BLOCKCHAIN

INDEX

Sr. No	Practicals	Sign
1.	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.	
b.	A transaction class to send and receive money and test it.	
c.	Create multiple transactions and display them.	
d.	Create a blockchain, a genesis block and execute it.	
e.	Create a mining function and test it.	
f.	Add blocks to the miner and dump the blockchain.	
2.	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.	
b.	Functions, Function Modifiers, View functions, Pure Functions, Fallback Function, Function Overloading, Mathematical functions, Cryptographic functions.	
3.	Implement and demonstrate the use of the following in Solidity:	
a.	Withdrawal Pattern, Restricted Access.	
b.	Contracts, Inheritance, Constructors, Abstract Contracts, Interfaces.	
c.	Libraries, Assembly, Events, Error handling.	
4.	Install hyperledger fabric and composer. Deploy and execute the application.	
5.	Demonstrate the running of the blockchain node.	
6.	Demonstrate the use of Bitcoin Core API.	
7.	Create your own blockchain and demonstrate its use.	

Practical 1

Aim: Write the following programs for Blockchain in Python:

1a) A simple client class that generates the private and public keys by using the built in Python RSA algorithm 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.PublicKey import RSA
from Crypto.Signature import PKCS1_v1_5
import binascii

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(se
                                        lf._private_key)
    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER'
)).decode('ascii')
Dinesh = Client()
print (Dinesh.identity)
```

Output:


30819f300d06092a864886f70d010101050003818d00308189028181 00eea39d3e40f737cff2050d40515a4833c80cdd9073d88a3629aef45ee77793c78197eb0f8bbf0688c0672e22ee74e691bd53668

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

Code:

```
class Transaction:
                                    def __init__(self,
                                        sender,
                                        recipient,
                                        value):
                                        self.sender =
                                        sender
                                        s
                                        e
                                        l
                                        f
                                        .
                                        r
                                        e
                                        c
                                        i
                                        p
                                        i
                                        e
                                        n
                                        t
                                        =
                                        r
                                        e
                                        c
                                        i
                                        p
                                        i
                                        e
                                        n
                                        t
                                        s
                                        e
                                        l
                                        f
                                        .
                                        v
                                        a
                                        l
                                        u
                                        e
                                        =
                                        v
                                        a
                                        l
```

```
u                                                    e
                                                     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')
Dinesh = Client()
Ramesh = Client()
t = Transaction(Dinesh,Ramesh.identity,5.0)

signature = t.sign_transaction()

print (signature)
```

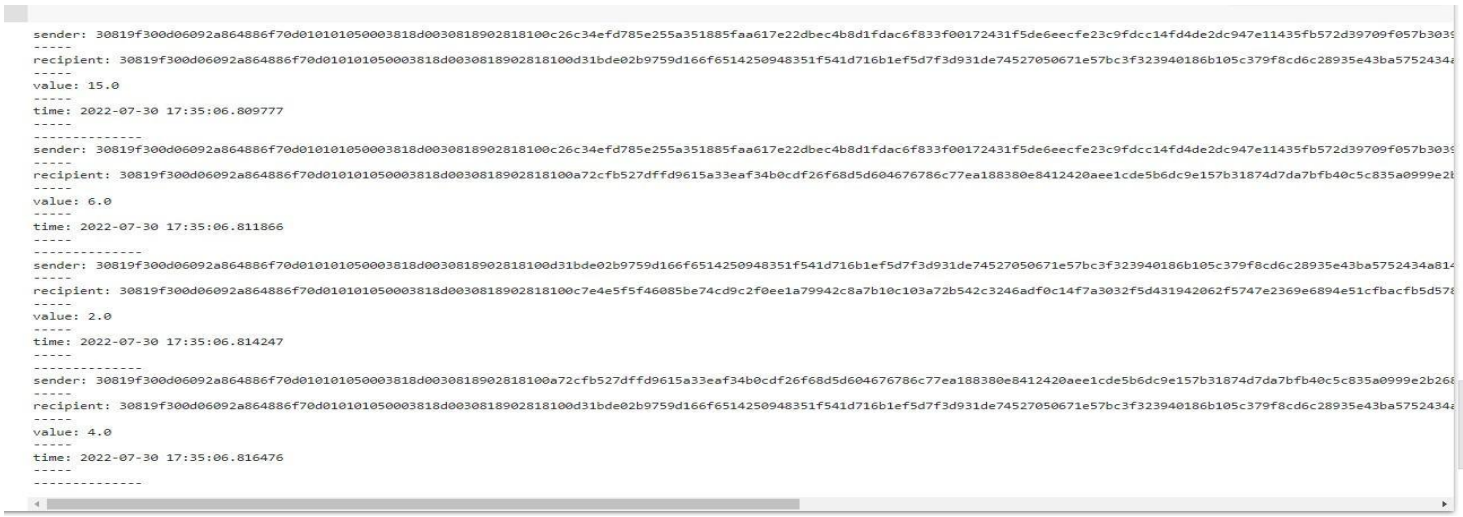
Output:

1c) Create Multiple Transaction and display them.

Code:

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

  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)

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

Output:



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

Code:

```
class Block:
                                    def __init(self):self.verified_transa
```

```
ctions=[]self.previous_block
                                    ock_hash=""self.Nonce=""
```

```
last_block_hash = ""

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 = []

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

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

Output:

```
Number of blocks in the chain: 1
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a4a79ac0645d8483c7c97fb6b67e1bf90f17d87d1f
-----
value: 500.0
-----
time: 2022-07-30 17:50:16.288802
-----
--------------
====================================
```

1e) Create a mining function and test it.

