~~	~~'the exchange'~~	~~0.0000000000~~	O	O	S
~~#2:1~~	~~archived~~	~~100.0000000000~~	~~"USD"~~	~~'the issuer'~~	~~'the holder'~~	~~'the exchange'~~	~~1.2000000000~~	O	O	S
#3:1	active	83.3333333333	"GBP"	'the issuer'	'the holder'	'the exchange'	1.2000000000	O	O	S

Question 3 - ERC20 and AMM Deployment

1. Create two tokens with the ERC20 interface



```
constructor() {
    name = "comp163_1";
    symbol = "comp1";
    decimals = 18;
    _totalSupply = 100*10**18;
    balances[msg.sender] = _totalSupply;
}
```

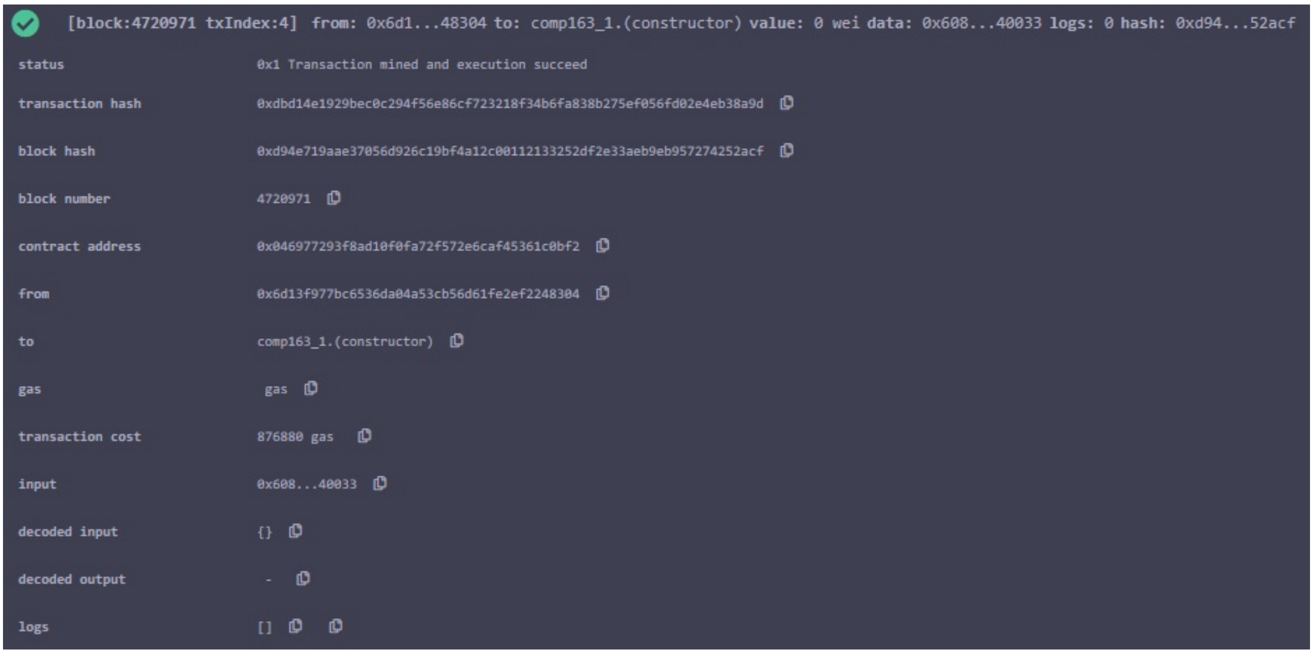


```
constructor() {
    name = "comp163_2";
    symbol = "comp2";
    decimals = 18;
    _totalSupply = 100*10**18;
    balances[msg.sender] = _totalSupply;
}
```

Please refer to files "comp163_1_ERC20.sol" and "comp163_2_ERC20.sol" under the folder "Q3 ERC20_AMM_Deloyment

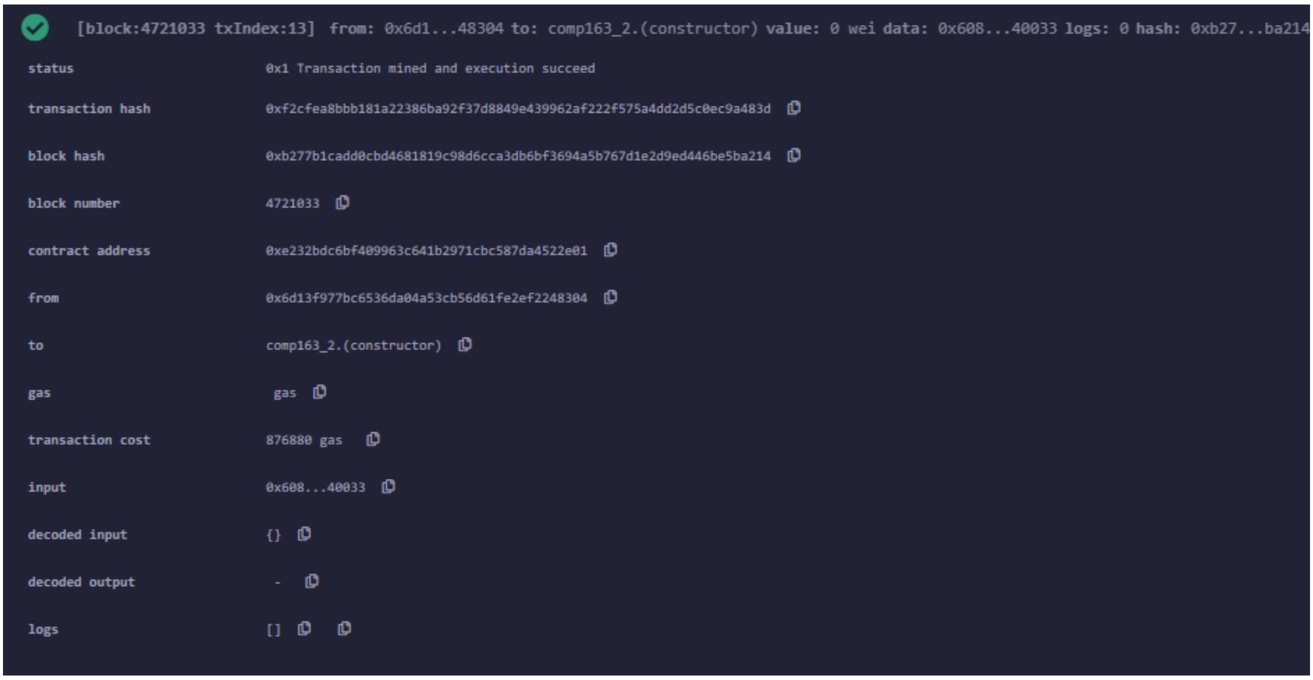
2. Test the ERC20 contracts by deploying it on the Remix VM

