BLOCKCHAIN

A PRACTICAL REPORT

ON

BLOCKCHAIN

SUBMITTED BY
Mr. **SAYED FARHAN SHAHID ALI**
Roll No: **IT21015**

UNDER THE GUIDANCE OF
PROF. **DINAAZ SHAIKH**

Submitted in fulfillment of the requirements for qualifying
MSc. IT Part II Semester - IV Examination 2022-2023

University of Mumbai
Department of Information Technology

R.D. & S.H National College of Arts, Commerce & S.W.A.
Science College Bandra (West), Mumbai – 400 050

R. D. & S. H. National & S. W. A. Science College

Bandra (W), Mumbai – 400050.

Department of Information Technology
M.Sc. (IT – SEMESTER IV)

Certificate

*This is to certify that **Blockchain Practicals** performed at R.D & S.H National & S.W.A. Science College by Mr. **Sayed Farhan** holding Seat No. _______ studying Master of Science in Information Technology Semester – IV has been satisfactorily completed as prescribed by the University of Mumbai, during the year 2022 – 2023.*

Subject In-Charge **Coordinator In-Charge** **External Examiner**

College Stamp

INDEX

Practical No 1

Aim: - Write the following programs for Blockchain in Python

Practical No 1

Aim: - Write the following programs for Blockchain in Python

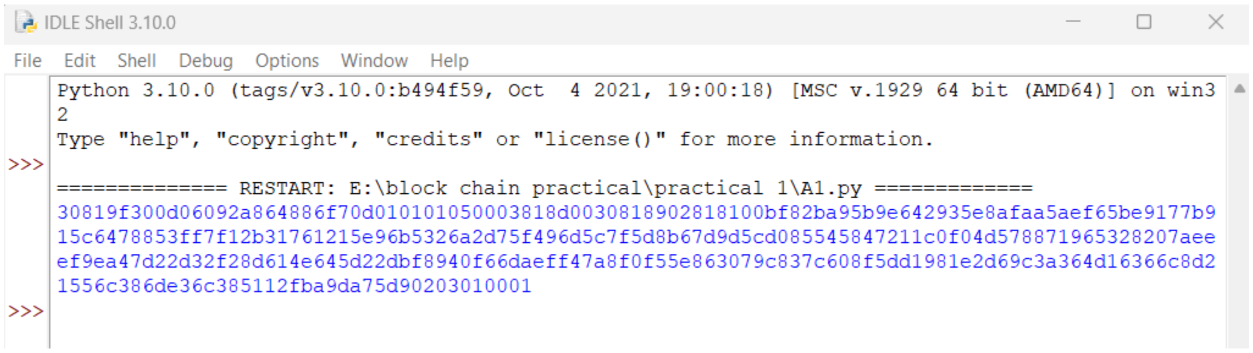
a. A simple client class that generates the private and public keys by using the built-in Python RSA algorithm and test it.

Code:

```
# import libraries
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections

# following imports are required by PKI
import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.PublicKey import RSA
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')

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

Output:



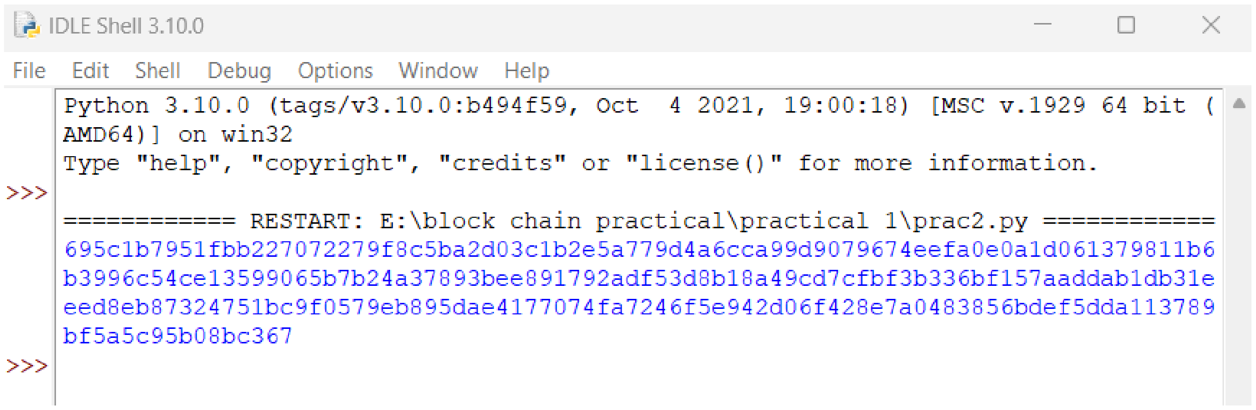
```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win3
2
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: E:\block chain practical\practical 1\A1.py =============
30819f300d06092a864886f70d010101050003818d0030818902818100bf82ba95b9e642935e8afaa5aef65be9177b9
15c6478853ff7f12b31761215e96b5326a2d75f496d5c7f5d8b67d9d5cd085545847211c0f04d578871965328207aee
ef9ea47d22d32f28d614e645d22dbf8940f66daeff47a8f0f55e863079c837c608f5dd1981e2d69c3a364d16366c8d2
1556c386de36c385112fba9da75d90203010001
>>>
```

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

Code:

```
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections
import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.PublicKey import RSA
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')

Farhan= Client()
Dealer= Client()

t = Transaction(
   Farhan,
   Dealer.identity,
   5.0
)

signature = t.sign_transaction()
print (signature)
```

Output:



IDLE Shell 3.10.0

```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============ RESTART: E:\block chain practical\practical 1\prac2.py ============
695c1b7951fbb227072279f8c5ba2d03c1b2e5a779d4a6cca99d9079674eefa0e0a1d061379811b6
b3996c54ce13599065b7b24a37893bee891792adf53d8b18a49cd7cfbf3b336bf157aaddab1db31e
eed8eb87324751bc9f0579eb895dae4177074fa7246f5e942d06f428e7a0483856bdef5dda113789
bf5a5c95b08bc367
>>>
```

c. Create multiple transactions and display them.

Code:

```
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections

import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.PublicKey import RSA
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')
```

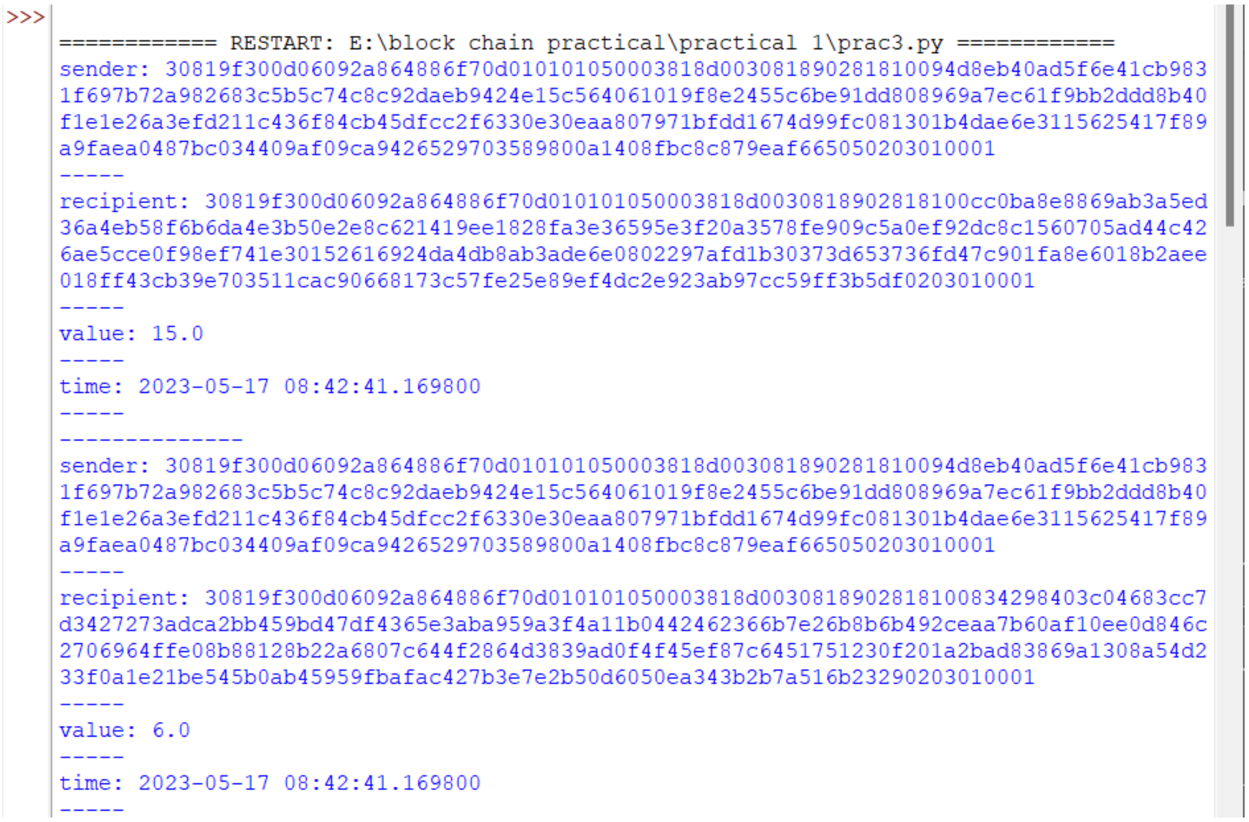
```
class Transaction:
   def__init_(self, sender, recipient, value):
    self.sender = sender
    self.recipient = recipient
    self.value = value
    self.time = datetime.datetime.now()

   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):
   #for transaction in transactions:
   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 ('-----')

transactions = []
Dinesh = Client()
Ramesh = Client()
Seema = Client()
Vijay = Client()

t1 = Transaction(
  Dinesh,
  Ramesh.identity,
  15.0
)
t1.sign_transaction()
transactions.append(t1)

t2 = Transaction(
  Dinesh,
  Seema.identity,
  6.0
)
t2.sign_transaction()
transactions.append(t2)
t3 = Transaction(
  Ramesh,
  Vijay.identity,
  2.0
)
```

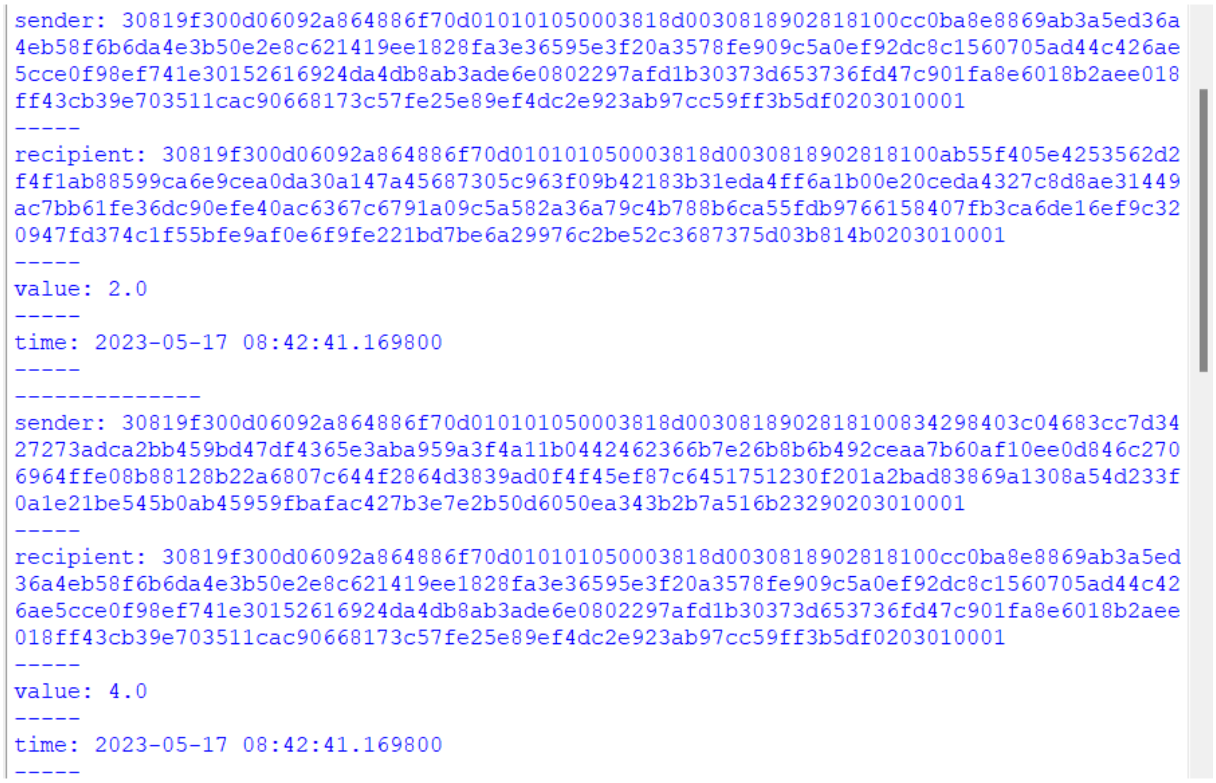
```
t3.sign_transaction()
transactions.append(t3)
t4 = Transaction(
  Seema,
  Ramesh.identity,
  4.0
)
t4.sign_transaction()
transactions.append(t4)
t5 = Transaction(
  Vijay,
  Seema.identity,
  7.0
)
t5.sign_transaction()
transactions.append(t5)
t6 = Transaction(
  Ramesh,
  Seema.identity,
  3.0
)
t6.sign_transaction()
transactions.append(t6)
t7 = Transaction(
  Seema,
  Dinesh.identity,
  8.0
)
t7.sign_transaction()
```

```
transactions.append(t7)
t8 = Transaction(
   Seema,
   Ramesh.identity,
   1.0
)
t8.sign_transaction()
transactions.append(t8)
t9 = Transaction(
   Vijay,
   Dinesh.identity,
   5.0
)
t9.sign_transaction()
transactions.append(t9)
t10 = Transaction(
   Vijay,
   Ramesh.identity,
   3.0
)
t10.sign_transaction()
transactions.append(t10)

for transaction in transactions:
   display_transaction (transaction)
   print ('.....................')
```

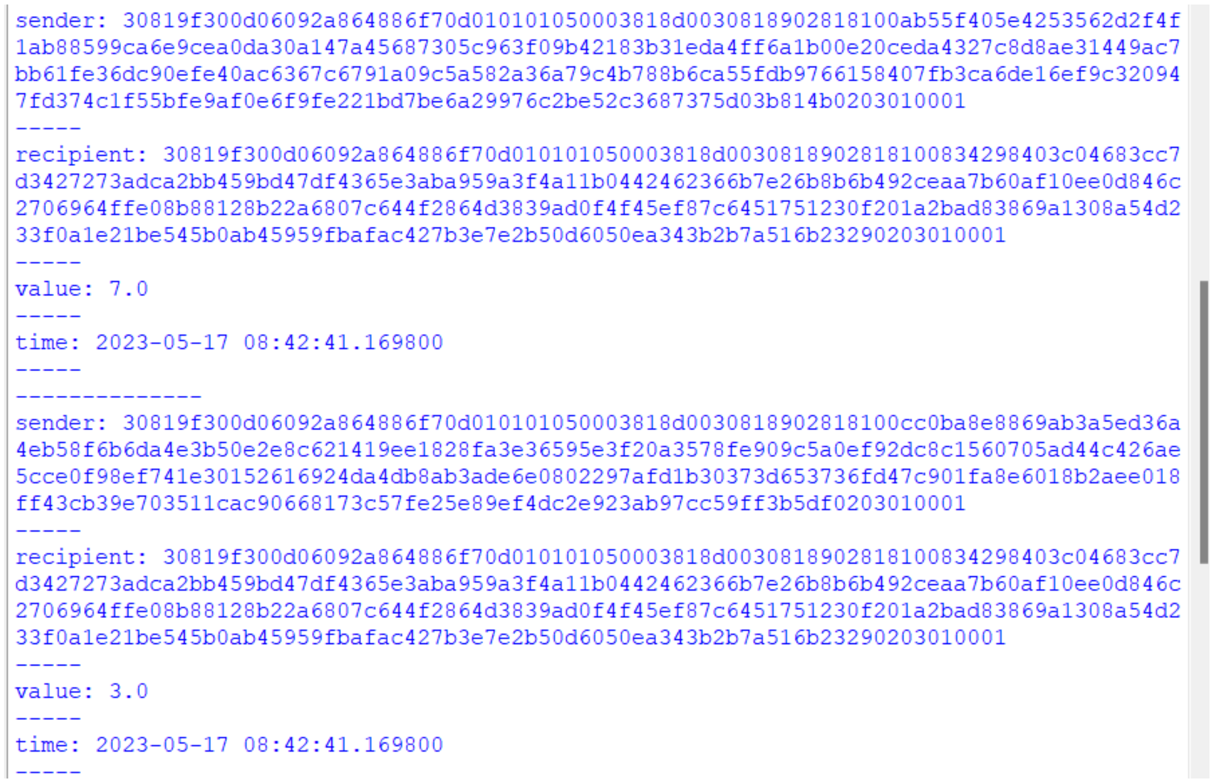
Output:



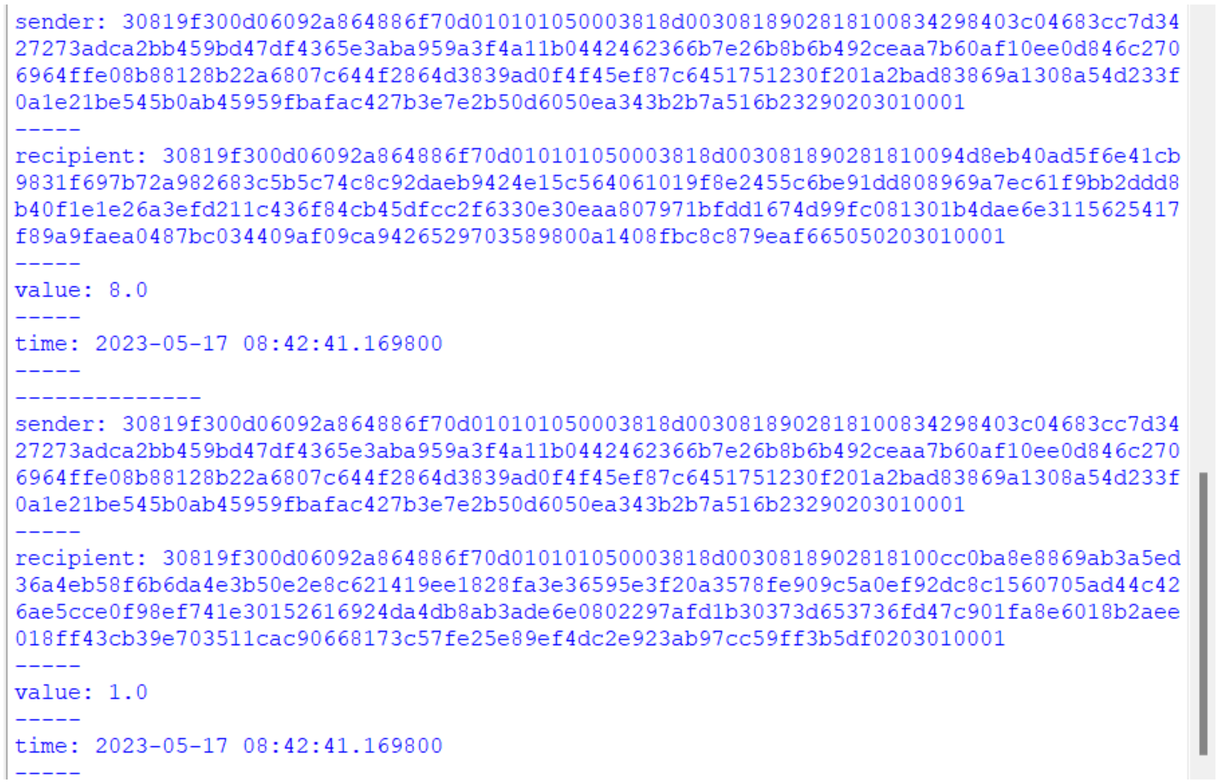
```
>>>
============ RESTART: E:\block chain practical\practical 1\prac3.py ============
sender: 30819f300d06092a864886f70d010101050003818d003081890281810094d8eb40ad5f6e41cb983
1f697b72a982683c5b5c74c8c92daeb9424e15c564061019f8e2455c6be91dd808969a7ec61f9bb2ddd8b40
f1e1e26a3efd211c436f84cb45dfcc2f6330e30eaa807971bfdd1674d99fc081301b4dae6e3115625417f89
a9faea0487bc034409af09ca9426529703589800a1408fbc8c879eaf665050203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100cc0ba8e8869ab3a5ed
36a4eb58f6b6da4e3b50e2e8c621419ee1828fa3e36595e3f20a3578fe909c5a0ef92dc8c1560705ad44c42
6ae5cce0f98ef741e30152616924da4db8ab3ade6e0802297afd1b30373d653736fd47c901fa8e6018b2aee
018ff43cb39e703511cac90668173c57fe25e89ef4dc2e923ab97cc59ff3b5df0203010001
-----
value: 15.0
-----
time: 2023-05-17 08:42:41.169800
-----
--------------
sender: 30819f300d06092a864886f70d010101050003818d003081890281810094d8eb40ad5f6e41cb983
1f697b72a982683c5b5c74c8c92daeb9424e15c564061019f8e2455c6be91dd808969a7ec61f9bb2ddd8b40
f1e1e26a3efd211c436f84cb45dfcc2f6330e30eaa807971bfdd1674d99fc081301b4dae6e3115625417f89
a9faea0487bc034409af09ca9426529703589800a1408fbc8c879eaf665050203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d00308189028181008342984O3c04683cc7
d3427273adca2bb459bd47df4365e3aba959a3f4a11b0442462366b7e26b8b6b492ceaa7b60af10ee0d846c
2706964ffe08b88128b22a6807c644f2864d3839ad0f4f45ef87c6451751230f201a2bad83869a1308a54d2
33f0a1e21be545b0ab45959fbafac427b3e7e2b50d6050ea343b2b7a516b23290203010001
-----
value: 6.0
-----
time: 2023-05-17 08:42:41.169800
-----
```



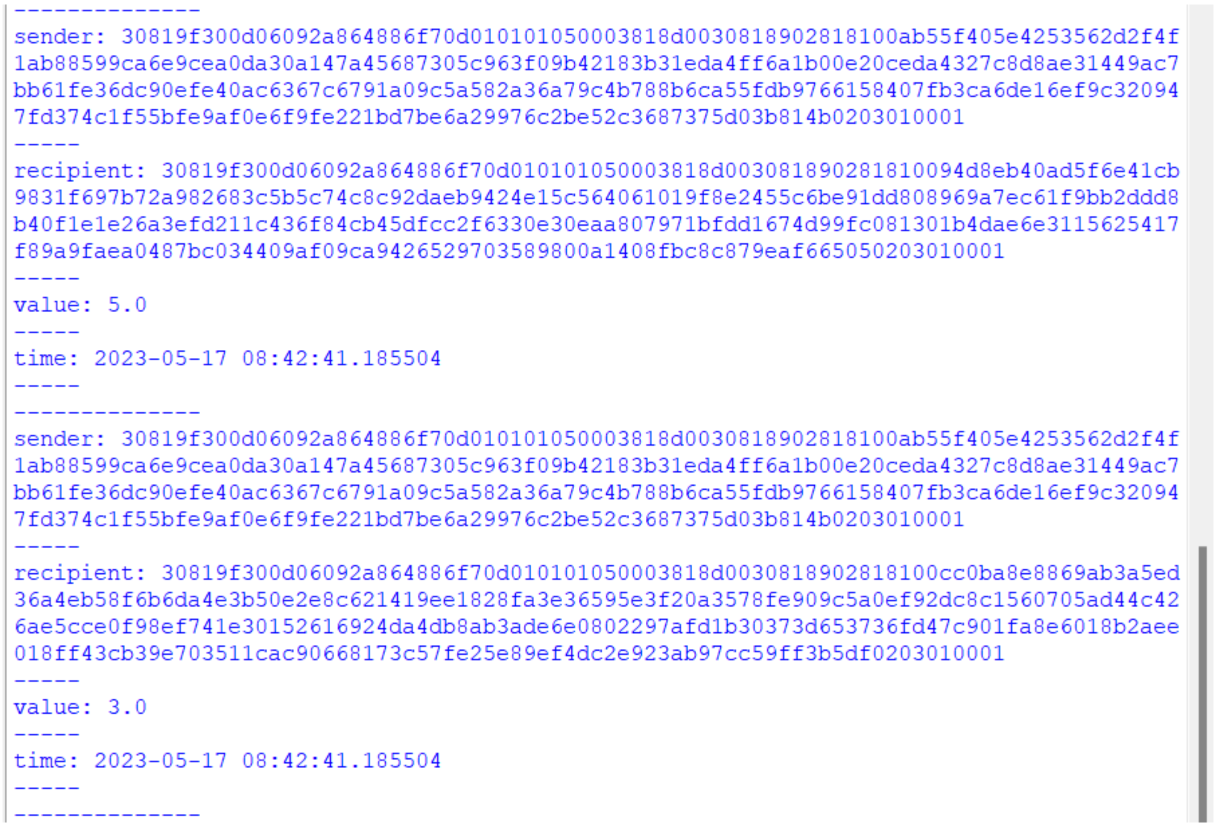
```
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100cc0ba8e8869ab3a5ed36a
4eb58f6b6da4e3b50e2e8c621419ee1828fa3e36595e3f20a3578fe909c5a0ef92dc8c1560705ad44c426ae
5cce0f98ef741e30152616924da4db8ab3ade6e0802297afd1b30373d653736fd47c901fa8e6018b2aee018
ff43cb39e703511cac90668173c57fe25e89ef4dc2e923ab97cc59ff3b5df0203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100ab55f405e4253562d2
f4f1ab88599ca6e9cea0da30a147a45687305c963f09b42183b31eda4ff6a1b00e20ceda4327c8d8ae31449
ac7bb61fe36dc90efe40ac6367c6791a09c5a582a36a79c4b788b6ca55fdb9766158407fb3ca6de16ef9c32
0947fd374c1f55bfe9af0e6f9fe221bd7be6a29976c2be52c3687375d03b814b0203010001
-----
value: 2.0
-----
time: 2023-05-17 08:42:41.169800
-----
--------------
sender: 30819f300d06092a864886f70d010101050003818d00308189028181008342984O3c04683cc7d34
27273adca2bb459bd47df4365e3aba959a3f4a11b0442462366b7e26b8b6b492ceaa7b60af10ee0d846c270
6964ffe08b88128b22a6807c644f2864d3839ad0f4f45ef87c6451751230f201a2bad83869a1308a54d233f
0a1e21be545b0ab45959fbafac427b3e7e2b50d6050ea343b2b7a516b23290203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100cc0ba8e8869ab3a5ed
36a4eb58f6b6da4e3b50e2e8c621419ee1828fa3e36595e3f20a3578fe909c5a0ef92dc8c1560705ad44c42
6ae5cce0f98ef741e30152616924da4db8ab3ade6e0802297afd1b30373d653736fd47c901fa8e6018b2aee
018ff43cb39e703511cac90668173c57fe25e89ef4dc2e923ab97cc59ff3b5df0203010001
-----
value: 4.0
-----
time: 2023-05-17 08:42:41.169800
-----
```



