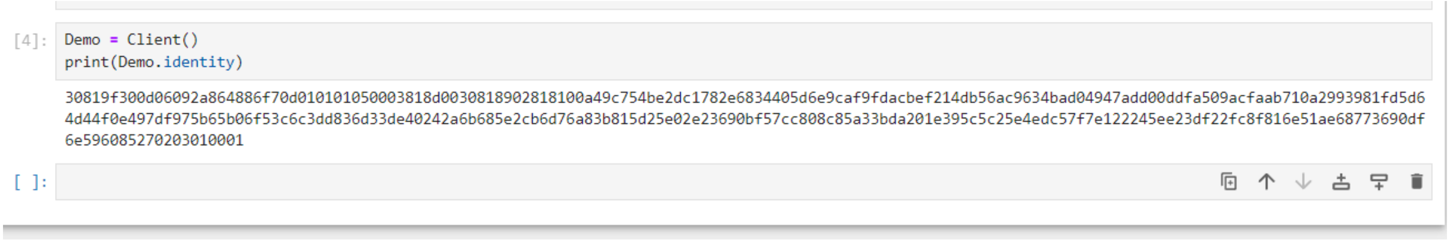
Practical 1

Aim 1a: A simple client class that generates the private and public keys by using the built-in Python RSA algorithm and test it.

Code:

```
!pip install pycryptodome

import Crypto
import binascii
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5


class Client:
    def __init__(self):
        #Creating a random number for key
        random = Crypto.Random.new().read
        #Creating a new public key and private key
        self._private_key = RSA.generate(1024,random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)

    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')


Demo = Client()
print(Demo.identity)
```

Output:



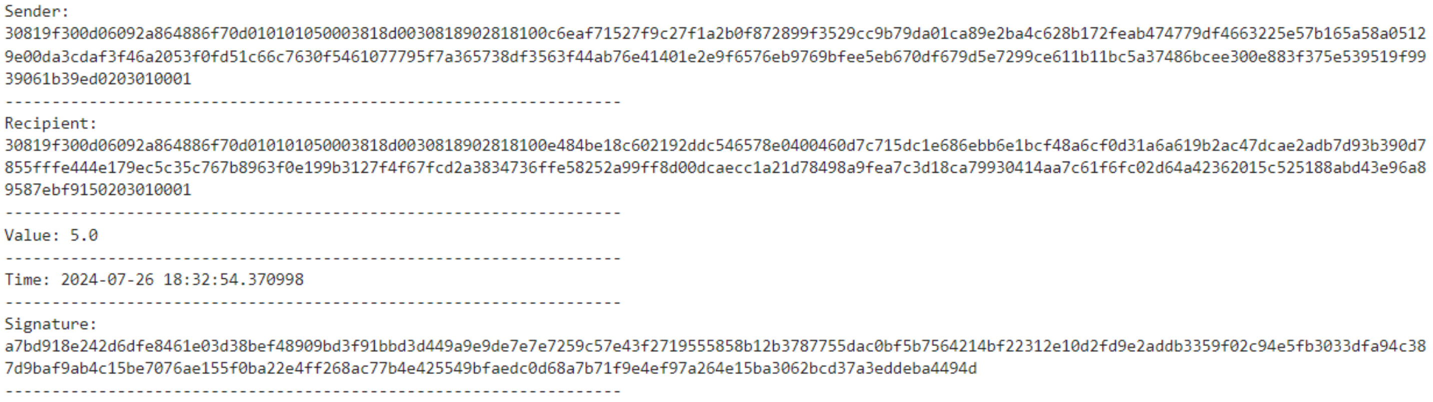
```
[4]: Demo = Client()
     print(Demo.identity)

     30819f300d06092a864886f70d010101050003818d0030818902818100a49c754be2dc1782e6834405d6e9caf9fdacbef214db56ac9634bad04947add00ddfa509acfaab710a2993981fd5d6
     4d44f0e497df975b65b06f53c6c3dd836d33de40242a6b685e2cb6d76a83b815d25e02e23690bf57cc808c85a33bda201e395c5c25e4edc57f7e122245ee23df22fc8f816e51ae68773690df
     6e596085270203010001

[ ]:
```

Aim 1b: A transaction class to send and receive money and test it.

Code:

```
!pip install pycryptodome

import collections
import datetime
import binascii
import Crypto
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5


class Client:
    def __init__(self):
        random = Crypto.Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)

    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')



class Transaction:
    def __init__(self, sender, recipient, value):
        self.sender = sender
        self.recipient = recipient
        self.value = value
        self.time = datetime.datetime.now()

    def to_dict(self):
        if self.sender == "Genesis":
            identity = "Genesis"
        else:
            identity = self.sender.identity
        return collections.OrderedDict({
            'sender': identity,
            'recipient': self.recipient,
            'value': self.value,
            'time' : self.time})

    def sign_transaction(self):
        private_key = self.sender._private_key
        signer = PKCS1_v1_5.new(private_key)
        h = SHA.new(str(self.to_dict()).encode('utf8'))
        return binascii.hexlify(signer.sign(h)).decode('ascii')



def display_transaction(transaction):
    dict = transaction.to_dict()
    print ("Sender: \n" + dict['sender'])
    print ('----------------------------------------------------------------')
    print ("Recipient: \n" + dict['recipient'])
    print ('----------------------------------------------------------------')
    print ("Value: " + str(dict['value']))
    print ('----------------------------------------------------------------')
    print ("Time: " + str(dict['time']))
    print ('----------------------------------------------------------------')
    print ("Signature: \n" + signature)
    print ('----------------------------------------------------------------')



Shlok = Client()
Jivesh = Client()

signature = Transaction(Shlok, Jivesh.identity, 5.0).sign_transaction()

display_transaction(Transaction(Shlok, Jivesh.identity, 5.0))
```

Output:



```
Sender:
30819f300d06092a864886f70d010101050003818d0030818902818100c6eaf71527f9c27f1a2b0f872899f3529cc9b79da01ca89e2ba4c628b172feab474779df4663225e57b165a58a0512
9e00da3cdaf3f46a2053f0fd51c66c7630f5461077795f7a365738df3563f44ab76e41401e2e9f6576eb9769bfee5eb670df679d5e7299ce611b11bc5a37486bcee300e883f375e539519f99
39061b39ed0203010001
----------------------------------------------------------------
Recipient:
30819f300d06092a864886f70d010101050003818d0030818902818100e484be18c602192ddc546578e0400460d7c715dc1e686ebb6e1bcf48a6cf0d31a6a619b2ac47dcae2adb7d93b390d7
855fffe444e179ec5c35c767b8963f0e199b3127f4f67fcd2a3834736ffe58252a99ff8d00dcaecc1a21d78498a9fea7c3d18ca79930414aa7c61f6fc02d64a42362015c525188abd43e96a8
9587ebf9150203010001
----------------------------------------------------------------
Value: 5.0
----------------------------------------------------------------
Time: 2024-07-26 18:32:54.370998
----------------------------------------------------------------
Signature:
a7bd918e242d6dfe8461e03d38bef48909bd3f91bbd3d449a9e9de7e7e7259c57e43f2719555858b12b3787755dac0bf5b7564214bf22312e10d2fd9e2addb3359f02c94e5fb3033dfa94c38
7d9baf9ab4c15be7076ae155f0ba22e4ff268ac77b4e425549bfaedc0d68a7b71f9e4ef97a264e15ba3062bcd37a3eddeba4494d
----------------------------------------------------------------
```

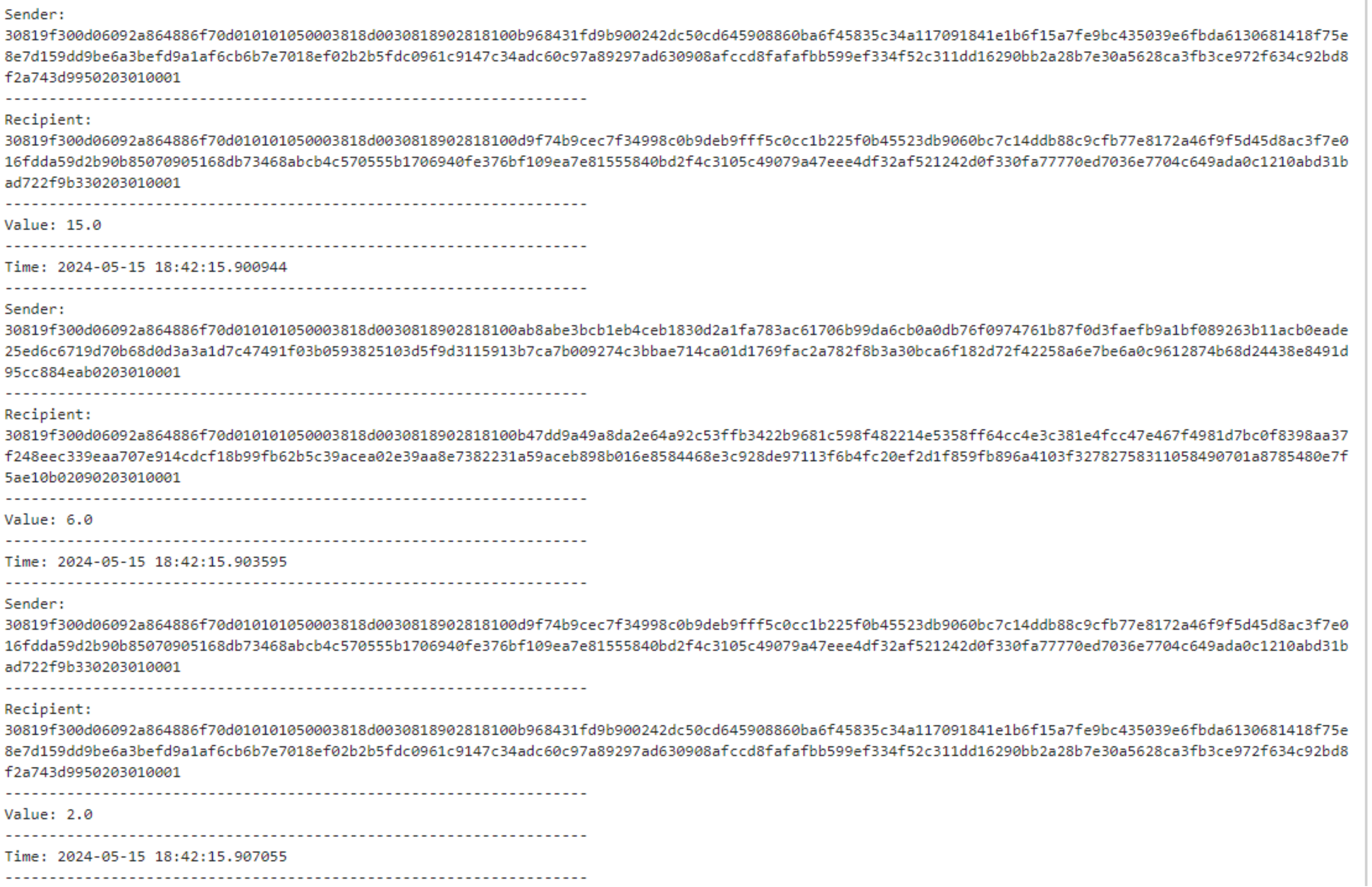
Aim 1c: Create multiple transactions and display them.

Code:

```
!pip install pycryptodome

import collections
import datetime
import binascii
import Crypto
import hashlib
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5



class Client:
    def __init__(self):
        random = Crypto.Random.new().read
        self._private_key = RSA.generate(1024, random) #create private key
        self._public_key = self._private_key.publickey() #create public key
        self._signer = PKCS1_v1_5.new(self._private_key) #create digital signature

    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')



class Transaction: #creating transaction
    def __init__(self, sender, recipient, value): # in python client used to create constructor
        self.sender = sender
        self.recipient = recipient
        self.value = value
        self.time = datetime.datetime.now()
    def to_dict(self): #record identity
        if self.sender == "Genesis": #base block in blockchain
            identity = "Genesis"
        else:
            identity = self.sender.identity
            return collections.OrderedDict({ # inserting in oredered manner \ storing | nothing but
an ordered dictionary
            'sender': identity,
            'recipient': self.recipient,
            'value': self.value,
            'time' : self.time})

    def sign_transaction(self): # verify sender and converting into hash value
```

```
        private_key = self.sender._private_key
        signer = PKCS1_v1_5.new(private_key)
        h = SHA.new(str(self.to_dict()).encode('utf8'))
        return binascii.hexlify(signer.sign(h)).decode('ascii')


def display_transaction(transaction):
    dict = transaction.to_dict()
    print ("Sender: \n" + dict['sender'])
    print ('----------------------------------------------------------------------')
    print ("Recipient: \n" + dict['recipient'])
    print ('----------------------------------------------------------------------')
    print ("Value: " + str(dict['value']))
    print ('----------------------------------------------------------------------')
    print ("Time: " + str(dict['time']))
    print ('----------------------------------------------------------------------')


transactions = []

Shlok = Client()
Jivesh = Client()
Shreyas = Client()
Himanshu = Client()


t1 = Transaction(Shlok, Jivesh.identity, 15.0)
t1.sign_transaction()
transactions.append(t1)

t2 = Transaction(Shreyas, Himanshu.identity,6.0)
t2.sign_transaction()
transactions.append(t2)

t3 = Transaction(Jivesh, Shlok.identity,2.0)
t3.sign_transaction()
transactions.append(t3)


for txn in transactions:
    display_transaction (txn)
```

Output:



```
Sender:
30819f300d06092a864886f70d010101050003818d0030818902818100b968431fd9b900242dc50cd645908860ba6f45835c34a117091841e1b6f15a7fe9bc435039e6fbda6130681418f75e
8e7d159dd9be6a3befd9a1af6cb6b7e7018ef02b2b5fdc0961c9147c34adc60c97a89297ad630908afccd8fafafbb599ef334f52c311dd16290bb2a28b7e30a5628ca3fb3ce972f634c92bd8
f2a743d9950203010001
------------------------------------------------------------------
Recipient:
30819f300d06092a864886f70d010101050003818d0030818902818100d9f74b9cec7f34998c0b9deb9fff5c0cc1b225f0b45523db9060bc7c14ddb88c9cfb77e8172a46f9f5d45d8ac3f7e0
16fdda59d2b90b85070905168db73468abcb4c570555b1706940fe376bf109ea7e81555840bd2f4c3105c49079a47eee4df32af521242d0f330fa77770ed7036e7704c649ada0c1210abd31b
ad722f9b330203010001
------------------------------------------------------------------
Value: 15.0
------------------------------------------------------------------
Time: 2024-05-15 18:42:15.900944
------------------------------------------------------------------
Sender:
30819f300d06092a864886f70d010101050003818d0030818902818100ab8abe3bcb1eb4ceb1830d2a1fa783ac61706b99da6cb0a0db76f0974761b87f0d3faefb9a1bf089263b11acb0eade
25ed6c6719d70b68d0d3a3a1d7c47491f03b0593825103d5f9d3115913b7ca7b009274c3bbae714ca01d1769fac2a782f8b3a30bca6f182d72f42258a6e7be6a0c9612874b68d24438e8491d
95cc884eab0203010001
------------------------------------------------------------------
Recipient:
30819f300d06092a864886f70d010101050003818d0030818902818100b47dd9a49a8da2e64a92c53ffb3422b9681c598f482214e5358ff64cc4e3c381e4fcc47e467f4981d7bc0f8398aa37
f248eec339eaa707e914cdcf18b99fb62b5c39acea02e39aa8e7382231a59aceb898b016e8584468e3c928de97113f6b4fc20ef2d1f859fb896a4103f327827583110584907 01a8785480e7f
5ae10b02090203010001
------------------------------------------------------------------
Value: 6.0
------------------------------------------------------------------
Time: 2024-05-15 18:42:15.903595
------------------------------------------------------------------
Sender:
30819f300d06092a864886f70d010101050003818d0030818902818100d9f74b9cec7f34998c0b9deb9fff5c0cc1b225f0b45523db9060bc7c14ddb88c9cfb77e8172a46f9f5d45d8ac3f7e0
16fdda59d2b90b85070905168db73468abcb4c570555b1706940fe376bf109ea7e81555840bd2f4c3105c49079a47eee4df32af521242d0f330fa77770ed7036e7704c649ada0c1210abd31b
ad722f9b330203010001
------------------------------------------------------------------
Recipient:
30819f300d06092a864886f70d010101050003818d0030818902818100b968431fd9b900242dc50cd645908860ba6f45835c34a117091841e1b6f15a7fe9bc435039e6fbda6130681418f75e
8e7d159dd9be6a3befd9a1af6cb6b7e7018ef02b2b5fdc0961c9147c34adc60c97a89297ad630908afccd8fafafbb599ef334f52c311dd16290bb2a28b7e30a5628ca3fb3ce972f634c92bd8
f2a743d9950203010001
------------------------------------------------------------------
Value: 2.0
------------------------------------------------------------------
Time: 2024-05-15 18:42:15.907055
------------------------------------------------------------------
```

Aim 1d: Create a blockchain, a genesis block and execute it.

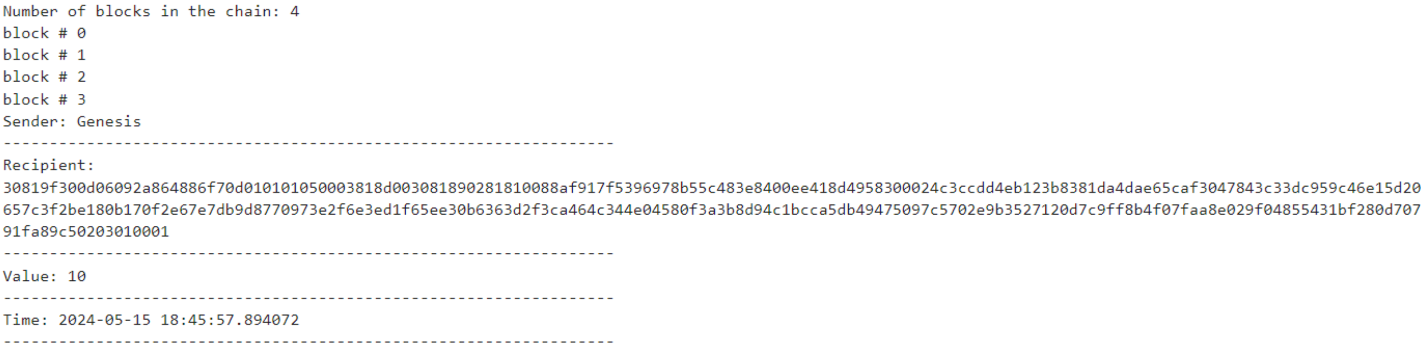
Code:

```
!pip install pycryptodome


import collections
import datetime
import binascii
import Crypto
import hashlib
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5


class Client:
  def __init__(self):
    random = Crypto.Random.new().read
    self._private_key = RSA.generate(1024, random)
    self._public_key = self._private_key.publickey()
    self._signer = PKCS1_v1_5.new(self._private_key)
  @property
  def identity(self):
    return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')



class Transaction:
  def __init__(self, sender, recipient, value):
    self.sender = sender
    self.recipient = recipient
    self.value = value
    self.time = datetime.datetime.now()

  def to_dict(self):
    if self.sender == "Genesis":
      identity = "Genesis"
    else:
      identity = self.sender.identity
    return collections.OrderedDict({
      'sender': identity,
      'recipient': self.recipient,
      'value': self.value,
      'time' : self.time})

  def sign_transaction(self):
    private_key = self.sender._private_key
```

```
        signer = PKCS1_v1_5.new(private_key)
        h = SHA.new(str(self.to_dict()).encode('utf8'))
        return binascii.hexlify(signer.sign(h)).decode('ascii')


class Block:
    def __init__(self):
        self.verified_transactions = []
        self.previous_block_hash = ""
        #self.Nonce = ""
    last_block_hash = ""


def blockchain (self):
        print ("Number of blocks in the chain: " + str(len (self)))
        for x in range (len(SampleCoins)):
            block_temp = SampleCoins[x]
            print ("block # " + str(x))
        for transaction in block_temp.verified_transactions:
            display_transaction (transaction)

def display_transaction(transaction):
    dict = transaction.to_dict()
    print ("Sender: " + dict['sender'])
    print ('-----------------------------------------------------------------')
    print ("Recipient: \n" + dict['recipient'])
    print ('-----------------------------------------------------------------')
    print ("Value: " + str(dict['value']))
    print ('-----------------------------------------------------------------')
    print ("Time: " + str(dict['time']))
    print ('-----------------------------------------------------------------')


SampleCoins = []

Shlok = Client()
Jivesh = Client()


txn0=Transaction("Genesis",Shlok.identity,10)

block0=Block()
block0.previous_block_hash = None
#Nonce = None
block0.verified_transactions.append(txn0)

last_block_hash = hash(block0)
```

```
SampleCoins.append(block0)
blockchain(SampleCoins)
```

Output:



```
Number of blocks in the chain: 4
block # 0
block # 1
block # 2
block # 3
Sender: Genesis
------------------------------------------------------------------
Recipient:
30819f300d06092a864886f70d010101050003818d0030818902818100 88af917f5396978b55c483e8400ee418d4958300024c3ccdd4eb123b8381da4dae65caf3047843c33dc959c46e15d20
657c3f2be180b170f2e67e7db9d8770973e2f6e3ed1f65ee30b6363d2f3ca464c344e04580f3a3b8d94c1bcca5db49475097c5702e9b3527120d7c9ff8b4f07faa8e029f04855431bf280d707
91fa89c50203010001
------------------------------------------------------------------
Value: 10
------------------------------------------------------------------
Time: 2024-05-15 18:45:57.894072
------------------------------------------------------------------
```

Aim 1e: Create a mining function and test it also add blocks to the miner and dump the blockchain.