comp1 transaction overview

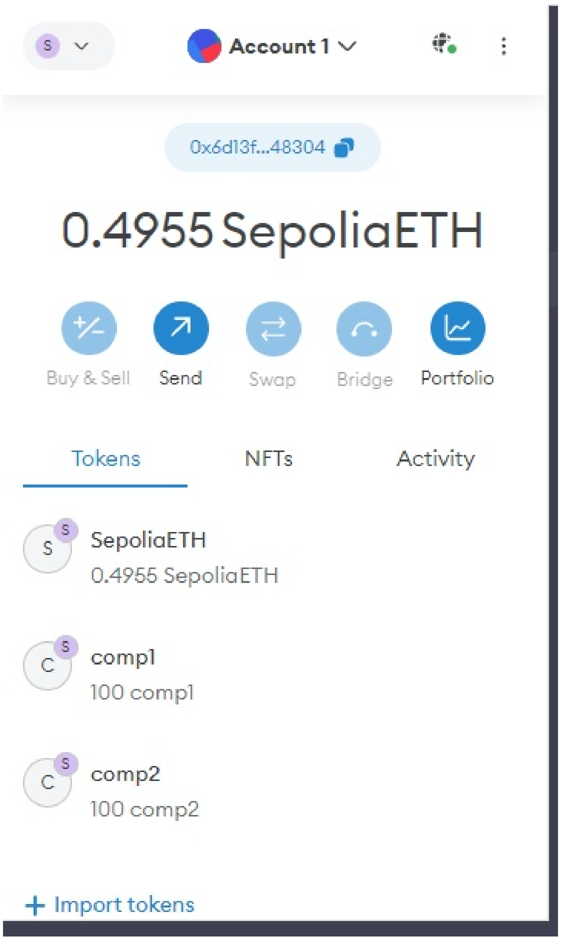


```
[block:4720971 txIndex:4]  from: 0x6d1...48304 to: comp163_1.(constructor) value: 0 wei data: 0x608...40033 logs: 0 hash: 0xd94...52acf
status               0x1 Transaction mined and execution succeed
transaction hash     0xdbd14e1929bec0c294f56e86cf723218f34b6fa838b275ef056fd02e4eb38a9d
block hash           0xd94e719aae37056d926c19bf4a12c00112133252df2e33aeb9eb957274252acf
block number         4720971
contract address     0x046977293f8ad10f0fa72f572e6caf45361c0bf2
from                 0x6d13f977bc6536da04a53cb56d61fe2ef2248304
to                   comp163_1.(constructor)
gas                   gas
transaction cost     876880 gas
input                0x608...40033
decoded input        {}
decoded output        -
logs                 []
```

comp2 transaction overview



```
[block:4721033 txIndex:13] from: 0x6d1...48304 to: comp163_2.(constructor) value: 0 wei data: 0x608...40033 logs: 0 hash: 0xb27...ba214
status                    0x1 Transaction mined and execution succeed
transaction hash          0xf2cfea8bbb181a22386ba92f37d8849e439962af222f575a4dd2d5c0ec9a483d
block hash                0xb277b1cadd0cbd4681819c98d6cca3db6bf3694a5b767d1e2d9ed446be5ba214
block number              4721033
contract address          0xe232bdc6bf409963c641b2971cbc587da4522e01
from                      0x6d13f977bc6536da04a53cb56d61fe2ef2248304
to                        comp163_2.(constructor)
gas                        gas
transaction cost          876880 gas
input                     0x608...40033
decoded input             {}
decoded output             -
logs                      []
```