```
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100ab55f405e4253562d2f4f
1ab88599ca6e9cea0da30a147a45687305c963f09b42183b31eda4ff6a1b00e20ceda4327c8d8ae31449ac7
bb61fe36dc90efe40ac6367c6791a09c5a582a36a79c4b788b6ca55fdb9766158407fb3ca6de16ef9c32094
7fd374c1f55bfe9af0e6f9fe221bd7be6a29976c2be52c3687375d03b814b0203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d00308189028181008342984O3c04683cc7
d3427273adca2bb459bd47df4365e3aba959a3f4a11b0442462366b7e26b8b6b492ceaa7b60af10ee0d846c
2706964ffe08b88128b22a6807c644f2864d3839ad0f4f45ef87c6451751230f201a2bad83869a1308a54d2
33f0a1e21be545b0ab45959fbafac427b3e7e2b50d6050ea343b2b7a516b23290203010001
-----
value: 7.0
-----
time: 2023-05-17 08:42:41.169800
-----
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100cc0ba8e8869ab3a5ed36a
4eb58f6b6da4e3b50e2e8c621419ee1828fa3e36595e3f20a3578fe909c5a0ef92dc8c1560705ad44c426ae
5cce0f98ef741e30152616924da4db8ab3ade6e0802297afd1b30373d653736fd47c901fa8e6018b2aee018
ff43cb39e703511cac90668173c57fe25e89ef4dc2e923ab97cc59ff3b5df0203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100834298403c04683cc7
d3427273adca2bb459bd47df4365e3aba959a3f4a11b0442462366b7e26b8b6b492ceaa7b60af10ee0d846c
2706964ffe08b88128b22a6807c644f2864d3839ad0f4f45ef87c6451751230f201a2bad83869a1308a54d2
33f0a1e21be545b0ab45959fbafac427b3e7e2b50d6050ea343b2b7a516b23290203010001
-----
value: 3.0
-----
time: 2023-05-17 08:42:41.169800
-----
```



```
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100834298403c04683cc7d34
27273adca2bb459bd47df4365e3aba959a3f4a11b0442462366b7e26b8b6b492ceaa7b60af10ee0d846c270
6964ffe08b88128b22a6807c644f2864d3839ad0f4f45ef87c6451751230f201a2bad83869a1308a54d233f
0a1e21be545b0ab45959fbafac427b3e7e2b50d6050ea343b2b7a516b23290203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d003081890281810094d8eb40ad5f6e41cb
9831f697b72a982683c5b5c74c8c92daeb9424e15c564061019f8e2455c6be91dd808969a7ec61f9bb2ddd8
b40f1e1e26a3efd211c436f84cb45dfcc2f6330e30eaa807971bfdd1674d99fc081301b4dae6e3115625417
f89a9faea0487bc034409af09ca9426529703589800a1408fbc8c879eaf665050203010001
-----
value: 8.0
-----
time: 2023-05-17 08:42:41.169800
-----
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100834298403c04683cc7d34
27273adca2bb459bd47df4365e3aba959a3f4a11b0442462366b7e26b8b6b492ceaa7b60af10ee0d846c270
6964ffe08b88128b22a6807c644f2864d3839ad0f4f45ef87c6451751230f201a2bad83869a1308a54d233f
0a1e21be545b0ab45959fbafac427b3e7e2b50d6050ea343b2b7a516b23290203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100cc0ba8e8869ab3a5ed
36a4eb58f6b6da4e3b50e2e8c621419ee1828fa3e36595e3f20a3578fe909c5a0ef92dc8c1560705ad44c42
6ae5cce0f98ef741e30152616924da4db8ab3ade6e0802297afd1b30373d653736fd47c901fa8e6018b2aee
018ff43cb39e703511cac90668173c57fe25e89ef4dc2e923ab97cc59ff3b5df0203010001
-----
value: 1.0
-----
time: 2023-05-17 08:42:41.169800
-----
```



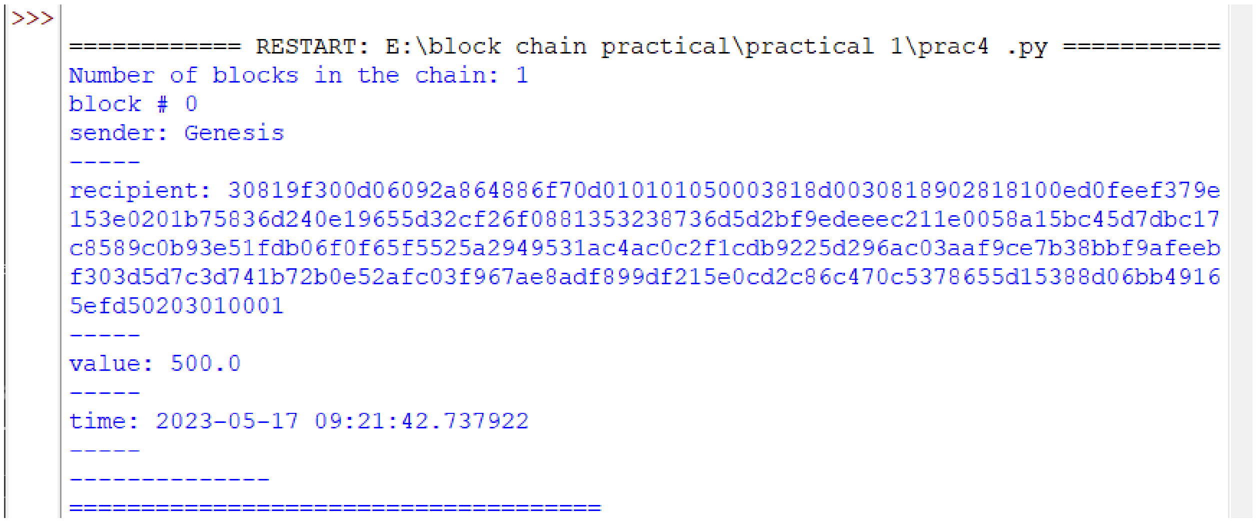
```
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100ab55f405e4253562d2f4f
1ab88599ca6e9cea0da30a147a45687305c963f09b42183b31eda4ff6a1b00e20ceda4327c8d8ae31449ac7
bb61fe36dc90efe40ac6367c6791a09c5a582a36a79c4b788b6ca55fdb9766158407fb3ca6de16ef9c32094
7fd374c1f55bfe9af0e6f9fe221bd7be6a29976c2be52c3687375d03b814b0203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d003081890281810094d8eb40ad5f6e41cb
9831f697b72a982683c5b5c74c8c92daeb9424e15c564061019f8e2455c6be91dd808969a7ec61f9bb2ddd8
b40f1e1e26a3efd211c436f84cb45dfcc2f6330e30eaa807971bfdd1674d99fc081301b4dae6e3115625417
f89a9faea0487bc034409af09ca9426529703589800a1408fbc8c879eaf665050203010001
-----
value: 5.0
-----
time: 2023-05-17 08:42:41.185504
-----
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100ab55f405e4253562d2f4f
1ab88599ca6e9cea0da30a147a45687305c963f09b42183b31eda4ff6a1b00e20ceda4327c8d8ae31449ac7
bb61fe36dc90efe40ac6367c6791a09c5a582a36a79c4b788b6ca55fdb9766158407fb3ca6de16ef9c32094
7fd374c1f55bfe9af0e6f9fe221bd7be6a29976c2be52c3687375d03b814b0203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100cc0ba8e8869ab3a5ed
36a4eb58f6b6da4e3b50e2e8c621419ee1828fa3e36595e3f20a3578fe909c5a0ef92dc8c1560705ad44c42
6ae5cce0f98ef741e30152616924da4db8ab3ade6e0802297afd1b30373d653736fd47c901fa8e6018b2aee
018ff43cb39e703511cac90668173c57fe25e89ef4dc2e923ab97cc59ff3b5df0203010001
-----
value: 3.0
-----
time: 2023-05-17 08:42:41.185504
-----
--------------
```

d. Create a blockchain, a genesis block and execute it.

Code:

```
import hashlib

import random

import string

import json

import binascii

import numpy as np

import pandas as pd

import pylab as pl

import logging

import datetime

import collections


import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.PublicKey import RSA
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):
  #for transaction in transactions:
  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 ('---- ')

class Block:
  def___init_(self):
    self.verified_transactions = []
    self.previous_block_hash = ""
    self.Nonce = ""
    last_block_hash = ""

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 ('=====================================')

Dinesh = Client()

t0 = Transaction (
 "Genesis",
 Dinesh.identity,
 500.0
)

block0 = Block()
block0.previous_block_hash = None
Nonce = None
block0.verified_transactions.append (t0)
digest = hash (block0)
last_block_hash = digest
TPCoins = []
TPCoins.append (block0)
dump_blockchain(TPCoins)
```

Output:



```
>>>
============ RESTART: E:\block chain practical\practical 1\prac4 .py ===========
Number of blocks in the chain: 1
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100ed0feef379e
153e0201b75836d240e19655d32cf26f0881353238736d5d2bf9edeeec211e0058a15bc45d7dbc17
c8589c0b93e51fdb06f0f65f5525a2949531ac4ac0c2f1cdb9225d296ac03aaf9ce7b38bbf9afeeb
f303d5d7c3d741b72b0e52afc03f967ae8adf899df215e0cd2c86c470c5378655d15388d06bb4916
5efd50203010001
-----
value: 500.0
-----
time: 2023-05-17 09:21:42.737922
-----
--------------
=====================================
```

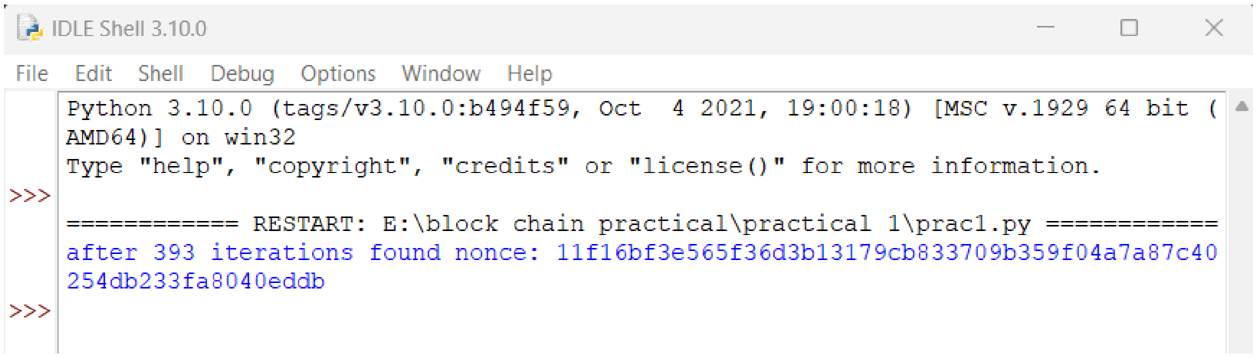
e. Create a mining function and test it.

Code:

```
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections

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

```
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) + " iterations found nonce: " + digest)
            return digest

mine("test message", 2)
```

Output:



```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============ RESTART: E:\block chain practical\practical 1\prac1.py ============
after 393 iterations found nonce: 11f16bf3e565f36d3b13179cb833709b359f04a7a87c40
254db233fa8040eddb
>>>
```

f. Add the block to the miner and dump the blokchain.

Code:

```
# import libraries
import hashlib
import random
import string
import json
import binascii
import numpy as np
import pandas as pd
import pylab as pl
import logging
import datetime
import collections

# following imports are required by PKI
import Crypto
import Crypto.Random
from Crypto.Hash import SHA
from Crypto.PublicKey import RSA
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):
   #for transaction in transactions:
   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 ('----')

transactions = []
Dinesh = Client()
Ramesh = Client()
Seema = Client()
Vijay = Client()

t1 = Transaction(
  Dinesh,
  Ramesh.identity,
  15.0
)
t1.sign_transaction()
transactions.append(t1)

t2 = Transaction(
  Dinesh,
  Seema.identity,
  6.0
)
t2.sign_transaction()
transactions.append(t2)
t3 = Transaction(
  Ramesh,
  Vijay.identity,
  2.0
)
t3.sign_transaction()
```

```
transactions.append(t3)
t4 = Transaction(
    Seema,
    Ramesh.identity,
    4.0
)
t4.sign_transaction()
transactions.append(t4)
t5 = Transaction(
    Vijay,
    Seema.identity,
    7.0
)
t5.sign_transaction()
transactions.append(t5)
t6 = Transaction(
    Ramesh,
    Seema.identity,
    3.0
)
t6.sign_transaction()
transactions.append(t6)
t7 = Transaction(
    Seema,
    Dinesh.identity,
    8.0
)
t7.sign_transaction()
transactions.append(t7)
```

```
t8 = Transaction(
   Seema,
   Ramesh.identity,
   1.0
)
t8.sign_transaction()
transactions.append(t8)
t9 = Transaction(
   Vijay,
   Dinesh.identity,
   5.0
)
t9.sign_transaction()
transactions.append(t9)
t10 = Transaction(
   Vijay,
   Ramesh.identity,
   3.0
)
t10.sign_transaction()
transactions.append(t10)

for transaction in transactions:
   display_transaction (transaction)
   print ('.....................')

class Block:
   def___init_(self):
      self.verified_transactions = []
```

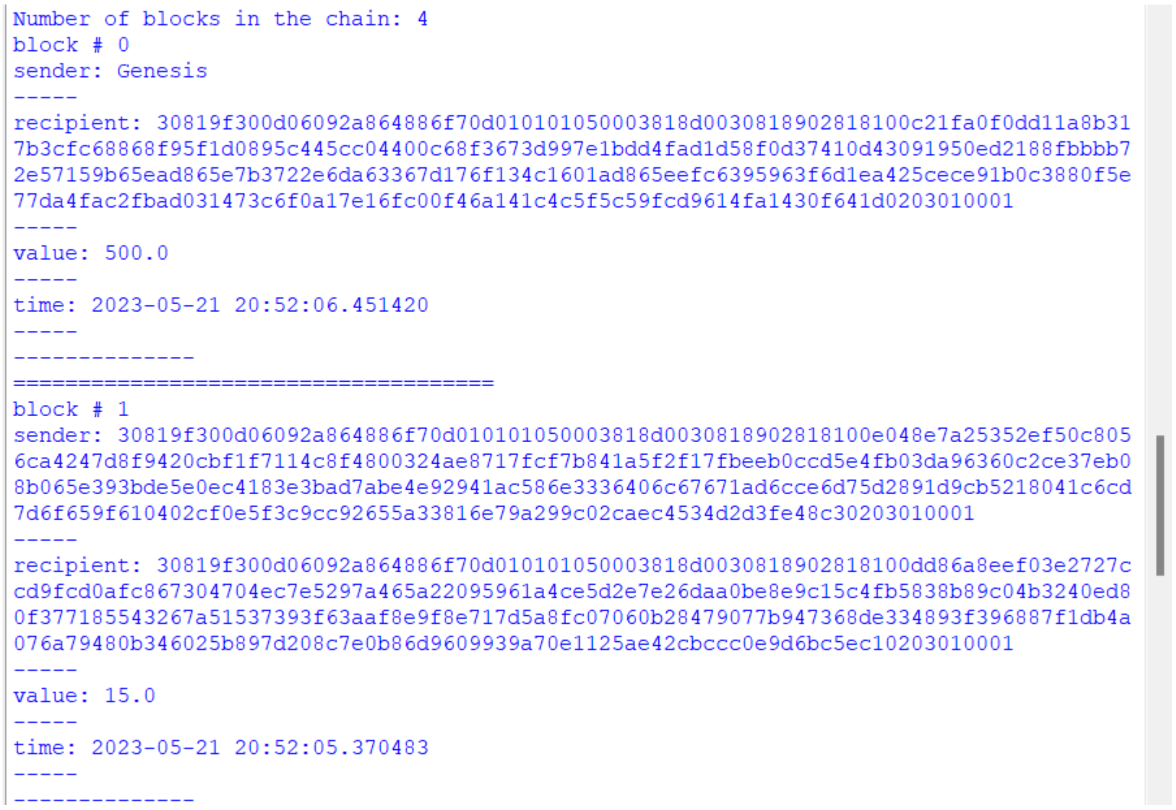
```
    self.previous_block_hash = ""
    self.Nonce = ""
    last_block_hash = ""

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 ('=====================================')

Dinesh = Client()

t0 = Transaction (
 "Genesis",
 Dinesh.identity,
 500.0
)

block0 = Block()
block0.previous_block_hash = None
Nonce = None
block0.verified_transactions.append (t0)
digest = hash (block0)
last_block_hash = digest
TPCoins = []
```

```
TPCoins.append (block0)
dump_blockchain(TPCoins)

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) + " iterations found nonce: " + digest)
            return digest

mine("test message", 2)

last_transaction_index = 0

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
   mine ("test message", 2)
   block.previous_block_hash = last_block_hash
block.Nonce = mine (block, 2)
```

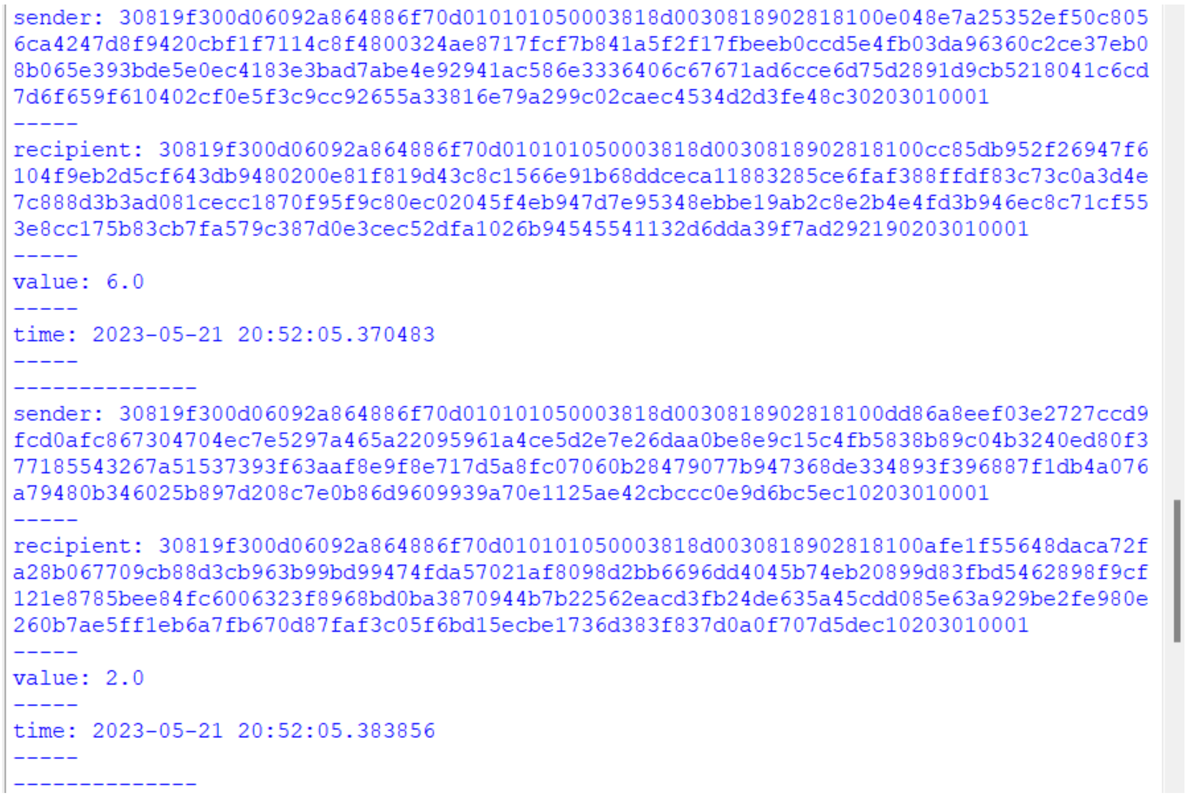
```
digest = hash (block)
TPCoins.append (block)
last_block_hash = digest

# Miner 2 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
# Miner 3 adds a block
block = Block()

for i in range(3):
   temp_transaction = transactions[last_transaction_index]
   #display_transaction (temp_transaction)
   # 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)
```

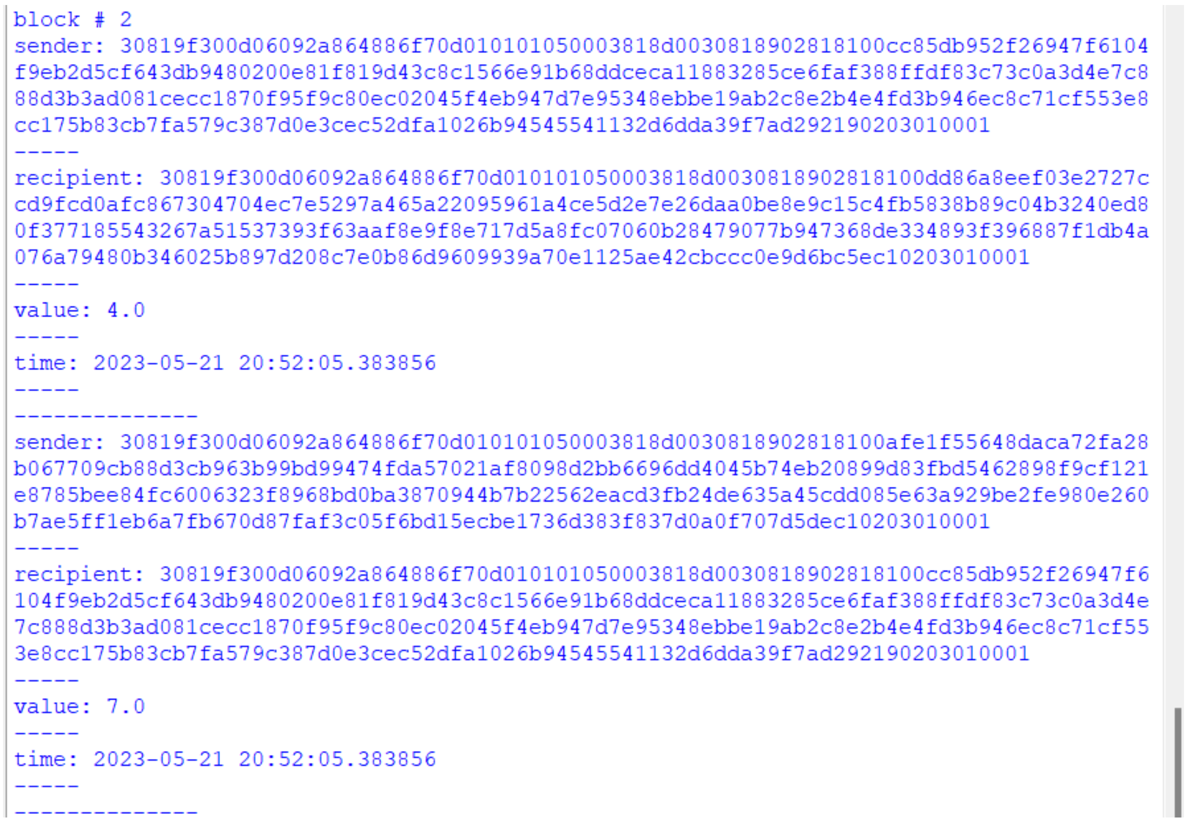
Output:



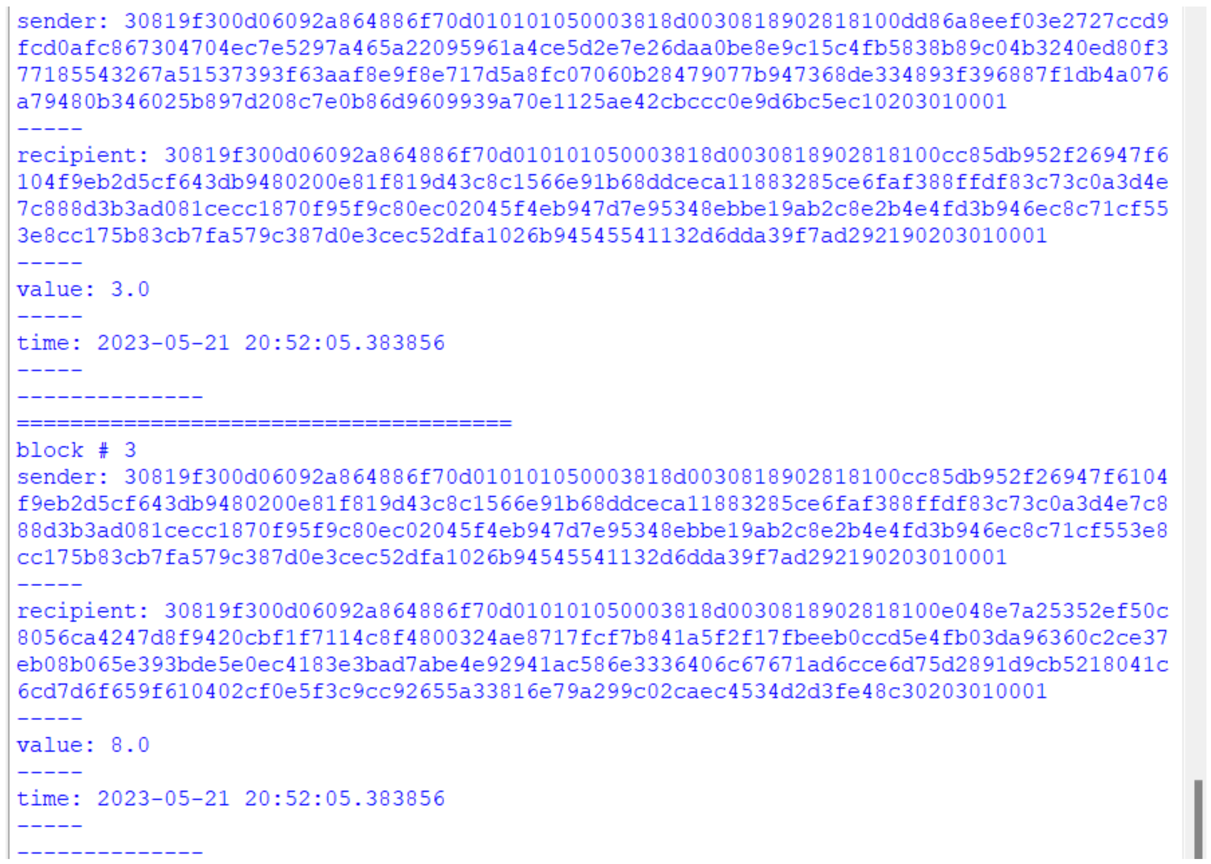
```
Number of blocks in the chain: 4
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100c21fa0f0dd11a8b31
7b3cfc68868f95f1d0895c445cc04400c68f3673d997e1bdd4fad1d58f0d37410d43091950ed2188fbbbb7
2e57159b65ead865e7b3722e6da63367d176f134c1601ad865eefc6395963f6d1ea425cece91b0c3880f5e
77da4fac2fbad031473c6f0a17e16fc00f46a141c4c5f5c59fcd9614fa1430f641d0203010001
-----
value: 500.0
-----
time: 2023-05-21 20:52:06.451420
-----
--------------
=====================================
block # 1
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100e048e7a25352ef50c805
6ca4247d8f9420cbf1f7114c8f4800324ae8717fcf7b841a5f2f17fbeeb0ccd5e4fb03da96360c2ce37eb0
8b065e393bde5e0ec4183e3bad7abe4e92941ac586e3336406c67671ad6cce6d75d2891d9cb5218041c6cd
7d6f659f610402cf0e5f3c9cc92655a33816e79a299c02caec4534d2d3fe48c30203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100dd86a8eef03e2727c
cd9fcd0afc867304704ec7e5297a465a22095961a4ce5d2e7e26daa0be8e9c15c4fb5838b89c04b3240ed8
0f377185543267a51537393f63aaf8e9f8e717d5a8fc07060b28479077b947368de334893f396887f1db4a
076a79480b346025b897d208c7e0b86d9609939a70e1125ae42cbccc0e9d6bc5ec10203010001
-----
value: 15.0
-----
time: 2023-05-21 20:52:05.370483
-----
--------------
```



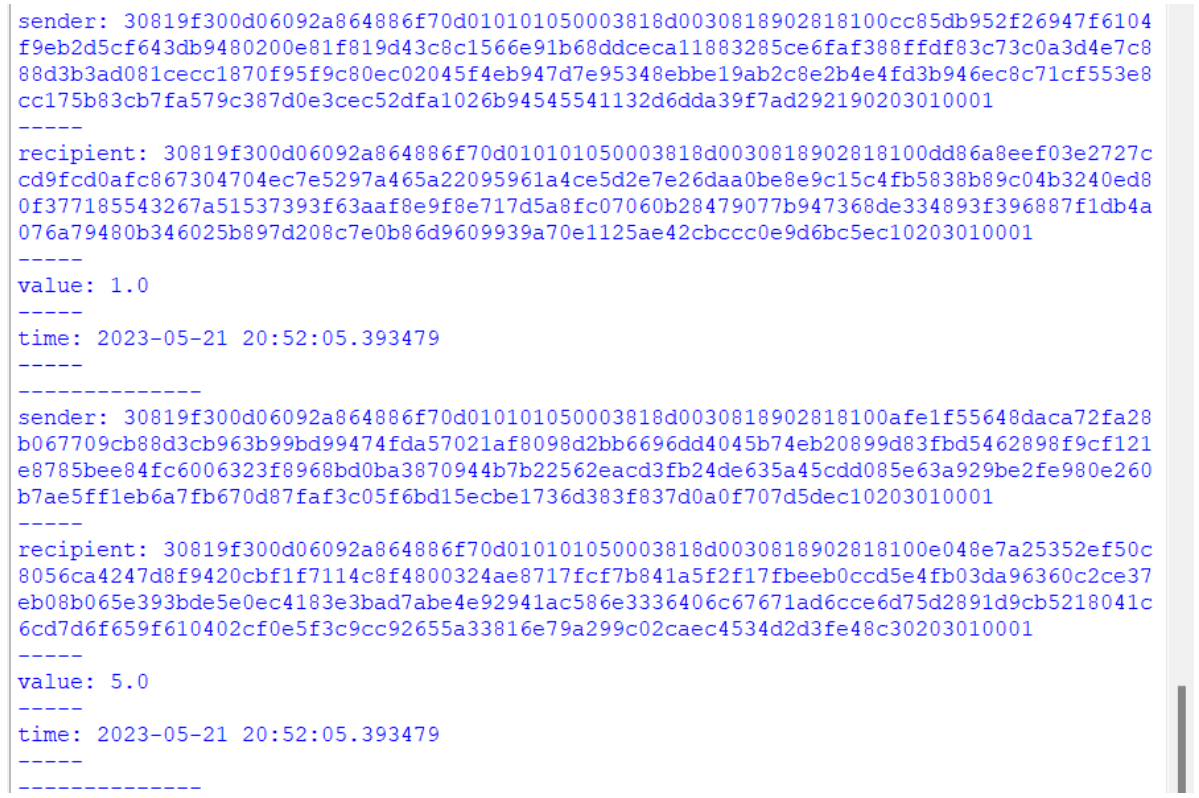
```
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100e048e7a25352ef50c805
6ca4247d8f9420cbf1f7114c8f4800324ae8717fcf7b841a5f2f17fbeeb0ccd5e4fb03da96360c2ce37eb0
8b065e393bde5e0ec4183e3bad7abe4e92941ac586e3336406c67671ad6cce6d75d2891d9cb5218041c6cd
7d6f659f610402cf0e5f3c9cc92655a33816e79a299c02caec4534d2d3fe48c30203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100cc85db952f26947f6
104f9eb2d5cf643db9480200e81f819d43c8c1566e91b68ddceca11883285ce6faf388ffdf83c73c0a3d4e
7c888d3b3ad081cecc1870f95f9c80ec02045f4eb947d7e95348ebbe19ab2c8e2b4e4fd3b946ec8c71cf55
3e8cc175b83cb7fa579c387d0e3cec52dfa1026b94545541132d6dda39f7ad292190203010001
-----
value: 6.0
-----
time: 2023-05-21 20:52:05.370483
-----
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100dd86a8eef03e2727ccd9
fcd0afc867304704ec7e5297a465a22095961a4ce5d2e7e26daa0be8e9c15c4fb5838b89c04b3240ed80f3
77185543267a51537393f63aaf8e9f8e717d5a8fc07060b28479077b947368de334893f396887f1db4a076
a79480b346025b897d208c7e0b86d9609939a70e1125ae42cbccc0e9d6bc5ec10203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100afe1f55648daca72f
a28b067709cb88d3cb963b99bd99474fda57021af8098d2bb6696dd4045b74eb20899d83fbd5462898f9cf
121e8785bee84fc6006323f8968bd0ba3870944b7b22562eacd3fb24de635a45cdd085e63a929be2fe980e
260b7ae5ff1eb6a7fb670d87faf3c05f6bd15ecbe1736d383f837d0a0f707d5dec10203010001
-----
value: 2.0
-----
time: 2023-05-21 20:52:05.383856
-----
--------------
```



```
block # 2
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100cc85db952f26947f6104
f9eb2d5cf643db9480200e81f819d43c8c1566e91b68ddceca11883285ce6faf388ffdf83c73c0a3d4e7c8
88d3b3ad081cecc1870f95f9c80ec02045f4eb947d7e95348ebbe19ab2c8e2b4e4fd3b946ec8c71cf553e8
cc175b83cb7fa579c387d0e3cec52dfa1026b94545541132d6dda39f7ad292190203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100dd86a8eef03e2727c
cd9fcd0afc867304704ec7e5297a465a22095961a4ce5d2e7e26daa0be8e9c15c4fb5838b89c04b3240ed8
0f377185543267a51537393f63aaf8e9f8e717d5a8fc07060b28479077b947368de334893f396887f1db4a
076a79480b346025b897d208c7e0b86d9609939a70e1125ae42cbccc0e9d6bc5ec10203010001
-----
value: 4.0
-----
time: 2023-05-21 20:52:05.383856
-----
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100afe1f55648daca72fa28
b067709cb88d3cb963b99bd99474fda57021af8098d2bb6696dd4045b74eb20899d83fbd5462898f9cf121
e8785bee84fc6006323f8968bd0ba3870944b7b22562eacd3fb24de635a45cdd085e63a929be2fe980e260
b7ae5ff1eb6a7fb670d87faf3c05f6bd15ecbe1736d383f837d0a0f707d5dec10203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100cc85db952f26947f6
104f9eb2d5cf643db9480200e81f819d43c8c1566e91b68ddceca11883285ce6faf388ffdf83c73c0a3d4e
7c888d3b3ad081cecc1870f95f9c80ec02045f4eb947d7e95348ebbe19ab2c8e2b4e4fd3b946ec8c71cf55
3e8cc175b83cb7fa579c387d0e3cec52dfa1026b94545541132d6dda39f7ad292190203010001
-----
value: 7.0
-----
time: 2023-05-21 20:52:05.383856
-----
--------------
```



```
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100dd86a8eef03e2727ccd9
fcd0afc867304704ec7e5297a465a22095961a4ce5d2e7e26daa0be8e9c15c4fb5838b89c04b3240ed80f3
77185543267a51537393f63aaf8e9f8e717d5a8fc07060b28479077b947368de334893f396887f1db4a076
a79480b346025b897d208c7e0b86d9609939a70e1125ae42cbccc0e9d6bc5ec10203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100cc85db952f26947f6
104f9eb2d5cf643db9480200e81f819d43c8c1566e91b68ddceca11883285ce6faf388ffdf83c73c0a3d4e
7c888d3b3ad081cecc1870f95f9c80ec02045f4eb947d7e95348ebbe19ab2c8e2b4e4fd3b946ec8c71cf55
3e8cc175b83cb7fa579c387d0e3cec52dfa1026b94545541132d6dda39f7ad292190203010001
-----
value: 3.0
-----
time: 2023-05-21 20:52:05.383856
-----
--------------
=====================================
block # 3
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100cc85db952f26947f6104
f9eb2d5cf643db9480200e81f819d43c8c1566e91b68ddceca11883285ce6faf388ffdf83c73c0a3d4e7c8
88d3b3ad081cecc1870f95f9c80ec02045f4eb947d7e95348ebbe19ab2c8e2b4e4fd3b946ec8c71cf553e8
cc175b83cb7fa579c387d0e3cec52dfa1026b94545541132d6dda39f7ad292190203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100e048e7a25352ef50c
8056ca4247d8f9420cbf1f7114c8f4800324ae8717fcf7b841a5f2f17fbeeb0ccd5e4fb03da96360c2ce37
eb08b065e393bde5e0ec4183e3bad7abe4e92941ac586e3336406c67671ad6cce6d75d2891d9cb5218041c
6cd7d6f659f610402cf0e5f3c9cc92655a33816e79a299c02caec4534d2d3fe48c30203010001
-----
value: 8.0
-----
time: 2023-05-21 20:52:05.383856
-----
--------------
```



```
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100cc85db952f26947f6104
f9eb2d5cf643db9480200e81f819d43c8c1566e91b68ddceca11883285ce6faf388ffdf83c73c0a3d4e7c8
88d3b3ad081cecc1870f95f9c80ec02045f4eb947d7e95348ebbe19ab2c8e2b4e4fd3b946ec8c71cf553e8
cc175b83cb7fa579c387d0e3cec52dfa1026b94545541132d6dda39f7ad292190203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100dd86a8eef03e2727c
cd9fcd0afc867304704ec7e5297a465a22095961a4ce5d2e7e26daa0be8e9c15c4fb5838b89c04b3240ed8
0f377185543267a51537393f63aaf8e9f8e717d5a8fc07060b28479077b947368de334893f396887f1db4a
076a79480b346025b897d208c7e0b86d9609939a70e1125ae42cbccc0e9d6bc5ec10203010001
-----
value: 1.0
-----
time: 2023-05-21 20:52:05.393479
-----
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100afe1f55648daca72fa28
b067709cb88d3cb963b99bd99474fda57021af8098d2bb6696dd4045b74eb20899d83fbd5462898f9cf121
e8785bee84fc6006323f8968bd0ba3870944b7b22562eacd3fb24de635a45cdd085e63a929be2fe980e260
b7ae5ff1eb6a7fb670d87faf3c05f6bd15ecbe1736d383f837d0a0f707d5dec10203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100e048e7a25352ef50c
8056ca4247d8f9420cbf1f7114c8f4800324ae8717fcf7b841a5f2f17fbeeb0ccd5e4fb03da96360c2ce37
eb08b065e393bde5e0ec4183e3bad7abe4e92941ac586e3336406c67671ad6cce6d75d2891d9cb5218041c
6cd7d6f659f610402cf0e5f3c9cc92655a33816e79a299c02caec4534d2d3fe48c30203010001
-----
value: 5.0
-----
time: 2023-05-21 20:52:05.393479
-----
--------------
```

Practical No 2

Aim: - Install and configure Go Ethereum and the Mist browser. Develop and test a sample application.

Practical No 2

Aim: - Install and configure Go Ethereum and the Mist browser. Develop and test a sample application.

Steps

Installing GETH (Go Ethereum)

Step 1: Go to website https://geth.ethereum.org/downloads/

Step 2: From stable releases Geth 1.5.8 (kind = installer)

Step 3: once downloaded run it then click next

Step 4: Select Geth and Development tools click next

Step 5: Select location to install click next

Step 6: Once Installation is finished Click Close and its done

Installing Mist Browser

Step 1: https://github.com/ethereum/mist/releases

Step 2: Under Ethereum Wallet and Mist 0.8.9 - "The Wizard" download mist-installer-0-8-9.exe

Step 3: For installation click, I agree -> next -> install

Run Mist

Step 1: Open the Mist from the start menu

Step 2: It will start downloading Blockchain data once you open it

Step 3: Once it finishes downloading it is ready to use

Run Geth

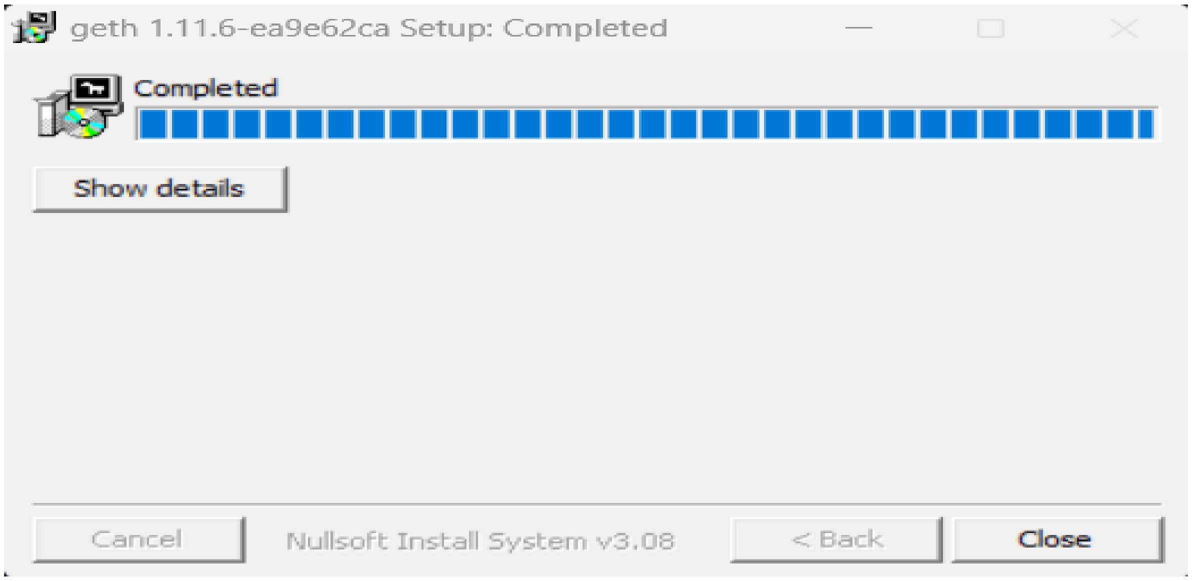
Step 1: Open CMD

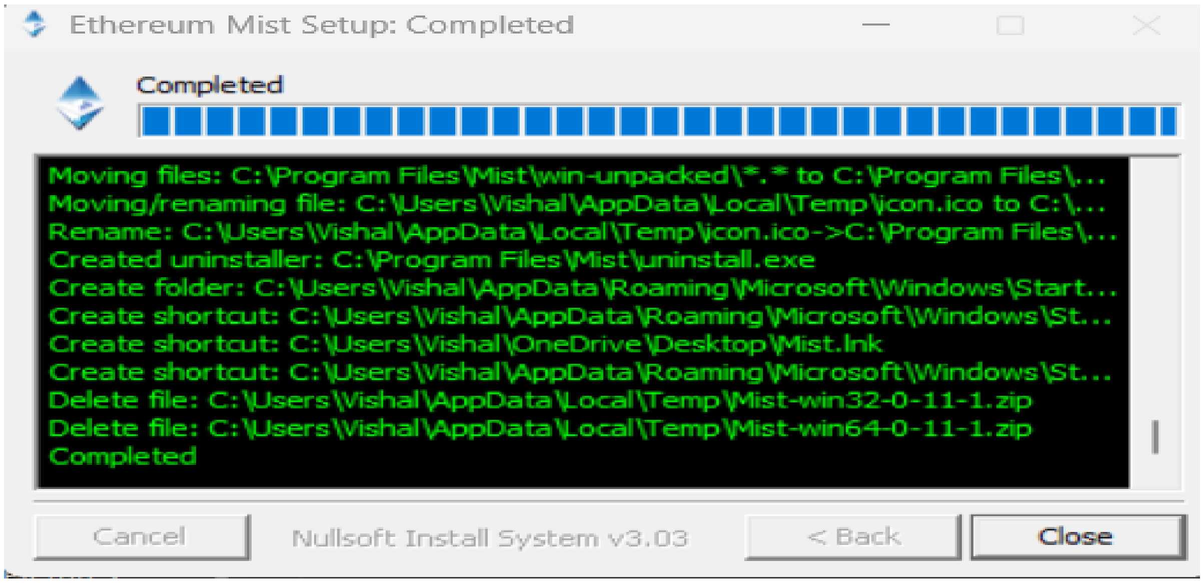
Step 2: Type GETH and press enter

Step 3: After it finishes loading press ctrl+c to exit the process.