Code:

```
import collections
import datetime
import binascii
!pip install pycryptodome
import Crypto
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5


class Client:
  def __init__(self):
    random=Crypto.Random.new().read
    self._private_key=RSA.generate(1024,random)
    self._public_key=self._private_key.publickey()
    self._signer=PKCS1_v1_5.new(self._private_key)
  @property
  def identity(self):
    return binascii.hexlify(self._public_key.exportKey(format='DER')).decode('ascii')


class Transaction:
  def __init__(self,sender,recipient,value):
    self.sender=sender
    self.recipient=recipient
    self.value=value
    self.time=datetime.datetime.now()

  def to_dict(self):
    if self.sender=="Genesis":
      identity="Genesis"
    else:
      identity=self.sender.identity

    return collections.OrderedDict({
      'sender':identity,
      'recipient':self.recipient,
      'value':self.value,
      'time':self.time})
  def sign_transaction(self):
    private_key=self.sender._private_key
    signer=PKCS1_v1_5.new(private_key)
    h=SHA.new(str(self.to_dict()).encode('utf8'))
```

```
        return binascii.hexlify(signer.sign(h)).decode('ascii')



import hashlib
def sha256(message):
    return hashlib.sha256(message.encode('ascii')).hexdigest()
def mine(message,difficulty=1):
    assert difficulty>=1
    prefix='1'*difficulty
    for i in range(1000):
        digest=sha256(str(hash(message))+str(i))
        if digest.startswith(prefix):
            print("after"+str(i)+"iterationsfoundnonce:"+digest)
            return digest


class Block:
    def __init__(self):
        self.verified_transactions=[]
        self.previous_block_hash=""
        self.Nonce=""


def display_transaction(transaction):

    dict=transaction.to_dict()
    print("sender : "+dict['sender'])
    print('-----')
    print("recipient : "+dict['recipient'])
    print('-----')
    print("value : "+str(dict['value']))
    print('-----')
    print("time : "+str(dict['time']))
    print('-----')


def dump_blockchain(self):
    print("Number of blocks in the chain :"+str(len(self)))
    for x in range(len(TPCoins)):
        block_temp=TPCoins[x]
        print("Block # "+str(x))
        for transaction in block_temp.verified_transactions:
            display_transaction(transaction)
            print('--------------')
            print('=====================================')


last_block_hash=""
TPCoins=[]
```

```
last_transaction_index=0
transactions=[]


Raja=Client()
Rani=Client()
Seema=Client()
Reema=Client()


t1=Transaction(Raja,Rani.identity,15.0)
t1.sign_transaction()
transactions.append(t1)

t2=Transaction(Raja,Seema.identity,6.0)
t2.sign_transaction()
transactions.append(t2)

t3=Transaction(Rani,Reema.identity,2.0)
t3.sign_transaction()
transactions.append(t3)

t4=Transaction(Seema,Rani.identity,4.0)
t4.sign_transaction()
transactions.append(t4)

t5=Transaction(Reema,Seema.identity,7.0)
t5.sign_transaction()
transactions.append(t5)

t6=Transaction(Rani,Seema.identity,3.0)
t6.sign_transaction()
transactions.append(t6)

t7=Transaction(Seema,Raja.identity,8.0)
t7.sign_transaction()
transactions.append(t7)

t8=Transaction(Seema,Rani.identity,1.0)
t8.sign_transaction()
transactions.append(t8)

t9=Transaction(Reema,Raja.identity,5.0)
t9.sign_transaction()
transactions.append(t9)

t10=Transaction(Reema,Rani.identity,3.0)
t10.sign_transaction()
transactions.append(t10)
```

```
#Miner1addsablock

block=Block()
for i in range(3):
    temp_transaction=transactions[last_transaction_index]
    #validatetransaction
    #if valid
    block.verified_transactions.append(temp_transaction)
    last_transaction_index+=1

block.previous_block_hash=last_block_hash
block.Nonce=mine(block,2)
digest=hash(block)
TPCoins.append(block)
last_block_hash=digest



#Miner2 adds a block

block=Block()
for i in range(3):
    temp_transaction=transactions[last_transaction_index]
    #validate transaction
    #if valid
    block.verified_transactions.append(temp_transaction)
    last_transaction_index+=1

block.previous_block_hash=last_block_hash
block.Nonce=mine(block,2)
digest=hash(block)
TPCoins.append(block)
last_block_hash=digest



#Miner3 adds a block

block=Block()
for i in range(3):
    temp_transaction=transactions[last_transaction_index]
    #validate transaction
    #if valid
    block.verified_transactions.append(temp_transaction)
    last_transaction_index+=1

block.previous_block_hash=last_block_hash
block.Nonce=mine(block,2)
digest=hash(block)
```

```
TPCoins.append(block)
last_block_hash=digest


dump_blockchain(TPCoins)
```

Output:

```
Number of blocks in the chain :3
Block # 0
sender :
30819f300d06092a864886f70d010101050003818d0030818902818100bb0d583a631d3afd2448d14dcc8b98e10420b08ce68e7b0e821cf6fb313b28fdf4f4e7b
41875f1f54944330b4f623c72f4684683f3a35298ff380c7a8b1662c13f2e3acdfc58bfef36e0952656d994fe2eb65b53fa0805e06f9cc9e354a1b3e8308559065c9
3381771fd6c80655f1c6d1f91db07f48787d54ca85ea65b4da0910203010001
-----
recipient :
30819f300d06092a864886f70d010101050003818d0030818902818100b483c3a8caacd702be7df4f82b87aeae3cecccd9caef9a69270fc7386b0c194675d4c755
82474bc006e1b73211434a3e3683a9f0d64da6f75c360b581af02bd2559c49715ffe87611e0a10f58d5c24f7ec6894eaccaac98a9d041a3f529585126a074eb39b0
cf65d34317c4806d8708b9c604c029a2be7720d811ada8e2ef0270203010001
-----
value : 15.0
-----
time : 2024-07-26 18:39:29.864381
-----
--------------
=====================================
sender :
30819f300d06092a864886f70d010101050003818d0030818902818100bb0d583a631d3afd2448d14dcc8b98e10420b08ce68e7b0e821cf6fb313b28fdf4f4e7b
41875f1f54944330b4f623c72f4684683f3a35298ff380c7a8b1662c13f2e3acdfc58bfef36e0952656d994fe2eb65b53fa0805e06f9cc9e354a1b3e8308559065c9
3381771fd6c80655f1c6d1f91db07f48787d54ca85ea65b4da0910203010001
-----
recipient :
30819f300d06092a864886f70d010101050003818d0030818902818100e0d2cf8c997cff17fa7cf6cd5396d6457a2b8a1dbc5e29a7e190aa0d0812b8dcf1d50877
871a8d3ef8a5740dd662decbfd64b7f90764e4b715eeb2cfe9a3a09f04ae0f80f91f7426db38083e4bef84698158a608762efd12d40ee61b53c8faaf9e042b67e719
fc671a333e5eafd9afaad0ac1e2437beb6a28c203099dd197a570203010001
-----
value : 6.0
-----
time : 2024-07-26 18:39:29.867377
-----
--------------
=====================================
sender :
30819f300d06092a864886f70d010101050003818d0030818902818100b483c3a8caacd702be7df4f82b87aeae3cecccd9caef9a69270fc7386b0c194675d4c755
82474bc006e1b73211434a3e3683a9f0d64da6f75c360b581af02bd2559c49715ffe87611e0a10f58d5c24f7ec6894eaccaac98a9d041a3f529585126a074eb39b0
cf65d34317c4806d8708b9c604c029a2be7720d811ada8e2ef0270203010001
-----
recipient :
30819f300d06092a864886f70d010101050003818d0030818902818100b1bcd63995fbcd7e1d0793dee29cfd92a5cff86a7a64f2c864e78b1853942ce47276d1d
dcc23c3435c1f1043ddeda52388703706af4bb71d1a0a4f9f7041e029e408d6f04f960b915a1776588c4d814334e3cca82938b28e7f7784ce183d019438e9524e1
2447f26f2d3655b37772cb8c7d5e5a1eddf70bbb1ef6d766e4842bf0203010001
-----
value : 2.0
-----
time : 2024-07-26 18:39:29.867377
-----
```

```
========================================
Block # 1
sender :
30819f300d06092a864886f70d010101050003818d0030818902818100e0d2cf8c997cff17fa7cf6cd5396d6457a2b8a1dbc5e29a7e190aa0d0812b8dcf1d50
877871a8d3ef8a5740dd662decbfd64b7f90764e4b715eeb2cfe9a3a09f04ae0f80f91f7426db38083e4bef84698158a608762efd12d40ee61b53c8faaf9e042b
67e719fc671a333e5eafd9afaad0ac1e2437beb6a28c203099dd197a570203010001
-----
recipient :
30819f300d06092a864886f70d010101050003818d0030818902818100b483c3a8caacd702be7df4f82b87aeae3cecccd9caef9a69270fc7386b0c194675d4c
75582474bc006e1b73211434a3e3683a9f0d64da6f75c360b581af02bd2559c49715ffe87611e0a10f58d5c24f7ec6894eaccaac98a9d041a3f529585126a074
eb39b0cf65d34317c4806d8708b9c604c029a2be7720d811ada8e2ef0270203010001
-----
value : 4.0
-----
time : 2024-07-26 18:39:29.867377
-----
--------------
========================================
sender :
30819f300d06092a864886f70d010101050003818d0030818902818100b1bcd63995fbcd7e1d0793dee29cfd92a5cff86a7a64f2c864e78b1853942ce47276d
1ddcc23c3435c1f1043ddeda52388703706af4bb71d1a0a4f9f7041e029e408d6f04f960b915a1776588c4d814334e3cca82938b28e7f7784ce183d019438e9
524e12447f26f2d3655b37772cb8c7d5e5a1eddf70bbb1ef6d766e4842bf0203010001
-----
recipient :
30819f300d06092a864886f70d010101050003818d0030818902818100e0d2cf8c997cff17fa7cf6cd5396d6457a2b8a1dbc5e29a7e190aa0d0812b8dcf1d50
877871a8d3ef8a5740dd662decbfd64b7f90764e4b715eeb2cfe9a3a09f04ae0f80f91f7426db38083e4bef84698158a608762efd12d40ee61b53c8faaf9e042b
67e719fc671a333e5eafd9afaad0ac1e2437beb6a28c203099dd197a570203010001
-----
value : 7.0
-----
time : 2024-07-26 18:39:29.870615
-----
--------------
========================================
sender :
30819f300d06092a864886f70d010101050003818d0030818902818100b483c3a8caacd702be7df4f82b87aeae3cecccd9caef9a69270fc7386b0c194675d4c
75582474bc006e1b73211434a3e3683a9f0d64da6f75c360b581af02bd2559c49715ffe87611e0a10f58d5c24f7ec6894eaccaac98a9d041a3f529585126a074
eb39b0cf65d34317c4806d8708b9c604c029a2be7720d811ada8e2ef0270203010001
-----
recipient :
30819f300d06092a864886f70d010101050003818d0030818902818100e0d2cf8c997cff17fa7cf6cd5396d6457a2b8a1dbc5e29a7e190aa0d0812b8dcf1d50
877871a8d3ef8a5740dd662decbfd64b7f90764e4b715eeb2cfe9a3a09f04ae0f80f91f7426db38083e4bef84698158a608762efd12d40ee61b53c8faaf9e042b
67e719fc671a333e5eafd9afaad0ac1e2437beb6a28c203099dd197a570203010001
-----
value : 3.0
-----
time : 2024-07-26 18:39:29.872114
-----
--------------
```

```
=====================================
Block # 2
sender :
30819f300d06092a864886f70d010101050003818d0030818902818100e0d2cf8c997cff17fa7cf6cd5396d6457a2b8a1dbc5e29a7e190aa0d0812b8dcf1d
50877871a8d3ef8a5740dd662decbfd64b7f90764e4b715eeb2cfe9a3a09f04ae0f80f91f7426db38083e4bef84698158a608762efd12d40ee61b53c8faaf9e
042b67e719fc671a333e5eafd9afaad0ac1e2437beb6a28c203099dd197a570203010001
-----
recipient :
30819f300d06092a864886f70d010101050003818d0030818902818100bb0d583a631d3afd2448d14dcc8b98e10420b08ce68e7b0e821cf6fb313b28fdf4
f4e7b41875f1f54944330b4f623c72f4684683f3a35298ff380c7a8b1662c13f2e3acdfc58bfef36e0952656d994fe2eb65b53fa0805e06f9cc9e354a1b3e830
8559065c93381771fd6c80655f1c6d1f91db07f48787d54ca85ea65b4da0910203010001
-----
value : 8.0
-----
time : 2024-07-26 18:39:29.873138
-----
--------------
=====================================
sender :
30819f300d06092a864886f70d010101050003818d0030818902818100e0d2cf8c997cff17fa7cf6cd5396d6457a2b8a1dbc5e29a7e190aa0d0812b8dcf1d
50877871a8d3ef8a5740dd662decbfd64b7f90764e4b715eeb2cfe9a3a09f04ae0f80f91f7426db38083e4bef84698158a608762efd12d40ee61b53c8faaf9e
042b67e719fc671a333e5eafd9afaad0ac1e2437beb6a28c203099dd197a570203010001
-----
recipient :
30819f300d06092a864886f70d010101050003818d0030818902818100b483c3a8caacd702be7df4f82b87aeae3cecccd9caef9a69270fc7386b0c194675d
4c75582474bc006e1b73211434a3e3683a9f0d64da6f75c360b581af02bd2559c49715ffe87611e0a10f58d5c24f7ec6894eaccaac98a9d041a3f529585126
a074eb39b0cf65d34317c4806d8708b9c604c029a2be7720d811ada8e2ef0270203010001
-----
value : 1.0
-----
time : 2024-07-26 18:39:29.873138
-----
--------------
=====================================
sender :
30819f300d06092a864886f70d010101050003818d0030818902818100b1bcd63995fbcd7e1d0793dee29cfd92a5cff86a7a64f2c864e78b1853942ce472
76d1ddcc23c3435c1f1043ddeda52388703706af4bb71d1a0a4f9f7041e029e408d6f04f960b915a1776588c4d814334e3cca82938b28e7f7784ce183d019
438e9524e12447f26f2d3655b37772cb8c7d5e5a1eddf70bbb1ef6d766e4842bf0203010001
-----
recipient :
30819f300d06092a864886f70d010101050003818d0030818902818100bb0d583a631d3afd2448d14dcc8b98e10420b08ce68e7b0e821cf6fb313b28fdf4
f4e7b41875f1f54944330b4f623c72f4684683f3a35298ff380c7a8b1662c13f2e3acdfc58bfef36e0952656d994fe2eb65b53fa0805e06f9cc9e354a1b3e830
8559065c93381771fd6c80655f1c6d1f91db07f48787d54ca85ea65b4da0910203010001
-----
value : 5.0
-----
time : 2024-07-26 18:39:29.875404
-----
--------------
=====================================
```

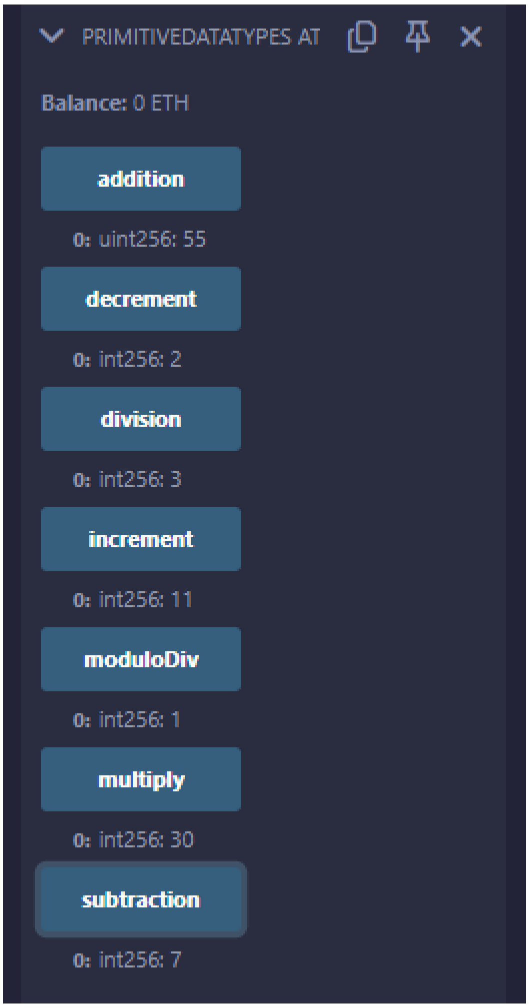
Practical 2

Aim 2a: Implement and demonstrate the use of Variables and Operators in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract PrimitiveDataTypes {

   //state variables (global variable)
   uint8   a = 20;
   uint256 b = 35;
   int     c = 10;
   int8    d = 3;

   bool    flag = true;
   address addr = 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c;

   // Operations in solidity
   uint public addition    = a + b;
   int  public subtraction = c - d;
   int  public multiply    = d * c;
   int  public division    = c / d;
   int  public moduloDiv   = c % d;
   int  public increment   = ++c;
   int  public decrement   = --d;

}
```

Output:



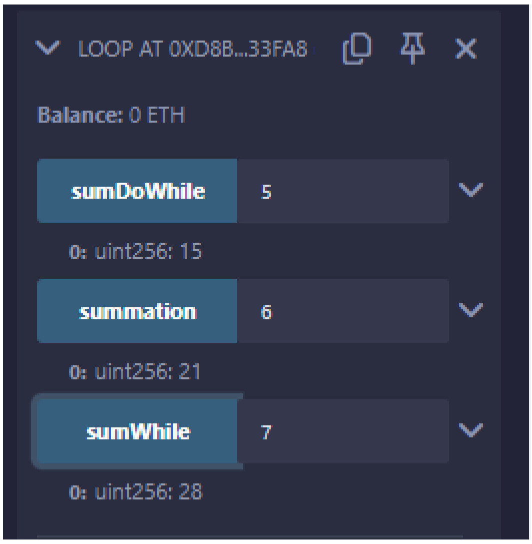
```
Balance: 0 ETH
addition
0: uint256: 55
decrement
0: int256: 2
division
0: int256: 3
increment
0: int256: 11
moduloDiv
0: int256: 1
multiply
0: int256: 30
subtraction
0: int256: 7
```

Aim 2b: Implement and demonstrate the use of Loops in Solidity:

Code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Loop {

   function summation(uint n) public pure returns (uint) {
      uint sum = 0;
      for (uint i = 1; i <= n; i++) {
         sum += i;
      }
      return sum;
   }

   function sumWhile(uint n) public pure returns (uint) {
      uint sum = 0;
      uint i = 1;
      while (i <= n) {
         sum += i;
         i++;
      }
      return sum;
   }

   function sumDoWhile(uint n) public pure returns (uint) {
      uint sum = 0;
      uint i = 1;
      do {
         sum += i;
         i++;
      } while (i <= n);
      return sum;
   }

}
```

Output:



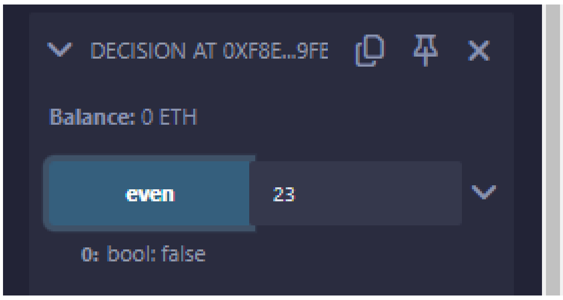
```
sumDoWhile 5
0: uint256: 15
summation 6
0: uint256: 21
sumWhile 7
0: uint256: 28
```

Aim 2c: Implement and demonstrate the use of Decision Making in Solidity:

Code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract decision{

    function even(uint n) public pure returns(bool){
        if(n%2==0){
            return true;
        }
        else{
            return false;
        }
    }
}
```

Output:



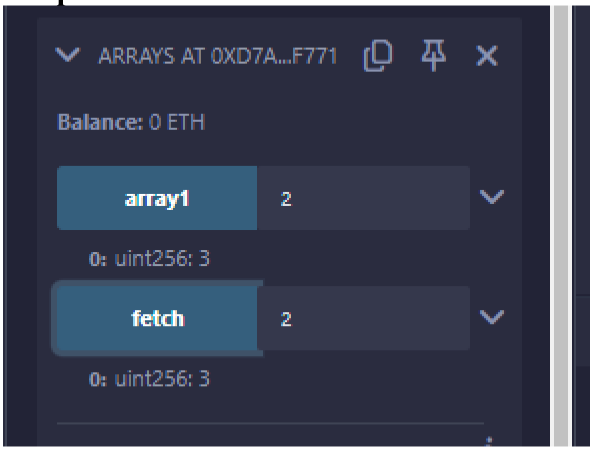
```
0: bool: false
```

Aim 2d: Implement and demonstrate the use of Arrays in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Arrays {

   // Declaring an array
   uint[] public array1 = [1, 2, 3, 4];

   function fetch(uint index) public view returns (uint) {
      require(index < array1.length, "Index out of bounds");
      return array1[index];
   }
}
```

Output:



```
0: uint256: 3
```

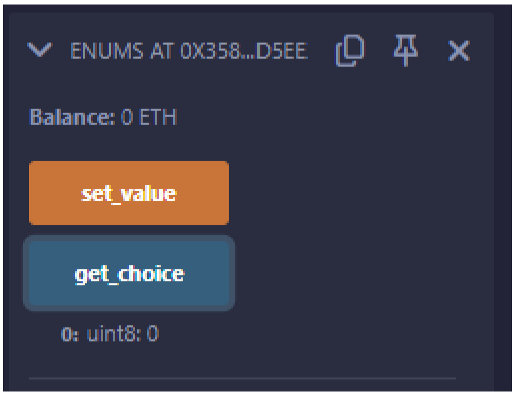
```
0: uint256: 3
```

Aim 2e: Implement and demonstrate the use of Enums in Solidity:

Code:

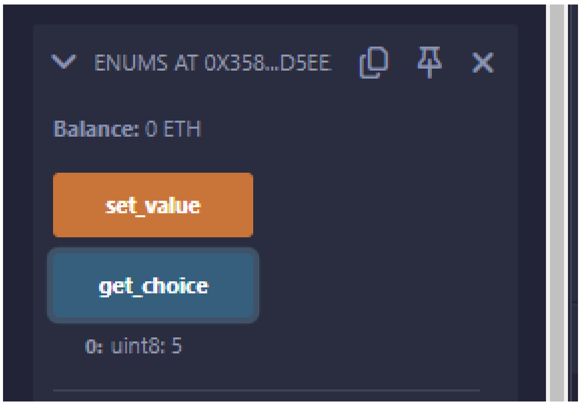
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Enums{

    //Define enum
    enum week_days {Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday}
    week_days choice;

    function set_value() public {
      choice = week_days.Friday;
    }

    // Defining a function to
    // return value of choice
    function get_choice(
    ) public view returns (week_days) {
      return choice;
    }
}
```

Output:



Balance: 0 ETH

0: uint8: 0



Balance: 0 ETH

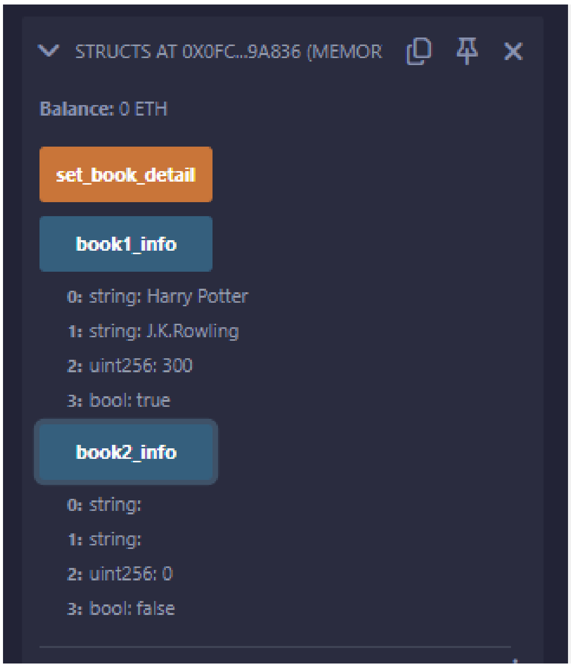
0: uint8: 5

Aim 2f: Implement and demonstrate the use of Structs in Solidity:

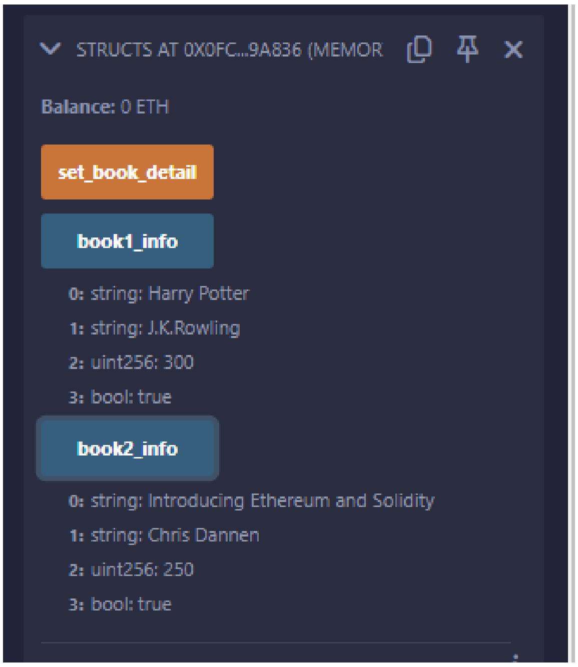
Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Structs{

   //declaring a struct
   struct Book {
      string name;
      string writer;
      uint price;
      bool available;
   }

   Book book1;

   //set book details like this
   Book book2 = Book ("Harry Potter","J.K.Rowling",300,true);

   //set book details like this
   function set_book_detail() public {
   book1 = Book("Introducing Ethereum and Solidity","Chris Dannen",250, true);
   }

   function book1_info() public view returns (string memory, string memory, uint, bool) {
      return(book2.name, book2.writer,book2.price, book2.available);
   }

     function book2_info() public view returns (string memory, string memory, uint, bool) {
     return (book1.name, book1.writer, book1.price, book1.available);
  }

}
```