0x6d13f...48304

0.4955 SepoliaETH

SepoliaETH
0.4955 SepoliaETH

comp1
100 comp1

comp2
100 comp2

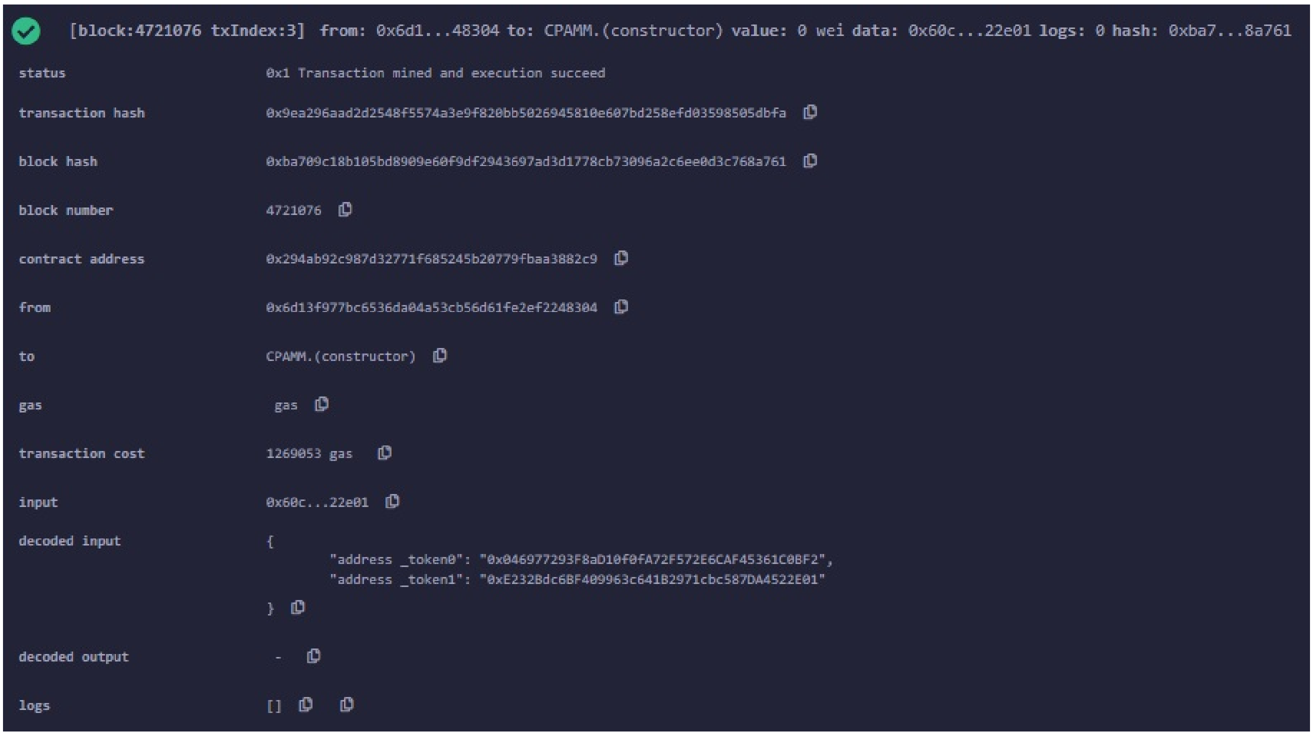
What's the balance of comp1 and comp2 for your address? 100 tokens each

3. Create a simple constant AMM contract by replacing "____" with the actual codes

Please refer to the file "AMM.sol"

4. Test the AMM contract by deploying it on the Remix VM. What's the transaction hash of your creatation of the contract

AMM Transaction overview

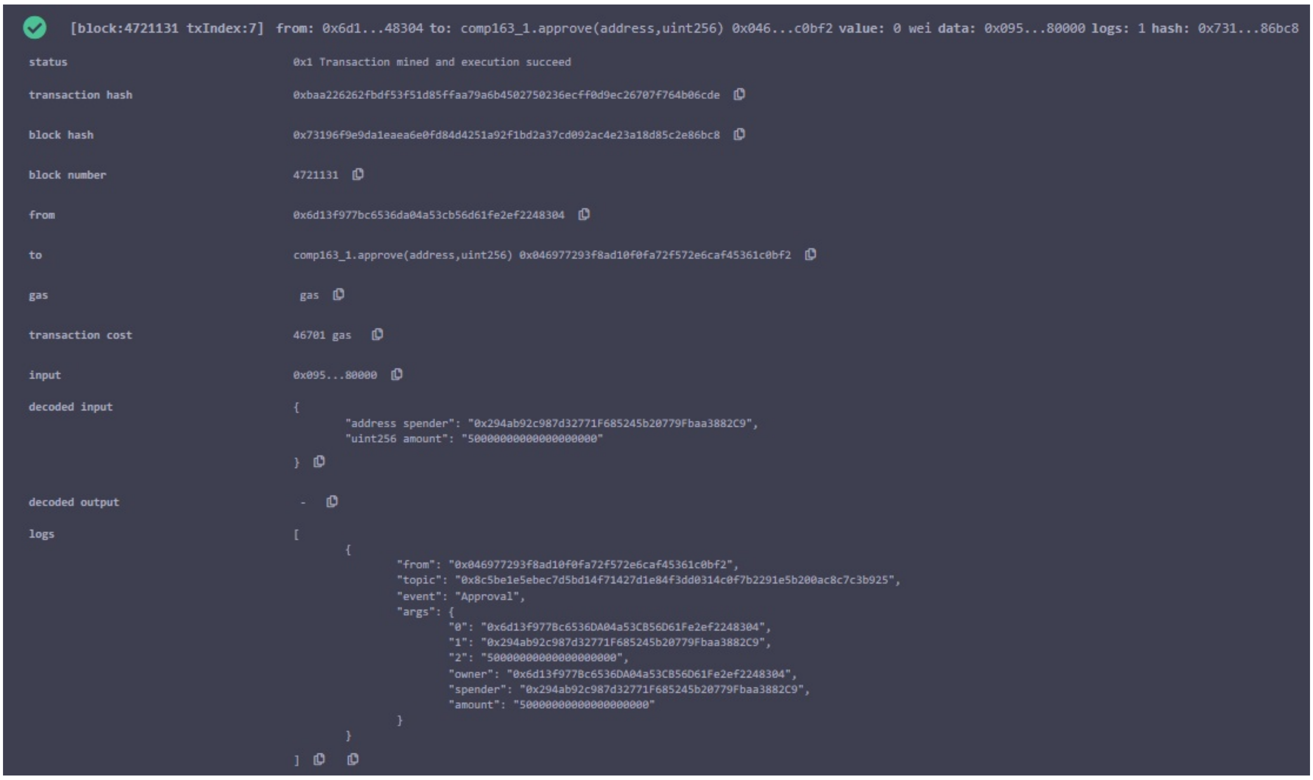


```
[block:4721076 txIndex:3] from: 0x6d1...48304 to: CPAMM.(constructor) value: 0 wei data: 0x60c...22e01 logs: 0 hash: 0xba7...8a761
status              0x1 Transaction mined and execution succeed
transaction hash    0x9ea296aad2d2548f5574a3e9f820bb5026945810e607bd258efd03598505dbfa
block hash          0xba709c18b105bd8909e60f9df2943697ad3d1778cb73096a2c6ee0d3c768a761
block number        4721076
contract address    0x294ab92c987d32771f685245b20779fbaa3882c9
from                0x6d13f977bc6536da04a53cb56d61fe2ef2248304
to                  CPAMM.(constructor)
gas                  gas
transaction cost    1269053 gas
input               0x60c...22e01
decoded input       {
                            "address _token0": "0x046977293F8aD10f0FA72F572E6CAF45361C0BF2",
                            "address _token1": "0xE232Bdc6BF409963c641B2971cbc587DA4522E01"
                    }
decoded output       -
logs                []
```