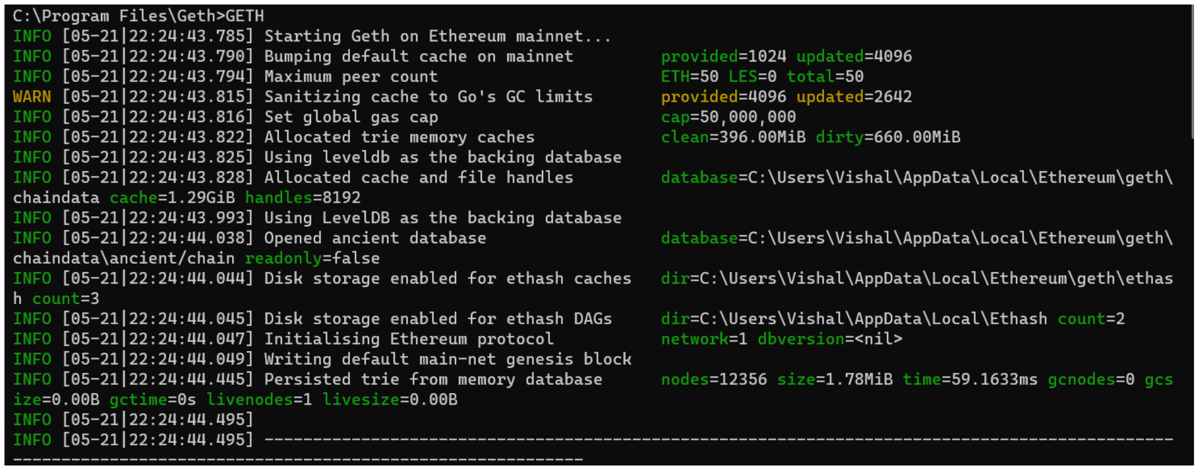
Step 4: Now it's ready to use

Output:


geth 1.11.6-ea9e62ca Setup: Completed
Completed
Show details
Cancel
Nullsoft Install System v3.08
< Back
Close



```
Moving files: C:\Program Files\Mist\win-unpacked\*.* to C:\Program Files\...
Moving/renaming file: C:\Users\Vishal\AppData\Local\Temp\icon.ico to C:\...
Rename: C:\Users\Vishal\AppData\Local\Temp\icon.ico->C:\Program Files\...
Created uninstaller: C:\Program Files\Mist\uninstall.exe
Create folder: C:\Users\Vishal\AppData\Roaming\Microsoft\Windows\Start...
Create shortcut: C:\Users\Vishal\AppData\Roaming\Microsoft\Windows\St...
Create shortcut: C:\Users\Vishal\OneDrive\Desktop\Mist.lnk
Create shortcut: C:\Users\Vishal\AppData\Roaming\Microsoft\Windows\St...
Delete file: C:\Users\Vishal\AppData\Local\Temp\Mist-win32-0-11-1.zip
Delete file: C:\Users\Vishal\AppData\Local\Temp\Mist-win64-0-11-1.zip
Completed
```



```
C:\Program Files\Geth>GETH
INFO [05-21|22:24:43.785] Starting Geth on Ethereum mainnet...
INFO [05-21|22:24:43.790] Bumping default cache on mainnet         provided=1024 updated=4096
INFO [05-21|22:24:43.794] Maximum peer count                       ETH=50 LES=0 total=50
WARN [05-21|22:24:43.815] Sanitizing cache to Go's GC limits       provided=4096 updated=2642
INFO [05-21|22:24:43.816] Set global gas cap                       cap=50,000,000
INFO [05-21|22:24:43.822] Allocated trie memory caches             clean=396.00MiB dirty=660.00MiB
INFO [05-21|22:24:43.825] Using leveldb as the backing database
INFO [05-21|22:24:43.828] Allocated cache and file handles         database=C:\Users\Vishal\AppData\Local\Ethereum\geth\
chaindata cache=1.29GiB handles=8192
INFO [05-21|22:24:43.993] Using LevelDB as the backing database
INFO [05-21|22:24:44.038] Opened ancient database                  database=C:\Users\Vishal\AppData\Local\Ethereum\geth\
chaindata\ancient/chain readonly=false
INFO [05-21|22:24:44.044] Disk storage enabled for ethash caches   dir=C:\Users\Vishal\AppData\Local\Ethereum\geth\ethas
h count=3
INFO [05-21|22:24:44.045] Disk storage enabled for ethash DAGs     dir=C:\Users\Vishal\AppData\Local\Ethash count=2
INFO [05-21|22:24:44.047] Initialising Ethereum protocol           network=1 dbversion=<nil>
INFO [05-21|22:24:44.049] Writing default main-net genesis block
INFO [05-21|22:24:44.445] Persisted trie from memory database      nodes=12356 size=1.78MiB time=59.1633ms gcnodes=0 gcs
ize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-21|22:24:44.495]
INFO [05-21|22:24:44.495] ----------------------------------------------------------------------------------------------
-----------------------------------------------------------
```

Practical No 3

Aim: - Implement and demonstrate the use of the following in Solidity

Practical No 3

Aim: - Implement and demonstrate the use of the following in Solidity

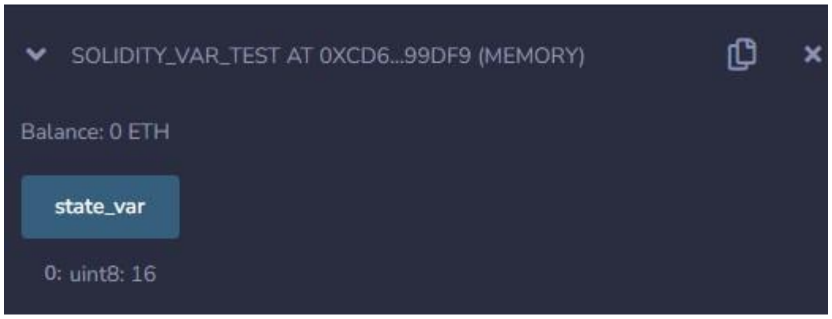
a. Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs, Mappings, Conversions, Ether Units, Special Variables.

Variable:

Code:

```
    // Solidity program to demonstrate state variables
    pragma solidity ^0.5.0;

    // Creating a contract
    contract Solidity_var_Test
    { // Declaring a state
    variable uint8 public
    state_var; // Defining a
    constructor constructor()
    public {
        state_var = 16;
        }
}
```

Output:



```
SOLIDITY_VAR_TEST AT 0XCD6...99DF9 (MEMORY)
Balance: 0 ETH
state_var
0: uint8: 16
```

Operators

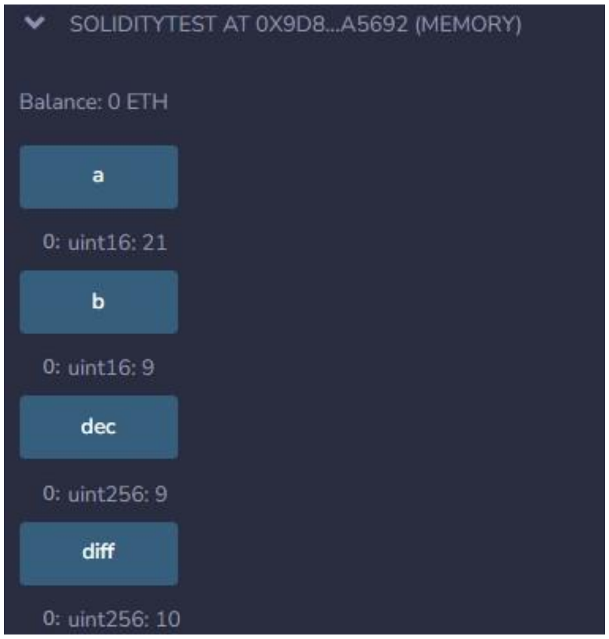
a. Arithmetic Operator:

Code:

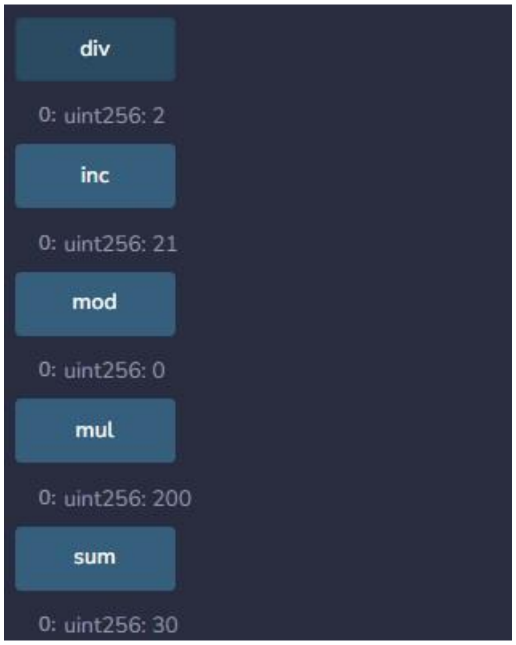
```
// Solidity contract to demonstrate
// Arithmetic Operator
pragma solidity ^0.5.0;

// Creating a contract
contract SolidityTest {

   // Initializing variables
   uint16 public a = 20;
   uint16 public b = 10;

   // Initializing a variable
   // with sum
   uint public sum = a + b;

   // Initializing a variable
   // with the difference
   uint public diff = a - b;

   // Initializing a variable
   // with product
   uint public mul = a * b;

   // Initializing a variable
   // with quotient
   uint public div = a / b;
   // Initializing a variable
```

```
    // with modulus
    uint public mod = a % b;

    // Initializing a variable
    // decrement value
    uint public dec = --b;

    // Initializing a variable
    // with increment value
    uint public inc = ++a;

}
```

Output:



```
SOLIDITYTEST AT 0X9D8...A5692 (MEMORY)

Balance: 0 ETH
a
0: uint16: 21
b
0: uint16: 9
dec
0: uint256: 9
diff
0: uint256: 10
```



```
div
0: uint256: 2
inc
0: uint256: 21
mod
0: uint256: 0
mul
0: uint256: 200
sum
0: uint256: 30
```

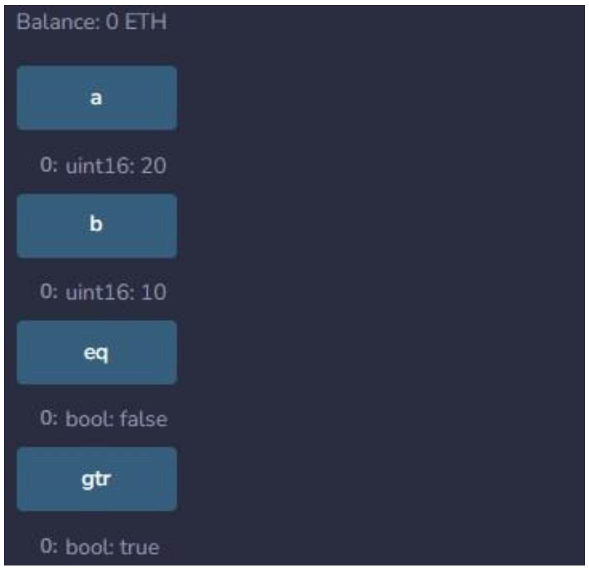
b. Relational Operator:

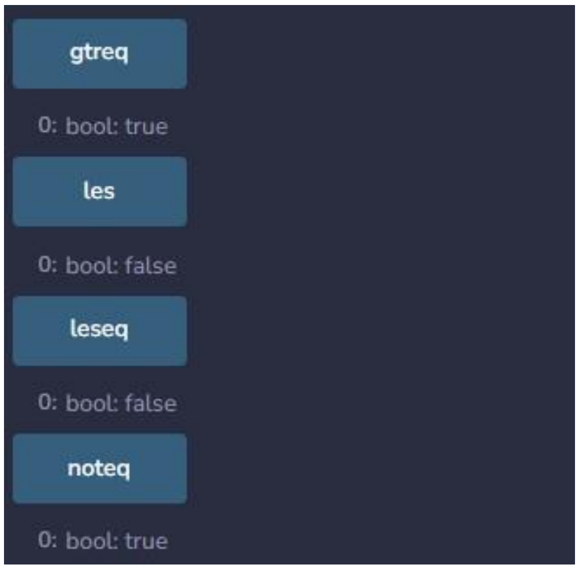
Code:

```
// Solidity program to demonstrate
// Relational Operator
pragma solidity ^0.5.0;

// Creating a contract
contract SolidityTest {

   // Declaring variables
   uint16 public a = 20;
   uint16 public b = 10;

   // Initializing a variable
   // with bool equal result
   bool public eq = a == b;

   // Initializing a variable
   // with bool not equal result
   bool public noteq = a != b;


   // Initializing a variable
   // with bool greater than result
   bool public gtr = a > b;

   // Initializing a variable
   // with bool less than result
   bool public les = a < b;


   // Initializing a variable
   // with bool greater than equal to result
   bool public gtreq = a >= b;
```

```
    // Initializing a variable
    // bool less than equal to result
    bool public leseq = a <= b;


}
```

Output:


```
Balance: 0 ETH
a
0: uint16: 20
b
0: uint16: 10
eq
0: bool: false
gtr
0: bool: true
```

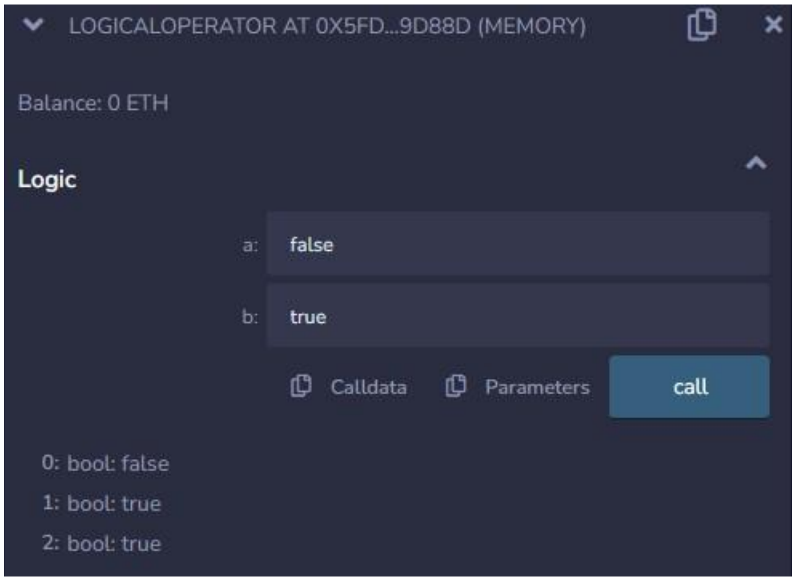

```
gtreq
0: bool: true
les
0: bool: false
leseq
0: bool: false
noteq
0: bool: true
```

c. Logical Operator:

Code:

```
// Solidity program to demonstrate
// Logical Operators
pragma solidity ^0.5.0;

// Creating a contract
contract logicalOperator{

        // Defining function to demonstrate
        // Logical operator
        function Logic(
        bool a, bool b) public view returns(
        bool, bool, bool){

        // Logical AND operator
        bool and = a&&b;

        // Logical OR operator
        bool or = a||b;


        // Logical NOT operator
        bool not = !a;
        return (and, or, not);
}
}
```

Output:



```
LOGICALOPERATOR AT 0X5FD...9D88D (MEMORY)
Balance: 0 ETH
Logic
a: false
b: true
Calldata  Parameters  call
0: bool: false
1: bool: true
2: bool: true
```

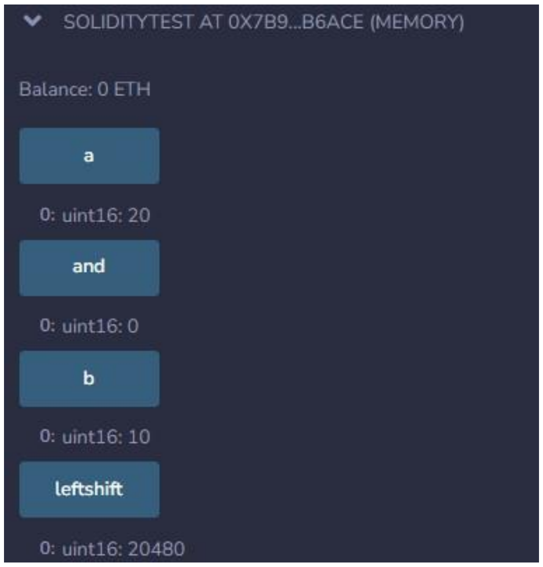
d. Bitwise Operator:

Code:

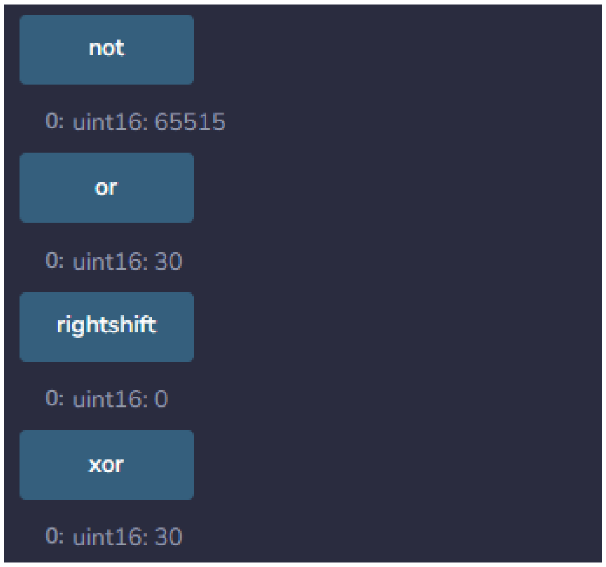
```
// Solidity program to demonstrate
// Bitwise Operator
pragma solidity ^0.5.0;
```

```
// Creating a contract
contract SolidityTest {

    // Declaring variables
    uint16 public a = 20;
    uint16 public b = 10;

    // Initializing a variable
    // to '&' value
    uint16 public and = a & b;

    // Initializing a variable
    // to '|' value
    uint16 public or = a | b;

    // Initializing a variable
    // to '^' value
    uint16 public xor = a ^ b;

    // Initializing a variable
    // to '<<' value
    uint16 public leftshift = a << b;

    // Initializing a variable
    // to '>>' value
    uint16 public rightshift = a >> b;

    // Initializing a variable
    // to '~' value
    uint16 public not = ~a ;


}
```

Output:



```
SOLIDITYTEST AT 0X7B9...B6ACE (MEMORY)
Balance: 0 ETH
a
0: uint16: 20
and
0: uint16: 0
b
0: uint16: 10
leftshift
0: uint16: 20480
```



```
not
0: uint16: 65515
or
0: uint16: 30
rightshift
0: uint16: 0
xor
0: uint16: 30
```

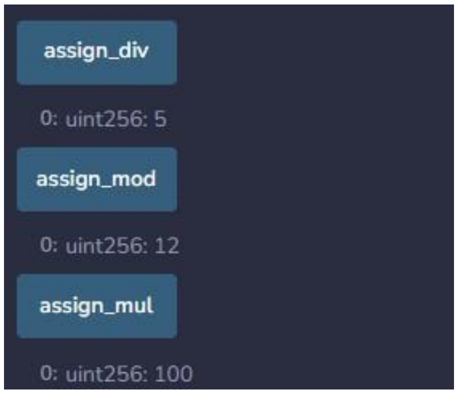
e. Assignment Operator:

Code:

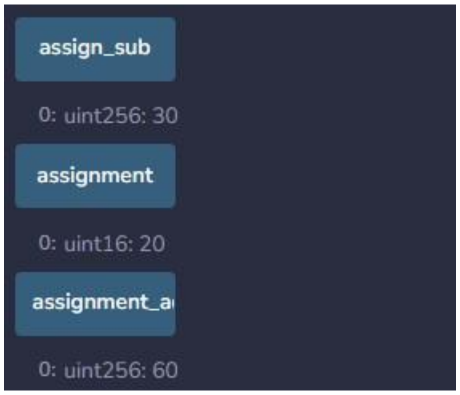
```
// Solidity program to demonstrate
// Assignment Operator
pragma solidity ^0.5.0;

// Creating a contract
contract SolidityTest {

    // Declaring variables
    uint16 public assignment = 20;
    uint public assignment_add = 50;
    uint public assign_sub = 50;
    uint public assign_mul = 10;
    uint public assign_div = 50;
    uint public assign_mod = 32;


    // Defining function to
    // demonstrate Assignment Operator
```

```
        function getResult() public{
        assignment_add += 10;
        assign_sub -= 20;
        assign_mul *= 10;
        assign_div /= 10;
        assign_mod %= 20;
        return ;
        }
}
```

Output:



```
0: uint256: 5
0: uint256: 12
0: uint256: 100
```



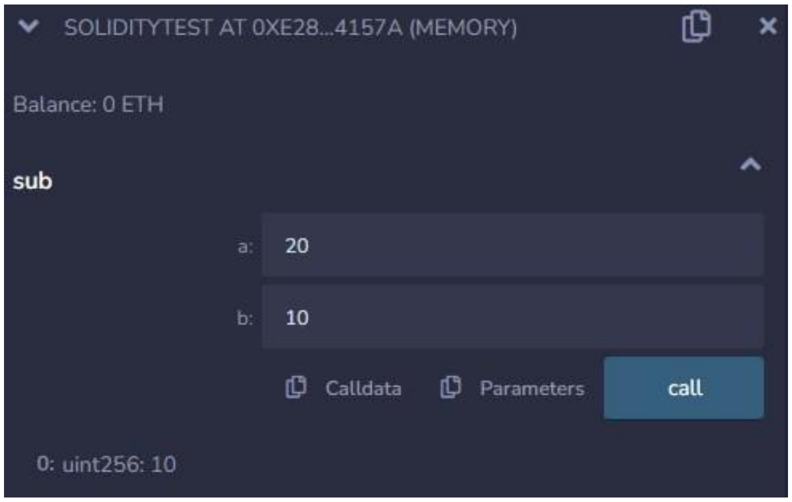
```
0: uint256: 30
0: uint16: 20
0: uint256: 60
```

f. Conditional Operator:

Code:

```
// Solidity program to demonstrate
// Conditional Operator
pragma solidity ^0.5.0;

// Creating a contract
contract SolidityTest{

    // Defining function to demonstrate
    // conditional operator
    function sub(
    uint a, uint b) public view returns(
    uint){
    uint result = (a > b? a-b : b-a);
    return result;
}
}
```

Output:



```
SOLIDITYTEST AT 0XE28...4157A (MEMORY)
Balance: 0 ETH
sub
a: 20
b: 10
0: uint256: 10
```

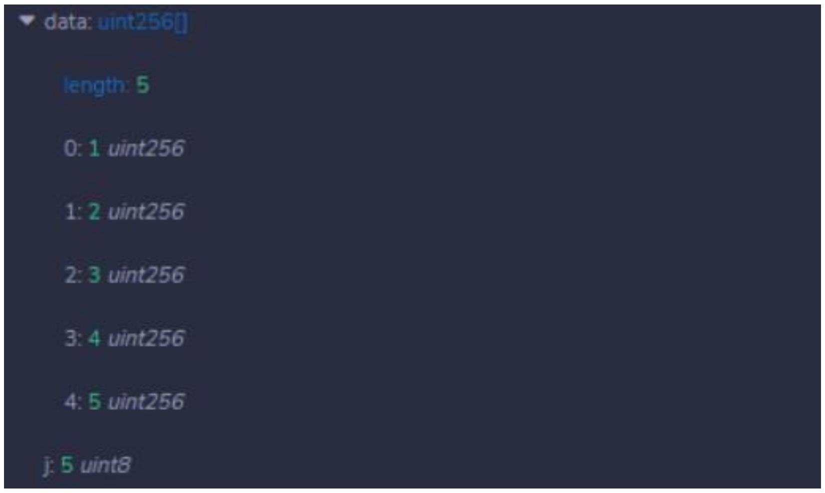
Loops

a. While Loop:

Code:

```
// Solidity program to
// demonstrate the use
// of 'While loop'
pragma solidity ^0.5.0;

// Creating a contract
contract Types {

        // Declaring a dynamic array
        uint[] data;

        // Declaring state variable
        uint8 j = 0;

        // Defining a function to
        // demonstrate While loop'
        function loop(
        ) public returns(uint[] memory){
        while(j < 5) {
                j++;
                data.push(j);
        }
        return data;
        }
}
```

Output:



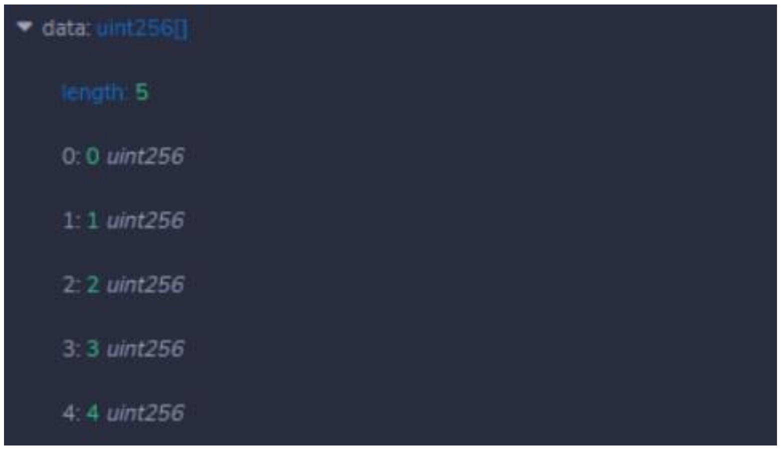
```
▾ data: uint256[]
    length: 5
    0: 1 uint256
    1: 2 uint256
    2: 3 uint256
    3: 4 uint256
    4: 5 uint256
  j: 5 uint8
```

b. Do-While Loop:

Code:

```
// Solidity program to
// demonstrate the use of
// 'Do-While loop'
pragma solidity ^0.5.0;

// Creating a contract
contract Types {

    // Declaring a dynamic array
    uint[] data;

    // Declaring state variable
    uint8 j = 0;

    // Defining function to demonstrate
    // 'Do-While loop'
    function loop(
    ) public returns(uint[] memory){
    do{
        j++;
        data.push(j);
    }while(j < 5) ;
    return data;
    }
}
```

Output:



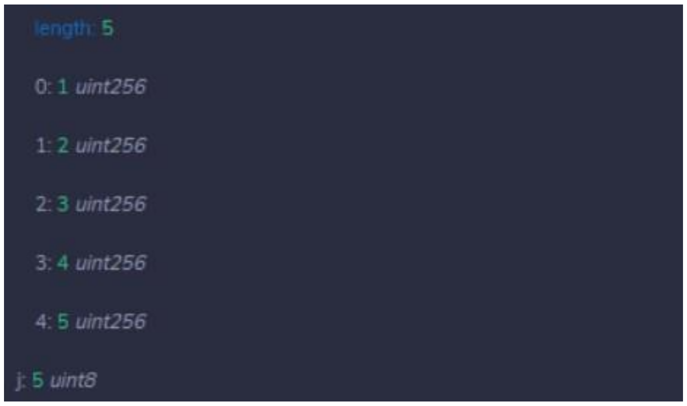
```
▾ data: uint256[]
    length: 5
    0: 0 uint256
    1: 1 uint256
    2: 2 uint256
    3: 3 uint256
    4: 4 uint256
```

c. For Loop:

Code:

```
// Solidity program to
// demonstrate the use
// of 'For loop'
pragma solidity ^0.5.0;

// Creating a contract
contract Types {

        // Declaring a dynamic array
        uint[] data;

        // Defining a function
        // to demonstrate 'For loop'
        function loop(
        ) public returns(uint[] memory){
        for(uint i=0; i<5; i++){
                data.push(i);
        }
        return data;
        }
}
```

Output:



```
  length: 5
  0: 1 uint256
  1: 2 uint256
  2: 3 uint256
  3: 4 uint256
  4: 5 uint256
j: 5 uint8
```

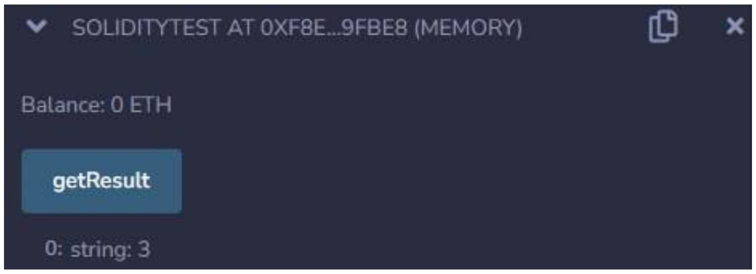
Decision Making:

If Statement

Code:

```
pragma solidity ^0.5.0;
contract SolidityTest {
  uint storedData;
  constructor() public {
    storedData = 10;
  }
  function getResult() public view returns(string memory){
    uint a = 1;
    uint b = 2;
    uint result = a + b;
    return integerToString(result);
  }
  function integerToString(uint _i) internal pure
    returns (string memory) {
    if (_i == 0) {  // if statement
      return "0";
    }
    uint j = _i;
    uint len;

    while (j != 0) {
      len++;
      j /= 10;
    }
    bytes memory bstr = new bytes(len);
    uint k = len - 1;

    while (_i != 0) {
      bstr[k--] = byte(uint8(48 + _i % 10));
      _i /= 10;
    }
    return string(bstr);//access local variable
  }
}
```

Output:



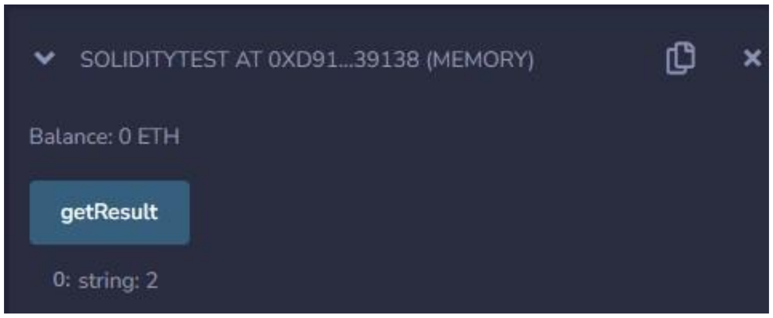
```
0: string: 3
```

If else statement:

Code:

```
pragma solidity ^0.5.0;
contract SolidityTest {
  uint storedData;
  constructor() public{
    storedData = 10;
  }
  function getResult() public view returns(string memory){
    uint a = 1;
    uint b = 2;
    uint result;
    if( a > b) { // if else statement
      result = a;
    } else {
      result = b;
    }
    return integerToString(result);
  }
  function integerToString(uint _i) internal pure
    returns (string memory) {
    if (_i == 0) {
      return "0";
    }
    uint j = _i;
    uint len;

    while (j != 0) {
      len++;
      j /= 10;
    }
    bytes memory bstr = new bytes(len);
    uint k = len - 1;

    while (_i != 0) {
      bstr[k--] = byte(uint8(48 + _i % 10));
      _i /= 10;
    }
    return string(bstr);//access local variable
  }
}
```

Output:



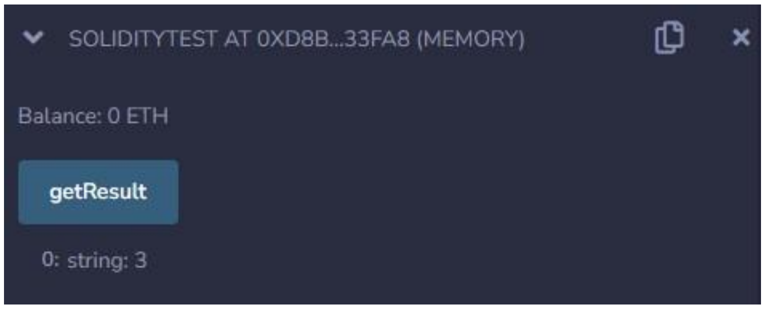
```
Balance: 0 ETH
0: string: 2
```

If-else-If statement:

Code:

```
pragma solidity ^0.5.0;
contract SolidityTest {
  uint storedData; // State variable
  constructor() public {
    storedData = 10;
  }
  function getResult() public view returns(string memory) {
    uint a = 1;
    uint b = 2;
    uint c = 3;
    uint result;

    if( a > b && a > c) { // if else statement
      result = a;
    } else if( b > a && b > c ){
      result = b;
    } else {
      result = c;
    }
    return integerToString(result);
  }
  function integerToString(uint _i) internal pure
    returns (string memory) {

    if (_i == 0) {
      return "0";
    }
    uint j = _i;
    uint len;

    while (j != 0) {
      len++;
      j /= 10;
    }
    bytes memory bstr = new bytes(len);
    uint k = len - 1;

    while (_i != 0) {
      bstr[k--] = byte(uint8(48 + _i % 10));
      _i /= 10;
    }
    return string(bstr);//access local variable
  }
}
```

Output:



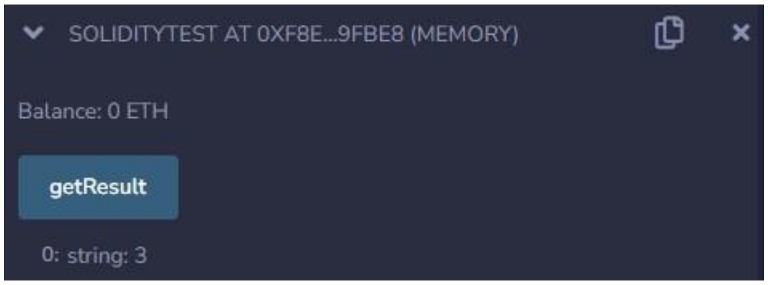
```
Balance: 0 ETH
0: string: 3
```

Strings:

Code:

```
pragma solidity ^0.5.0;
contract SolidityTest {
  constructor() public{
  }
  function getResult() public view returns(string memory){
    uint a = 1;
    uint b = 2;
    uint result = a + b;
    return integerToString(result);
  }
  function integerToString(uint _i) internal pure
    returns (string memory) {

    if (_i == 0) {
      return "0";
    }
    uint j = _i;
    uint len;

    while (j != 0) {
      len++;
      j /= 10;
    }
    bytes memory bstr = new bytes(len);
    uint k = len - 1;

    while (_i != 0) {
      bstr[k--] = byte(uint8(48 + _i % 10));
      _i /= 10;
    }
    return string(bstr);
  }
}
```

Output:


SOLIDITYTEST AT 0XF8E...9FBE8 (MEMORY)

Balance: 0 ETH

```
0: string: 3
```

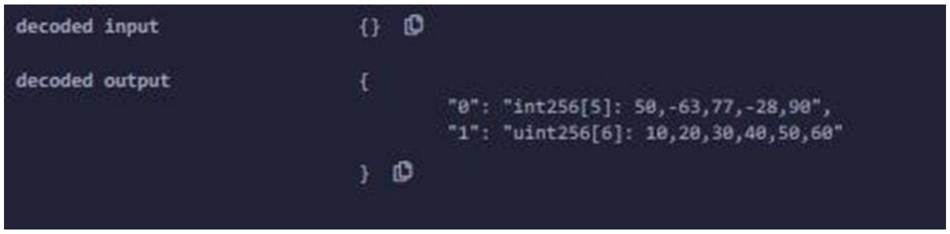
Arrays:

Code:

```
pragma solidity ^0.5.0;
contract test {
  function testArray() public pure{
    uint len = 7;

    //dynamic array
    uint[] memory a = new uint[](7);

    //bytes is same as byte[]
    bytes memory b = new bytes(len);
```

```
      assert(a.length == 7);
      assert(b.length == len);

      //access array variable
      a[6] = 8;

      //test array variable
      assert(a[6] == 8);

      //static array
      uint[3] memory c = [uint(1) , 2, 3];
      assert(c.length == 3);
   }
}
```

Output:



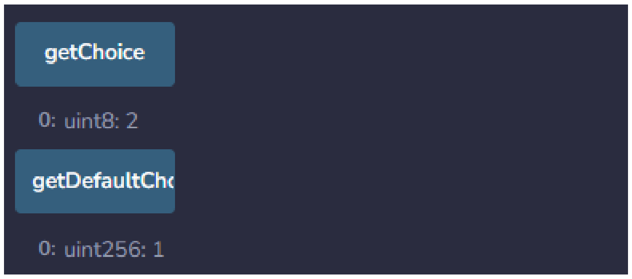
```
decoded input          {}
decoded output         {
                              "0": "int256[5]: 50,-63,77,-28,90",
                              "1": "uint256[6]: 10,20,30,40,50,60"
                       }
```

Enums:

Code:

```
pragma solidity ^0.5.0;
contract test {
   enum FreshJuiceSize{ SMALL, MEDIUM, LARGE }
   FreshJuiceSize choice;
   FreshJuiceSize constant defaultChoice = FreshJuiceSize.MEDIUM;

   function setLarge() public {
      choice = FreshJuiceSize.LARGE;
   }
   function getChoice() public view returns (FreshJuiceSize) {
    return choice;
  }
  function getDefaultChoice() public pure returns (uint) {
    return uint(defaultChoice);
  }
}
```

Output:



getChoice

```
0: uint8: 2
```

getDefaultCho

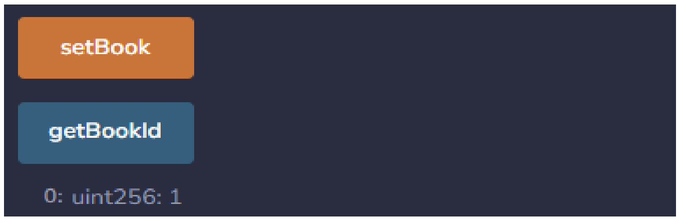
```
0: uint256: 1
```

Structs:

Code:

```
pragma solidity ^0.5.0;
contract test {
  struct Book {
    string title;
    string author;
    uint book_id;
  }
  Book book;

  function setBook() public {
    book = Book('Learn Java', 'TP', 1);
  }
  function getBookId() public view returns (uint) {
    return book.book_id;
  }
}
```

Output:



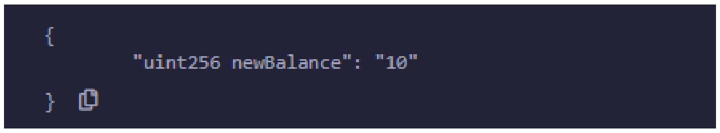
```
0: uint256: 1
```

Mapping:

Code:

```
pragma solidity ^0.5.0;

contract LedgerBalance {
  mapping(address => uint) public balances;

  function updateBalance(uint newBalance) public {
    balances[msg.sender] = newBalance;
  }
}
contract Updater {
  function updateBalance() public returns (uint) {
    LedgerBalance ledgerBalance = new LedgerBalance();
    ledgerBalance.updateBalance(10);
    return ledgerBalance.balances(address(this));
  }
}
```

Output:



```
{
	"uint256 newBalance": "10"
}
```

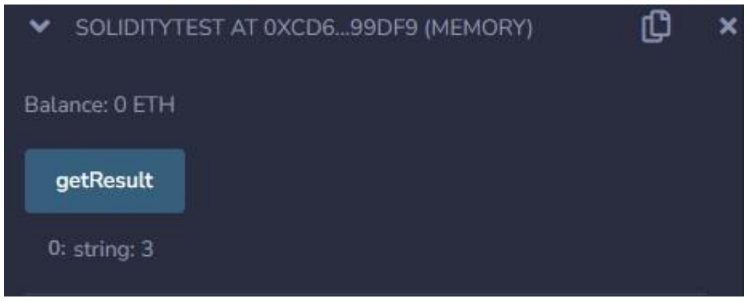
b. Functions, Function Modifiers, View functions, Pure Functions, Fallback Function, Function Overloading, Mathematical functions, Cryptographic functions.

Functions:

Code:

```
pragma solidity ^0.5.0;
contract SolidityTest {
  constructor() public{
  }
  function getResult() public view returns(string memory){
    uint a = 1;
    uint b = 2;
    uint result = a + b;
    return integerToString(result);
  }
  function integerToString(uint _i) internal pure
    returns (string memory) {

    if (_i == 0) {
      return "0";
    }
    uint j = _i;
    uint len;

    while (j != 0) {
      len++;
      j /= 10;
    }
    bytes memory bstr = new bytes(len);
    uint k = len - 1;
    while (_i != 0) {
      bstr[k--] = byte(uint8(48 + _i % 10));
      _i /= 10;
    }
    return string(bstr);//access local variable
  }
}
```

Output:



SOLIDITYTEST AT 0XCD6...99DF9 (MEMORY)

Balance: 0 ETH

getResult

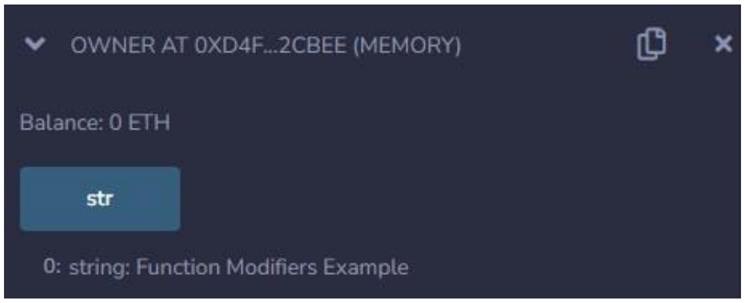
```
0: string: 3
```

Function Modifiers:

Code:

```
pragma solidity ^0.5.0;
contract Owner {    address owner;
  string public str = "Function Modifiers Example";
  constructor() public {
    owner = msg.sender;
  }
  modifier onlyOwner {
    require(msg.sender == owner);
    _;
  }
  modifier costs(uint price) {      if (msg.value >= price) {
      _;
    }
  } }
contract Register is Owner {    mapping (address => bool) registeredAddresses;
  uint price;
  constructor(uint initialPrice) public { price = initialPrice; }

  function register() public payable costs(price) {
    registeredAddresses[msg.sender] = true;
  }
  function changePrice(uint _price) public onlyOwner {
    price = _price;
  }
}
```

Output:



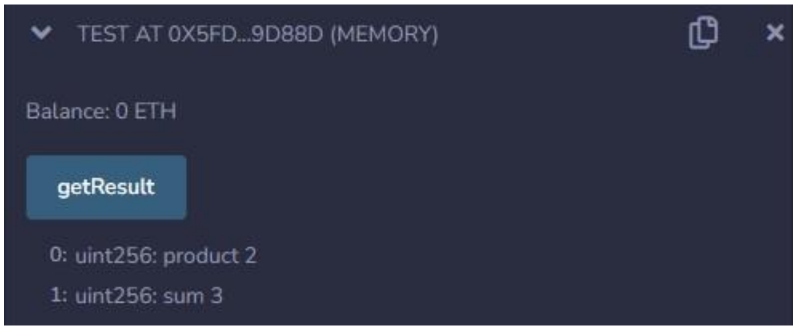
```
0: string: Function Modifiers Example
```

View Function:

Code:

```
pragma solidity ^0.5.0;

contract Test {
  function getResult() public view returns(uint product, uint sum){
    uint a = 1; // local variable
    uint b = 2;
    product = a * b;
    sum = a + b;
  }
}
```

Output:



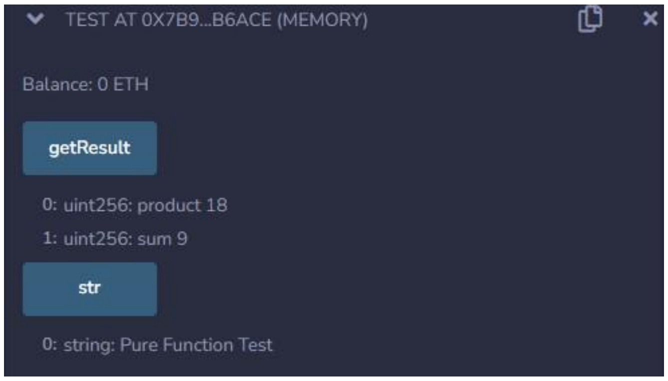
```
TEST AT 0X5FD...9D88D (MEMORY)
Balance: 0 ETH
getResult
0: uint256: product 2
1: uint256: sum 3
```

Pure Function:

Code:

```
pragma solidity ^0.5.0;
contract Test {

  function getResult() public pure returns(uint product, uint sum){
      uint a = 3;
      uint b = 6;
      product = a * b;
      sum = a + b;
  }
  string public str = "Pure Function Test";
}
```

Output:



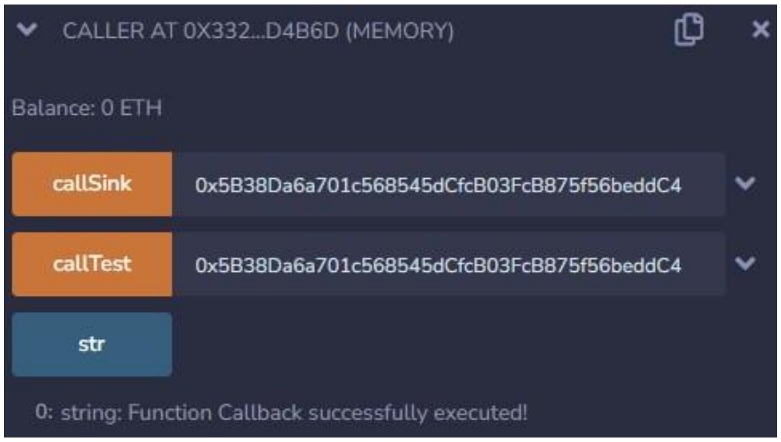
```
TEST AT 0X7B9...B6ACE (MEMORY)
Balance: 0 ETH
getResult
0: uint256: product 18
1: uint256: sum 9
str
0: string: Pure Function Test
```

Fallback Function:

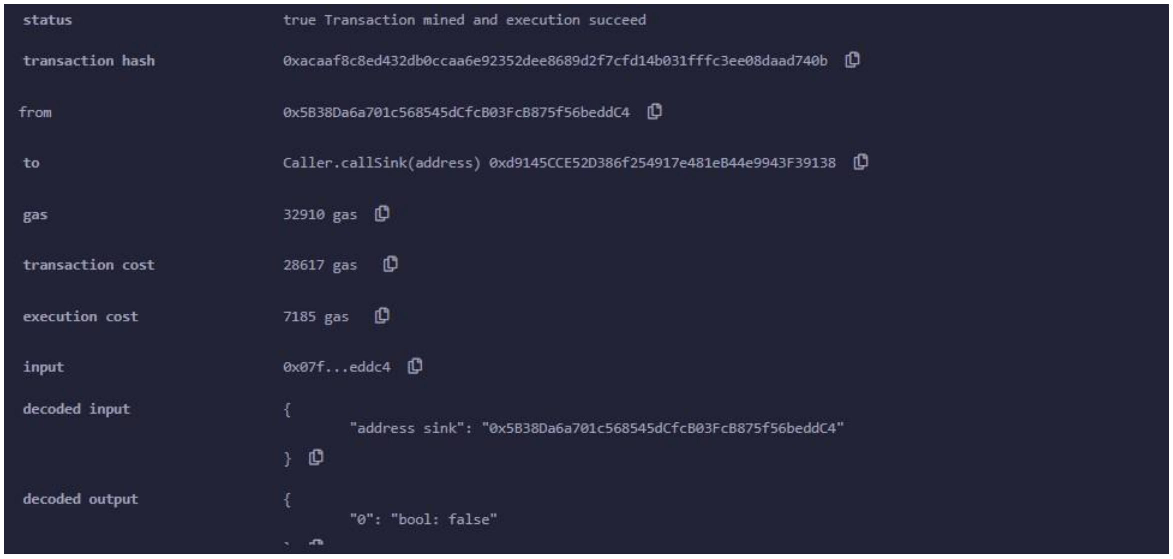
Code:

```
pragma solidity ^0.5.0;
contract Test {
   uint public x ;
   function() external { x = 1; }
}
contract Sink {
  function() external payable { }
}
contract Caller {
  function callTest(Test test) public returns (bool) {
    (bool success,) = address(test).call(abi.encodeWithSignature("nonExistingFunction()"));
    require(success);
    // test.x is now 1

    address payable testPayable = address(uint160(address(test)));

    // Sending ether to Test contract,
    // the transfer will fail, i.e. this returns false here.
    return (testPayable.send(2 ether));
  }
  function callSink(Sink sink) public returns (bool) {
     address payable sinkPayable = address(sink);
     return (sinkPayable.send(2 ether));
   }
   string public str = "Function Callback successfully executed!";
   }
```

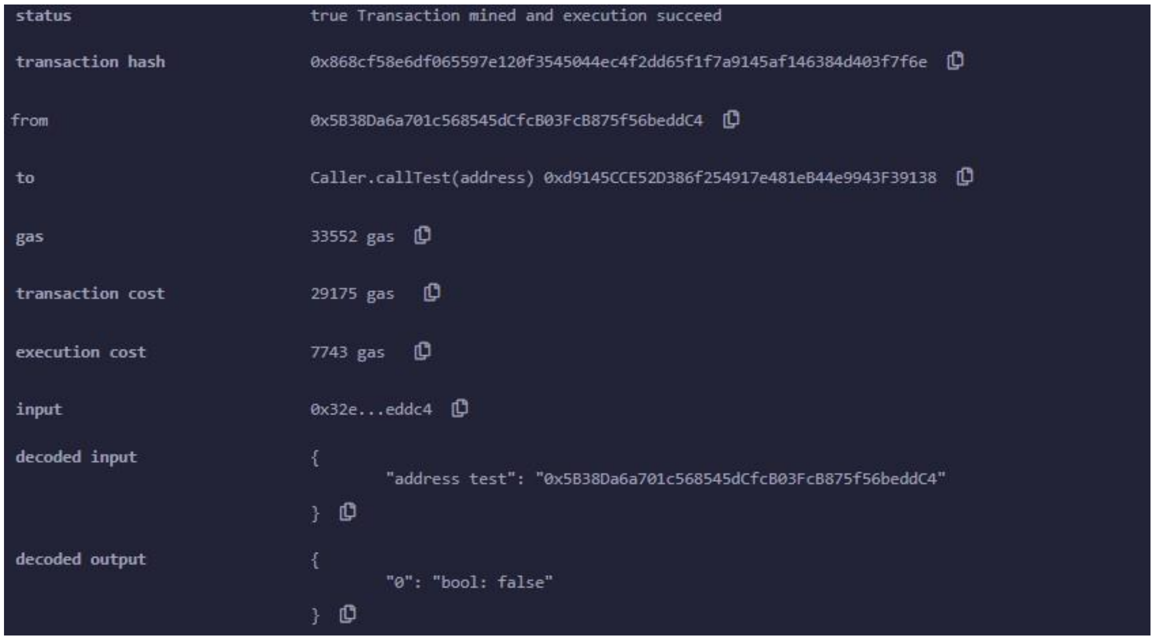
Output:


CALLER AT 0X332...D4B6D (MEMORY)
Balance: 0 ETH
callSink
0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
callTest
0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
str

```
0: string: Function Callback successfully executed!
```



```
status                true Transaction mined and execution succeed
transaction hash      0xacaaf8c8ed432db0ccaa6e92352dee8689d2f7cfd14b031fffc3ee08daad740b
from                  0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                    Caller.callSink(address) 0xd9145CCE52D386f254917e481eB44e9943F39138
gas                   32910 gas
transaction cost      28617 gas
execution cost        7185 gas
input                 0x07f...eddc4
decoded input         {
                              "address sink": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4"
                      }
decoded output        {
                              "0": "bool: false"
```