Output:



```
Balance: 0 ETH
book1_info
0: string: Harry Potter
1: string: J.K.Rowling
2: uint256: 300
3: bool: true
book2_info
0: string:
1: string:
2: uint256: 0
3: bool: false
```



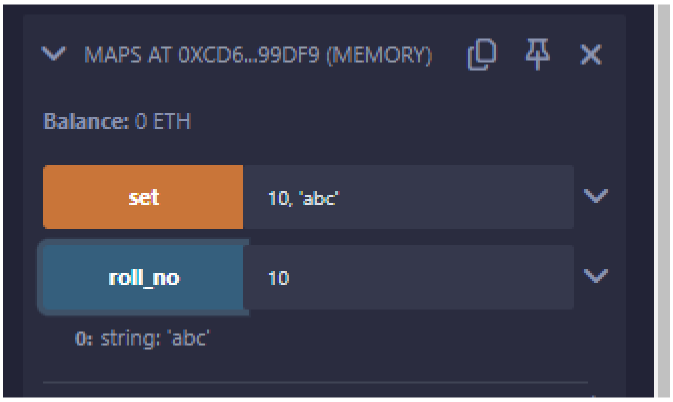
```
Balance: 0 ETH
book1_info
0: string: Harry Potter
1: string: J.K.Rowling
2: uint256: 300
3: bool: true
book2_info
0: string: Introducing Ethereum and Solidity
1: string: Chris Dannen
2: uint256: 250
3: bool: true
```

Aim 2g: Implement and demonstrate the use of Mappings in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract maps{

    mapping (uint=>string) public roll_no;

    function set(uint keys, string memory value) public {
        roll_no[keys]=value;
    }

}
```

Output:



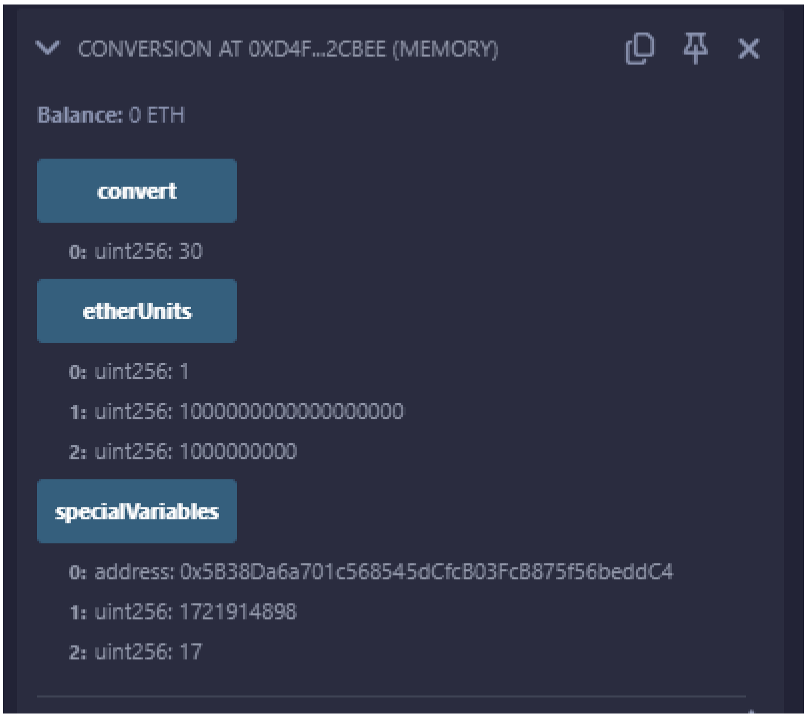
```
0: string: 'abc'
```

Aim 2h: Implement and demonstrate the use of Conversions, Ether Units, Special Variables in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Conversion {

   uint   a = 5;
   uint8  b = 10;
   uint16 c = 15;

   function convert() public view returns (uint) {
      uint result = a + uint(b) + uint(c);
      return result;
   }

   // Demonstrating Ether Units
   function etherUnits() public pure returns (uint, uint, uint) {
      uint oneWei = 1 wei;
      uint oneEther = 1 ether;
      uint oneGwei = 1 gwei;
      return (oneWei, oneEther, oneGwei);
   }

   // Demonstrating Special Variables
   function specialVariables() public view returns (address, uint, uint) {
      address sender = msg.sender; // Sender of the message (current call)
      uint timestamp = block.timestamp; // Current block timestamp
      uint blockNumber = block.number; // Current block number
      return (sender, timestamp, blockNumber);
   }
}
```

Output:



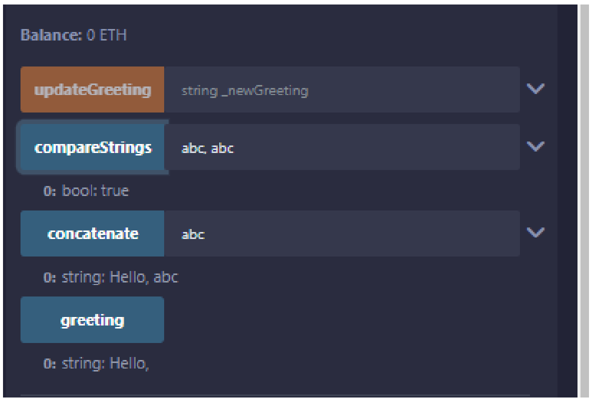
```
CONVERSION AT 0XD4F...2CBEE (MEMORY)

Balance: 0 ETH

0: uint256: 30

0: uint256: 1
1: uint256: 1000000000000000000
2: uint256: 1000000000

0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
1: uint256: 1721914898
2: uint256: 17
```

Aim 2i: Implement and demonstrate the use of Strings in Solidity:

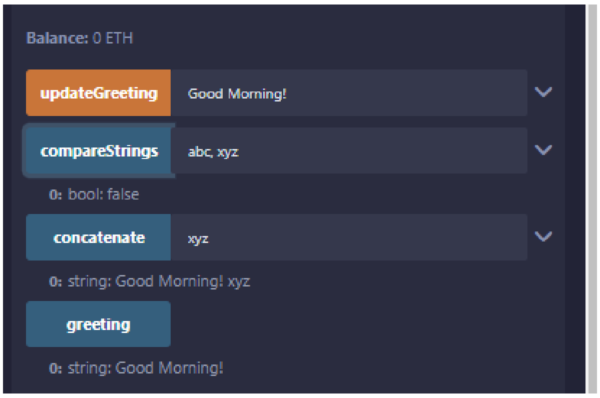
Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract StringExample {
    // State variable to store a string
    string public greeting = "Hello, ";

    // Function to concatenate strings
    function concatenate(string memory _name) public view returns (string memory) {
        return string(abi.encodePacked(greeting, _name));
    }

    // Function to compare two strings
    function compareStrings(string memory _a, string memory _b) public pure returns (bool) {
        return keccak256(abi.encodePacked(_a)) == keccak256(abi.encodePacked(_b));
    }

    // Function to update the greeting
    function updateGreeting(string memory _newGreeting) public {
        greeting = _newGreeting;
    }
}
```

Output:



```
Balance: 0 ETH
updateGreeting   string _newGreeting
compareStrings   abc, abc
0: bool: true
concatenate      abc
0: string: Hello, abc
greeting
0: string: Hello,
```



```
Balance: 0 ETH
updateGreeting   Good Morning!
compareStrings   abc, xyz
0: bool: false
concatenate      xyz
0: string: Good Morning! xyz
greeting
0: string: Good Morning!
```

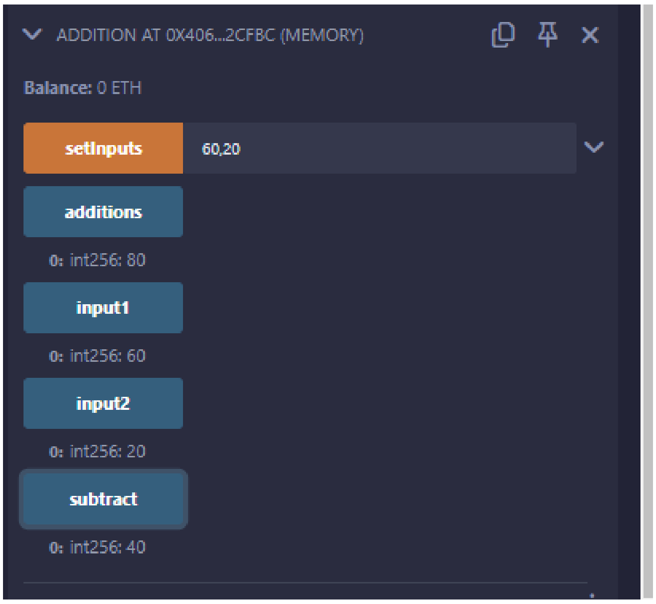
Practical 3

Aim 3a: Implement and demonstrate the use of Functions in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Addition {

    int public input1;
    int public input2;

    function setInputs(int _input1, int _input2) public {
        input1 = _input1;
        input2 = _input2;
    }

    function additions() public view returns(int) {
        return input1 + input2;
    }

    function subtract() public view returns(int) {
        return input1 - input2;
    }
}
```

Output:



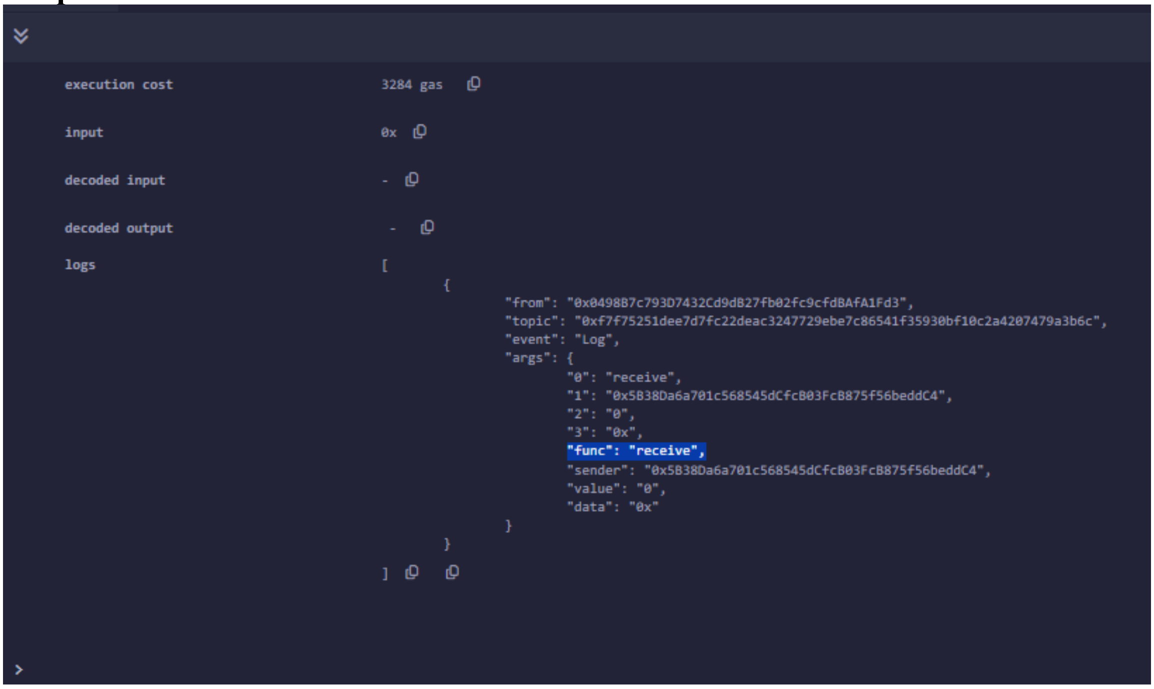
```
ADDITION AT 0X406...2CFBC (MEMORY)
Balance: 0 ETH
setInputs  60,20
additions
0: int256: 80
input1
0: int256: 60
input2
0: int256: 20
subtract
0: int256: 40
```

Aim 3b: Implement and demonstrate the use of Fallback Functions in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract fallbackfn
{
    event Log(string func,address sender, uint value, bytes data);

    fallback() external payable{
        emit Log("fallback",msg.sender,msg.value,msg.data);
    }

    receive() external payable{
        emit Log("receive",msg.sender,msg.value,"");
        //msg.data is empty hence no need to specify it and mark it as empty string
    }
}
```

Output:



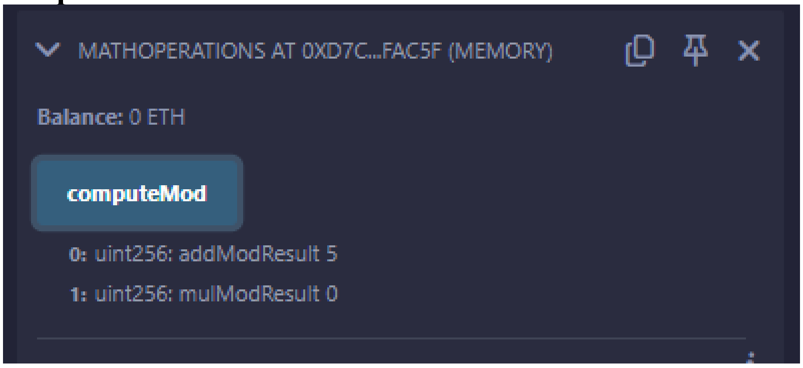
```
execution cost          3284 gas

input                   0x

decoded input           -

decoded output          -

logs                    [
                            {
                                "from": "0x0498B7c793D7432Cd9dB27fb02fc9cfdBAfA1Fd3",
                                "topic": "0xf7f75251dee7d7fc22deac3247729ebe7c86541f35930bf10c2a4207479a3b6c",
                                "event": "Log",
                                "args": {
                                        "0": "receive",
                                        "1": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                        "2": "0",
                                        "3": "0x",
                                        "func": "receive",
                                        "sender": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                        "value": "0",
                                        "data": "0x"
                                }
                            }
                        ]
```

Aim 3c: Implement and demonstrate the use of Mathematical functions in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract MathOperations {
   // addMod computes (x + y) % k
   // mulMod computes (x * y) % k

   function computeMod() public pure returns (uint addModResult, uint mulModResult) {
      uint x = 3;
      uint y = 2;
      uint k = 6;
      addModResult = addmod(x, y, k);
      mulModResult = mulmod(x, y, k);
   }
}
```

Output:



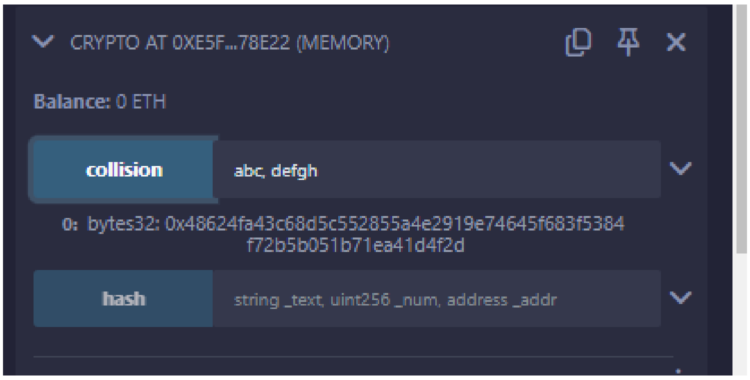
```
Balance: 0 ETH
0: uint256: addModResult 5
1: uint256: mulModResult 0
```

Aim 3d: Implement and demonstrate the use of Cryptographic functions in Solidity:

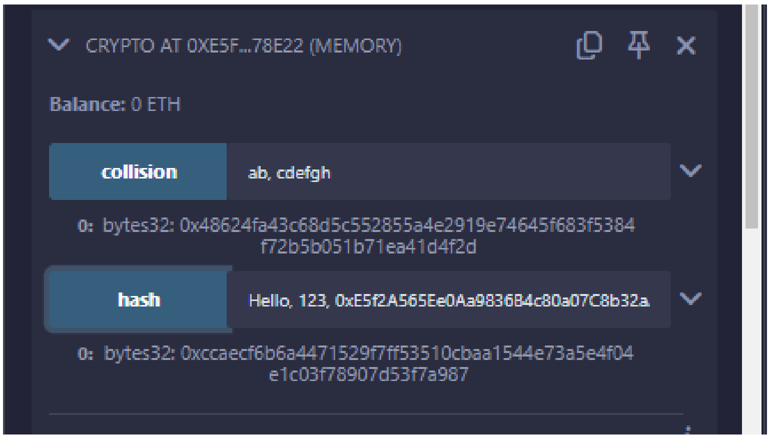
Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Crypto {
   function hash(string memory _text,uint _num,address _addr) public pure returns (bytes32)
{
         return keccak256(abi.encodePacked(_text, _num, _addr));
         }

   function collision(string memory _text, string memory _anotherText)public pure returns
(bytes32){
            return keccak256(abi.encodePacked(_text, _anotherText));
         }
}

 //hash is same for collision
 //0x5f38993891425af42a69bd3cbabdc916f093d4f444455134d4371f4ddd17bd08 - shlok
shivkar
 //0x5f38993891425af42a69bd3cbabdc916f093d4f444455134d4371f4ddd17bd08 - shl
okshivkar

//abc, defgh
//0x48624fa43c68d5c552855a4e2919e74645f683f5384f72b5b051b71ea41d4f2d

//ab, cdefgh
//0x48624fa43c68d5c552855a4e2919e74645f683f5384f72b5b051b71ea41d4f2d


contract GuessTheWord {
   bytes32 public answer =
0x054d6026be33f8ebb0dbd5e7ee11b97bd98f59d6261e53559798f3f81e63dc30;

   function guess(string memory _word) public view returns (bool) {
    return keccak256(abi.encodePacked(_word)) == answer;
   }
}
```

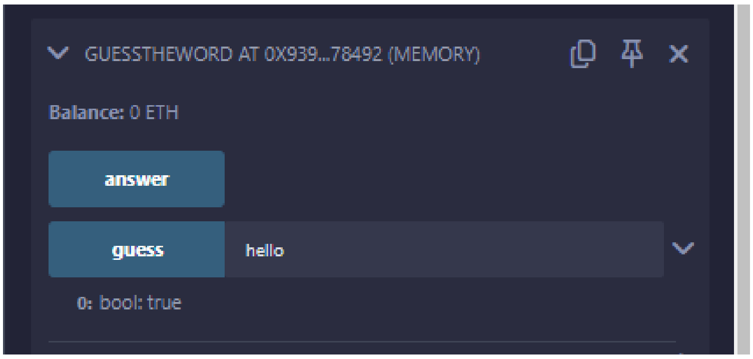
Output:



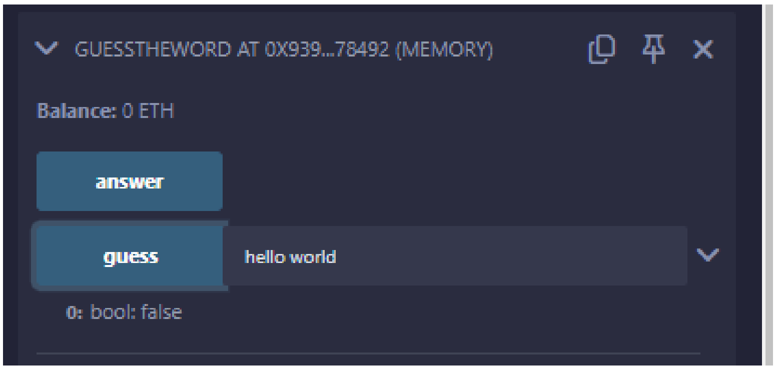
```
CRYPTO AT 0XE5F...78E22 (MEMORY)
Balance: 0 ETH
collision  abc, defgh
0: bytes32: 0x48624fa43c68d5c552855a4e2919e74645f683f5384
            f72b5b051b71ea41d4f2d
hash  string _text, uint256 _num, address _addr
```



```
CRYPTO AT 0XE5F...78E22 (MEMORY)
Balance: 0 ETH
collision  ab, cdefgh
0: bytes32: 0x48624fa43c68d5c552855a4e2919e74645f683f5384
            f72b5b051b71ea41d4f2d
hash  Hello, 123, 0xE5f2A565Ee0Aa983684c80a07C8b32a
0: bytes32: 0xccaecf6b6a4471529f7ff53510cbaa1544e73a5e4f04
            e1c03f78907d53f7a987
```