AMM contract Transaction Hash: 0x9ea296aad2d2548f5574a3e9f820bb5026945810e607bd258efd03598505dbfa

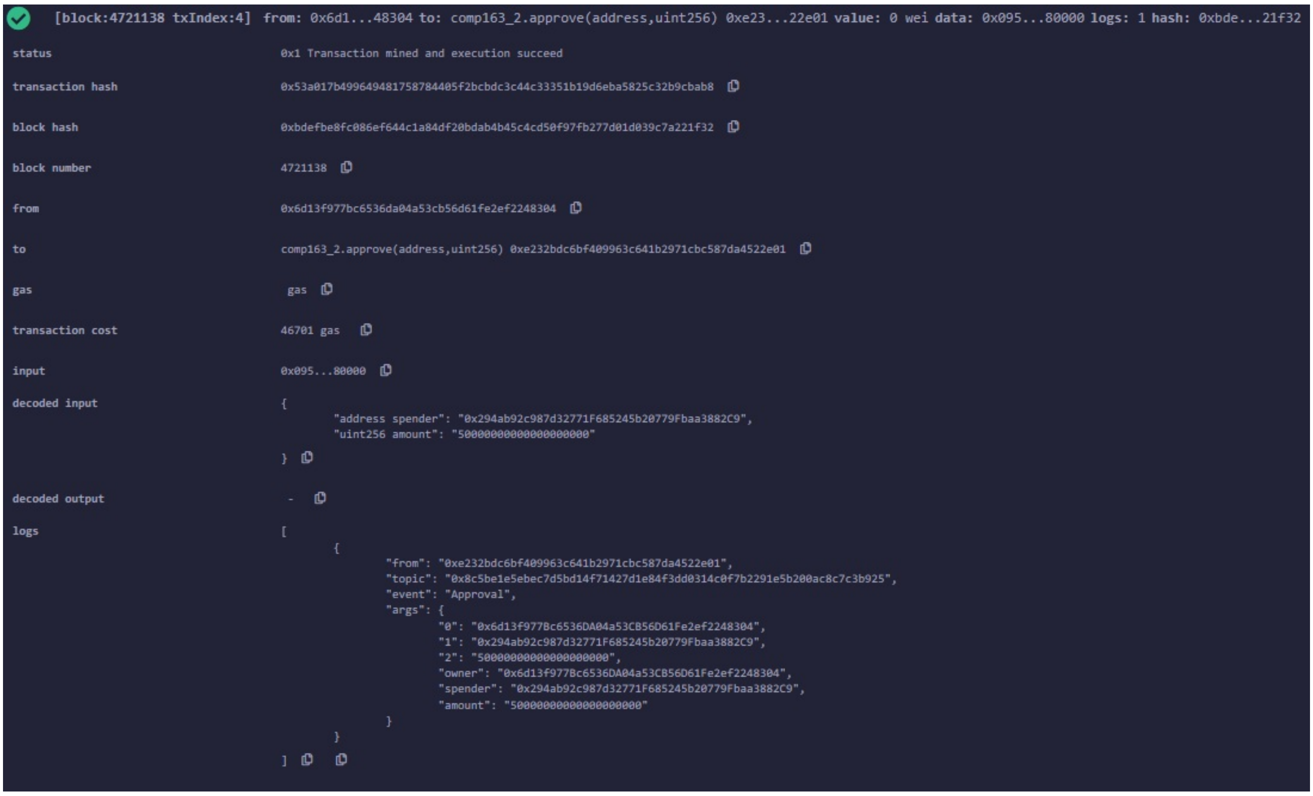
5. Approve 50 comp1 and 50 comp2 to the AMM contract, then add liquidity to the contract. Approve another 10 comp1 to the AMM contract, then swap comp1 for comp2 . Remove the liquidity.