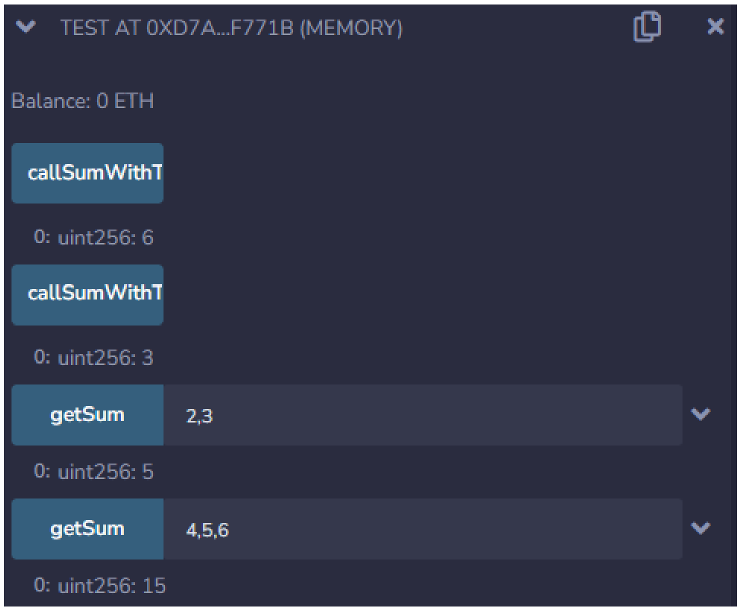
```
status                true Transaction mined and execution succeed
transaction hash      0x868cf58e6df065597e120f3545044ec4f2dd65f1f7a9145af146384d403f7f6e
from                  0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                    Caller.callTest(address) 0xd9145CCE52D386f254917e481eB44e9943F39138
gas                   33552 gas
transaction cost      29175 gas
execution cost        7743 gas
input                 0x32e...eddc4
decoded input         {
                              "address test": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4"
                      }
decoded output        {
                              "0": "bool: false"
                      }
```

Function Overloading:

Code:

```
pragma solidity ^0.5.0;

contract Test {
  function getSum(uint a, uint b) public pure returns(uint){
    return a + b;
  }
  function getSum(uint a, uint b, uint c) public pure returns(uint){
    return a + b + c;
  }
  function callSumWithTwoArguments() public pure returns(uint){
    return getSum(1,2);
  }
  function callSumWithThreeArguments() public pure returns(uint){
    return getSum(1,2,3);
  }
}
```

Output:



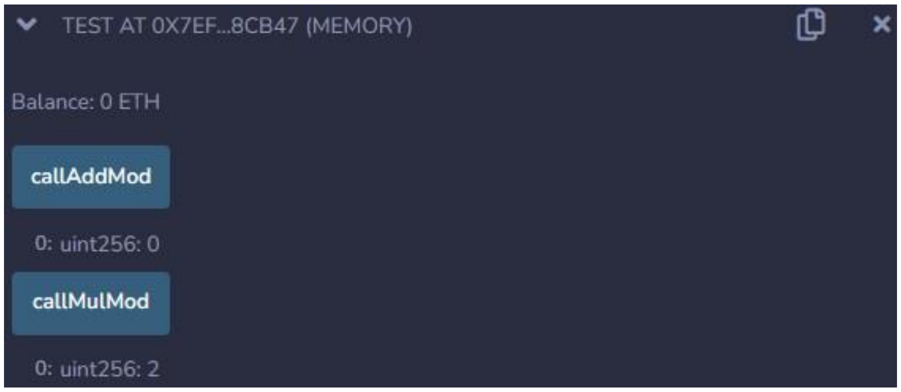
```
TEST AT 0XD7A...F771B (MEMORY)
Balance: 0 ETH
callSumWithT
0: uint256: 6
callSumWithT
0: uint256: 3
getSum  2,3
0: uint256: 5
getSum  4,5,6
0: uint256: 15
```

Mathematical Function:

Code:

```
pragma solidity ^0.5.0;

contract Test {
  function callAddMod() public pure returns(uint){
    return addmod(4, 5, 3);
  }
  function callMulMod() public pure returns(uint){
    return mulmod(4, 5, 3);
  }
}
```

Output:



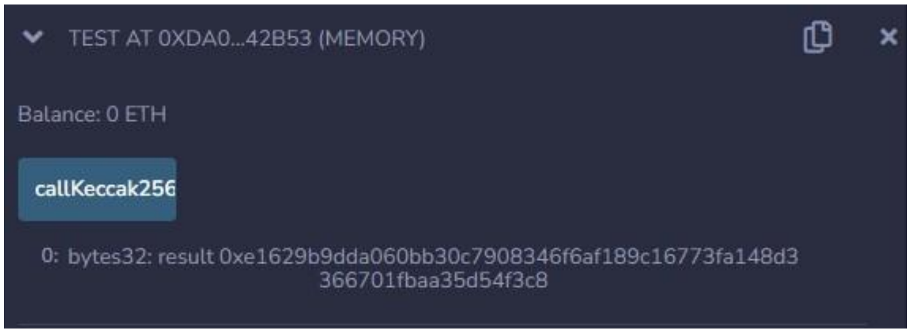
```
TEST AT 0X7EF...8CB47 (MEMORY)

Balance: 0 ETH

callAddMod
0: uint256: 0
callMulMod
0: uint256: 2
```

Cryptographic Functions

Code:

```
pragma solidity ^0.5.0;

contract Test {
  function callKeccak256() public pure returns(bytes32 result){
    return keccak256("ABC");
  }
}
```

Output:



```
Balance: 0 ETH

0: bytes32: result 0xe1629b9dda060bb30c7908346f6af189c16773fa148d3
                   366701fbaa35d54f3c8
```

Practical No 4

Aim: - Implement and demonstrate the use of the following in Solidity:

Practical No 4

Aim: - Implement and demonstrate the use of the following in Solidity:

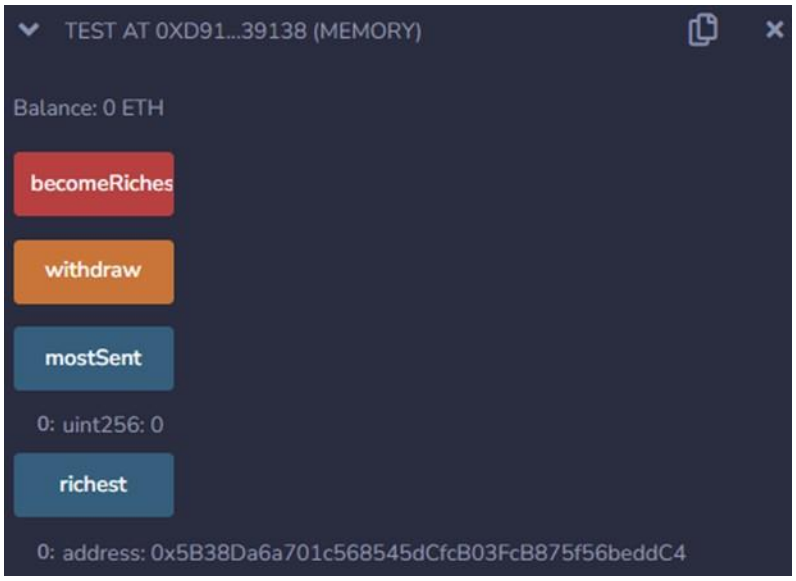
a. Withdrawal Pattern, Restricted Access.

Withdrawal Pattern:

Code:

```
pragma solidity ^0.5.0;

contract Test {
  address public richest;
  uint public mostSent;

  mapping (address => uint) pendingWithdrawals;

  constructor() public payable {
    richest = msg.sender;
    mostSent = msg.value;
  }
  function becomeRichest() public payable returns (bool) {
    if (msg.value > mostSent) {
      pendingWithdrawals[richest] += msg.value;
      richest = msg.sender;
      mostSent = msg.value;
      return true;
    } else  {
      return false;
    }
  }
  function withdraw() public {
    uint amount = pendingWithdrawals[msg.sender];
    pendingWithdrawals[msg.sender] = 0;
    msg.sender.transfer(amount);
  }
}
```

Output:



```
Balance: 0 ETH
0: uint256: 0
0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
```

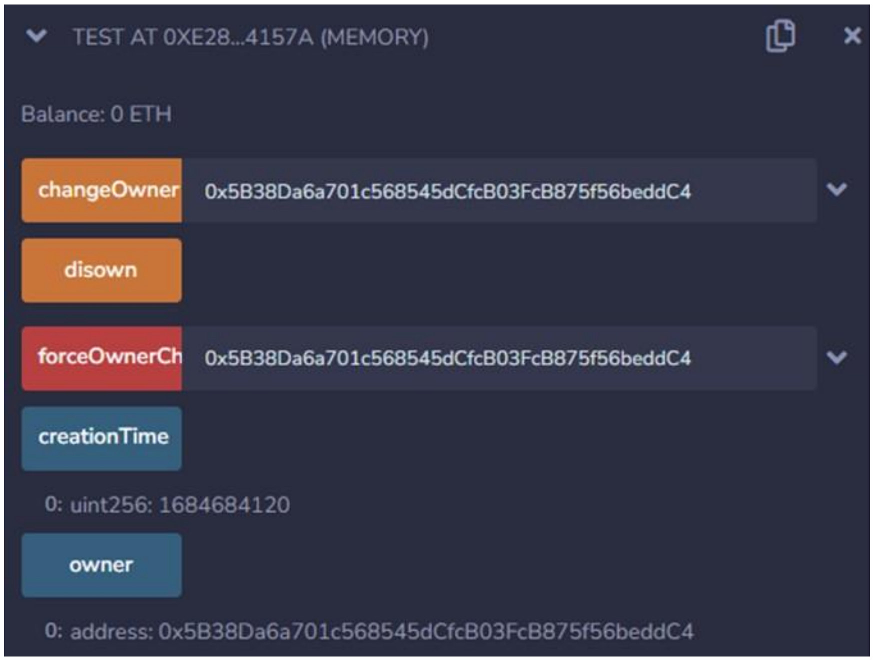
Restricted Access:

Code:

```
pragma solidity ^0.5.0;

contract Test {
  address public owner = msg.sender;
  uint public creationTime = now;

  modifier onlyBy(address _account) {
    require(
      msg.sender == _account,
      "Sender not authorized."
    );
```

```
        _;
    }
    function changeOwner(address _newOwner) public onlyBy(owner) {
        owner = _newOwner;
    }
    modifier onlyAfter(uint _time) {
        require(
            now >= _time,
            "Function called too early."
        );
        _;
    }
    function disown() public onlyBy(owner) onlyAfter(creationTime + 6 weeks) {
        delete owner;
    }
    modifier costs(uint _amount) {
        require(
            msg.value >= _amount,
            "Not enough Ether provided."
        );
        _;
        if (msg.value > _amount)
            msg.sender.transfer(msg.value - _amount);
    }
    function forceOwnerChange(address _newOwner) public payable costs(200 ether) {
        owner = _newOwner;
        if (uint(owner) & 0 == 1) return;
    }
}
```

Output:



```
Balance: 0 ETH
changeOwner 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
forceOwnerCh 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
0: uint256: 1684684120
0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
```

b. Contracts, Inheritance, Interfaces.

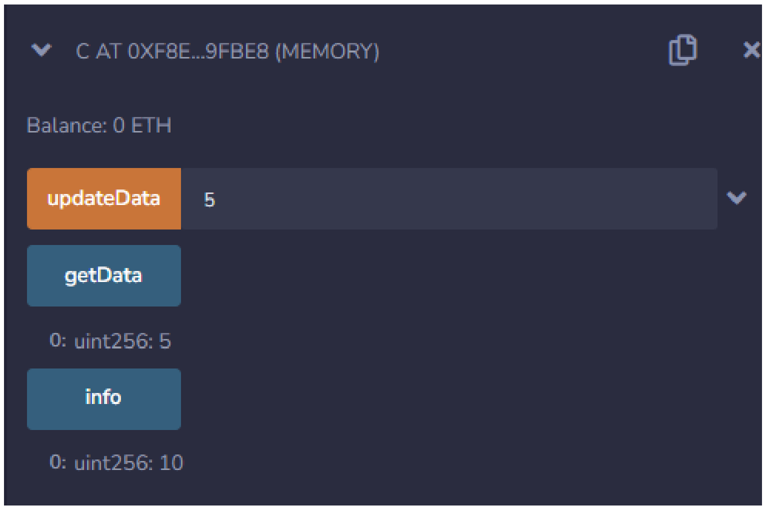
Contracts:

Code:

```
pragma solidity ^0.5.0;

contract C {
  //private state variable
  uint private data;

  //public state variable
  uint public info;

  //constructor
  constructor() public {
    info = 10;
}
```

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

  //public function
  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; }
}
//External Contract
contract D {
  function readData() public returns(uint) {
    C c = new C();
    c.updateData(7);
    return c.getData();
  }
}
//Derived Contract
contract E is C {
  uint private result;
  C private c;

  constructor() public {
    c = new C();
  }
  function getComputedResult() public {
    result = compute(3, 5);
  }
  function getResult() public view returns(uint) { return result; }
  function getData() public view returns(uint) { return c.info(); }
}
```

Output:



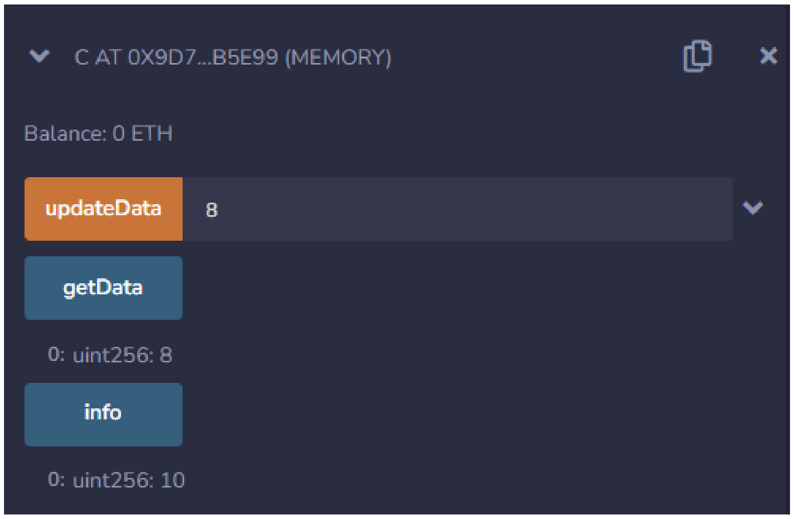
```
C AT 0XF8E...9FBE8 (MEMORY)
Balance: 0 ETH
updateData 5
getData
0: uint256: 5
info
0: uint256: 10
```

Inheritance:

Code:

```
pragma solidity ^0.5.0;

contract C {
  //private state variable
  uint private data;

  //public state variable
  uint public info;

  //constructor
  constructor() public {
    info = 10;
  }
  //private function
  function increment(uint a) private pure returns(uint) { return a + 1; }

  //public function
  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; }
}
//Derived Contract
contract E is C {
  uint private result;
  C private c;
  constructor() public {
    c = new C();
  }
  function getComputedResult() public {
    result = compute(3, 5);
  }
  function getResult() public view returns(uint) { return result; }
  function getData() public view returns(uint) { return c.info(); }
}
```

Output:



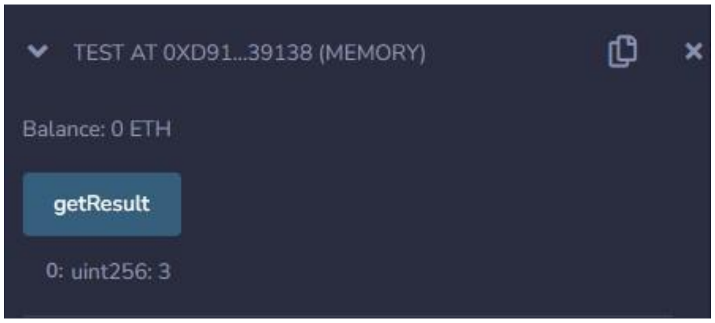
```
Balance: 0 ETH
updateData 8
0: uint256: 8
0: uint256: 10
```

Interfaces:

Code:

```
pragma solidity ^0.5.0;

interface Calculator {
  function getResult() external view returns(uint);
}
contract Test is Calculator {
  constructor() public {}
  function getResult() external view returns(uint){
    uint a = 1;
    uint b = 2;
    uint result = a + b;
    return result;
  }
}
```

Output:



```
TEST AT 0XD91...39138 (MEMORY)
Balance: 0 ETH
getResult
0: uint256: 3
```

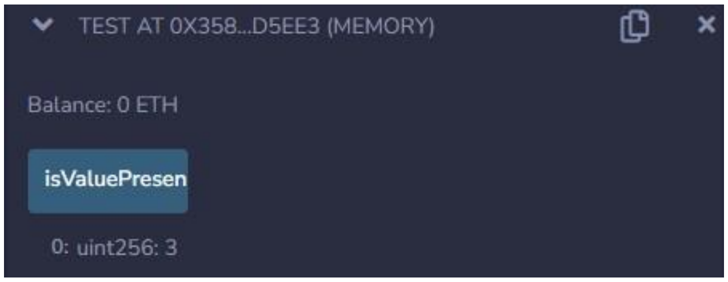
c. Libraries, Assembly, Error handling.

Libraries:

Code:

```
pragma solidity ^0.5.0;

library Search {
  function indexOf(uint[] storage self, uint value) public view returns (uint) {
    for (uint i = 0; i < self.length; i++) if (self[i] == value) return i;
    return uint(-1);
  }
}
contract Test {
  uint[] data;
  constructor() public {
    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;
  }
}
```

Output:



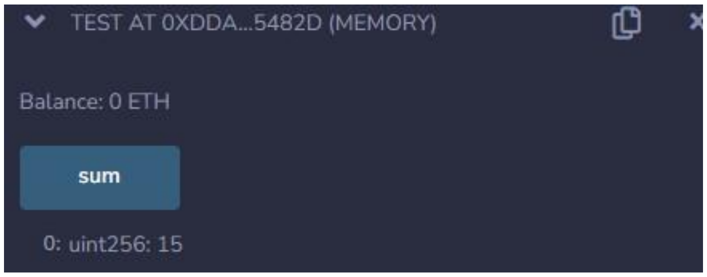
```
TEST AT 0X358...D5EE3 (MEMORY)
Balance: 0 ETH
isValuePresen
0: uint256: 3
```

Assembly:

Code:

```
pragma solidity ^0.5.0;

library Sum {
  function sumUsingInlineAssembly(uint[] memory _data) public pure returns (uint o_sum) {
    for (uint i = 0; i < _data.length; ++i) {
      assembly {
        o_sum := add(o_sum, mload(add(add(_data, 0x20), mul(i, 0x20))))
      }
    }
  }
}
contract Test {
  uint[] data;

  constructor() public {
    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:



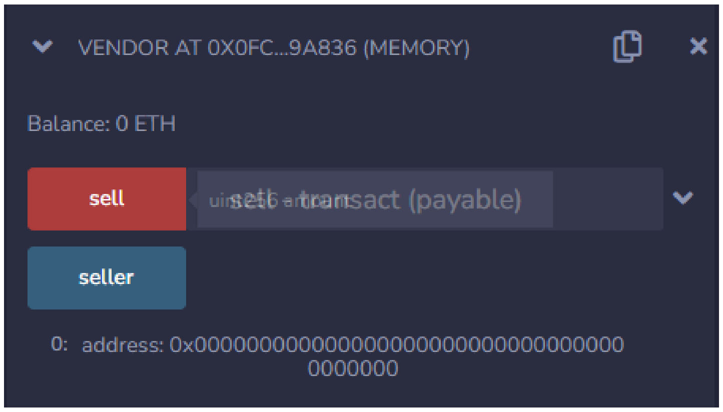
```
Balance: 0 ETH
0: uint256: 15
```

Error Handling:

Code:

```
pragma solidity ^0.5.0;

contract Vendor {
  address public seller;
  modifier onlySeller() {
    require(
      msg.sender == seller,
      "Only seller can call this."
    );
    _;
  }
  function sell(uint amount) public payable onlySeller {
    if (amount > msg.value / 2 ether)
      revert("Not enough Ether provided.");
    // Perform the sell operation.
  }
}
```

Output:



VENDOR AT 0X0FC...9A836 (MEMORY)

Balance: 0 ETH

```
0:  address: 0x0000000000000000000000000000000000000000
```

Practical No 5

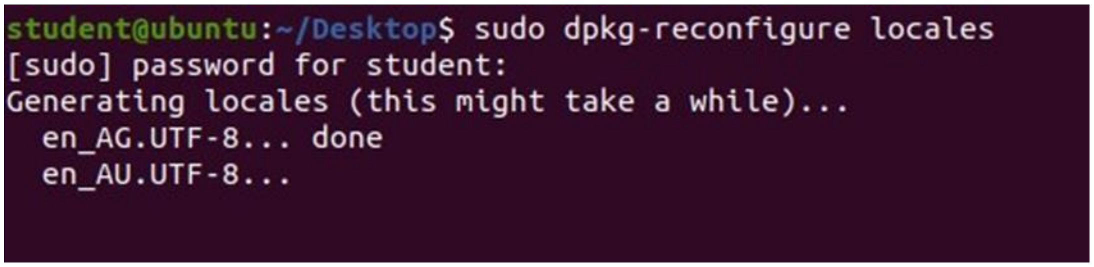
Aim: - Install hyperledger fabric and composer. Deploy and execute the application.

Practical No 5

Aim: - Install hyperledger fabric and composer. Deploy and execute the application.