```
GUESSTHEWORD AT 0X939...78492 (MEMORY)
Balance: 0 ETH
answer
guess  hello
0: bool: true
```



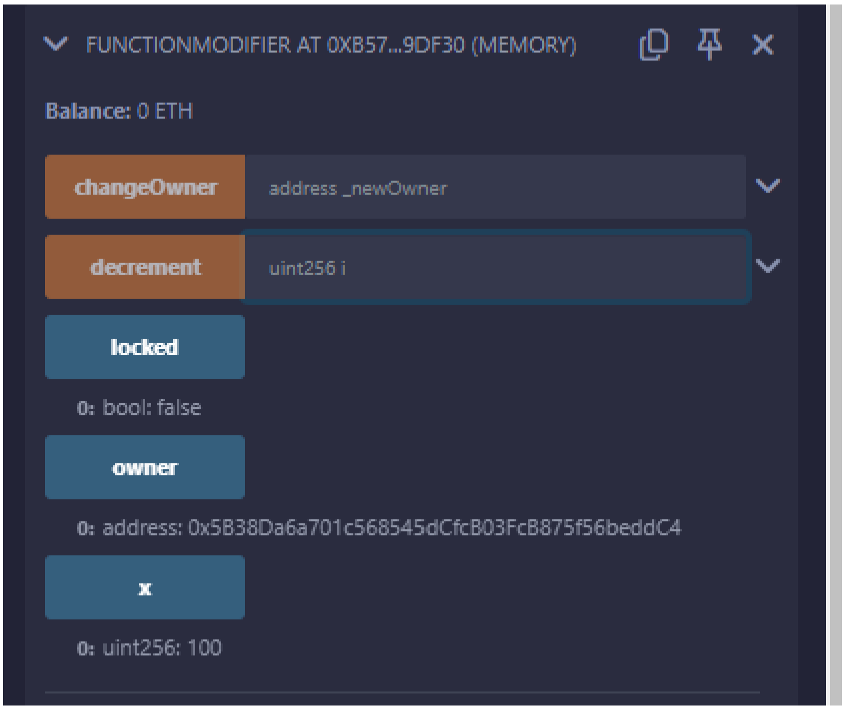
```
GUESSTHEWORD AT 0X939...78492 (MEMORY)
Balance: 0 ETH
answer
guess  hello world
0: bool: false
```

Aim 3e: Implement and demonstrate the use of Function Modifiers in Solidity:

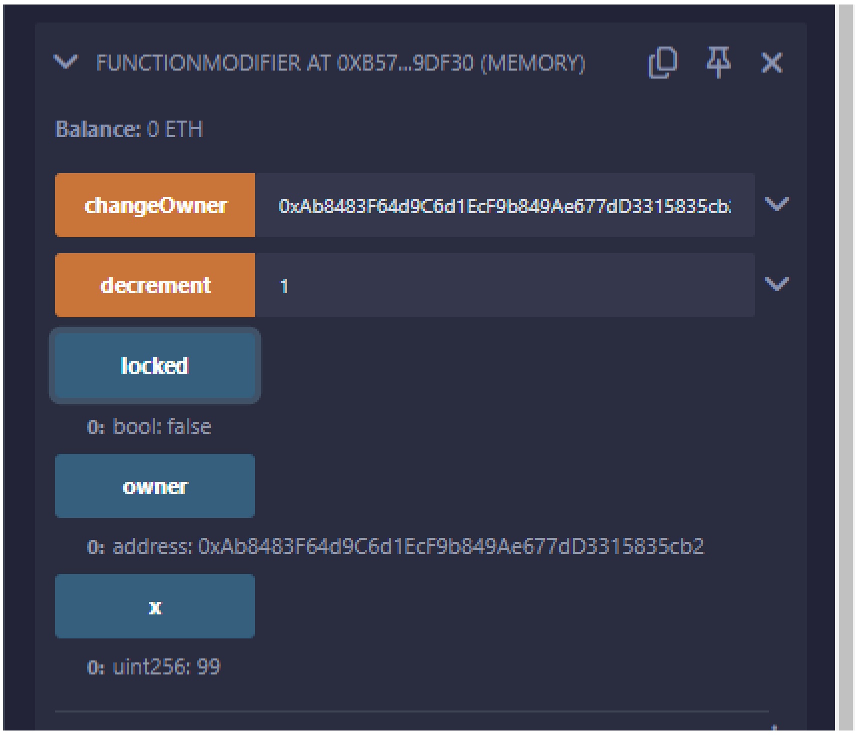
Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

contract FunctionModifier{

    address public owner;
    uint public x = 100;
    bool public locked;

    constructor() {
        // Set the transaction sender as the owner of the contract.
        owner = msg.sender;
        }

        modifier onlyOwner() {
            require(msg.sender == owner, "Not owner");
            _;
            }

        modifier validAddress(address _addr) {
            require(_addr != address(0), "Not valid address");
            _;
            }

    function changeOwner(address _newOwner) public onlyOwner validAddress(_newOwner)
{
        owner = _newOwner;
        }

        modifier noReentrancy() {
            require(!locked, "No reentrancy");
            locked = true;
            _;
            locked = false;
        }

    function decrement(uint i) public noReentrancy {
        x -= i;
        if (i > 1) {
            decrement(i - 1);
        }
    }
}
```

Output:



```
FUNCTIONMODIFIER AT 0XB57...9DF30 (MEMORY)
Balance: 0 ETH
0: bool: false
0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
0: uint256: 100
```



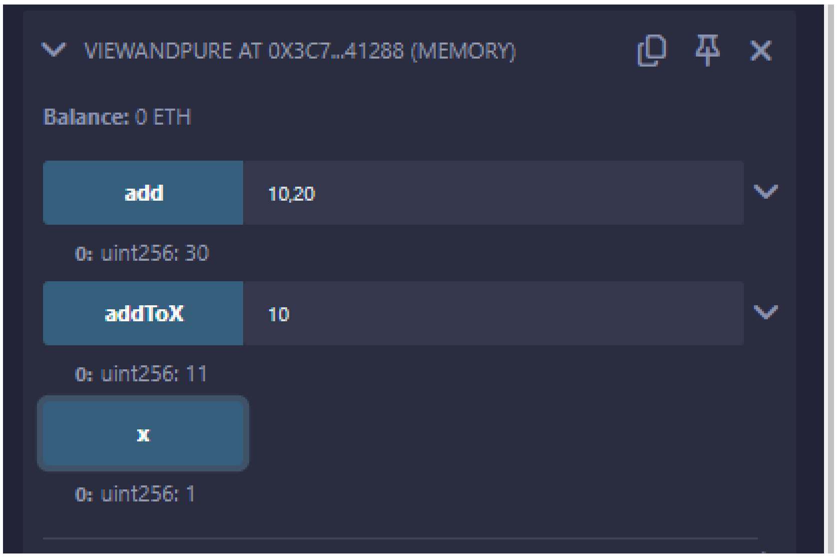
```
FUNCTIONMODIFIER AT 0XB57...9DF30 (MEMORY)
Balance: 0 ETH
changeOwner 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb
decrement 1
0: bool: false
0: address: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
0: uint256: 99
```

Aim 3f: Implement and demonstrate the use of View and Pure Functions in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.3;

contract ViewAndPure {
   uint public x = 1;

   // Promise not to modify the state.
   function addToX(uint y) public view returns (uint) {
      return x + y;
   }

   // Promise not to modify or read from the state.
   function add(uint i, uint j) public pure returns (uint) {
      return i + j;
   }
}
```

Output:



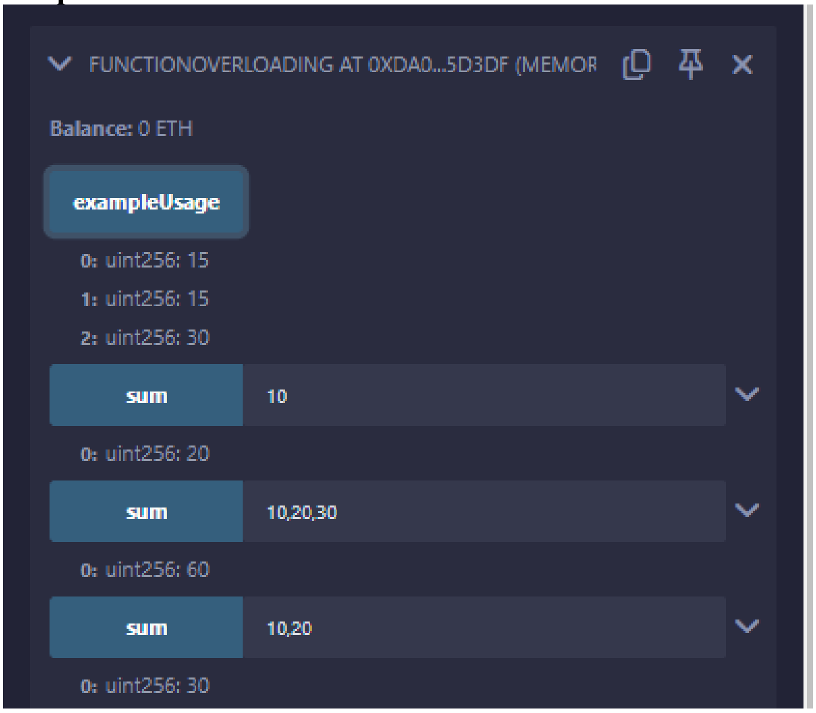
```
add 10,20
0: uint256: 30
addToX 10
0: uint256: 11
x
0: uint256: 1
```

Aim 3g: Implement and demonstrate the use of Function Overloading in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract FunctionOverloading {
    // Function with one parameter
    function sum(uint a) public pure returns (uint) { return a + 10; }

    // Overloaded function with two parameters
    function sum(uint a, uint b) public pure returns (uint) { return a + b; }

    // Overloaded function with three parameters
    function sum(uint a, uint b, uint c) public pure returns (uint) { return a + b + c; }

    // Examples of calling overloaded functions
    function exampleUsage() public pure returns (uint, uint, uint) {
        uint result1 = sum(5);           // Calls the first sum function
        uint result2 = sum(5, 10);       // Calls the second sum function
        uint result3 = sum(5, 10, 15);   // Calls the third sum function

        return (result1, result2, result3);
    }
}
```

Output:



```
FUNCTIONOVERLOADING AT 0XDA0...5D3DF (MEMOR
Balance: 0 ETH
exampleUsage
0: uint256: 15
1: uint256: 15
2: uint256: 30
sum  10
0: uint256: 20
sum  10,20,30
0: uint256: 60
sum  10,20
0: uint256: 30
```

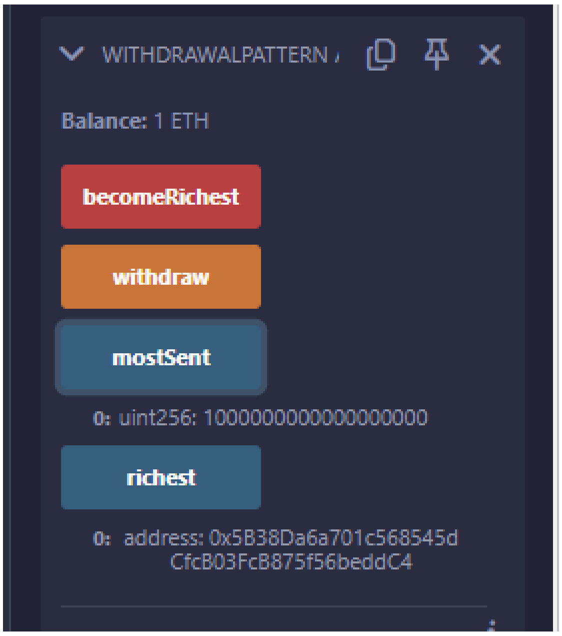
Practical 4

Aim 4a: Implement and demonstrate the use of Withdrawal Pattern in Solidity:

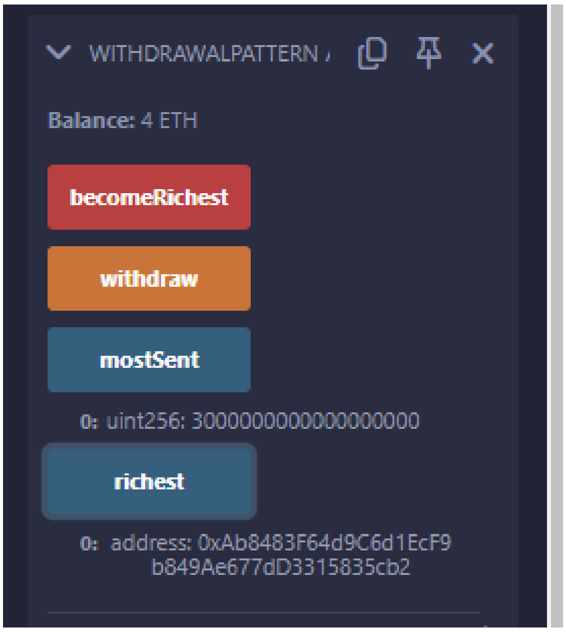
Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

contract withdrawalPattern{
    address public richest;
    uint public mostSent;

    mapping (address=>uint) pendingWithdrawals;
    error NotEnoughEther();

    constructor() payable{
        richest = msg.sender;
        mostSent = msg.value;
    }

    function becomeRichest() public payable{
        if (msg.value <= mostSent) revert NotEnoughEther();
        pendingWithdrawals[richest] += msg.value;
        richest = msg.sender;
        mostSent = msg.value;
    }

    function withdraw() public {
        uint amount = pendingWithdrawals[msg.sender];
        pendingWithdrawals[msg.sender] = 0;
        payable (msg.sender).transfer(amount);
    }

}
```

Output:



```
Balance: 1 ETH
0: uint256: 1000000000000000000
0: address: 0x5B38Da6a701c568545d
            CfcB03FcB875f56beddC4
```



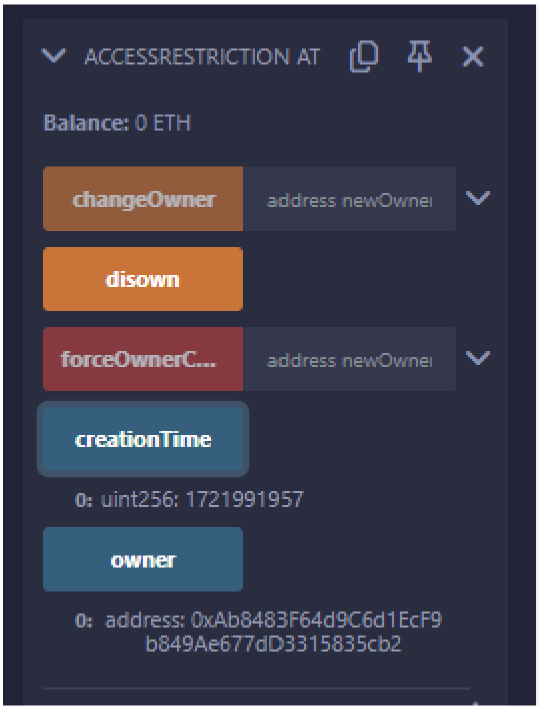
```
Balance: 4 ETH
0: uint256: 3000000000000000000
0: address: 0xAb8483F64d9C6d1EcF9
            b849Ae677dD3315835cb2
```

Aim 4b: Implement and demonstrate the use of Restricted Access in Solidity:

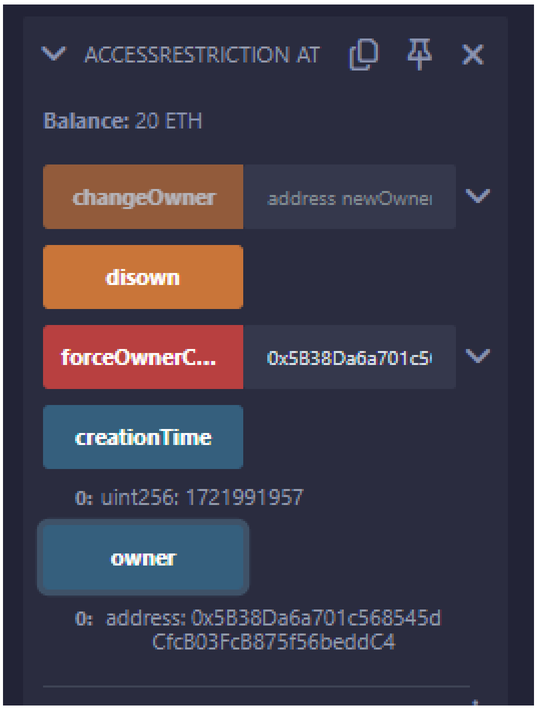
Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;
contract AccessRestriction {

   address public owner = msg.sender;
   uint public creationTime = block.timestamp;

   error Unauthorized();
   error TooEarly();
   error NotEnoughEther();

   modifier onlyBy(address account){
      if (msg.sender != account)
      revert Unauthorized();
      _;
   }

   modifier costs(uint amount) {
      if (msg.value < amount)
         revert NotEnoughEther();
         _;
      if (msg.value > amount)
         payable(msg.sender).transfer(msg.value - amount);
   }

   modifier onlyAfter(uint time) {
      if (block.timestamp < time)
         revert TooEarly();
         _;
   }

   function changeOwner(address newOwner)public onlyBy(owner){
      owner = newOwner;
   }

   function disown()public onlyBy(owner) onlyAfter(creationTime + 6 weeks){
      delete owner;
   }

   function forceOwnerChange(address newOwner)public payable costs(20 ether){
      owner = newOwner;
      // just some example condition
      if (uint160(owner) & 0 == 1)
         return;
   }
}
```

Output:



```
Balance: 0 ETH
0: uint256: 1721991957
0: address: 0xAb8483F64d9C6d1EcF9
   b849Ae677dD3315835cb2
```



```
Balance: 20 ETH
0: uint256: 1721991957
0: address: 0x5B38Da6a701c568545d
   CfcB03FcB875f56beddC4
```

Practical 5

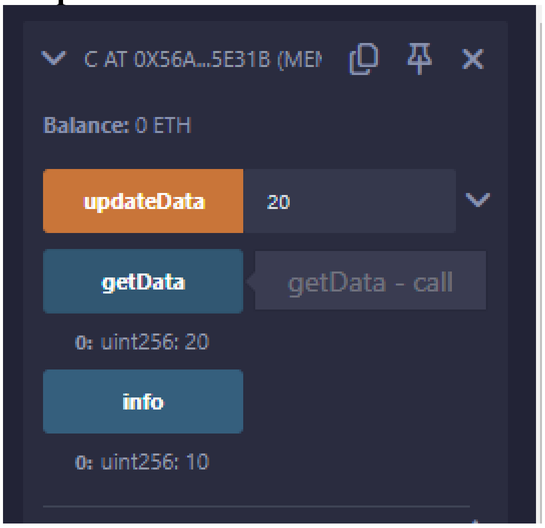
Aim 5a: Implement and demonstrate the use of Contracts and Inheritance in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract C{

   uint private data;
   uint public info;

   constructor()  {
      info = 10;
      }

      function increment(uint a) private pure returns(uint){
         return a + 1;
      }

      function updateData(uint a) public {
         data = a;
      }

      function getData() public view returns(uint) {
         return data;
      }
      function compute(uint a, uint b) internal pure returns (uint) {
         return a + b;
      }
}


contract D {

   function readData() public returns(uint) {
      C c = new C();
      c.updateData(7);
      return c.getData();
   }
}


contract E is C {

   uint private result;
   C private c;
```

```
  constructor()  {
     c = new C();
  }

  function getComputedResult() public {
     result = compute(3, 6);
  }

  function getResult() public view returns(uint) {
     return result;
  }
}
```

Output:

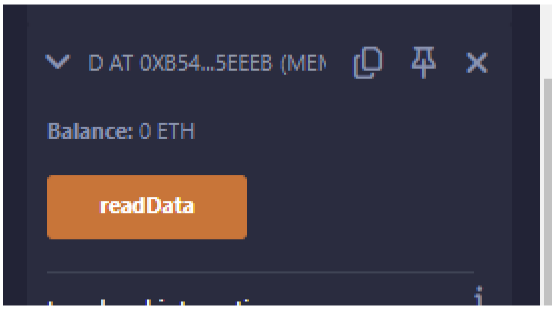


C AT 0X56A...5E31B (MEI

Balance: 0 ETH

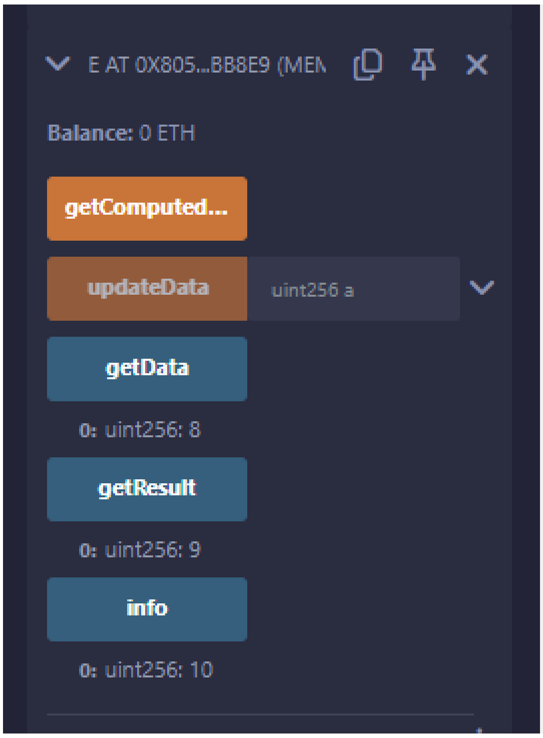
0: uint256: 20

0: uint256: 10



D AT 0XB54...5EEEB (MEI

Balance: 0 ETH



E AT 0X805...BB8E9 (MEN

Balance: 0 ETH

0: uint256: 8

0: uint256: 9

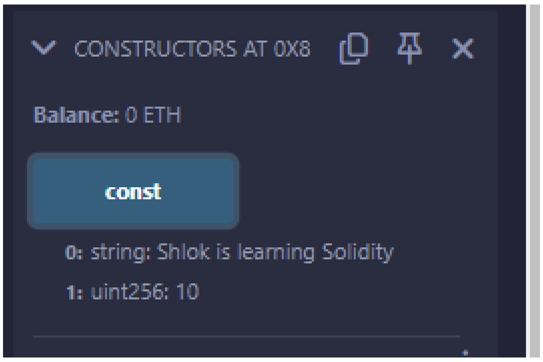
0: uint256: 10

Aim 5b: Implement and demonstrate the use of Constructors in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract constructors{

    string str;
    uint amount;

    constructor(){
        str  = "Shlok is learning Solidity";
        amount = 10;
    }

    function const()public view returns(string memory,uint){
        return (str,amount);

    }

}
```

Output:



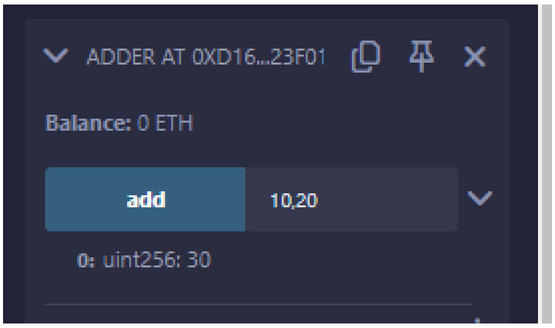
```
0: string: Shlok is learning Solidity
1: uint256: 10
```

Aim 5c: Implement and demonstrate the use of Abstract Contracts in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

abstract contract Main {
   // Define an abstract function that can be overridden
   function add(uint a, uint b) public virtual pure returns (uint);
}

contract Adder is Main {
   // Override the add function from the Main contract
   function add(uint a, uint b) public override pure returns (uint) {
      return a + b;
   }
}
```

Output:



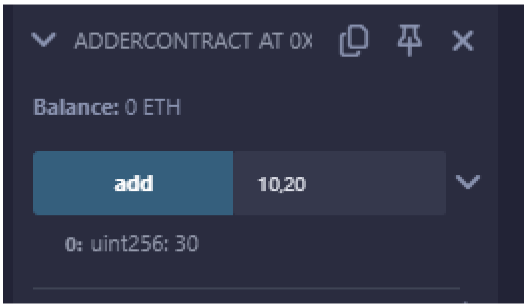
```
Balance: 0 ETH
10,20
0: uint256: 30
```

Aim 5d: Implement and demonstrate the use of Abstract Contracts in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

interface adder{
    function add(uint a, uint b)external pure returns(uint);
}

contract adderContract is adder{
    function add(uint a, uint b)external pure returns(uint){
        return a+b;
    }
}
```

Output:



```
10,20
0: uint256: 30
```

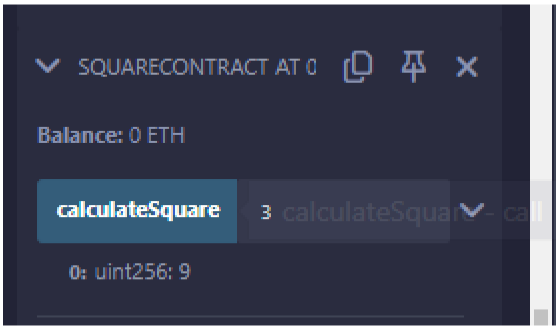
Practical 6

Aim 6a: Implement and demonstrate the use of Libraries in Solidity:

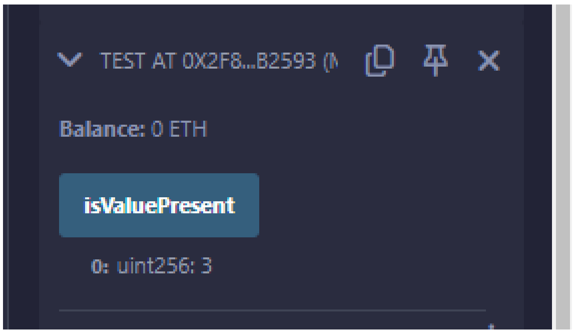
Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

library Search {
  function indexOf(uint[] storage self, uint value) internal view returns (uint) {
    for (uint i = 0; i < self.length; i++) {
      if (self[i] == value) {
        return i;
      }
    }
    return type(uint).max;

  }
}

contract Test {
  uint[] data;

  constructor() {
    data.push(1);
    data.push(2);
    data.push(3);
    data.push(4);
    data.push(5);
  }

  function isValuePresent() external view returns (uint) {
    uint value = 4;

    // Search if value is present in the array using Library function
    uint index = Search.indexOf(data, value);
    return index;
  }
}

library MathLibrary {
  function square(uint num) internal pure returns (uint) {
    return num * num;
  }
}

contract SquareContract {
  using MathLibrary for uint;
  function calculateSquare(uint num) external pure returns (uint) {
    return num.square();
  }
}
```

Output:



```
0: uint256: 9
```



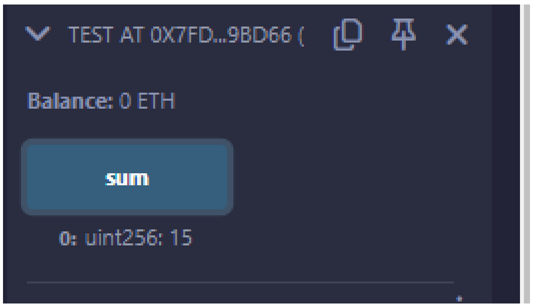
```
0: uint256: 3
```

Aim 6b: Implement and demonstrate the use of Assembly in Solidity:

Code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

library Sum {
  function sumUsingInlineAssembly(uint[] memory _data) public pure returns (uint sum) {
    for (uint i = 0; i < _data.length; ++i) {
      assembly {
        // Load the value from memory at the current index
        let value := mload(add(add(_data, 0x20), mul(i, 0x20)))
        // Add the value to the sum
        sum := add(sum, value)
      }
    }
    // Return the calculated sum
    return sum;
  }
}

contract Test {
  uint[] data;

  constructor() {
    data.push(1);
    data.push(2);
    data.push(3);
    data.push(4);
    data.push(5);
  }

  function sum() external view returns (uint) {
    return Sum.sumUsingInlineAssembly(data);
  }
}
```

Output:



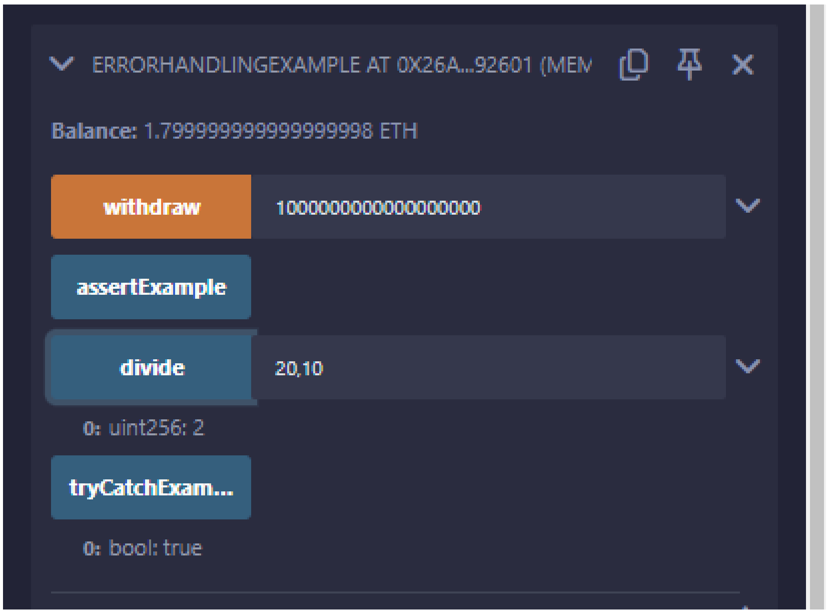
```
Balance: 0 ETH
0: uint256: 15
```

Aim 6c: Implement and demonstrate the use of Error handling in Solidity:

Code:

```
pragma solidity ^0.8.17;

contract ErrorHandlingExample {
    constructor() payable {
        // Allow the contract to receive Ether during deployment
    }

    function divide(uint256 numerator, uint256 denominator) external pure returns (uint256) {
        require(denominator != 0, "Division by zero is not allowed");
        return numerator / denominator;
    }

    function withdraw(uint256 amount) external {
        require(amount <= address(this).balance, "Insufficient balance");

        payable(msg.sender).transfer(amount);
    }

    function assertExample() external pure {
        uint256 x = 5;
        uint256 y = 10;
        assert(x < y);
    }

    function tryCatchExample() external view returns (bool) {
        try this.divide(10, 5) returns (uint256 result) {
            // Handle successful division
            return true;
        } catch Error(string memory errorMessage) {
            // Handle division error
            return false;
        }
    }
}
```

Output:



ERRORHANDLINGEXAMPLE AT 0X26A...92601 (MEM

Balance: 1.799999999999999998 ETH

withdraw 1000000000000000000

assertExample

divide 20,10

0: uint256: 2

tryCatchExam...

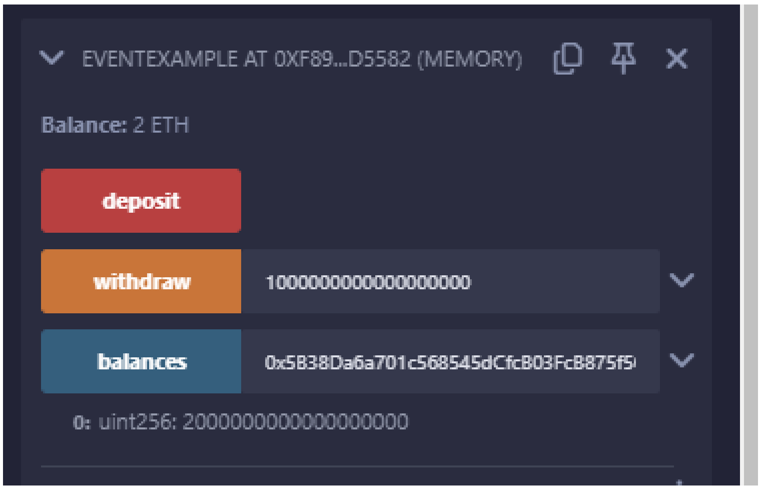
0: bool: true

Aim 6d: Implement and demonstrate the use of Events in Solidity:

Code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract EventExample {

   // Define an event
   event Deposit(address indexed from, uint256 amount);
   event Withdraw(address indexed to, uint256 amount);

   // Mapping to keep track of user balances
   mapping(address => uint256) public balances;

   // Function to deposit ether into the contract
   function deposit() public payable {
      require(msg.value > 0, "Must deposit more than 0 ether");

      // Update the balance
      balances[msg.sender] += msg.value;

      // Emit the Deposit event
      emit Deposit(msg.sender, msg.value);
   }

   // Function to withdraw ether from the contract
   function withdraw(uint256 amount) public {
      require(balances[msg.sender] >= amount, "Insufficient balance");

      // Update the balance
      balances[msg.sender] -= amount;

      // Transfer the ether
      payable(msg.sender).transfer(amount);

      // Emit the Withdraw event
      emit Withdraw(msg.sender, amount);
   }
}
```

Output:



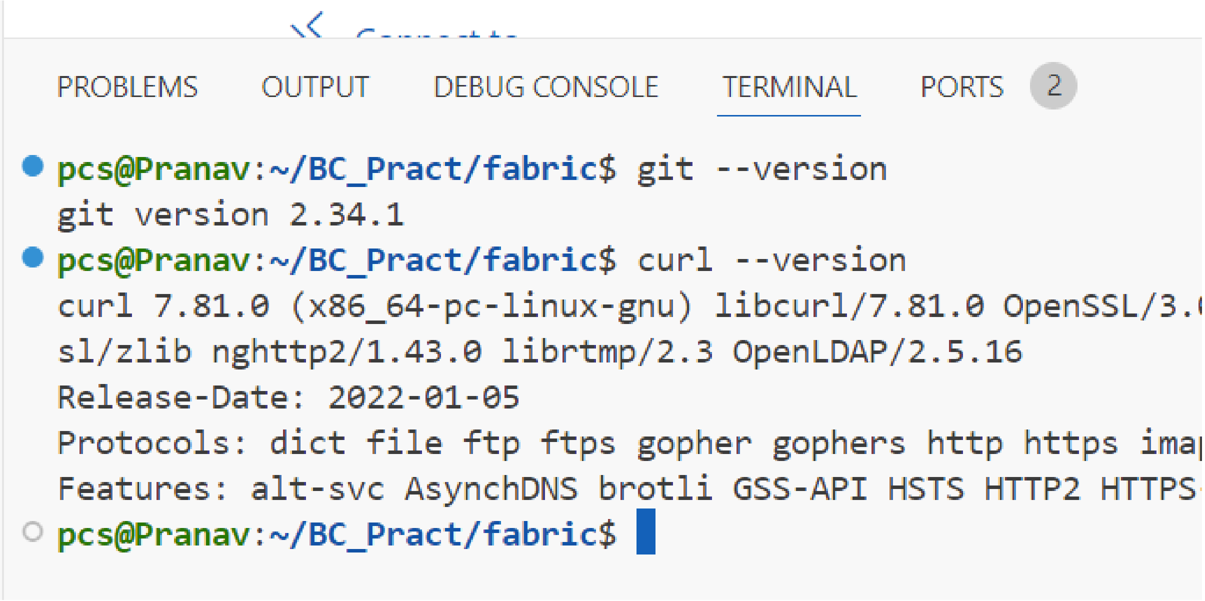
Balance: 2 ETH

```
withdraw: 1000000000000000000
balances: 0x5B38Da6a701c568545dCfcB03FcB875f5
0: uint256: 2000000000000000000
```

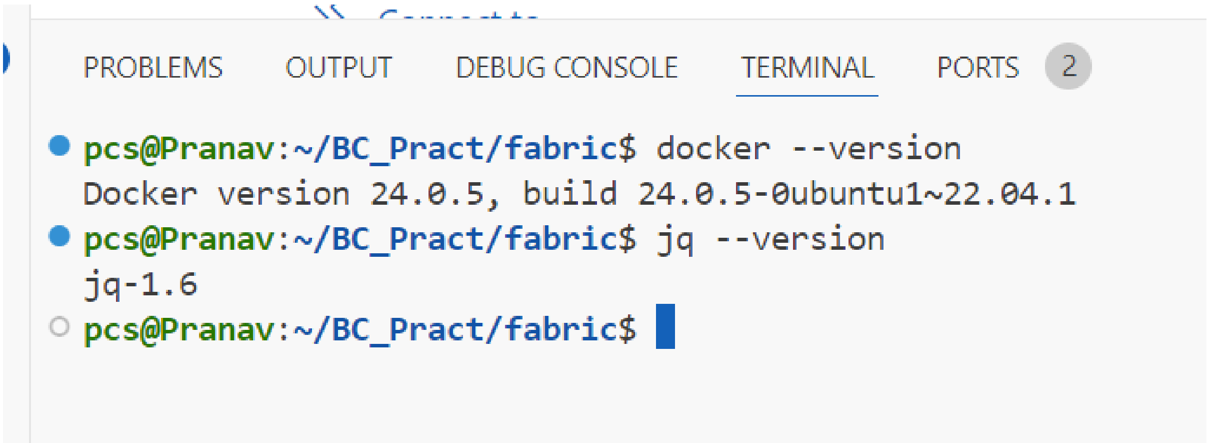
Practical 7

Aim: Install hyperledger fabric

Commands and Output:



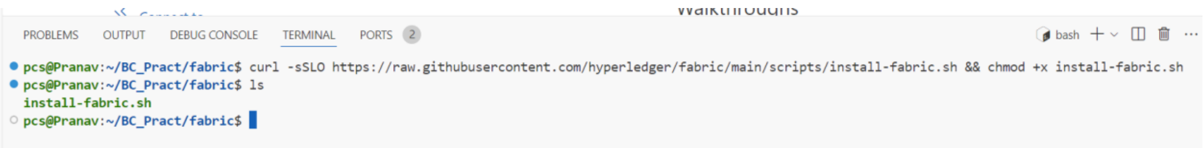
```
pcs@Pranav:~/BC_Pract/fabric$ git --version
git version 2.34.1
pcs@Pranav:~/BC_Pract/fabric$ curl --version
curl 7.81.0 (x86_64-pc-linux-gnu) libcurl/7.81.0 OpenSSL/3.
sl/zlib nghttp2/1.43.0 librtmp/2.3 OpenLDAP/2.5.16
Release-Date: 2022-01-05
Protocols: dict file ftp ftps gopher gophers http https ima
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS
pcs@Pranav:~/BC_Pract/fabric$
```



```
pcs@Pranav:~/BC_Pract/fabric$ docker --version
Docker version 24.0.5, build 24.0.5-0ubuntu1~22.04.1
pcs@Pranav:~/BC_Pract/fabric$ jq --version
jq-1.6
pcs@Pranav:~/BC_Pract/fabric$
```

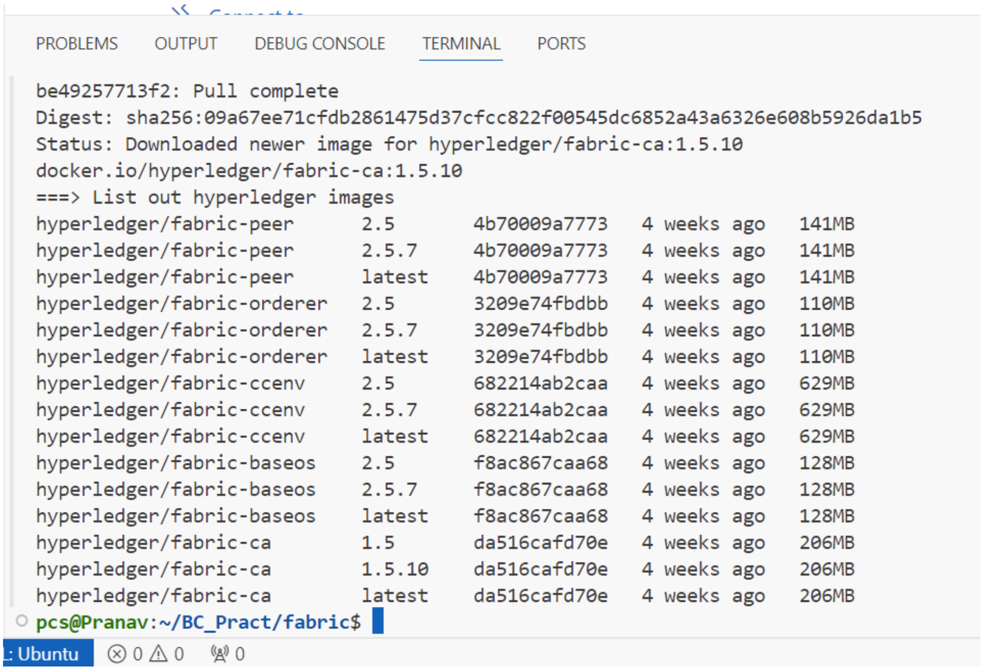
Download fabric samples

curl -sSLO https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh && chmod +x install-fabric.sh



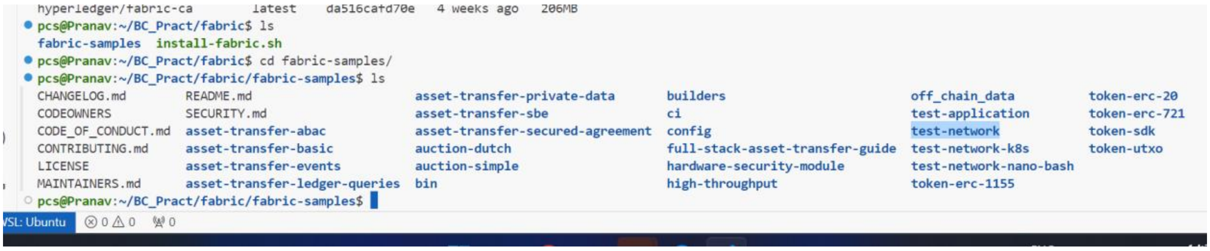
```
pcs@Pranav:~/BC_Pract/fabric$ curl -sSLO https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh && chmod +x install-fabric.sh
pcs@Pranav:~/BC_Pract/fabric$ ls
install-fabric.sh
pcs@Pranav:~/BC_Pract/fabric$
```

Pull the docker containers
./install-fabric.sh



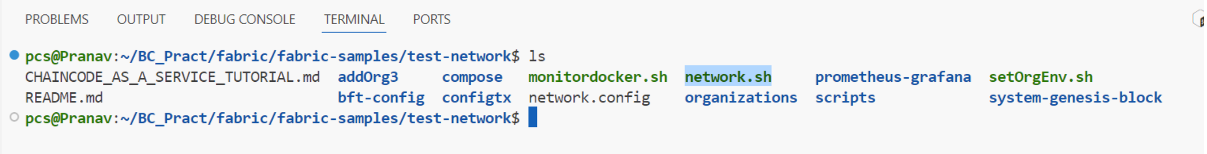
```
be49257713f2: Pull complete
Digest: sha256:09a67ee71cfdb2861475d37cfcc822f00545dc6852a43a6326e608b5926da1b5
Status: Downloaded newer image for hyperledger/fabric-ca:1.5.10
docker.io/hyperledger/fabric-ca:1.5.10
===> List out hyperledger images
hyperledger/fabric-peer      2.5      4b70009a7773   4 weeks ago   141MB
hyperledger/fabric-peer      2.5.7    4b70009a7773   4 weeks ago   141MB
hyperledger/fabric-peer      latest   4b70009a7773   4 weeks ago   141MB
hyperledger/fabric-orderer   2.5      3209e74fbdbb   4 weeks ago   110MB
hyperledger/fabric-orderer   2.5.7    3209e74fbdbb   4 weeks ago   110MB
hyperledger/fabric-orderer   latest   3209e74fbdbb   4 weeks ago   110MB
hyperledger/fabric-ccenv     2.5      682214ab2caa   4 weeks ago   629MB
hyperledger/fabric-ccenv     2.5.7    682214ab2caa   4 weeks ago   629MB
hyperledger/fabric-ccenv     latest   682214ab2caa   4 weeks ago   629MB
hyperledger/fabric-baseos    2.5      f8ac867caa68   4 weeks ago   128MB
hyperledger/fabric-baseos    2.5.7    f8ac867caa68   4 weeks ago   128MB
hyperledger/fabric-baseos    latest   f8ac867caa68   4 weeks ago   128MB
hyperledger/fabric-ca        1.5      da516cafd70e   4 weeks ago   206MB
hyperledger/fabric-ca        1.5.10   da516cafd70e   4 weeks ago   206MB
hyperledger/fabric-ca        latest   da516cafd70e   4 weeks ago   206MB
pcs@Pranav:~/BC_Pract/fabric$
```

Navigate to test network directory
ls
cd fabric-samples
ls



```
hyperledger/fabric-ca        latest   da516cafd70e   4 weeks ago   206MB
pcs@Pranav:~/BC_Pract/fabric$ ls
fabric-samples  install-fabric.sh
pcs@Pranav:~/BC_Pract/fabric$ cd fabric-samples/
pcs@Pranav:~/BC_Pract/fabric/fabric-samples$ ls
CHANGELOG.md        README.md                      asset-transfer-private-data        builders                         off_chain_data            token-erc-20
CODEOWNERS          SECURITY.md                    asset-transfer-sbe                 ci                               test-application          token-erc-721
CODE_OF_CONDUCT.md  asset-transfer-abac            asset-transfer-secured-agreement   config                           test-network              token-sdk
CONTRIBUTING.md     asset-transfer-basic           auction-dutch                      full-stack-asset-transfer-guide  test-network-k8s          token-utxo
LICENSE             asset-transfer-events          auction-simple                     hardware-security-module         test-network-nano-bash
MAINTAINERS.md      asset-transfer-ledger-queries  bin                                high-throughput                  token-erc-1155
pcs@Pranav:~/BC_Pract/fabric/fabric-samples$
```

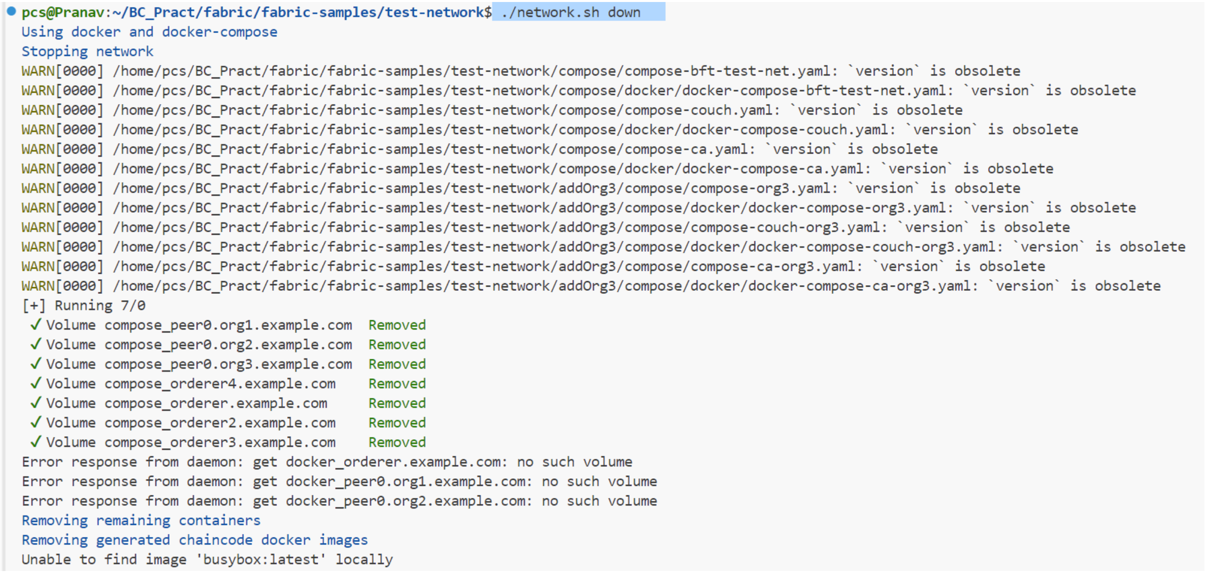
cd test-network
ls