Approve 50 comp1 to AMM transaction overview



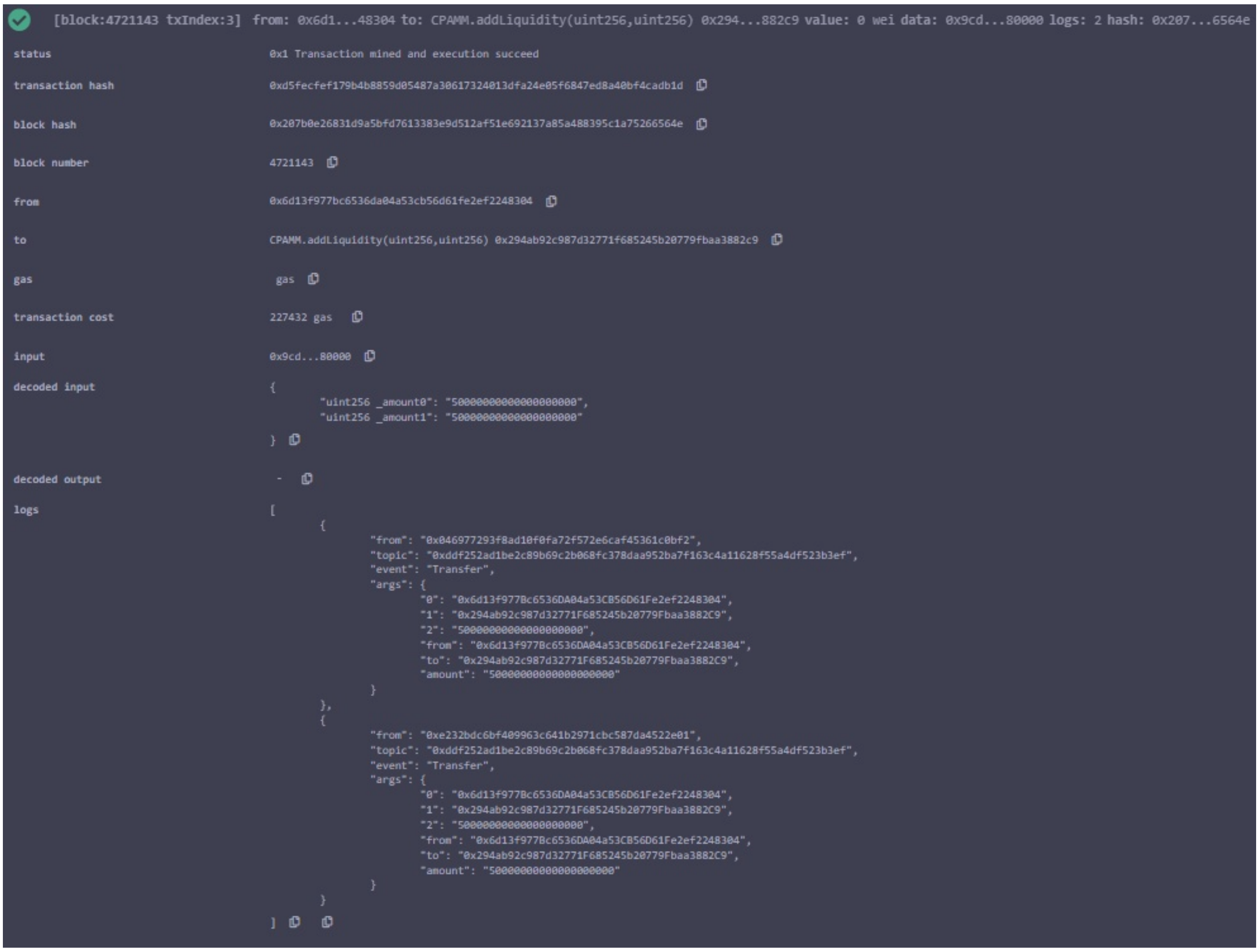
```
[block:4721131 txIndex:7] from: 0x6d1...48304 to: comp163_1.approve(address,uint256) 0x046...c0bf2 value: 0 wei data: 0x095...80000 logs: 1 hash: 0x731...86bc8
status              0x1 Transaction mined and execution succeed
transaction hash    0xbaa226262fbdf53f51d85ffaa79a6b4502750236ecff0d9ec26707f764b06cde
block hash          0x73196f9e9da1eaea6e0fd84d4251a92f1bd2a37cd092ac4e23a18d85c2e86bc8
block number        4721131
from                0x6d13f977bc6536da04a53cb56d61fe2ef2248304
to                  comp163_1.approve(address,uint256) 0x046977293f8ad10f0fa72f572e6caf45361c0bf2
gas                  gas
transaction cost    46701 gas
input               0x095...80000
decoded input       {
                            "address spender": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                            "uint256 amount": "50000000000000000000"
                    }
decoded output       -
logs                [
                            {
                                    "from": "0x046977293f8ad10f0fa72f572e6caf45361c0bf2",
                                    "topic": "0x8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b925",
                                    "event": "Approval",
                                    "args": {
                                            "0": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                            "1": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                            "2": "50000000000000000000",
                                            "owner": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                            "spender": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                            "amount": "50000000000000000000"
                                    }
                            }
                    ]
```