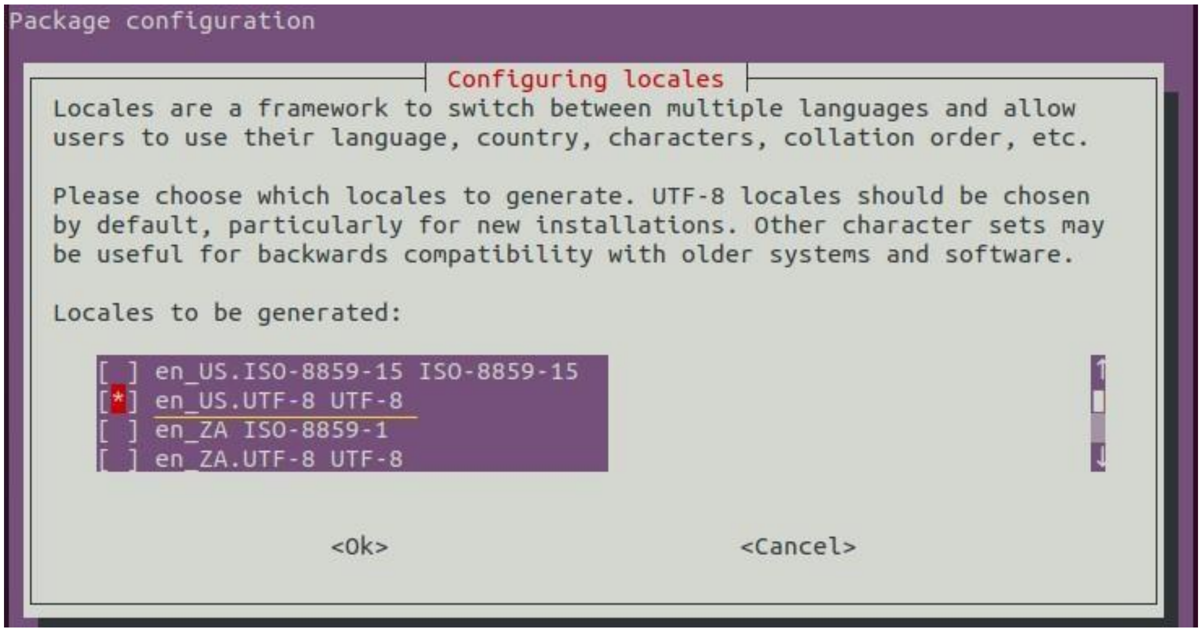
1. Create VM
2. Download VMware Player.
3. Download Ubuntu ISO
4. Install vmware player
5. Create VM of Ubuntu using vmware player

$ sudo dpkg-reconfigure locales // choose en_US.UTF-8 if in doubt

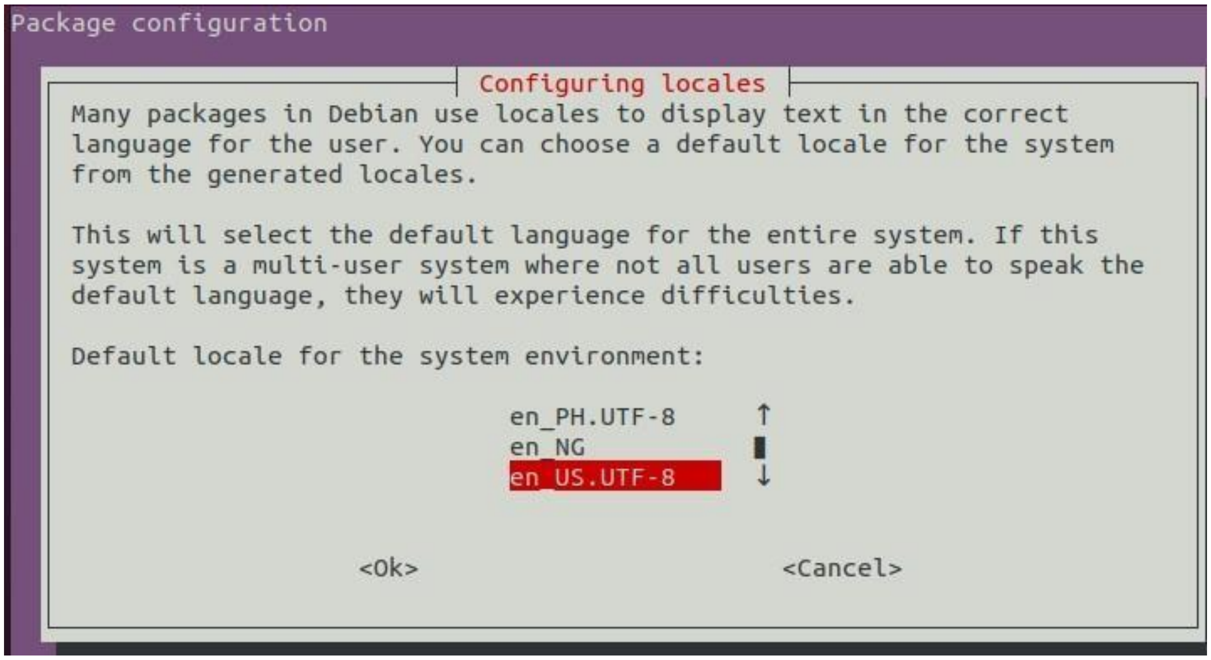


```
student@ubuntu:~/Desktop$ sudo dpkg-reconfigure locales
[sudo] password for student:
Generating locales (this might take a while)...
  en_AG.UTF-8... done
  en_AU.UTF-8...
```

$ sudo apt-get update

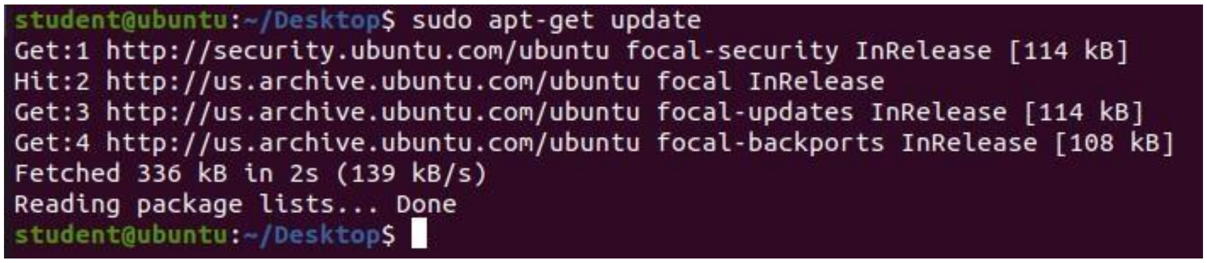


```
Package configuration

                         Configuring locales
 Locales are a framework to switch between multiple languages and allow
 users to use their language, country, characters, collation order, etc.

 Please choose which locales to generate. UTF-8 locales should be chosen
 by default, particularly for new installations. Other character sets may
 be useful for backwards compatibility with older systems and software.

 Locales to be generated:

    [ ] en_US.ISO-8859-15 ISO-8859-15
    [*] en_US.UTF-8 UTF-8
    [ ] en_ZA ISO-8859-1
    [ ] en_ZA.UTF-8 UTF-8

                 <Ok>                         <Cancel>
```



```
Package configuration

Configuring locales
Many packages in Debian use locales to display text in the correct
language for the user. You can choose a default locale for the system
from the generated locales.

This will select the default language for the entire system. If this
system is a multi-user system where not all users are able to speak the
default language, they will experience difficulties.

Default locale for the system environment:

                        en_PH.UTF-8
                        en_NG
                        en_US.UTF-8

            <Ok>                            <Cancel>
```

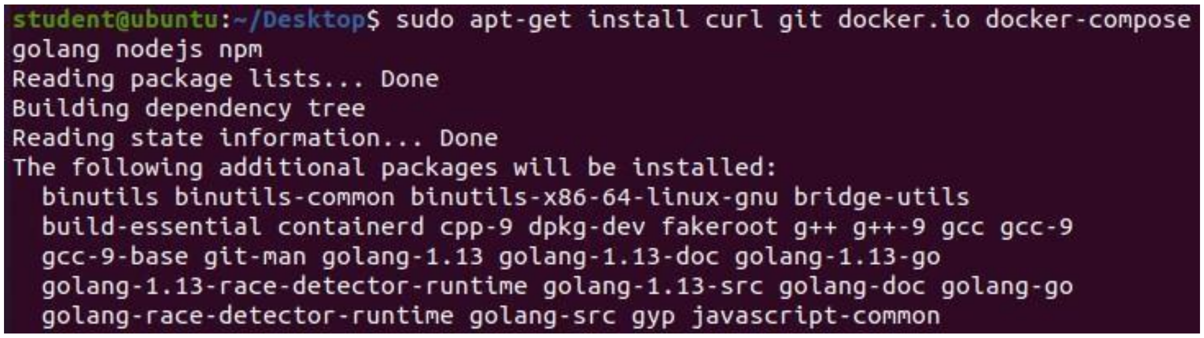
$ sudo apt-get upgrade



```
student@ubuntu:~/Desktop$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Fetched 336 kB in 2s (139 kB/s)
Reading package lists... Done
student@ubuntu:~/Desktop$
```

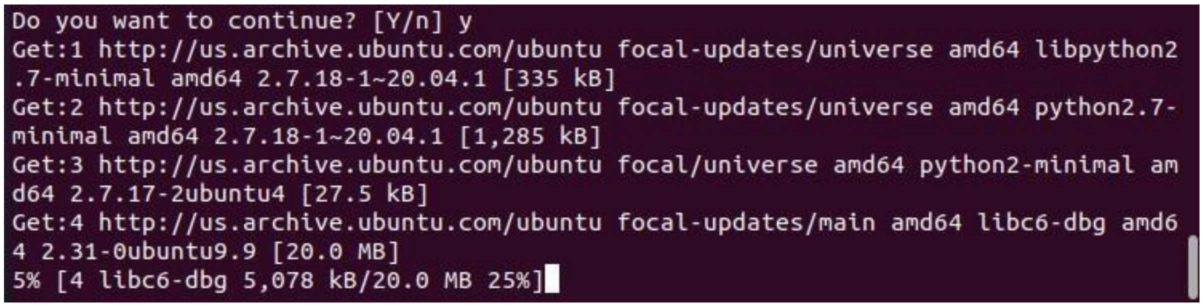
Install pre-requists

$ sudo apt-get install curl git docker.io docker-compose golang nodejs npm



```
student@ubuntu:~/Desktop$ sudo apt-get install curl git docker.io docker-compose
golang nodejs npm
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu bridge-utils
  build-essential containerd cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9
  gcc-9-base git-man golang-1.13 golang-1.13-doc golang-1.13-go
  golang-1.13-race-detector-runtime golang-1.13-src golang-doc golang-go
  golang-race-detector-runtime golang-src gyp javascript-common
```

Type Y for yes



```
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libpython2
.7-minimal amd64 2.7.18-1~20.04.1 [335 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python2.7-
minimal amd64 2.7.18-1~20.04.1 [1,285 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 python2-minimal am
d64 2.7.17-2ubuntu4 [27.5 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libc6-dbg amd6
4 2.31-0ubuntu9.9 [20.0 MB]
5% [4 libc6-dbg 5,078 kB/20.0 MB 25%]
```

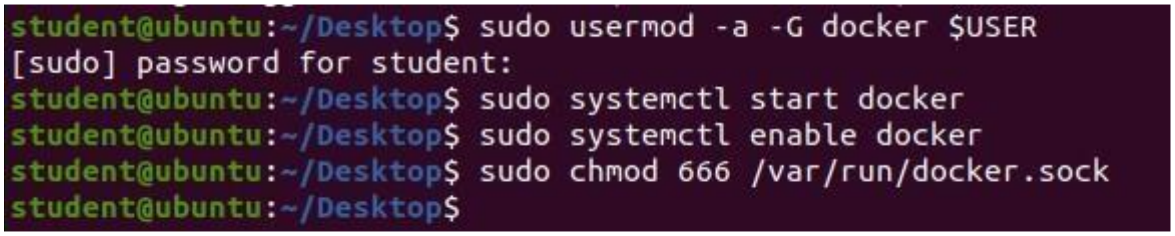
Install Docker

$ sudo usermod -a -G docker $USER

$ sudo systemctl start docker

$ sudo systemctl enable docker

$ sudo chmod 666 /var/run/docker.sock

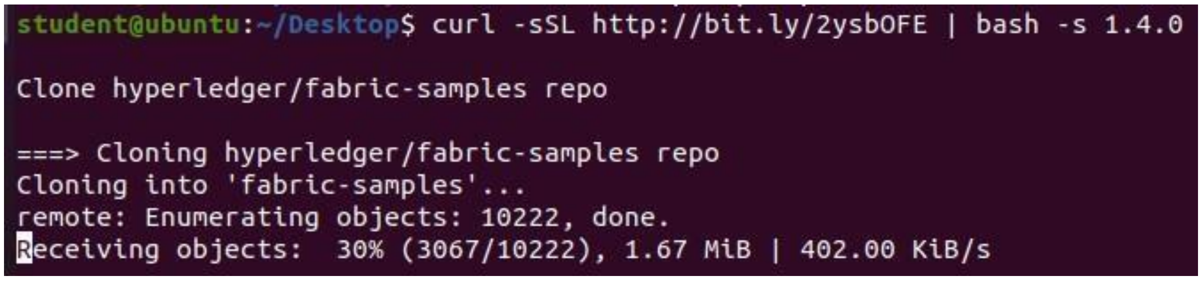


```
student@ubuntu:~/Desktop$ sudo usermod -a -G docker $USER
[sudo] password for student:
student@ubuntu:~/Desktop$ sudo systemctl start docker
student@ubuntu:~/Desktop$ sudo systemctl enable docker
student@ubuntu:~/Desktop$ sudo chmod 666 /var/run/docker.sock
student@ubuntu:~/Desktop$
```

Install Hyperledger Fabric

1. Check the latest version of fabric repository
2. Install Fabric

$ curl -sSL http://bit.ly/2ysbOFE | bash -s 1.4.0

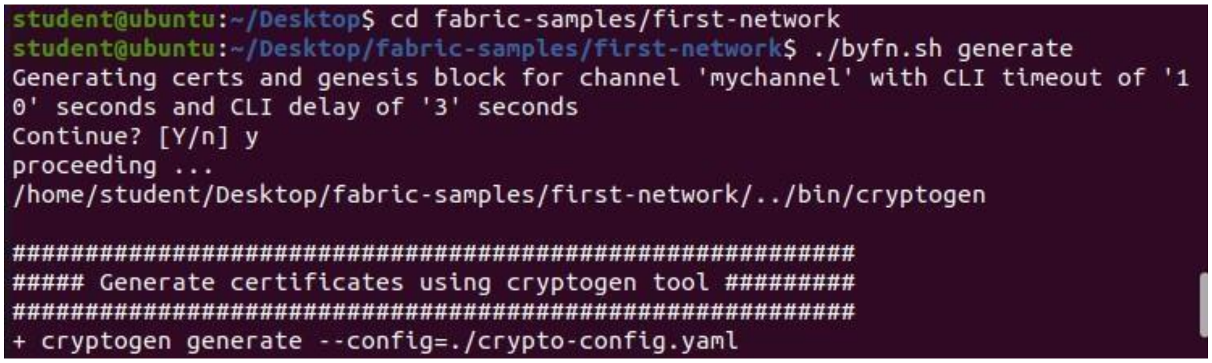


```
student@ubuntu:~/Desktop$ curl -sSL http://bit.ly/2ysbOFE | bash -s 1.4.0

Clone hyperledger/fabric-samples repo

===> Cloning hyperledger/fabric-samples repo
Cloning into 'fabric-samples'...
remote: Enumerating objects: 10222, done.
Receiving objects:  30% (3067/10222), 1.67 MiB | 402.00 KiB/s
```

3. Check if fabric is installed, you should see big "END" once done

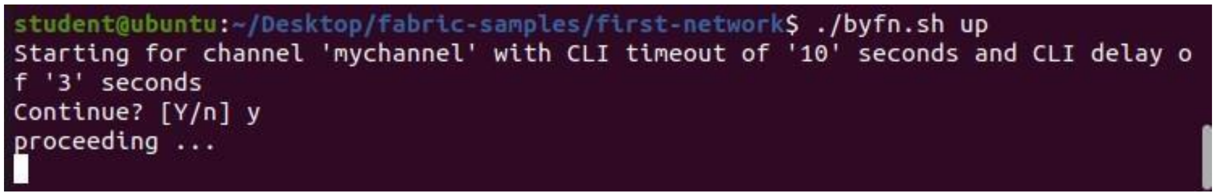
$ cd fabric-samples/first-network

$ ./byfn.sh generate



```
student@ubuntu:~/Desktop$ cd fabric-samples/first-network
student@ubuntu:~/Desktop/fabric-samples/first-network$ ./byfn.sh generate
Generating certs and genesis block for channel 'mychannel' with CLI timeout of '1
0' seconds and CLI delay of '3' seconds
Continue? [Y/n] y
proceeding ...
/home/student/Desktop/fabric-samples/first-network/../bin/cryptogen

##########################################################
##### Generate certificates using cryptogen tool #########
##########################################################
+ cryptogen generate --config=./crypto-config.yaml
```

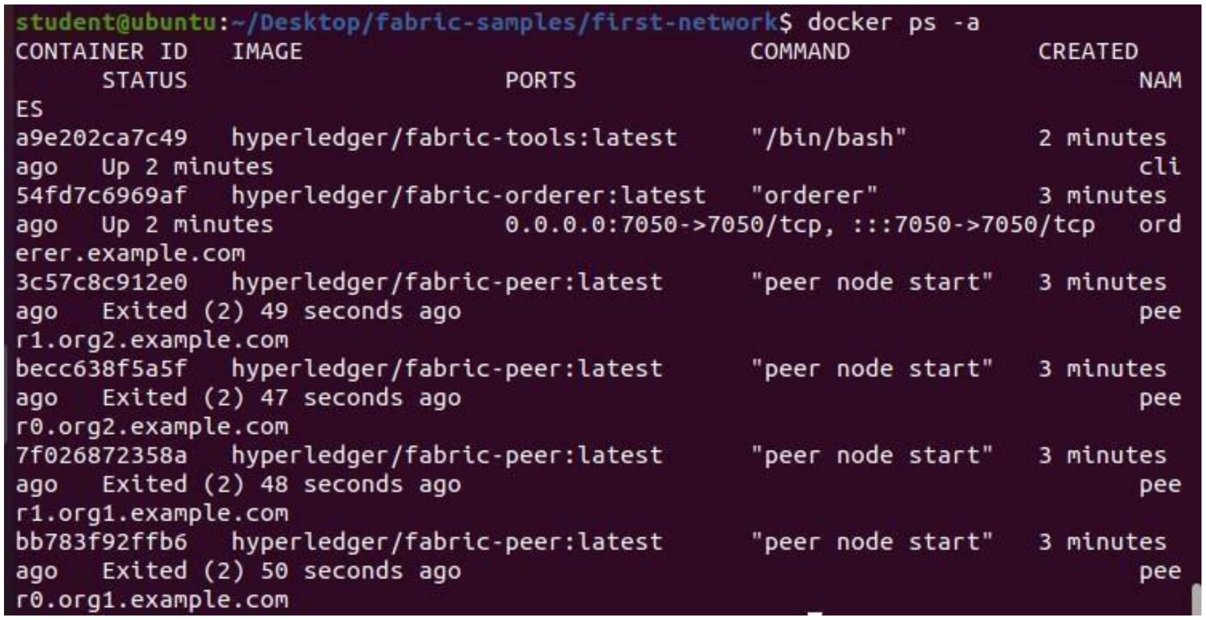
$ ./byfn.sh up



```
student@ubuntu:~/Desktop/fabric-samples/first-network$ ./byfn.sh up
Starting for channel 'mychannel' with CLI timeout of '10' seconds and CLI delay o
f '3' seconds
Continue? [Y/n] y
proceeding ...
```

4. Check if fabric docker is running smoothly

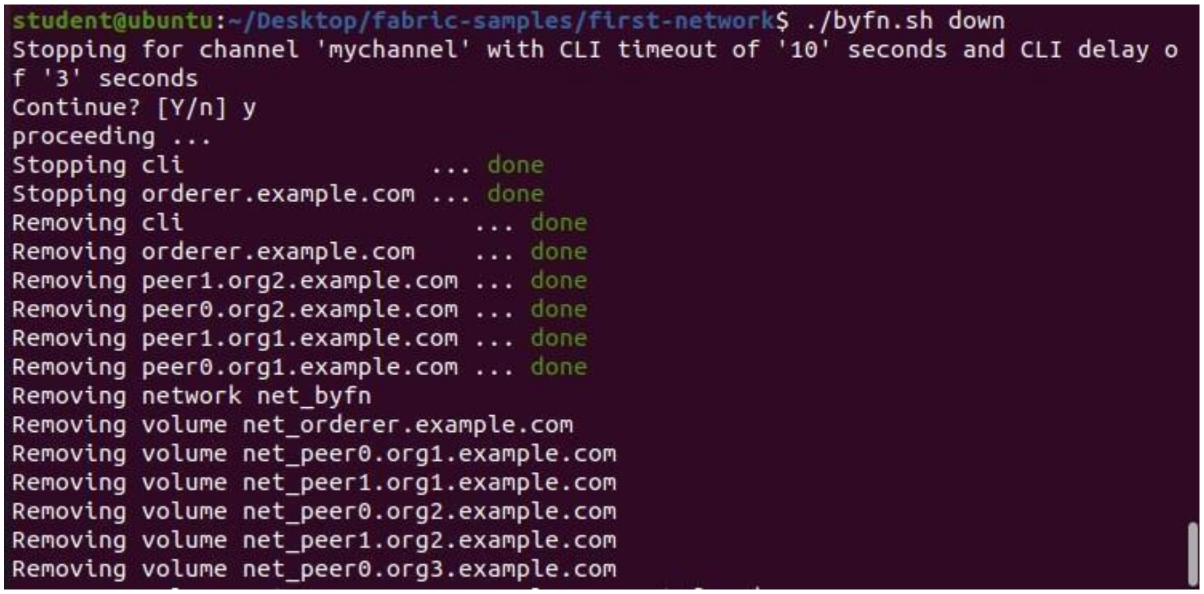
$ docker ps -a



```
student@ubuntu:~/Desktop/fabric-samples/first-network$ docker ps -a
CONTAINER ID   IMAGE                               COMMAND             CREATED
      STATUS                     PORTS                                       NAM
ES
a9e202ca7c49   hyperledger/fabric-tools:latest     "/bin/bash"         2 minutes
ago   Up 2 minutes                                                           cli
54fd7c6969af   hyperledger/fabric-orderer:latest   "orderer"           3 minutes
ago   Up 2 minutes               0.0.0.0:7050->7050/tcp, :::7050->7050/tcp   ord
erer.example.com
3c57c8c912e0   hyperledger/fabric-peer:latest      "peer node start"   3 minutes
ago   Exited (2) 49 seconds ago                                              pee
r1.org2.example.com
becc638f5a5f   hyperledger/fabric-peer:latest      "peer node start"   3 minutes
ago   Exited (2) 47 seconds ago                                              pee
r0.org2.example.com
7f026872358a   hyperledger/fabric-peer:latest      "peer node start"   3 minutes
ago   Exited (2) 48 seconds ago                                              pee
r1.org1.example.com
bb783f92ffb6   hyperledger/fabric-peer:latest      "peer node start"   3 minutes
ago   Exited (2) 50 seconds ago                                              pee
r0.org1.example.com
```

5. Stop the network

$ ./byfn.sh down

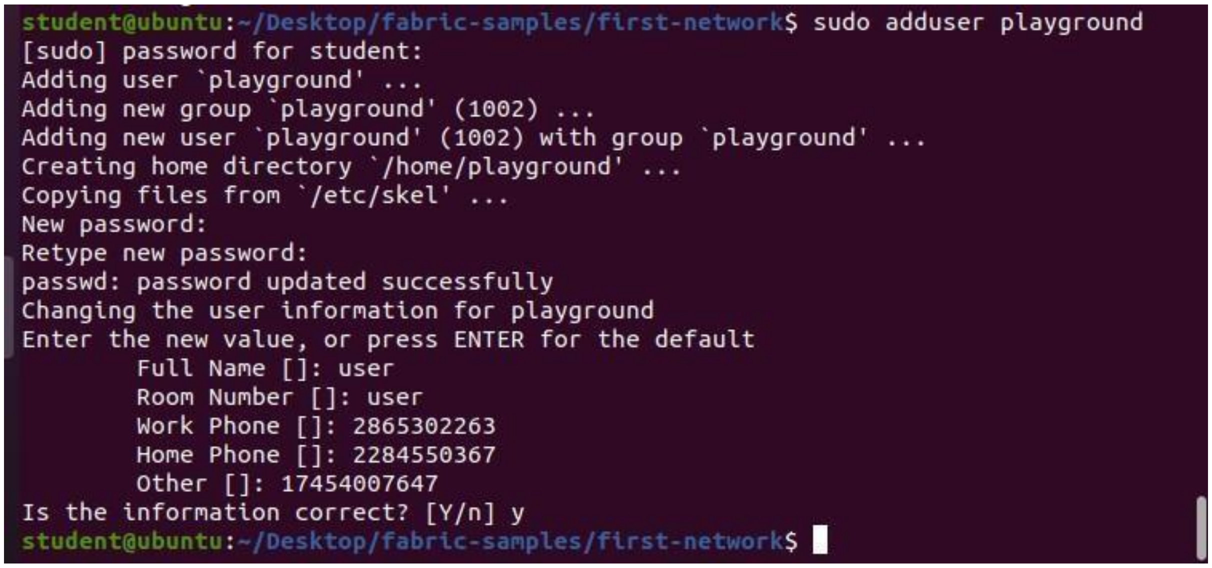


```
student@ubuntu:~/Desktop/fabric-samples/first-network$ ./byfn.sh down
Stopping for channel 'mychannel' with CLI timeout of '10' seconds and CLI delay o
f '3' seconds
Continue? [Y/n] y
proceeding ...
Stopping cli                 ... done
Stopping orderer.example.com ... done
Removing cli                    ... done
Removing orderer.example.com    ... done
Removing peer1.org2.example.com ... done
Removing peer0.org2.example.com ... done
Removing peer1.org1.example.com ... done
Removing peer0.org1.example.com ... done
Removing network net_byfn
Removing volume net_orderer.example.com
Removing volume net_peer0.org1.example.com
Removing volume net_peer1.org1.example.com
Removing volume net_peer0.org2.example.com
Removing volume net_peer1.org2.example.com
Removing volume net_peer0.org3.example.com
```

Install Composer

1. Create new user, when asked about the full name, use something different than the full name used of the main user, to avoid confusion next time you are logging on.