```
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ ls
CHAINCODE_AS_A_SERVICE_TUTORIAL.md  addOrg3     compose   monitordocker.sh  network.sh     prometheus-grafana  setOrgEnv.sh
README.md                           bft-config  configtx  network.config    organizations  scripts             system-genesis-block
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
```

Remove any containers or artifacts

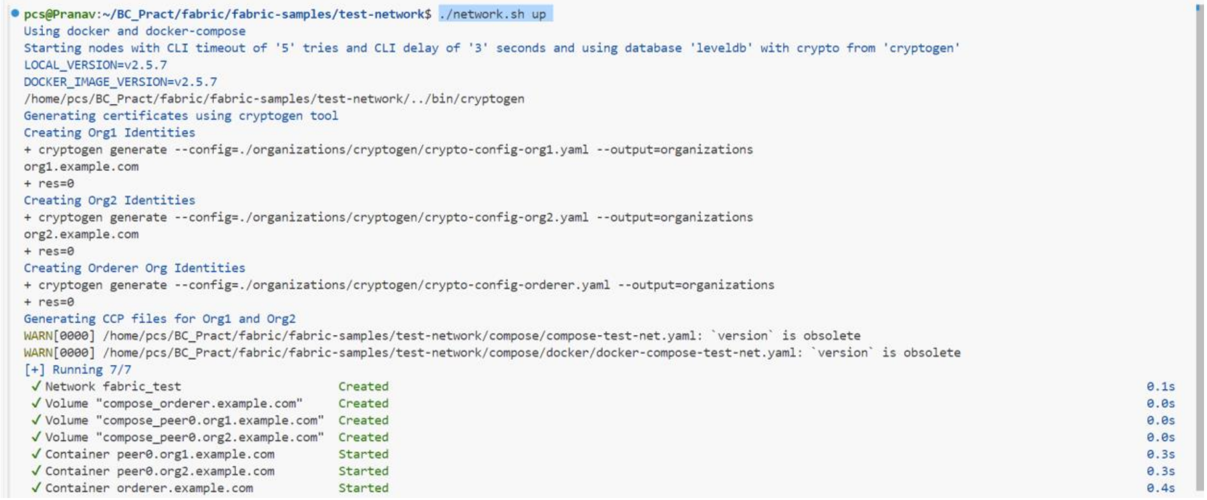
./network.sh down



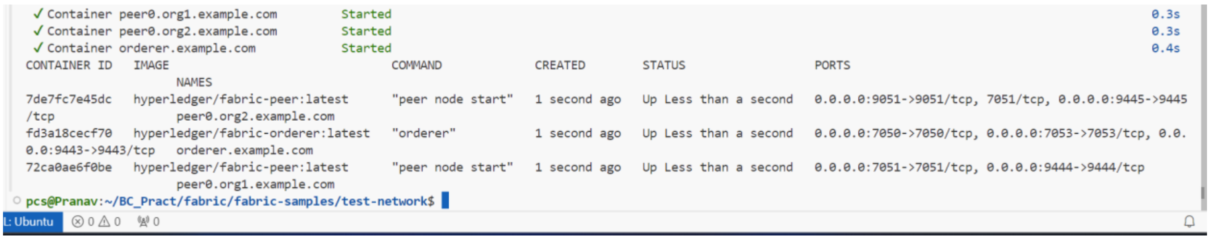
```
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ ./network.sh down
Using docker and docker-compose
Stopping network
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/compose-bft-test-net.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/docker/docker-compose-bft-test-net.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/compose-couch.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/docker/docker-compose-couch.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/compose-ca.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/docker/docker-compose-ca.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/addOrg3/compose/compose-org3.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/addOrg3/compose/docker/docker-compose-org3.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/addOrg3/compose/compose-couch-org3.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/addOrg3/compose/docker/docker-compose-couch-org3.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/addOrg3/compose/compose-ca-org3.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/addOrg3/compose/docker/docker-compose-ca-org3.yaml: `version` is obsolete
[+] Running 7/0
 ✓ Volume compose_peer0.org1.example.com  Removed
 ✓ Volume compose_peer0.org2.example.com  Removed
 ✓ Volume compose_peer0.org3.example.com  Removed
 ✓ Volume compose_orderer4.example.com    Removed
 ✓ Volume compose_orderer.example.com     Removed
 ✓ Volume compose_orderer2.example.com    Removed
 ✓ Volume compose_orderer3.example.com    Removed
Error response from daemon: get docker_orderer.example.com: no such volume
Error response from daemon: get docker_peer0.org1.example.com: no such volume
Error response from daemon: get docker_peer0.org2.example.com: no such volume
Removing remaining containers
Removing generated chaincode docker images
Unable to find image 'busybox:latest' locally
```

Up the network

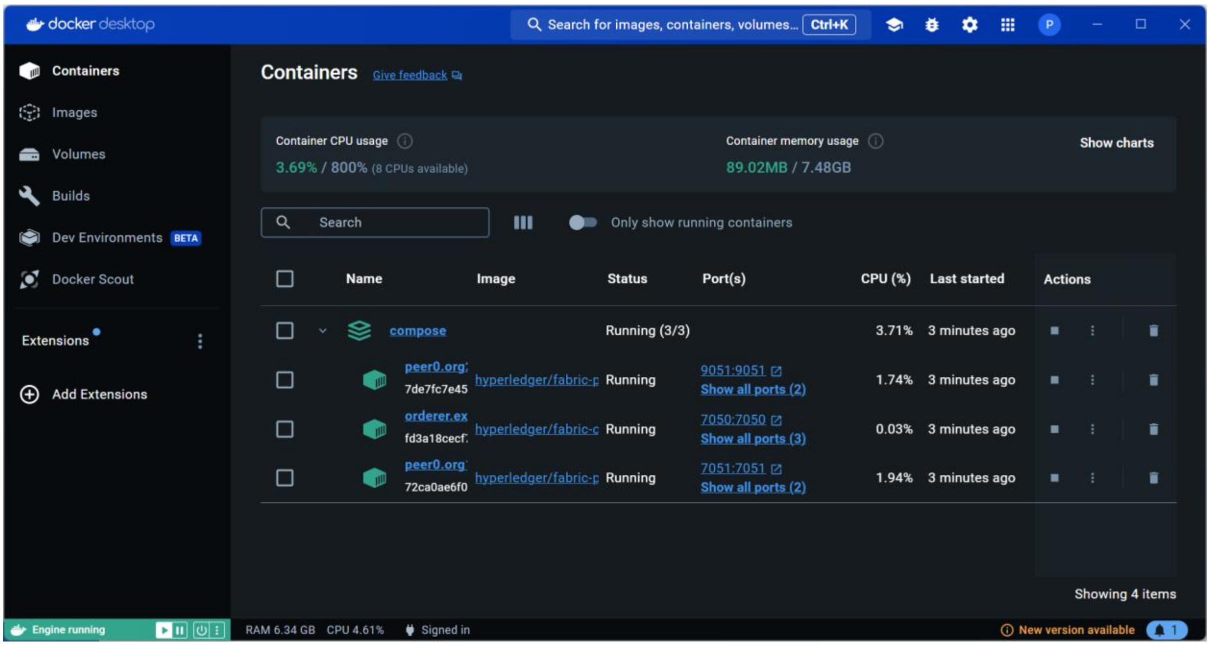
./network.sh up



```
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ ./network.sh up
Using docker and docker-compose
Starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb' with crypto from 'cryptogen'
LOCAL_VERSION=v2.5.7
DOCKER_IMAGE_VERSION=v2.5.7
/home/pcs/BC_Pract/fabric/fabric-samples/test-network/../bin/cryptogen
Generating certificates using cryptogen tool
Creating Org1 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organizations
org1.example.com
+ res=0
Creating Org2 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organizations
org2.example.com
+ res=0
Creating Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
+ res=0
Generating CCP files for Org1 and Org2
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/compose-test-net.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/docker/docker-compose-test-net.yaml: `version` is obsolete
[+] Running 7/7
 ✓ Network fabric_test                     Created                                                              0.1s
 ✓ Volume "compose_orderer.example.com"    Created                                                              0.0s
 ✓ Volume "compose_peer0.org1.example.com" Created                                                              0.0s
 ✓ Volume "compose_peer0.org2.example.com" Created                                                              0.0s
 ✓ Container peer0.org1.example.com        Started                                                              0.3s
 ✓ Container peer0.org2.example.com        Started                                                              0.3s
 ✓ Container orderer.example.com           Started                                                              0.4s
```



```
 ✓ Container peer0.org1.example.com        Started                                                              0.3s
 ✓ Container peer0.org2.example.com        Started                                                              0.3s
 ✓ Container orderer.example.com           Started                                                              0.4s
CONTAINER ID   IMAGE                               COMMAND             CREATED        STATUS                  PORTS
                 NAMES
7de7fc7e45dc   hyperledger/fabric-peer:latest      "peer node start"   1 second ago   Up Less than a second   0.0.0.0:9051->9051/tcp, 7051/tcp, 0.0.0.0:9445->9445
/tcp             peer0.org2.example.com
fd3a18cecf70   hyperledger/fabric-orderer:latest   "orderer"           1 second ago   Up Less than a second   0.0.0.0:7050->7050/tcp, 0.0.0.0:7053->7053/tcp, 0.0.
0.0:9443->9443/tcp   orderer.example.com
72ca0ae6f0be   hyperledger/fabric-peer:latest      "peer node start"   1 second ago   Up Less than a second   0.0.0.0:7051->7051/tcp, 0.0.0.0:9444->9444/tcp
                 peer0.org1.example.com
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
```



Containers

Container CPU usage 3.69% / 800% (8 CPUs available)

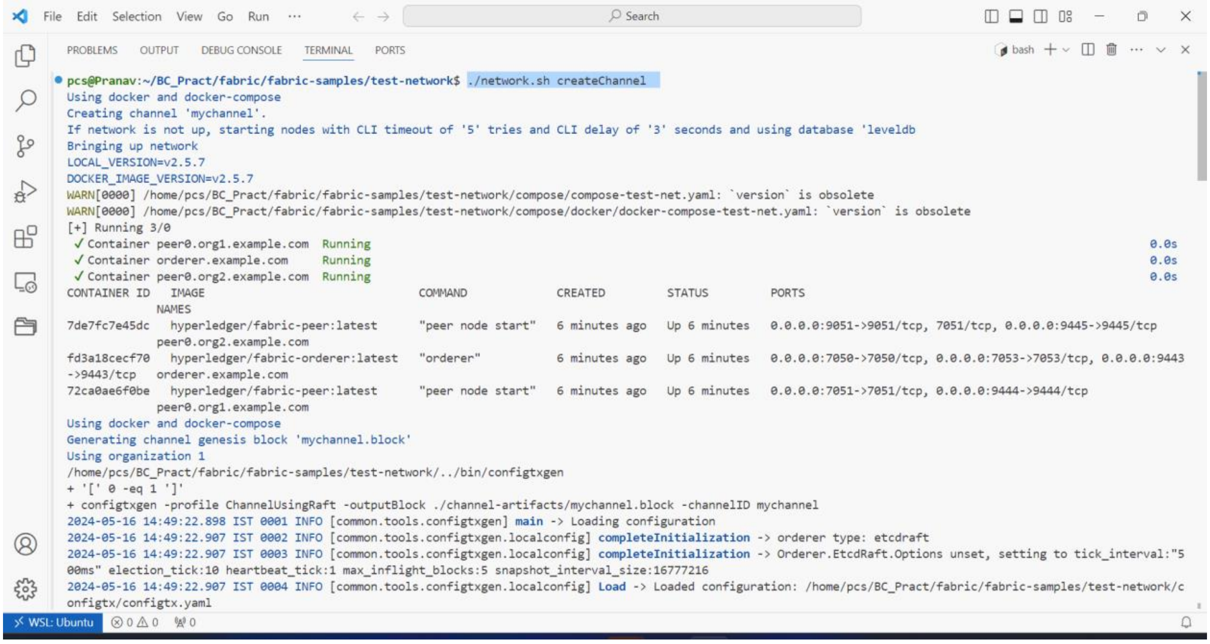
Container memory usage 89.02MB / 7.48GB

	Name	Image	Status	Port(s)	CPU (%)	Last started
□	compose		Running (3/3)		3.71%	3 minutes ago
□	peer0.org: 7de7fc7e45	hyperledger/fabric-p	Running	9051:9051 Show all ports (2)	1.74%	3 minutes ago
□	orderer.ex fd3a18cecf:	hyperledger/fabric-c	Running	7050:7050 Show all ports (3)	0.03%	3 minutes ago
□	peer0.org: 72ca0ae6f0	hyperledger/fabric-p	Running	7051:7051 Show all ports (2)	1.94%	3 minutes ago

Showing 4 items

Create a channel

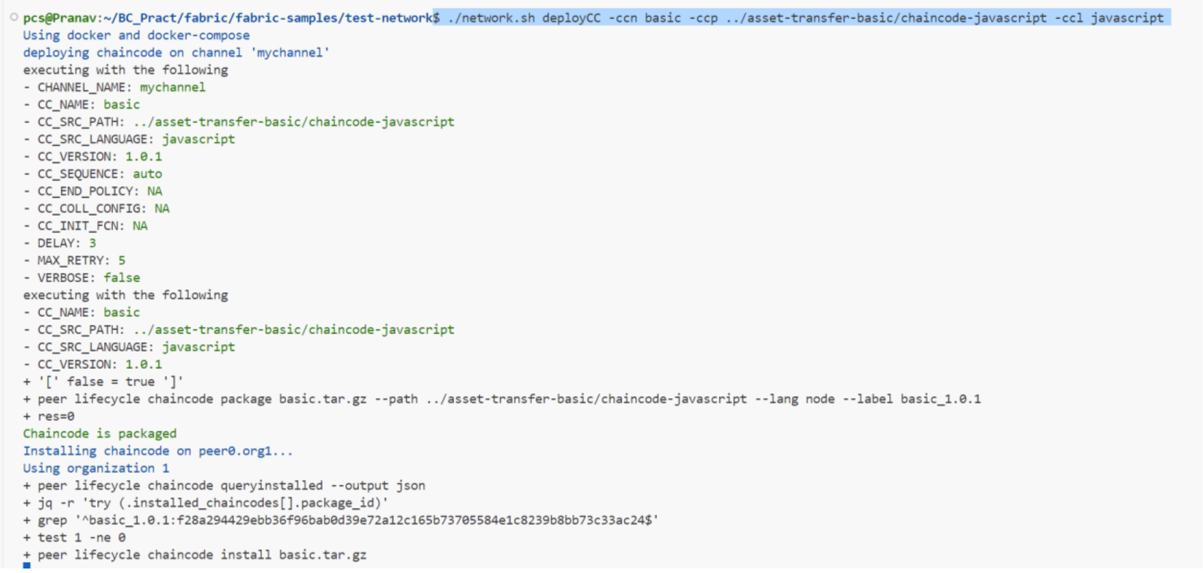
./network.sh createChannel



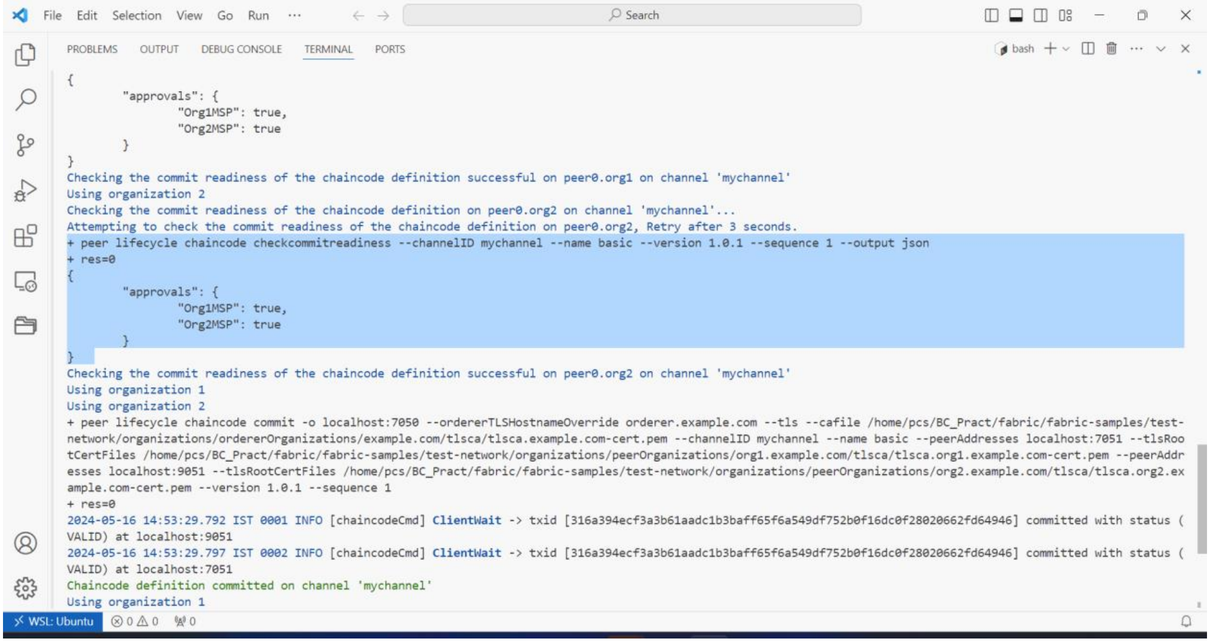
```
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ ./network.sh createChannel
Using docker and docker-compose
Creating channel 'mychannel'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds and using database 'leveldb
Bringing up network
LOCAL_VERSION=v2.5.7
DOCKER_IMAGE_VERSION=v2.5.7
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/compose-test-net.yaml: `version` is obsolete
WARN[0000] /home/pcs/BC_Pract/fabric/fabric-samples/test-network/compose/docker/docker-compose-test-net.yaml: `version` is obsolete
[+] Running 3/0
 ✓ Container peer0.org1.example.com  Running                                                                      0.0s
 ✓ Container orderer.example.com     Running                                                                      0.0s
 ✓ Container peer0.org2.example.com  Running                                                                      0.0s
CONTAINER ID   IMAGE                               COMMAND             CREATED         STATUS         PORTS
          NAMES
7de7fc7e45dc   hyperledger/fabric-peer:latest      "peer node start"   6 minutes ago   Up 6 minutes   0.0.0.0:9051->9051/tcp, 7051/tcp, 0.0.0.0:9445->9445/tcp
          peer0.org2.example.com
fd3a18cecf70   hyperledger/fabric-orderer:latest   "orderer"           6 minutes ago   Up 6 minutes   0.0.0.0:7050->7050/tcp, 0.0.0.0:7053->7053/tcp, 0.0.0.0:9443
->9443/tcp   orderer.example.com
72ca0ae6f0be   hyperledger/fabric-peer:latest      "peer node start"   6 minutes ago   Up 6 minutes   0.0.0.0:7051->7051/tcp, 0.0.0.0:9444->9444/tcp
          peer0.org1.example.com
Using docker and docker-compose
Generating channel genesis block 'mychannel.block'
Using organization 1
/home/pcs/BC_Pract/fabric/fabric-samples/test-network/../bin/configtxgen
+ '[' 0 -eq 1 ']'
+ configtxgen -profile ChannelUsingRaft -outputBlock ./channel-artifacts/mychannel.block -channelID mychannel
2024-05-16 14:49:22.898 IST 0001 INFO [common.tools.configtxgen] main -> Loading configuration
2024-05-16 14:49:22.907 IST 0002 INFO [common.tools.configtxgen.localconfig] completeInitialization -> orderer type: etcdraft
2024-05-16 14:49:22.907 IST 0003 INFO [common.tools.configtxgen.localconfig] completeInitialization -> Orderer.EtcdRaft.Options unset, setting to tick_interval:"5
00ms" election_tick:10 heartbeat_tick:1 max_inflight_blocks:5 snapshot_interval_size:16777216
2024-05-16 14:49:22.907 IST 0004 INFO [common.tools.configtxgen.localconfig] Load -> Loaded configuration: /home/pcs/BC_Pract/fabric/fabric-samples/test-network/c
onfigtx/configtx.yaml
```

Deploy chaincode on peers and channel

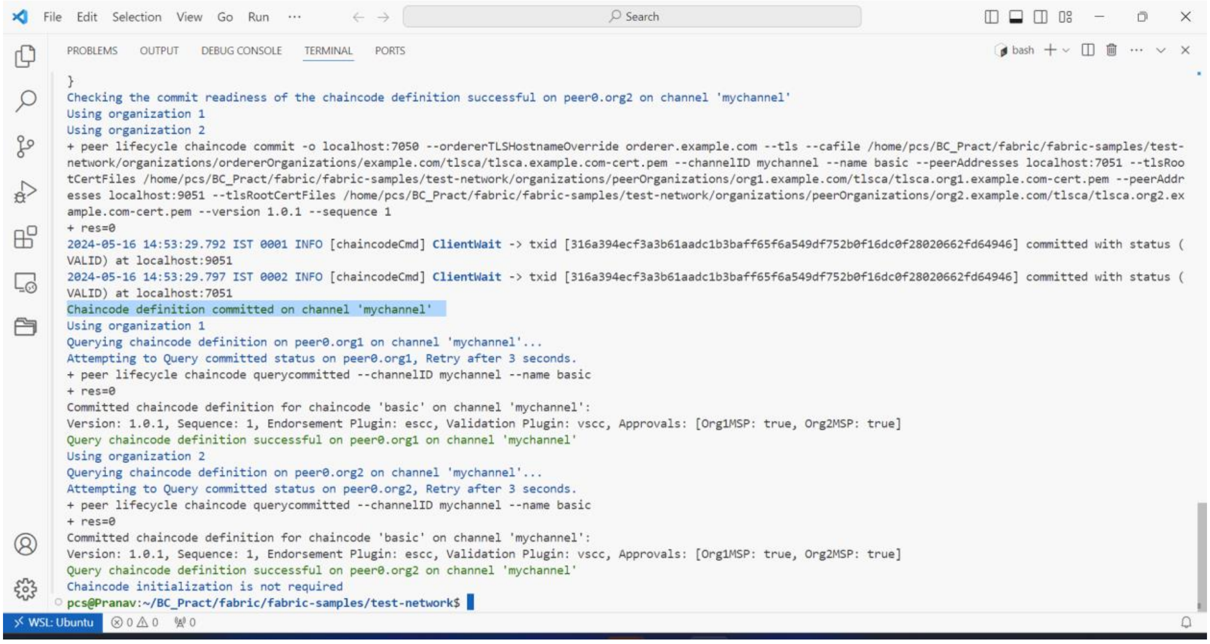
./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-javascript -ccl javascript



```
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ ./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-javascript -ccl javascript
Using docker and docker-compose
deploying chaincode on channel 'mychannel'
executing with the following
- CHANNEL_NAME: mychannel
- CC_NAME: basic
- CC_SRC_PATH: ../asset-transfer-basic/chaincode-javascript
- CC_SRC_LANGUAGE: javascript
- CC_VERSION: 1.0.1
- CC_SEQUENCE: auto
- CC_END_POLICY: NA
- CC_COLL_CONFIG: NA
- CC_INIT_FCN: NA
- DELAY: 3
- MAX_RETRY: 5
- VERBOSE: false
executing with the following
- CC_NAME: basic
- CC_SRC_PATH: ../asset-transfer-basic/chaincode-javascript
- CC_SRC_LANGUAGE: javascript
- CC_VERSION: 1.0.1
+ '[' false = true ']'
+ peer lifecycle chaincode package basic.tar.gz --path ../asset-transfer-basic/chaincode-javascript --lang node --label basic_1.0.1
+ res=0
Chaincode is packaged
Installing chaincode on peer0.org1...
Using organization 1
+ peer lifecycle chaincode queryinstalled --output json
+ jq -r 'try (.installed_chaincodes[].package_id)'
+ grep '^basic_1.0.1:f28a294429ebb36f96bab0d39e72a12c165b73705584e1c8239b8bb73c33ac24$'
+ test 1 -ne 0
+ peer lifecycle chaincode install basic.tar.gz
```