Approve 50 comp2 to AMM transaction overview



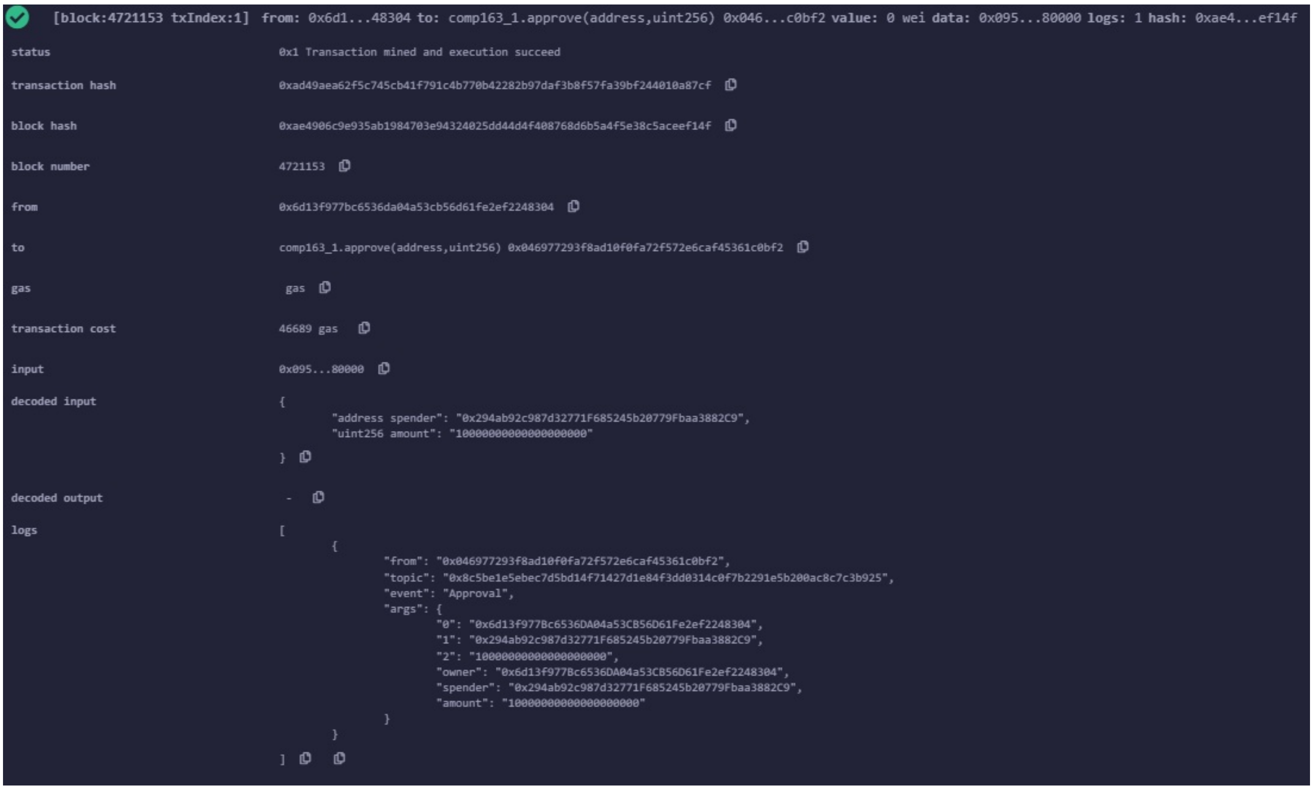
```
[block:4721138 txIndex:4]  from: 0x6d1...48304 to: comp163_2.approve(address,uint256) 0xe23...22e01 value: 0 wei data: 0x095...80000 logs: 1 hash: 0xbde...21f32
status                    0x1 Transaction mined and execution succeed
transaction hash          0x53a017b499649481758784405f2bcbdc3c44c33351b19d6eba5825c32b9cbab8
block hash                0xbdefbe8fc086ef644c1a84df20bdab4b45c4cd50f97fb277d01d039c7a221f32
block number              4721138
from                      0x6d13f977bc6536da04a53cb56d61fe2ef2248304
to                        comp163_2.approve(address,uint256) 0xe232bdc6bf409963c641b2971cbc587da4522e01
gas                        gas
transaction cost          46701 gas
input                     0x095...80000
decoded input             {
                                  "address spender": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                  "uint256 amount": "5000000000000000000"
                          }
decoded output             -
logs                      [
                                  {
                                          "from": "0xe232bdc6bf409963c641b2971cbc587da4522e01",
                                          "topic": "0x8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b925",
                                          "event": "Approval",
                                          "args": {
                                                  "0": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "1": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "2": "5000000000000000000",
                                                  "owner": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "spender": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "amount": "5000000000000000000"
                                          }
                                  }
                          ]
```

Add liquidity transaction overview



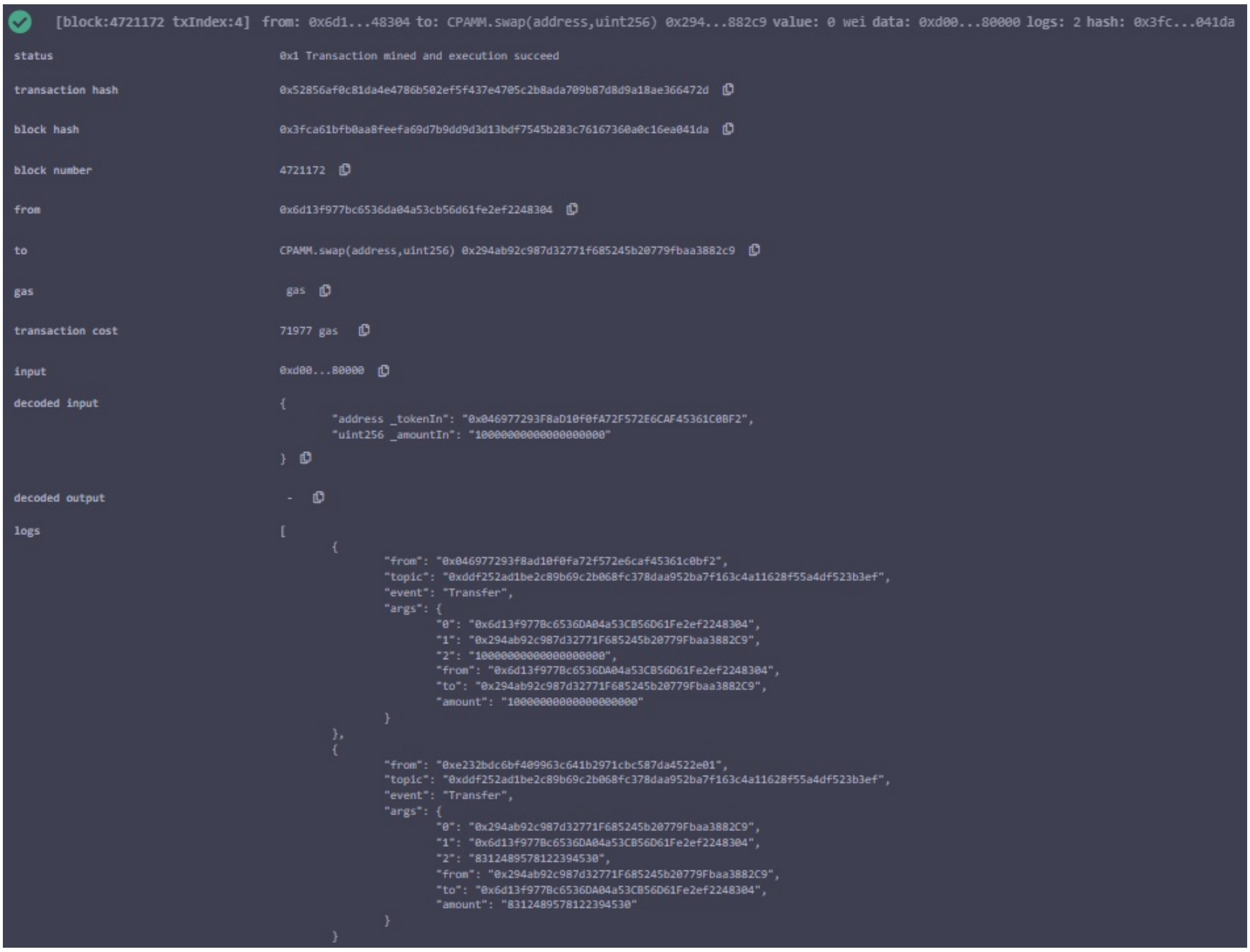
```
[block:4721143 txIndex:3]  from: 0x6d1...48304 to: CPAMM.addLiquidity(uint256,uint256) 0x294...882c9 value: 0 wei data: 0x9cd...80000 logs: 2 hash: 0x207...6564e
status                    0x1 Transaction mined and execution succeed
transaction hash          0xd5fecfef179b4b8859d05487a30617324013dfa24e05f6847ed8a40bf4cadb1d
block hash                0x207b0e26831d9a5bfd7613383e9d512af51e692137a85a488395c1a75266564e
block number              4721143
from                      0x6d13f977bc6536da04a53cb56d61fe2ef2248304
to                        CPAMM.addLiquidity(uint256,uint256) 0x294ab92c987d32771f685245b20779fbaa3882c9
gas                        gas
transaction cost          227432 gas
input                     0x9cd...80000
decoded input             {
                                  "uint256 _amount0": "5000000000000000000",
                                  "uint256 _amount1": "5000000000000000000"
                          }
decoded output             -
logs                      [
                                  {
                                          "from": "0x046977293f8ad10f0fa72f572e6caf45361c0bf2",
                                          "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                          "event": "Transfer",
                                          "args": {
                                                  "0": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "1": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "2": "5000000000000000000",
                                                  "from": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "to": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "amount": "5000000000000000000"
                                          }
                                  },
                                  {
                                          "from": "0xe232bdc6bf409963c641b2971cbc587da4522e01",
                                          "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                          "event": "Transfer",
                                          "args": {
                                                  "0": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "1": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "2": "5000000000000000000",
                                                  "from": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "to": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "amount": "5000000000000000000"
                                          }
                                  }
                          ]
```