$ sudo adduser playground



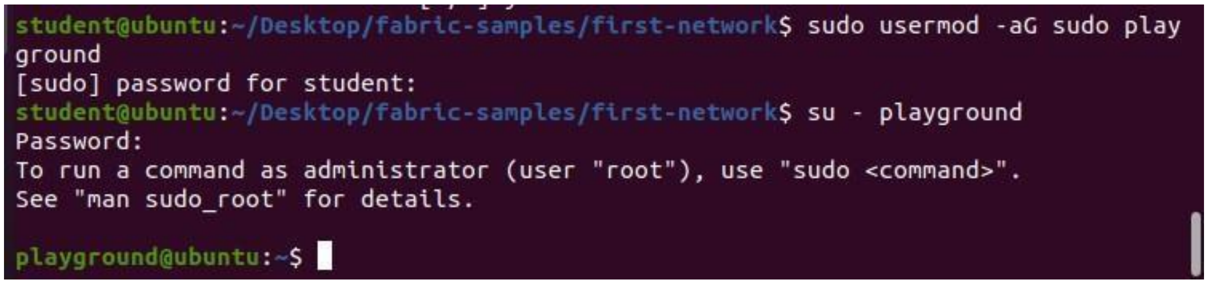
```
student@ubuntu:~/Desktop/fabric-samples/first-network$ sudo adduser playground
[sudo] password for student:
Adding user `playground' ...
Adding new group `playground' (1002) ...
Adding new user `playground' (1002) with group `playground' ...
Creating home directory `/home/playground' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for playground
Enter the new value, or press ENTER for the default
        Full Name []: user
        Room Number []: user
        Work Phone []: 2865302263
        Home Phone []: 2284550367
        Other []: 17454007647
Is the information correct? [Y/n] y
student@ubuntu:~/Desktop/fabric-samples/first-network$ 
```

2. Set permission for the new user

   $ sudo usermod -aG sudo playground

3. Login as the new user

   $ su – playground v

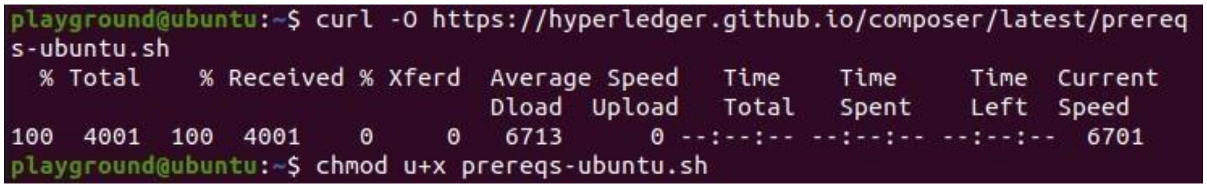


```
student@ubuntu:~/Desktop/fabric-samples/first-network$ sudo usermod -aG sudo play
ground
[sudo] password for student:
student@ubuntu:~/Desktop/fabric-samples/first-network$ su - playground
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

playground@ubuntu:~$ 
```

4. Install the prerequisites by getting and running the script from github. It will ask for the password of "playground" account to proceed.

   $ curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh

   $ chmod u+x prereqs-ubuntu.sh



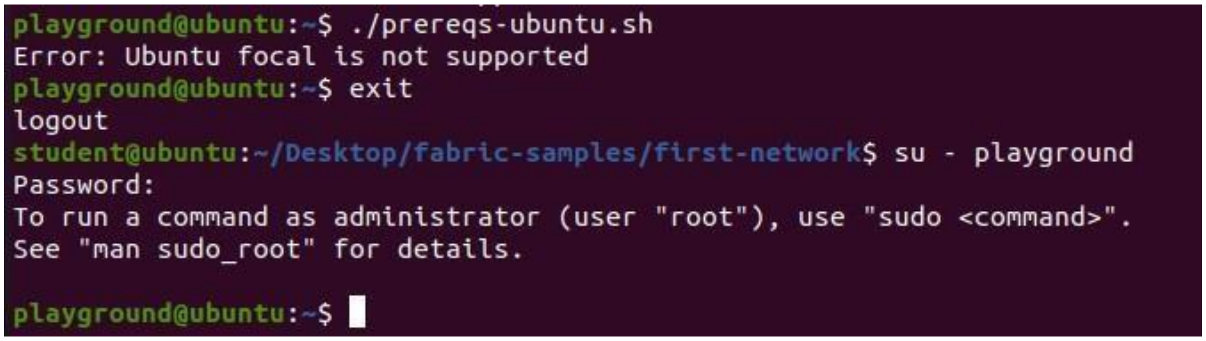
```
playground@ubuntu:~$ curl -O https://hyperledger.github.io/composer/latest/prereq
s-ubuntu.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  4001  100  4001    0     0   6713      0 --:--:-- --:--:-- --:--:--  6701
playground@ubuntu:~$ chmod u+x prereqs-ubuntu.sh
```

   $ ./prereqs-ubuntu.sh

5. Logout and login with the new user to get things activated properly

   $ exit

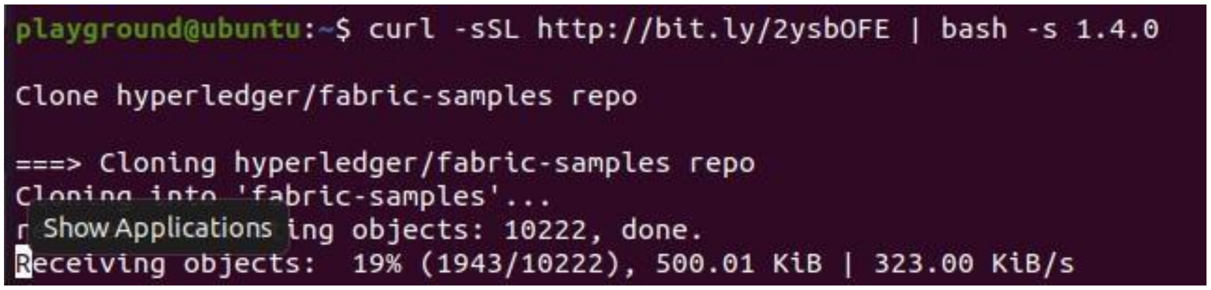
   $ su - playground



```
playground@ubuntu:~$ ./prereqs-ubuntu.sh
Error: Ubuntu focal is not supported
playground@ubuntu:~$ exit
logout
student@ubuntu:~/Desktop/fabric-samples/first-network$ su - playground
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

playground@ubuntu:~$ 
```

6. Install components needed for running Hyperledger Fabric

   $ curl -sSL http://bit.ly/2ysbOFE | bash -s 1.4.0



```
playground@ubuntu:~$ curl -sSL http://bit.ly/2ysbOFE | bash -s 1.4.0

Clone hyperledger/fabric-samples repo

===> Cloning hyperledger/fabric-samples repo
Cloning into 'fabric-samples'...
r Show Applications ing objects: 10222, done.
Receiving objects:  19% (1943/10222), 500.01 KiB | 323.00 KiB/s
```

7. Install components needed for running Hyperledger Composer

   $ npm install -g composer-cli composer-rest-server generator-hyperledger- composer yo composer-playground

8. Start Composer

   $ composer-playground

9. Open your browser and check it:

   http://localhost:8080

Practical No 6

Aim: - Demonstrate the use of Bitcoin Core API.

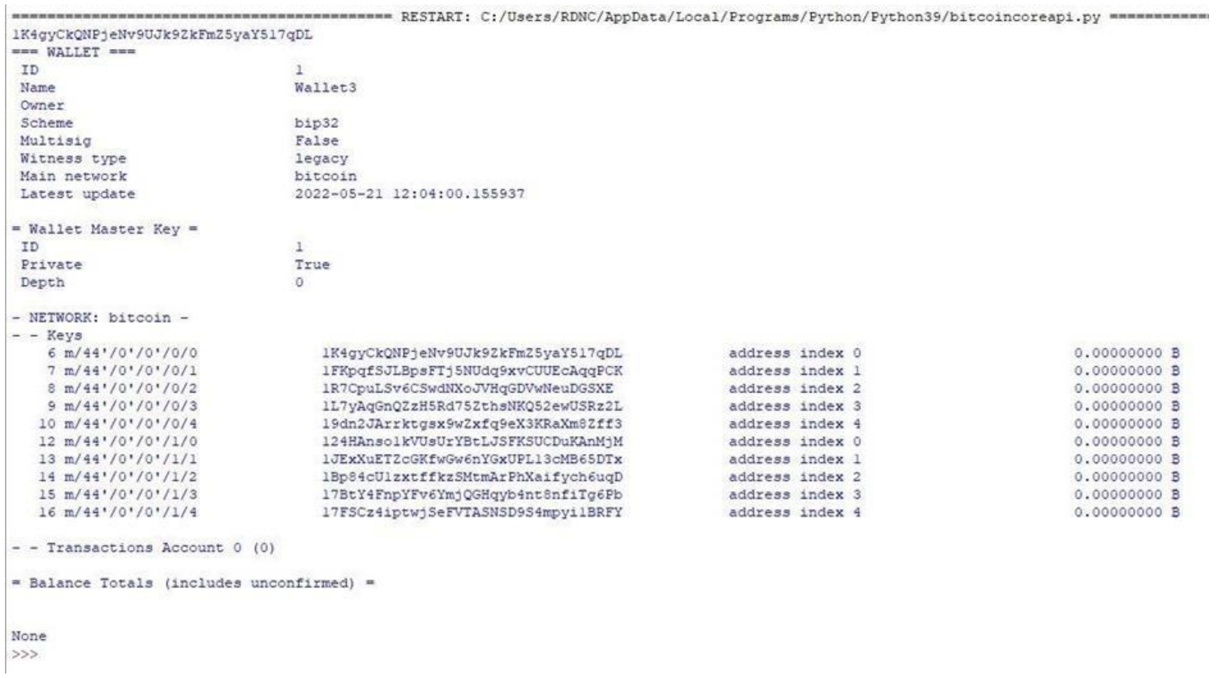
Practical No 6

Aim: - Demonstrate the use of Bitcoin Core API.

Code:

```
from bitcoinlib.wallets import Wallet

w = Wallet.create('Wallet3')

key1 = w.get_key()

print(key1.address)

# Send a small transaction to your wallet and use the scan() method to update transactions and UTXO’s

w.scan()

print(w.info())
```

Output:



```
============================================ RESTART: C:/Users/RDNC/AppData/Local/Programs/Python/Python39/bitcoincoreapi.py ===========
1K4gyCkQNPjeNv9UJk9ZkFmZ5yaY517qDL
=== WALLET ===
 ID                              1
 Name                            Wallet3
 Owner
 Scheme                          bip32
 Multisig                        False
 Witness type                    legacy
 Main network                    bitcoin
 Latest update                   2022-05-21 12:04:00.155937

= Wallet Master Key =
 ID                              1
 Private                         True
 Depth                           0

- NETWORK: bitcoin -
- - Keys
    6 m/44'/0'/0'/0/0               1K4gyCkQNPjeNv9UJk9ZkFmZ5yaY517qDL          address index 0                         0.00000000 B
    7 m/44'/0'/0'/0/1               1FKpqfSJLBpsFTj5NUdq9xvCUUEcAqqPCK          address index 1                         0.00000000 B
    8 m/44'/0'/0'/0/2               1R7CpuLSv6CSwdNXoJVHqGDVwNeuDGSXE           address index 2                         0.00000000 B
    9 m/44'/0'/0'/0/3               1L7yAqGnQZzH5Rd75ZthsNKQ52ewUSRz2L          address index 3                         0.00000000 B
   10 m/44'/0'/0'/0/4               19dn2JArrktgsx9wZxfq9eX3KRaXm8Zff3          address index 4                         0.00000000 B
   12 m/44'/0'/0'/1/0               124HAnsolkVUsUrYBtLJSFKSUCDuKAnMjM          address index 0                         0.00000000 B
   13 m/44'/0'/0'/1/1               1JExXuETZcGKfwGw6nYGxUPL13cMB65DTx          address index 1                         0.00000000 B
   14 m/44'/0'/0'/1/2               1Bp84cU1zxtffkzSMtmArPhXaifych6uqD          address index 2                         0.00000000 B
   15 m/44'/0'/0'/1/3               17BtY4FnpYFv6YmjQGHqyb4nt8nfiTg6Pb          address index 3                         0.00000000 B
   16 m/44'/0'/0'/1/4               17FSCz4iptwjSeFVTASNSD9S4mpyi1BRFY          address index 4                         0.00000000 B

- - Transactions Account 0 (0)

= Balance Totals (includes unconfirmed) =


None
>>>
```

Practical No 7

Aim: - Create your own blockchain and demonstrate its use.

Practical No 7

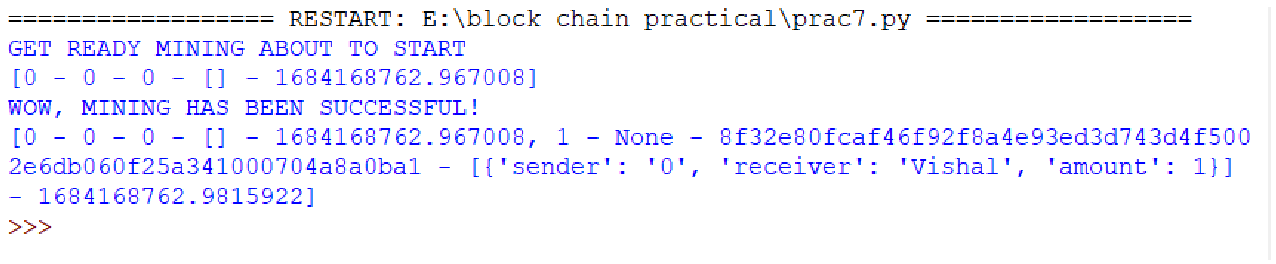
Aim: - Create your own blockchain and demonstrate its use.

Code:

```
import hashlib
import time
class Block(object):
    def___init_(self, index, proof_number, previous_hash, data, timestamp=None):
        self.index = index
        self.proof_number = proof_number
        self.previous_hash = previous_hash
        self.data = data
        self.timestamp = timestamp or time.time()
    @property
    def compute_hash(self):
        string_block = "{}{}{}{}{}".format(self.index, self.proof_number, self.previous_hash,
self.data, self.timestamp)
        return hashlib.sha256(string_block.encode()).hexdigest()
    def___repr_(self):
        return "{} - {} - {} - {} - {}".format(self.index, self.proof_number, self.previous_hash,
self.data, self.timestamp)
class BlockChain(object):
    def___init_(self):
        self.chain = []
        self.current_data = []
        self.nodes = set()
        self.build_genesis()
    def build_genesis(self):
        self.build_block(proof_number=0, previous_hash=0)
    def build_block(self, proof_number, previous_hash):
        block = Block(
            index=len(self.chain),
            proof_number=proof_number,
            previous_hash=previous_hash,
            data=self.current_data
        )
        self.current_data = []
        self.chain.append(block)
        return block
    @staticmethod
    def confirm_validity(block, previous_block):
        if previous_block.index + 1 != block.index:
            return False
        elif previous_block.compute_hash != block.previous_hash
            return False
        elif block.timestamp <= previous_block.timestamp:
            return False
        return True
    def get_data(self, sender, receiver, amount):
        self.current_data.append({
            'sender': sender,
            'receiver': receiver,
            'amount': amount
        })
        return True
    @staticmethod
    def proof_of_work(last_proof):
        pass
    @property
    def latest_block(self):
        return self.chain[-1]
    def chain_validity(self):
        pass
    def block_mining(self, details_miner):
        self.get_data(
            sender="0", #it implies that this node has created a new block
            receiver=details_miner,
            quantity=1, #creating a new block (or identifying the proof number) is awared with 1
        )
        last_block = self.latest_block
        last_proof_number = last_block.proof_number
        proof_number = self.proof_of_work(last_proof_number)
        last_hash = last_block.compute_hash
        block = self.build_block(proof_number, last_hash)
        return vars(block)
    def create_node(self, address):
        self.nodes.add(address)
        return True
    @staticmethod
    def get_block_object(block_data):
        return Block(
            block_data['index'],
            block_data['proof_number'],
            block_data['previous_hash'],
            block_data['data'],
            timestamp=block_data['timestamp']
        )
blockchain = BlockChain()
```

```
print("GET READY MINING ABOUT TO START")
print(blockchain.chain)
last_block = blockchain.latest_block
last_proof_number = last_block.proof_number
proof_number = blockchain.proof_of_work(last_proof_number)
blockchain.get_data(
    sender="0", #this means that this node has constructed another block
    receiver="Farhan",
    amount=1, #building a new block (or figuring out the proof number) is awarded with 1
)
last_hash = last_block.compute_hash
block = blockchain.build_block(proof_number, last_hash)
print("WOW, MINING HAS BEEN SUCCESSFUL!")
print(blockchain.chain)
```

Output:



```
================== RESTART: E:\block chain practical\prac7.py ==================
GET READY MINING ABOUT TO START
[0 - 0 - 0 - [] - 1684168762.967008]
WOW, MINING HAS BEEN SUCCESSFUL!
[0 - 0 - 0 - [] - 1684168762.967008, 1 - None - 8f32e80fcaf46f92f8a4e93ed3d743d4f500
2e6db060f25a341000704a8a0ba1 - [{'sender': '0', 'receiver': 'Vishal', 'amount': 1}]
- 1684168762.9815922]
>>>
```

Practical No 8

Aim: - Builds Dapps using Moralis and MetaMask.

Practical No 8

Aim: - Builds Dapps using Moralis and MetaMask.

Code:

Index.html

```
<html>
  <head>
    <!-- Moralis SDK code -->
    <script src="https://cdn.jsdelivr.net/npm/web3@latest/dist/web3.min.js"></script>
    <script src="https://unpkg.com/moralis/dist/moralis.js"></script>
  </head>
  <body>
    <h1>Moralis Gas Stats</h1>
    <button id="btn-login">Moralis Login</button>
    <button id="btn-logout">Logout</button>
    <button id="btn-get-stats">Refresh Stats</button>
    <!-- stats will go here -->
    <ul id="gas-stats"></ul>
    <script>
      // connect to Moralis server
      const serverUrl = "https://smqaqyghrcmw.usemoralis.com:2053/server";
      const appId = "VOorzYHtVy8A7LxIzj8ig4eIA2Kz9iYYCeTfO7Lk";
      Moralis.start({ serverUrl, appId });
      // LOG IN WITH METAMASK
      async function login() {
      let user = Moralis.User.current();
      if (!user) {
      user = await Moralis.authenticate();
      }
      console.log("logged in user:", user);
      getStats();
      }
      // LOG OUT
      async function logOut() {
      await Moralis.User.logOut();
      console.log("logged out");
      }
      // bind button click handlers
      document.getElementById("btn-login").onclick = login;
      document.getElementById("btn-logout").onclick = logOut;
      document.getElementById("btn-get-stats").onclick = getStats;
      // refresh stats
      function getStats() {
      const user = Moralis.User.current();
      if (user) {
      getUserTransactions(user);
```

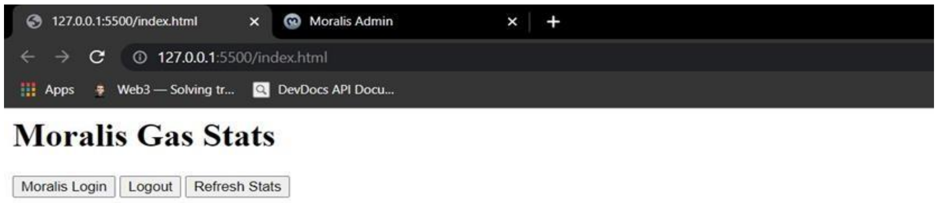
```
        }
        getAverageGasPrices();
        }
        // HISTORICAL TRANSACTIONS
        async function getUserTransactions(user) {
        // create query
        const query = new Moralis.Query("EthTransactions");
        query.equalTo("from_address", user.get("ethAddress"));
        // subscribe to query updates
        const subscription = await query.subscribe();
        handleNewTransaction(subscription);
        // run query
        const results = await query.find();
        console.log("user transactions:", results);
        }
        // REAL-TIME TRANSACTIONS
        async function handleNewTransaction(subscription) {
        // log each new transaction
        subscription.on("create", function (data) {
        console.log("new transaction: ", data);
        });
        }
        // CLOUD FUNCTION
        async function getAverageGasPrices() {
        const results = await Moralis.Cloud.run("getAvgGas");
        console.log("average user gas prices:", results);
        renderGasStats(results);
        }
        function renderGasStats(data) {
        const container = document.getElementById("gas-stats");
        container.innerHTML = data
        .map(function (row, rank) {
        return `<li>#${rank + 1}: ${Math.round(row.avgGas)} gwei</li>`;
        })
        .join("");
        }
        //get stats on page load
        getStats();
      </script>
  </body>
</html>
```

Cloud Function on moralis server

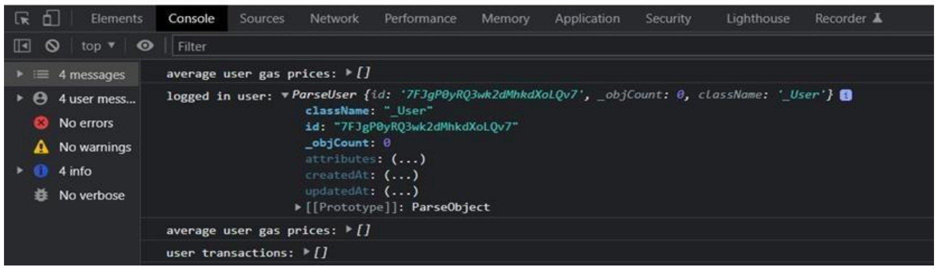
```
Moralis.Cloud.define("getAvgGas", async function (request) {

const query = new Moralis.Query("EthTransactions");
```

```
  const pipeline = [
    {
      group: {
        // group by "from_address"
        objectId: "$from_address",
        // add computed property avgGas
        // get average and convert wei to gwei
        avgGas: { $avg: { $divide: ["$gas_price", 1000000000] } },
      },
    },
    { sort: { avgGas: -1 } }, // sort by avgGas high to low
    { limit: 10 }, // only return top 10 results
  ];
  // the master key is required for aggregate queries
  const results = await query.aggregate(pipeline, { useMasterKey: true });
  return results;
});
```

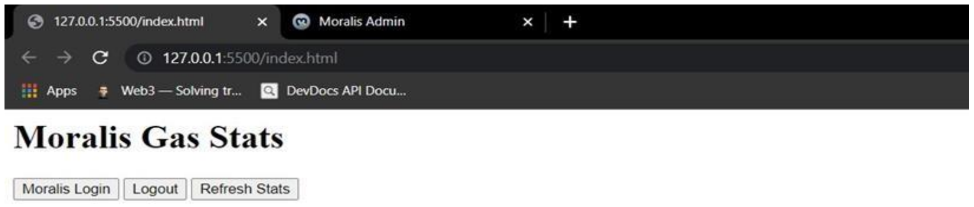
Output:

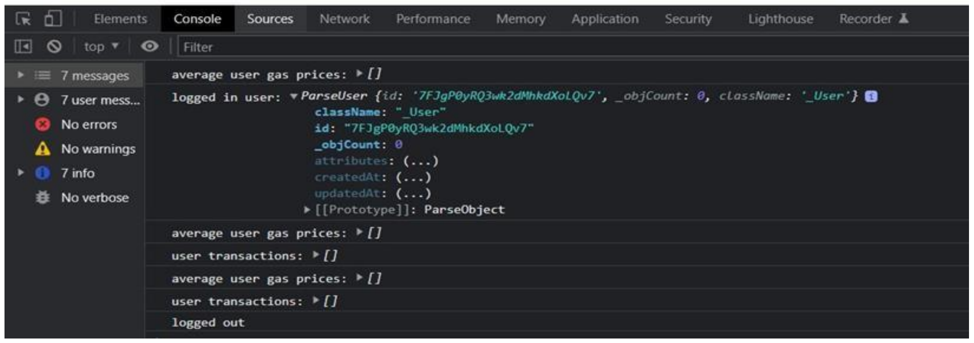


Moralis Gas Stats



```
average user gas prices: ▸[]
logged in user: ▾ParseUser {id: '7FJgP0yRQ3wk2dMhkdXoLQv7', _objCount: 0, className: '_User'}
                    className: "_User"
                    id: "7FJgP0yRQ3wk2dMhkdXoLQv7"
                    _objCount: 0
                    attributes: (...)
                    createdAt: (...)
                    updatedAt: (...)
                  ▸[[Prototype]]: ParseObject
average user gas prices: ▸[]
user transactions: ▸[]
```





```
average user gas prices: ▸[]
logged in user: ▾ParseUser {id: '7FJgP0yRQ3wk2dMhkdXoLQv7', _objCount: 0, className: '_User'}
                    className: "_User"
                    id: "7FJgP0yRQ3wk2dMhkdXoLQv7"
                    _objCount: 0
                    attributes: (...)
                    createdAt: (...)
                    updatedAt: (...)
                  ▸[[Prototype]]: ParseObject
average user gas prices: ▸[]
user transactions: ▸[]
average user gas prices: ▸[]
user transactions: ▸[]
logged out
```