```
{
        "approvals": {
                "Org1MSP": true,
                "Org2MSP": true
        }
}
Checking the commit readiness of the chaincode definition successful on peer0.org1 on channel 'mychannel'
Using organization 2
Checking the commit readiness of the chaincode definition on peer0.org2 on channel 'mychannel'...
Attempting to check the commit readiness of the chaincode definition on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode checkcommitreadiness --channelID mychannel --name basic --version 1.0.1 --sequence 1 --output json
+ res=0
{
        "approvals": {
                "Org1MSP": true,
                "Org2MSP": true
        }
}
Checking the commit readiness of the chaincode definition successful on peer0.org2 on channel 'mychannel'
Using organization 1
Using organization 2
+ peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/pcs/BC_Pract/fabric/fabric-samples/test-
network/organizations/ordererOrganizations/example.com/tlsca/tlsca.example.com-cert.pem --channelID mychannel --name basic --peerAddresses localhost:7051 --tlsRoo
tCertFiles /home/pcs/BC_Pract/fabric/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/tlsca/tlsca.org1.example.com-cert.pem --peerAddr
esses localhost:9051 --tlsRootCertFiles /home/pcs/BC_Pract/fabric/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/tlsca/tlsca.org2.ex
ample.com-cert.pem --version 1.0.1 --sequence 1
+ res=0
2024-05-16 14:53:29.792 IST 0001 INFO [chaincodeCmd] ClientWait -> txid [316a394ecf3a3b61aadc1b3baff65f6a549df752b0f16dc0f28020662fd64946] committed with status (
VALID) at localhost:9051
2024-05-16 14:53:29.797 IST 0002 INFO [chaincodeCmd] ClientWait -> txid [316a394ecf3a3b61aadc1b3baff65f6a549df752b0f16dc0f28020662fd64946] committed with status (
VALID) at localhost:7051
Chaincode definition committed on channel 'mychannel'
Using organization 1
```



```
}
Checking the commit readiness of the chaincode definition successful on peer0.org2 on channel 'mychannel'
Using organization 1
Using organization 2
+ peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/pcs/BC_Pract/fabric/fabric-samples/test-
network/organizations/ordererOrganizations/example.com/tlsca/tlsca.example.com-cert.pem --channelID mychannel --name basic --peerAddresses localhost:7051 --tlsRoo
tCertFiles /home/pcs/BC_Pract/fabric/fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/tlsca/tlsca.org1.example.com-cert.pem --peerAddr
esses localhost:9051 --tlsRootCertFiles /home/pcs/BC_Pract/fabric/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/tlsca/tlsca.org2.ex
ample.com-cert.pem --version 1.0.1 --sequence 1
+ res=0
2024-05-16 14:53:29.792 IST 0001 INFO [chaincodeCmd] ClientWait -> txid [316a394ecf3a3b61aadc1b3baff65f6a549df752b0f16dc0f28020662fd64946] committed with status (
VALID) at localhost:9051
2024-05-16 14:53:29.797 IST 0002 INFO [chaincodeCmd] ClientWait -> txid [316a394ecf3a3b61aadc1b3baff65f6a549df752b0f16dc0f28020662fd64946] committed with status (
VALID) at localhost:7051
Chaincode definition committed on channel 'mychannel'
Using organization 1
Querying chaincode definition on peer0.org1 on channel 'mychannel'...
Attempting to Query committed status on peer0.org1, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name basic
+ res=0
Committed chaincode definition for chaincode 'basic' on channel 'mychannel':
Version: 1.0.1, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org1 on channel 'mychannel'
Using organization 2
Querying chaincode definition on peer0.org2 on channel 'mychannel'...
Attempting to Query committed status on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name basic
+ res=0
Committed chaincode definition for chaincode 'basic' on channel 'mychannel':
Version: 1.0.1, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org2 on channel 'mychannel'
Chaincode initialization is not required
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
```

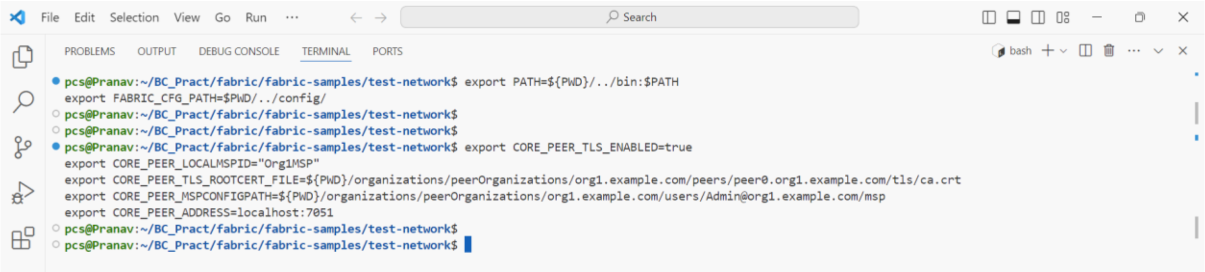
Interacting with the network

Set the path for peer binary and config for core.yaml

```
export PATH=${PWD}/../bin:$PATH
export FABRIC_CFG_PATH=$PWD/../config/
```

Set the environment variables to operate Peer as Org1

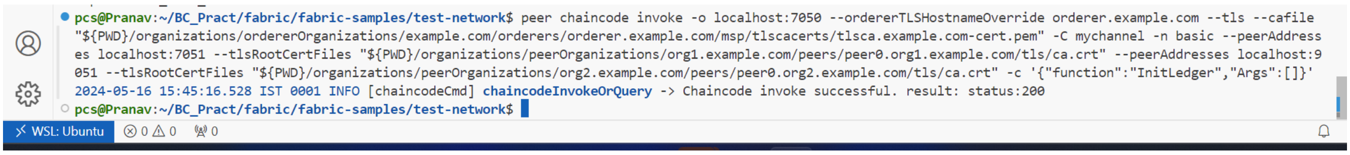
```
export CORE_PEER_TLS_ENABLED=true
export CORE_PEER_LOCALMSPID="Org1MSP"
export
CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
export
CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
export CORE_PEER_ADDRESS=localhost:7051
```



```
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ export PATH=${PWD}/../bin:$PATH
export FABRIC_CFG_PATH=$PWD/../config/
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ export CORE_PEER_TLS_ENABLED=true
export CORE_PEER_LOCALMSPID="Org1MSP"
export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
export CORE_PEER_ADDRESS=localhost:7051
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
```

Command to initialize the ledger with assets

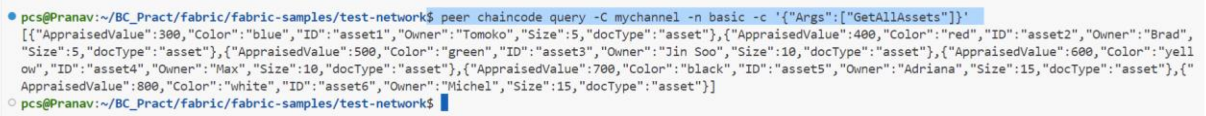
peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"InitLedger","Args":[]}'



```
● pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile
  "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddress
  es localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9
  051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"InitLedger","Args":[]}'
  2024-05-16 15:45:16.528 IST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
○ pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
```

Query the ledger

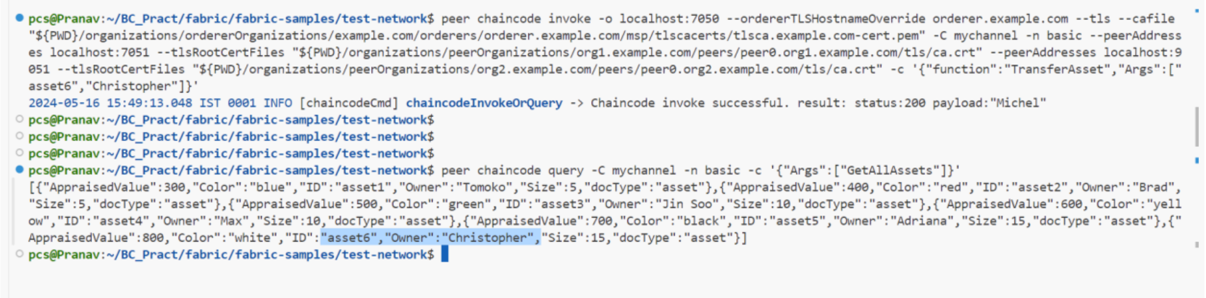
peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'



```
● pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'
  [{"AppraisedValue":300,"Color":"blue","ID":"asset1","Owner":"Tomoko","Size":5,"docType":"asset"},{"AppraisedValue":400,"Color":"red","ID":"asset2","Owner":"Brad",
  "Size":5,"docType":"asset"},{"AppraisedValue":500,"Color":"green","ID":"asset3","Owner":"Jin Soo","Size":10,"docType":"asset"},{"AppraisedValue":600,"Color":"yell
  ow","ID":"asset4","Owner":"Max","Size":10,"docType":"asset"},{"AppraisedValue":700,"Color":"black","ID":"asset5","Owner":"Adriana","Size":15,"docType":"asset"},{"
  AppraisedValue":800,"Color":"white","ID":"asset6","Owner":"Michel","Size":15,"docType":"asset"}]
○ pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
```

Transfer the asset

peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"TransferAsset","Args":["asset6","Christopher"]}'

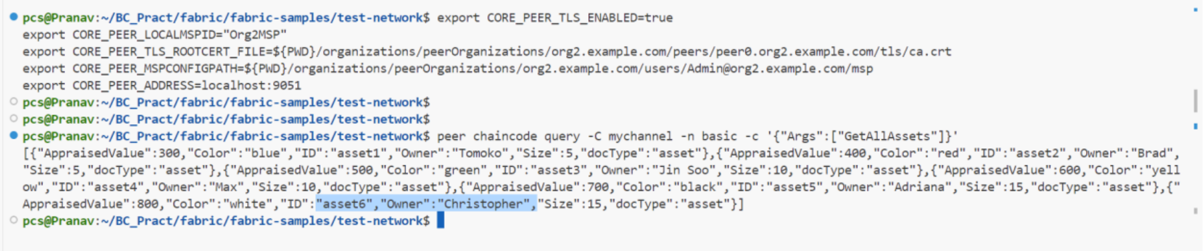


```
● pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile
  "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddress
  es localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9
  051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"TransferAsset","Args":["
  asset6","Christopher"]}'
  2024-05-16 15:49:13.048 IST 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200 payload:"Michel"
○ pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
○ pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
○ pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
● pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'
  [{"AppraisedValue":300,"Color":"blue","ID":"asset1","Owner":"Tomoko","Size":5,"docType":"asset"},{"AppraisedValue":400,"Color":"red","ID":"asset2","Owner":"Brad",
  "Size":5,"docType":"asset"},{"AppraisedValue":500,"Color":"green","ID":"asset3","Owner":"Jin Soo","Size":10,"docType":"asset"},{"AppraisedValue":600,"Color":"yell
  ow","ID":"asset4","Owner":"Max","Size":10,"docType":"asset"},{"AppraisedValue":700,"Color":"black","ID":"asset5","Owner":"Adriana","Size":15,"docType":"asset"},{"
  AppraisedValue":800,"Color":"white","ID":"asset6","Owner":"Christopher","Size":15,"docType":"asset"}]
○ pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
```

Lets query the ledger from Org2 peer

Set the environment variables to operate Peer as Org2

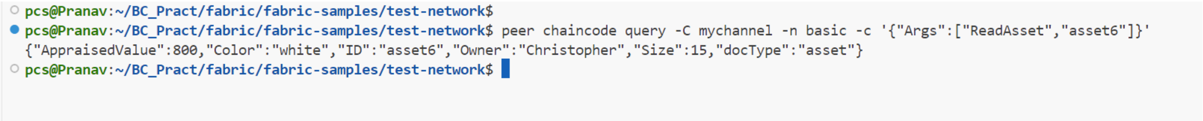
```
export CORE_PEER_TLS_ENABLED=true
export CORE_PEER_LOCALMSPID="Org2MSP"
export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp
export CORE_PEER_ADDRESS=localhost:9051
```



```
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ export CORE_PEER_TLS_ENABLED=true
export CORE_PEER_LOCALMSPID="Org2MSP"
export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt
export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp
export CORE_PEER_ADDRESS=localhost:9051
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'
[{"AppraisedValue":300,"Color":"blue","ID":"asset1","Owner":"Tomoko","Size":5,"docType":"asset"},{"AppraisedValue":400,"Color":"red","ID":"asset2","Owner":"Brad",
"Size":5,"docType":"asset"},{"AppraisedValue":500,"Color":"green","ID":"asset3","Owner":"Jin Soo","Size":10,"docType":"asset"},{"AppraisedValue":600,"Color":"yell
ow","ID":"asset4","Owner":"Max","Size":10,"docType":"asset"},{"AppraisedValue":700,"Color":"black","ID":"asset5","Owner":"Adriana","Size":15,"docType":"asset"},{"
AppraisedValue":800,"Color":"white","ID":"asset6","Owner":"Christopher","Size":15,"docType":"asset"}]
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
```

Query the ledger

```
peer chaincode query -C mychannel -n basic -c '{"Args":["ReadAsset","asset6"]}'
```



```
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$ peer chaincode query -C mychannel -n basic -c '{"Args":["ReadAsset","asset6"]}'
{"AppraisedValue":800,"Color":"white","ID":"asset6","Owner":"Christopher","Size":15,"docType":"asset"}
pcs@Pranav:~/BC_Pract/fabric/fabric-samples/test-network$
```

Bring the network down

```
./network.sh down
```

Practical 8

Aim: Demonstrate the running of the blockchain node

Code and Output:

To check if the prerequisites (Node.js, npm, and Truffle) are installed, you can run the following commands:

Step 1: Prerequisites

Install Node.js

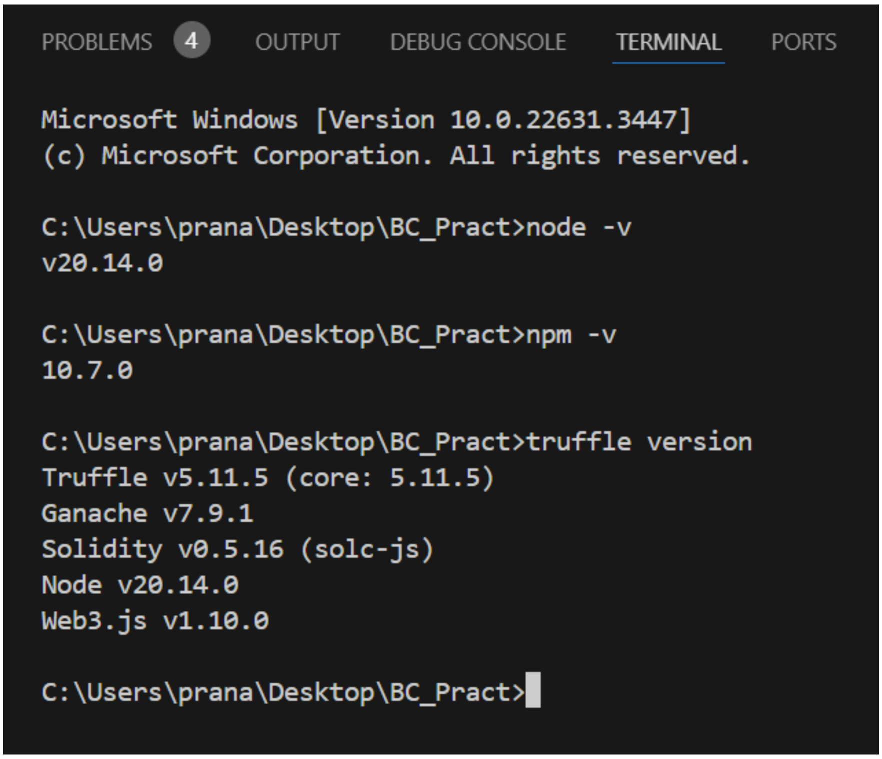
https://nodejs.org/en/download/prebuilt-installer

Execute the following Commands:

```
npm install -g truffle
npm install -g ganache-cli
```

1) **Check Node.js and npm installation:**
   node -v
   npm -v

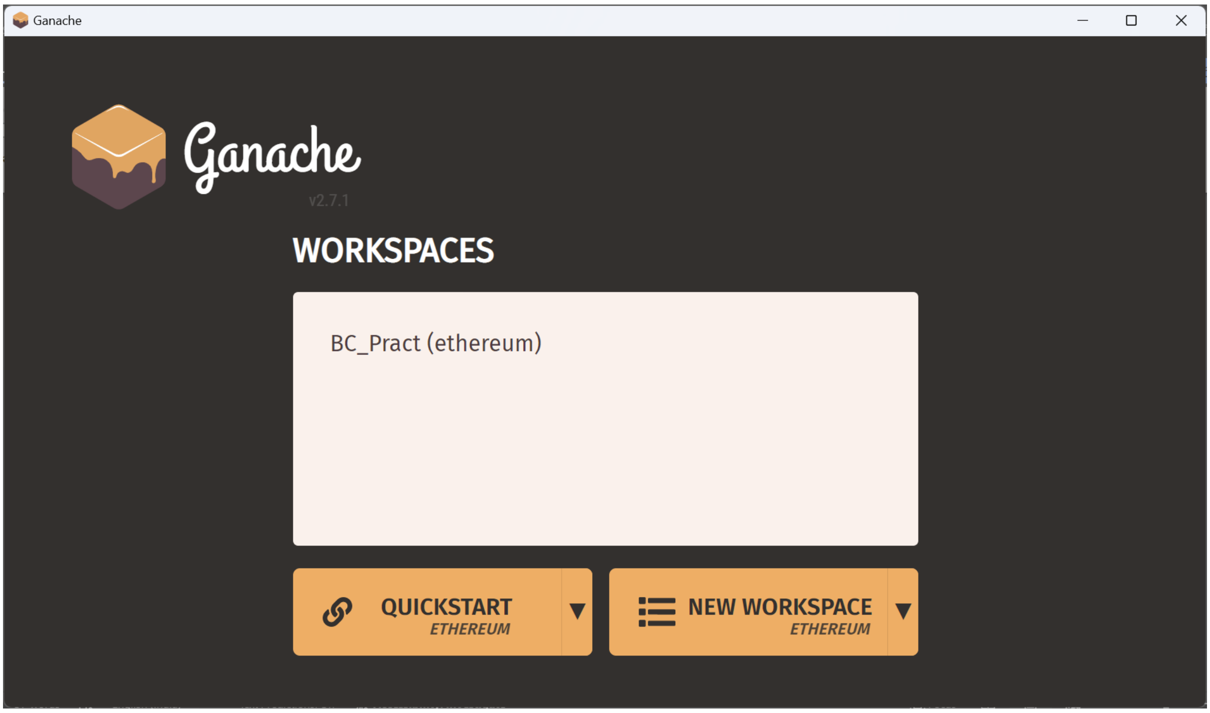
2) **Check Truffle installation:**
   truffle version



```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prana\Desktop\BC_Pract>node -v
v20.14.0

C:\Users\prana\Desktop\BC_Pract>npm -v
10.7.0

C:\Users\prana\Desktop\BC_Pract>truffle version
Truffle v5.11.5 (core: 5.11.5)
Ganache v7.9.1
Solidity v0.5.16 (solc-js)
Node v20.14.0
Web3.js v1.10.0

C:\Users\prana\Desktop\BC_Pract>
```

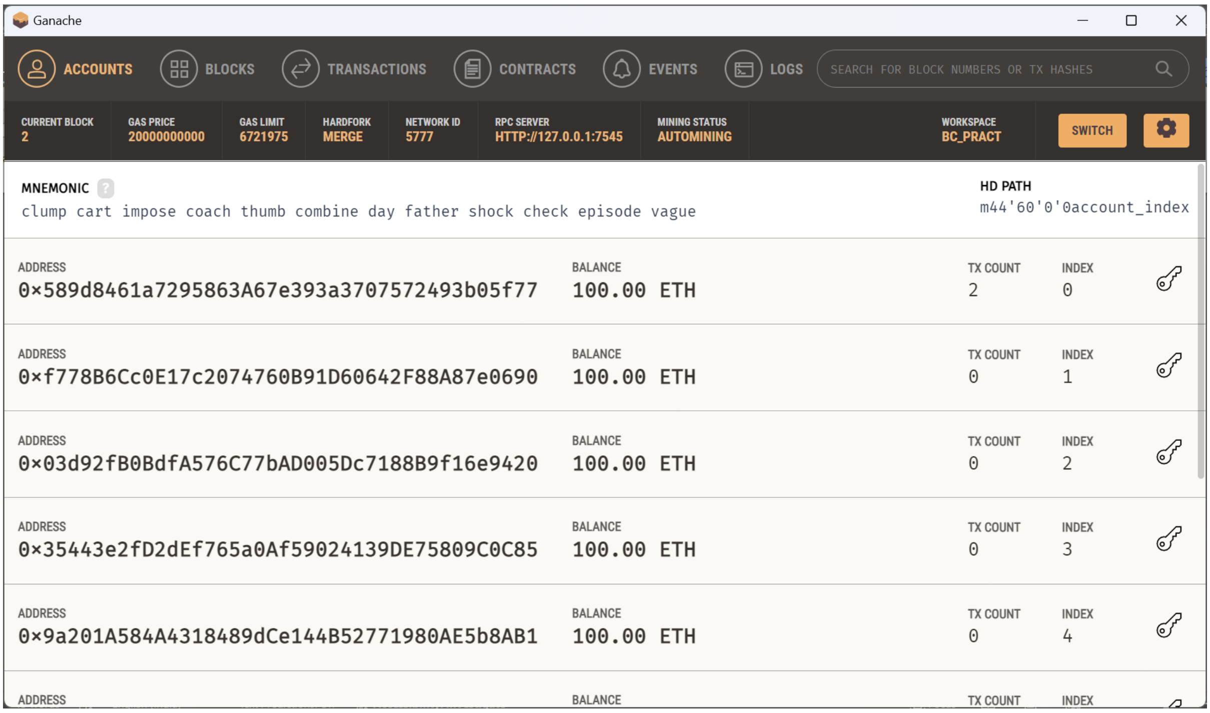
3) Install Ganache

https://archive.trufflesuite.com/ganache/



BC_Pract (ethereum)

4) Create a new Workspace (BC_Pract)



CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
2	20000000000	6721975	MERGE	5777	HTTP://127.0.0.1:7545	AUTOMINING	BC_PRACT

MNEMONIC
clump cart impose coach thumb combine day father shock check episode vague

HD PATH
m44'60'0'0account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×589d8461a7295863A67e393a3707572493b05f77	100.00 ETH	2	0
0×f778B6Cc0E17c2074760B91D60642F88A87e0690	100.00 ETH	0	1
0×03d92fB0BdfA576C77bAD005Dc7188B9f16e9420	100.00 ETH	0	2
0×35443e2fD2dEf765a0Af59024139DE75809C0C85	100.00 ETH	0	3
0×9a201A584A4318489dCe144B52771980AE5b8AB1	100.00 ETH	0	4

Step 2: Initialize a Truffle Project

1) **Create a new directory for your project:**

```
mkdir myProj
cd myProj
```

2) **Initialize the Truffle project:**

```
truffle init
```

Step 3: Create a Solidity Smart Contract

1) **Navigate to the Contracts directory(myProj/contracts):**

SimpleStorage.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract SimpleStorage {
   uint256 public storedData;

   function set(uint256 x) public {
      storedData = x;
   }

   function get() public view returns (uint256) {
      return storedData;
   }
}
```