Approve another 10 comp1 to AMM transaction overview



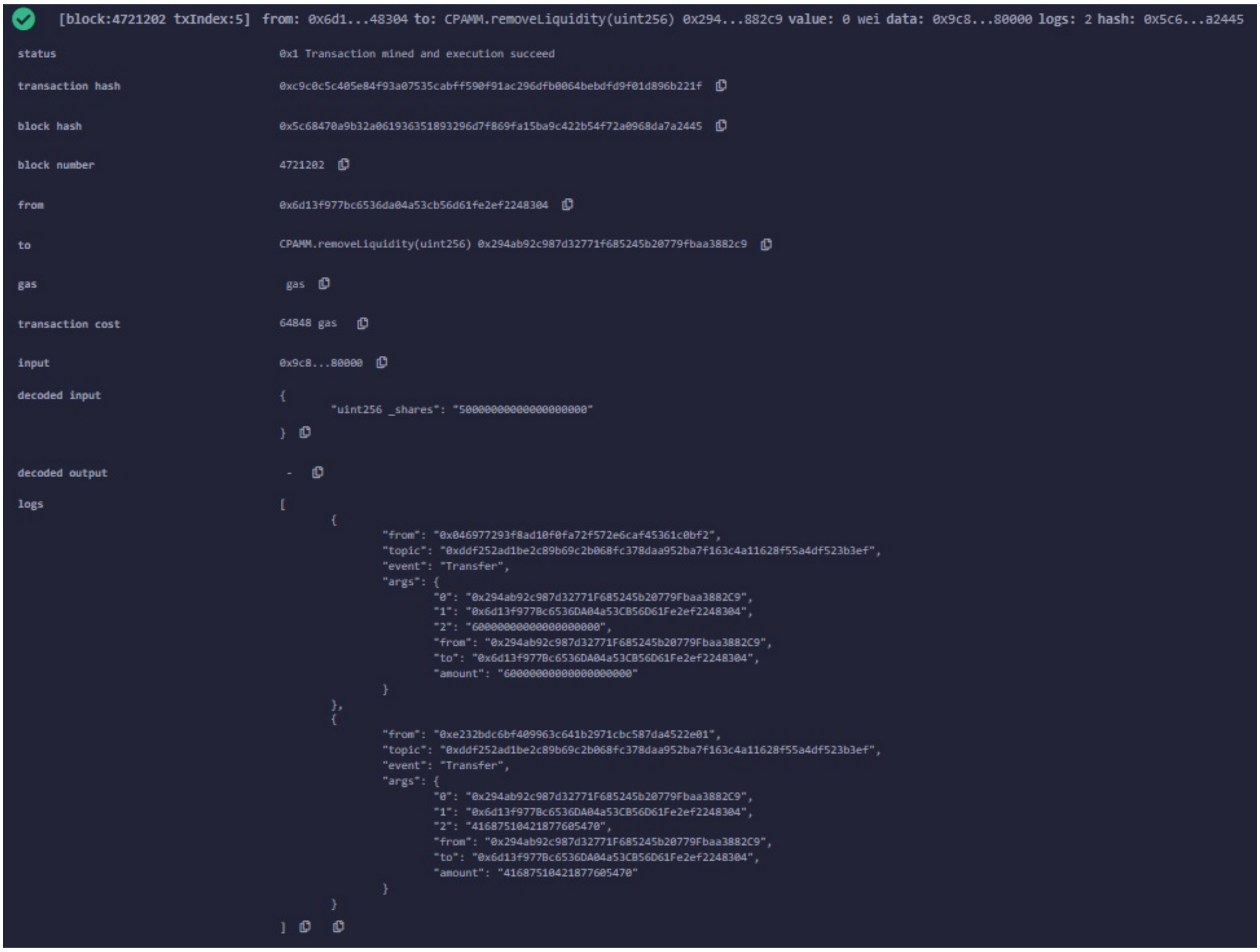
```
[block:4721153 txIndex:1] from: 0x6d1...48304 to: comp163_1.approve(address,uint256) 0x046...c0bf2 value: 0 wei data: 0x095...80000 logs: 1 hash: 0xae4...ef14f
status                  0x1 Transaction mined and execution succeed
transaction hash        0xad49aea62f5c745cb41f791c4b770b42282b97daf3b8f57fa39bf244010a87cf
block hash              0xae4906c9e935ab1984703e94324025dd44d4f408768d6b5a4f5e38c5aceef14f
block number            4721153
from                    0x6d13f977bc6536da04a53cb56d61fe2ef2248304
to                      comp163_1.approve(address,uint256) 0x046977293f8ad10f0fa72f572e6caf45361c0bf2
gas                      gas
transaction cost        46689 gas
input                   0x095...80000
decoded input           {
                                "address spender": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                "uint256 amount": "1000000000000000000"
                        }
decoded output           -
logs                    [
                                {
                                        "from": "0x046977293f8ad10f0fa72f572e6caf45361c0bf2",
                                        "topic": "0x8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b925",
                                        "event": "Approval",
                                        "args": {
                                                "0": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                "1": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                "2": "1000000000000000000",
                                                "owner": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                "spender": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                "amount": "1000000000000000000"
                                        }
                                }
                        ]
```

Swap comp1 for comp2 transaction overview



```
[block:4721172 txIndex:4] from: 0x6d1...48304 to: CPAMM.swap(address,uint256) 0x294...882c9 value: 0 wei data: 0xd00...80000 logs: 2 hash: 0x3fc...041da
status                  0x1 Transaction mined and execution succeed
transaction hash        0x52856af0c81da4e4786b502ef5f437e4705c2b8ada709b87d8d9a18ae366472d
block hash              0x3fca61bfb0aa8feefa69d7b9dd9d3d13bdf7545b283c76167360a0c16ea041da
block number            4721172
from                    0x6d13f977bc6536da04a53cb56d61fe2ef2248304
to                      CPAMM.swap(address,uint256) 0x294ab92c987d32771f685245b20779fbaa3882c9
gas                      gas
transaction cost        71977 gas
input                   0xd00...80000
decoded input           {
                                "address _tokenIn": "0x046977293F8aD10f0fA72F572E6CAF45361C0BF2",
                                "uint256 _amountIn": "1000000000000000000"
                        }
decoded output           -
logs                    [
                                {
                                        "from": "0x046977293f8ad10f0fa72f572e6caf45361c0bf2",
                                        "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                        "event": "Transfer",
                                        "args": {
                                                "0": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                "1": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                "2": "1000000000000000000",
                                                "from": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                "to": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                "amount": "1000000000000000000"
                                        }
                                },
                                {
                                        "from": "0xe232bdc6bf409963c641b2971cbc587da4522e01",
                                        "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                        "event": "Transfer",
                                        "args": {
                                                "0": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                "1": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                "2": "831248957812239453O",
                                                "from": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                "to": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                "amount": "8312489578122394530"
                                        }
                                }
```

Remove liquidity transaction overview



```
[block:4721202 txIndex:5] from: 0x6d1...48304 to: CPAMM.removeLiquidity(uint256) 0x294...882c9 value: 0 wei data: 0x9c8...80000 logs: 2 hash: 0x5c6...a2445
status                    0x1 Transaction mined and execution succeed
transaction hash          0xc9c0c5c405e84f93a07535cabff590f91ac296dfb0064bebdfd9f01d896b221f
block hash                0x5c68470a9b32a061936351893296d7f869fa15ba9c422b54f72a0968da7a2445
block number              4721202
from                      0x6d13f977bc6536da04a53cb56d61fe2ef2248304
to                        CPAMM.removeLiquidity(uint256) 0x294ab92c987d32771f685245b20779fbaa3882c9
gas                        gas
transaction cost          64848 gas
input                     0x9c8...80000
decoded input             {
                                  "uint256 _shares": "50000000000000000000"
                          }
decoded output             -
logs                      [
                                  {
                                          "from": "0x046977293f8ad10f0fa72f572e6caf45361c0bf2",
                                          "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                          "event": "Transfer",
                                          "args": {
                                                  "0": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "1": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "2": "60000000000000000000",
                                                  "from": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "to": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "amount": "60000000000000000000"
                                          }
                                  },
                                  {
                                          "from": "0xe232bdc6bf409963c641b2971cbc587da4522e01",
                                          "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                          "event": "Transfer",
                                          "args": {
                                                  "0": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "1": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "2": "41687510421877605470",
                                                  "from": "0x294ab92c987d32771F685245b20779Fbaa3882C9",
                                                  "to": "0x6d13f977Bc6536DA04a53CB56D61Fe2ef2248304",
                                                  "amount": "41687510421877605470"
                                          }
                                  }
                          ]
```

6. What other traditional financial agent could be replaced by the smart contract?

Credit Scoring Agencies (etc Experian) Traditional credit scoring agencies evaluate an individuals credit against a set criteria not known to the individual, only outputting a score that deems how creditworthy they are. Smart contracts can use a user's transaction history on the blockchain to assess how creditworthy they are. Providing transparency and an objective assessment on creditworthiness, it would also require a lot less information about the individual as traditional agencies will ask for full address history and a lot more data.