2) **Compile the Smart Contract**

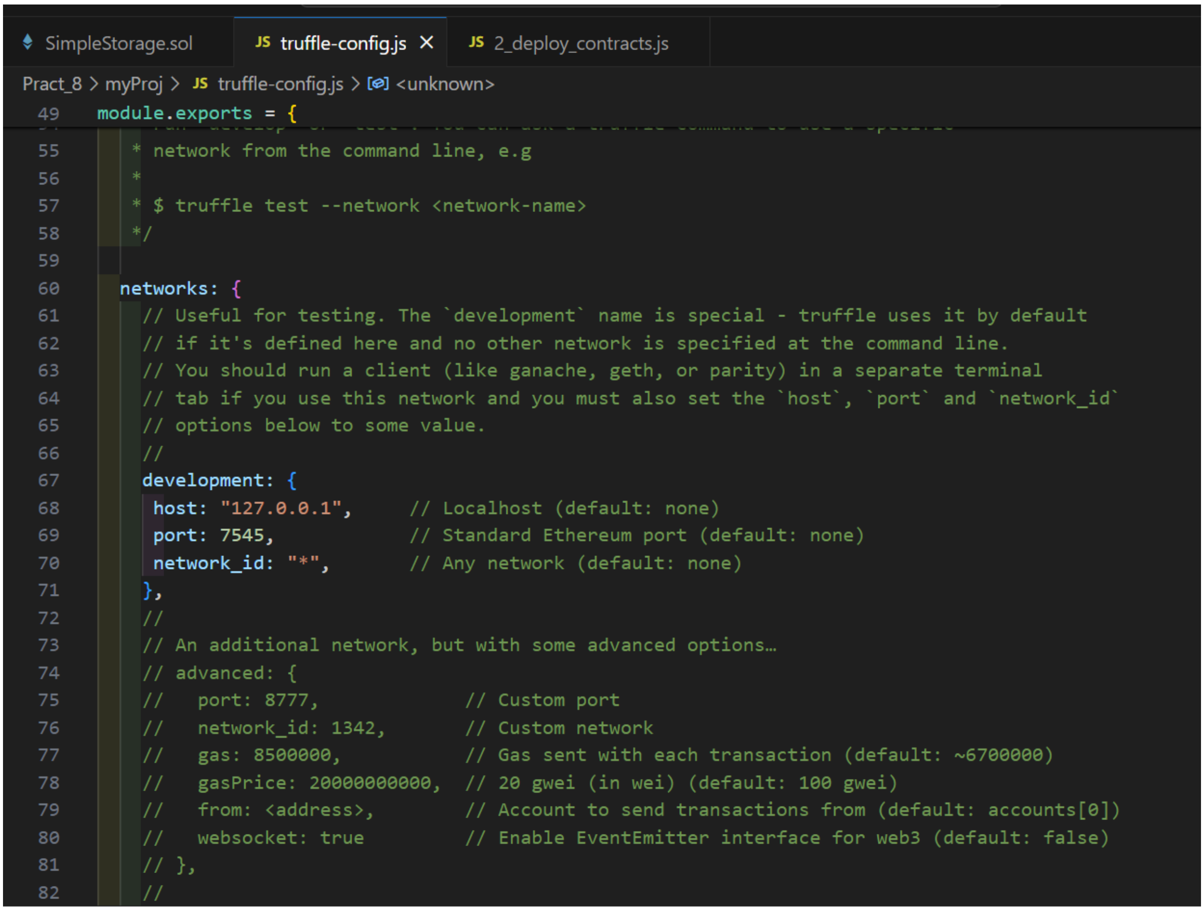
Command: truffle compile

```
C:\Users\prana\Desktop\BC_Pract\Pract_8\myProj>truffle compile
```

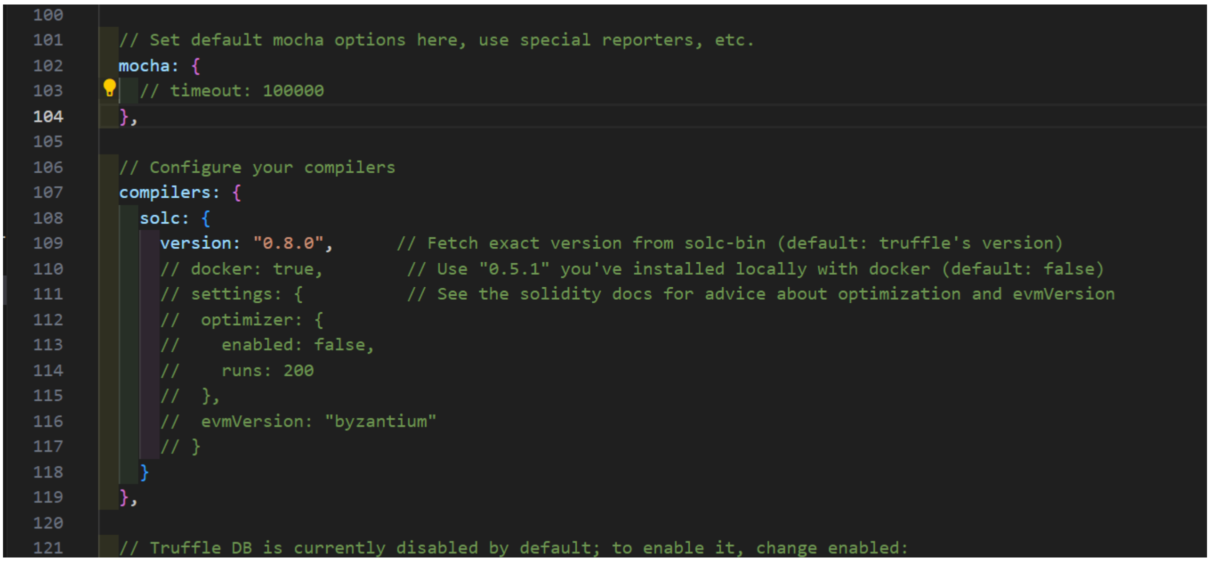
Step 4: Configure Truffle to Use Ganache

Open the truffle-config.js file and configure the development network to use Ganache. Update the networks section:

```
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1",
      port: 7545,  // Match the port Ganache is using
      network_id: "*" // Match any network id
    }
  },
  compilers: {
    solc: {
      version: "0.8.0"  // Specify the Solidity compiler version
    }
  }
};
```



```
module.exports = {
   * network from the command line, e.g
   *
   * $ truffle test --network <network-name>
   */

  networks: {
    // Useful for testing. The `development` name is special - truffle uses it by default
    // if it's defined here and no other network is specified at the command line.
    // You should run a client (like ganache, geth, or parity) in a separate terminal
    // tab if you use this network and you must also set the `host`, `port` and `network_id`
    // options below to some value.
    //
    development: {
     host: "127.0.0.1",     // Localhost (default: none)
     port: 7545,            // Standard Ethereum port (default: none)
     network_id: "*",       // Any network (default: none)
    },
    //
    // An additional network, but with some advanced options…
    // advanced: {
    //   port: 8777,             // Custom port
    //   network_id: 1342,       // Custom network
    //   gas: 8500000,           // Gas sent with each transaction (default: ~6700000)
    //   gasPrice: 20000000000,  // 20 gwei (in wei) (default: 100 gwei)
    //   from: <address>,        // Account to send transactions from (default: accounts[0])
    //   websocket: true         // Enable EventEmitter interface for web3 (default: false)
    // },
    //
```



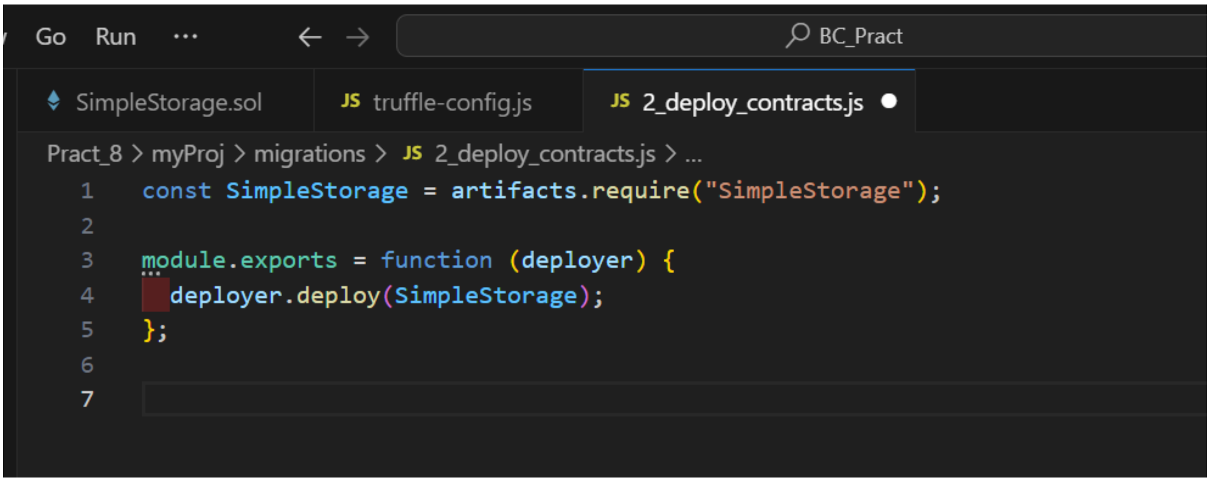
```
// Set default mocha options here, use special reporters, etc.
mocha: {
  // timeout: 100000
},

// Configure your compilers
compilers: {
  solc: {
    version: "0.8.0",      // Fetch exact version from solc-bin (default: truffle's version)
    // docker: true,        // Use "0.5.1" you've installed locally with docker (default: false)
    // settings: {          // See the solidity docs for advice about optimization and evmVersion
    //  optimizer: {
    //    enabled: false,
    //    runs: 200
    //  },
    //  evmVersion: "byzantium"
    // }
  }
},

// Truffle DB is currently disabled by default; to enable it, change enabled:
```

Step 5: Migrate the Smart Contract to Ganache

1) **Start Ganache (open the Ganache application and start a new workspace(BC_Pract)).**
2) **Create a migration script in the migrations directory (e.g., deploy_contracts.js):**

Pract_8\myProj\migrations\2_deploy_contracts.js

```
const SimpleStorage = artifacts.require("SimpleStorage");

module.exports = function (deployer) {
  deployer.deploy(SimpleStorage);
};
```

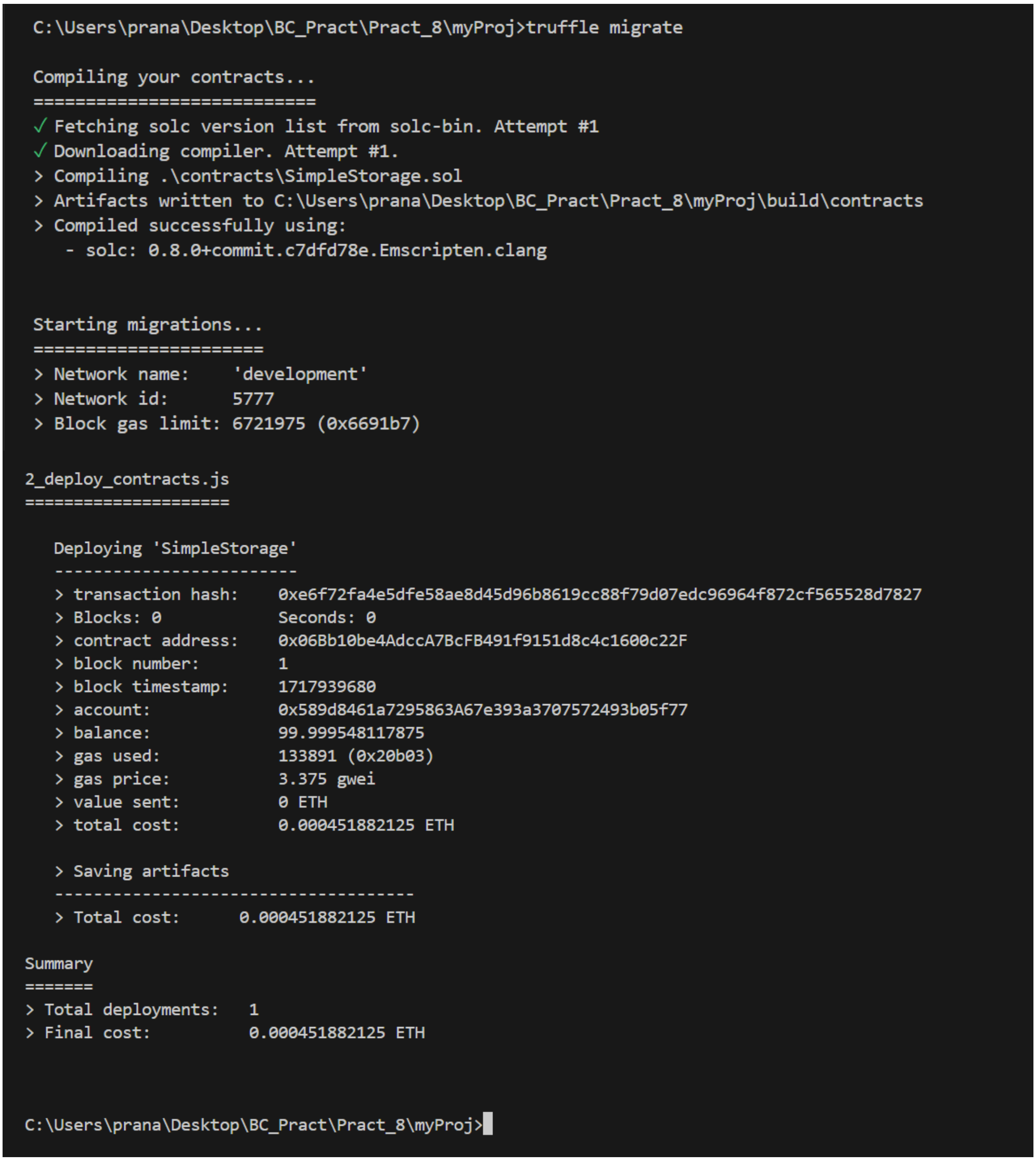


```
const SimpleStorage = artifacts.require("SimpleStorage");

module.exports = function (deployer) {
  deployer.deploy(SimpleStorage);
};
```

3) Run the migration:

Command: truffle migrate

C:\Users\prana\Desktop\BC_Pract\Pract_8\myProj>truffle migrate



```
 C:\Users\prana\Desktop\BC_Pract\Pract_8\myProj>truffle migrate

 Compiling your contracts...
 ===========================
 ✓ Fetching solc version list from solc-bin. Attempt #1
 ✓ Downloading compiler. Attempt #1.
 > Compiling .\contracts\SimpleStorage.sol
 > Artifacts written to C:\Users\prana\Desktop\BC_Pract\Pract_8\myProj\build\contracts
 > Compiled successfully using:
    - solc: 0.8.0+commit.c7dfd78e.Emscripten.clang


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

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

   Deploying 'SimpleStorage'
   -------------------------
   > transaction hash:    0xe6f72fa4e5dfe58ae8d45d96b8619cc88f79d07edc96964f872cf565528d7827
   > Blocks: 0            Seconds: 0
   > contract address:    0x06Bb10be4AdccA7BcFB491f9151d8c4c1600c22F
   > block number:        1
   > block timestamp:     1717939680
   > account:             0x589d8461a7295863A67e393a3707572493b05f77
   > balance:             99.999548117875
   > gas used:            133891 (0x20b03)
   > gas price:           3.375 gwei
   > value sent:          0 ETH
   > total cost:          0.000451882125 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:      0.000451882125 ETH

Summary
=======
> Total deployments:   1
> Final cost:          0.000451882125 ETH


C:\Users\prana\Desktop\BC_Pract\Pract_8\myProj>
```

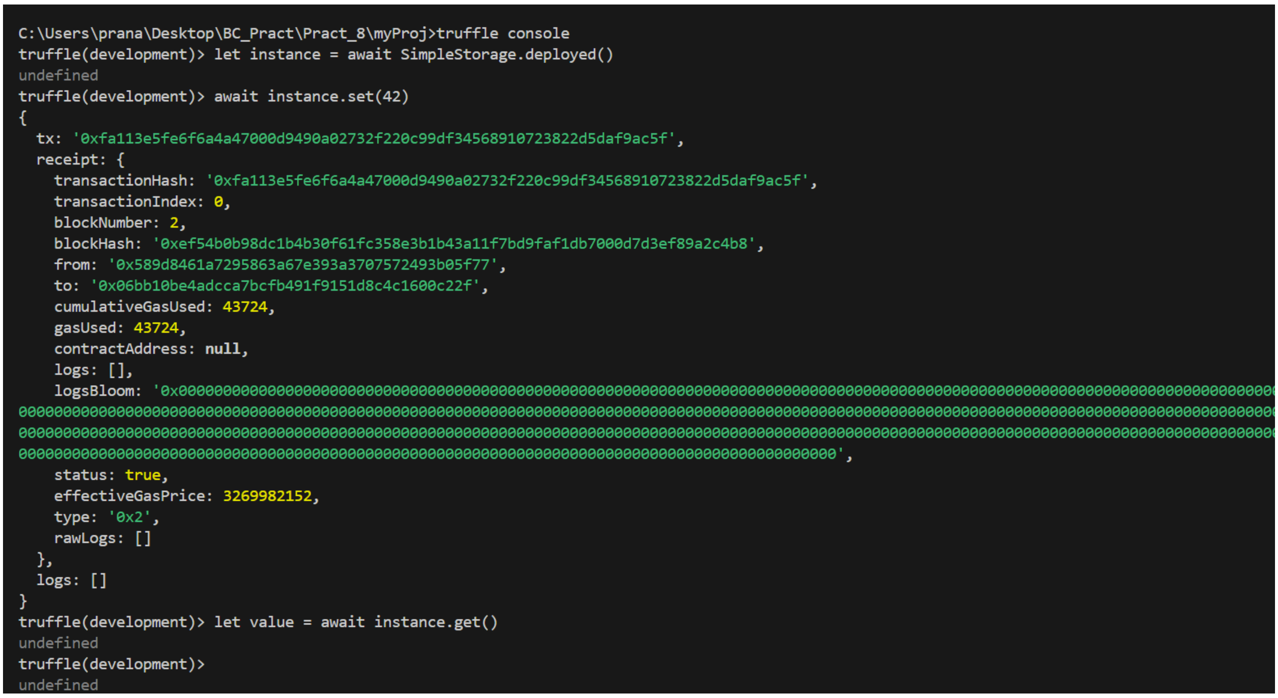
Step 6: Interact with the Deployed Contract

1) **Open the new command prompt:**
   Command: truffle console

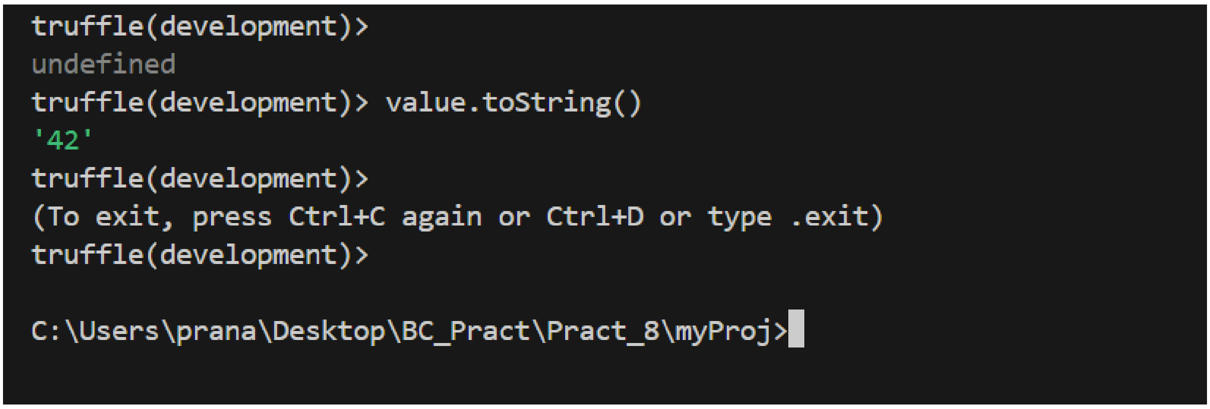
   C:\Users\prana\Desktop\BC_Pract\Pract_8\myProj>truffle console

2) **Interact with the deployed contract:**
   Execute the following commands one-by-one

   let instance = await SimpleStorage.deployed()
   await instance.set(42)
   let value = await instance.get()
   value.toString() // Output should be '42'



```
C:\Users\prana\Desktop\BC_Pract\Pract_8\myProj>truffle console
truffle(development)> let instance = await SimpleStorage.deployed()
undefined
truffle(development)> await instance.set(42)
{
  tx: '0xfa113e5fe6f6a4a47000d9490a02732f220c99df34568910723822d5daf9ac5f',
  receipt: {
    transactionHash: '0xfa113e5fe6f6a4a47000d9490a02732f220c99df34568910723822d5daf9ac5f',
    transactionIndex: 0,
    blockNumber: 2,
    blockHash: '0xef54b0b98dc1b4b30f61fc358e3b1b43a11f7bd9faf1db7000d7d3ef89a2c4b8',
    from: '0x589d8461a7295863a67e393a3707572493b05f77',
    to: '0x06bb10be4adcca7bcfb491f9151d8c4c1600c22f',
    cumulativeGasUsed: 43724,
    gasUsed: 43724,
    contractAddress: null,
    logs: [],
    logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000',
    status: true,
    effectiveGasPrice: 3269982152,
    type: '0x2',
    rawLogs: []
  },
  logs: []
}
truffle(development)> let value = await instance.get()
undefined
truffle(development)>
undefined
```



```
truffle(development)>
undefined
truffle(development)> value.toString()
'42'
truffle(development)>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
truffle(development)>

C:\Users\prana\Desktop\BC_Pract\Pract_8\myProj>
```

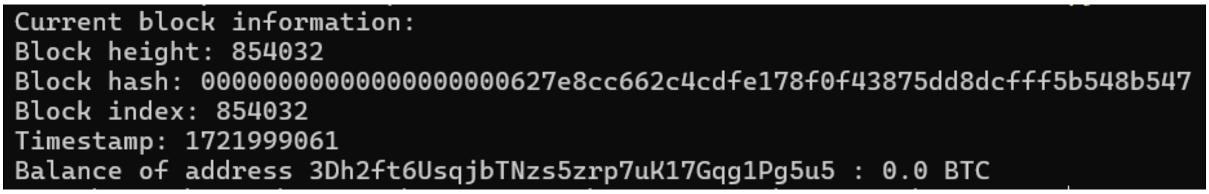
Practical 9

Aim: Demonstrate the use of Bitcoin API.

Code:

```
import requests

# Task 1: Get information regarding the current block
def get_current_block_info():
    response = requests.get("https://blockchain.info/latestblock")
    block_info = response.json()
    print("Current block information:")
    print("Block height:", block_info['height'])
    print("Block hash:", block_info['hash'])
    print("Block index:", block_info['block_index'])
    print("Timestamp:", block_info['time'])

# Task 3: Get balance of an address
def get_address_balance(address):
    response = requests.get(f"https://blockchain.info/q/addressbalance/{address}")
    balance = float(response.text) / 10**8
    print("Balance of address", address, ":", balance, "BTC")

# Example usage
if __name__ == "__main__":
    # Task 1: Get information regarding the current block
    get_current_block_info()

    # Task 3: Get balance of an address
    address = "3Dh2ft6UsqjbTNzs5zrp7uK17Gqg1Pg5u5"
    get_address_balance(address)
```

Output:



```
Current block information:
Block height: 854032
Block hash: 00000000000000000000627e8cc662c4cdfe178f0f43875dd8dcfff5b548b547
Block index: 854032
Timestamp: 1721999061
Balance of address 3Dh2ft6UsqjbTNzs5zrp7uK17Gqg1Pg5u5 : 0.0 BTC
```