Code:

```
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: "+ dig
est)
    return digest

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

Output:

```
after 238 iterations found nonce: 1124711db6185591392c6a06c24a3c2ebbaeca647fb8374824f939ada27c09e9
after 353 iterations found nonce: 11e0a4b57bb76496ecc6ab5a3c126165bc9dbbaf80f664e7e192a422360c3884
after 419 iterations found nonce: 11280dfd9ab05b3dbfa8699901537329414O8faa4a3b0832819b161f52321c08
after 511 iterations found nonce: 1150944bd7ea429acd052da390b148f27eb881686e0f8bfcdf77f833af878e2e
after 822 iterations found nonce: 110e61d41d94b48f6dfcb6f43f9ceafe2ab7168456dab088e05126d43d0366ef
after 924 iterations found nonce: 11147bf32bafeee4505b5cc40c3b227366d8a41ab3bd740e437c74400b7a8127
'6d80c4cf3cd4fb49aad25deced696c3a48e17974ecbfb1441e2128517f0152b2'
```

1f) Add blocks to the miner and dump the blockchain

Code:

```
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

    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:

```
after 308 iterations found nonce: 111b1715b007a9920e76b241e9a4dde15a50b00faeed353bf9279311deb59804
after 398 iterations found nonce: 11a65418aca0201df85c2eeacabbcf0624092abb64bf6e159ccf5539cbb1cfe1
after 527 iterations found nonce: 11310267f14590e45be2375fe94e881c65f86020e3fcf7844a849241783680a7
after 718 iterations found nonce: 117ba8286d575afe73007e6f588b6f541580bec444d5165a3f71d9e4b623f3fa
after 308 iterations found nonce: 111b1715b007a9920e76b241e9a4dde15a50b00faeed353bf9279311deb59804
after 398 iterations found nonce: 11a65418aca0201df85c2eeacabbcf0624092abb64bf6e159ccf5539cbb1cfe1
after 527 iterations found nonce: 11310267f14590e45be2375fe94e881c65f86020e3fcf7844a849241783680a7
after 718 iterations found nonce: 117ba8286d575afe73007e6f588b6f541580bec444d5165a3f71d9e4b623f3fa
after 308 iterations found nonce: 111b1715b007a9920e76b241e9a4dde15a50b00faeed353bf9279311deb59804
after 398 iterations found nonce: 11a65418aca0201df85c2eeacabbcf0624092abb64bf6e159ccf5539cbb1cfe1
after 527 iterations found nonce: 11310267f14590e45be2375fe94e881c65f86020e3fcf7844a849241783680a7
after 718 iterations found nonce: 117ba8286d575afe73007e6f588b6f541580bec444d5165a3f71d9e4b623f3fa
```

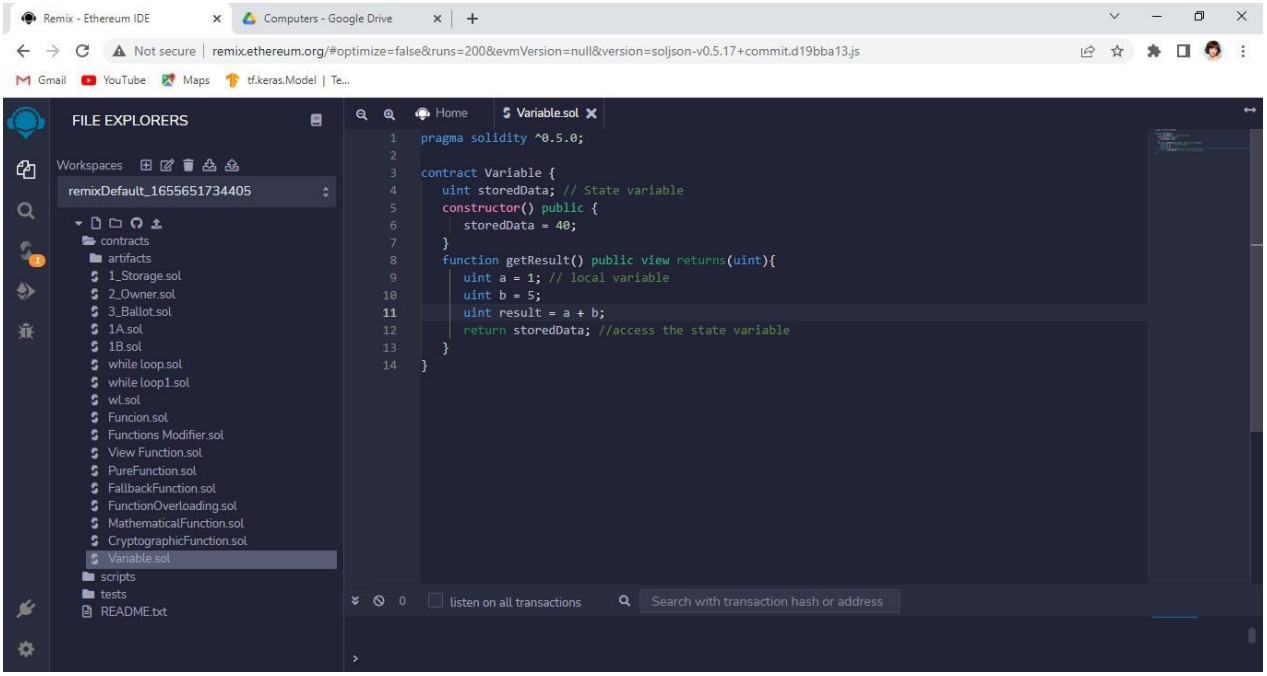
Practical 2

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

2a) Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs, Mappings, Conversions, Ether Units, Special Variables. Variable

Variable

Code:

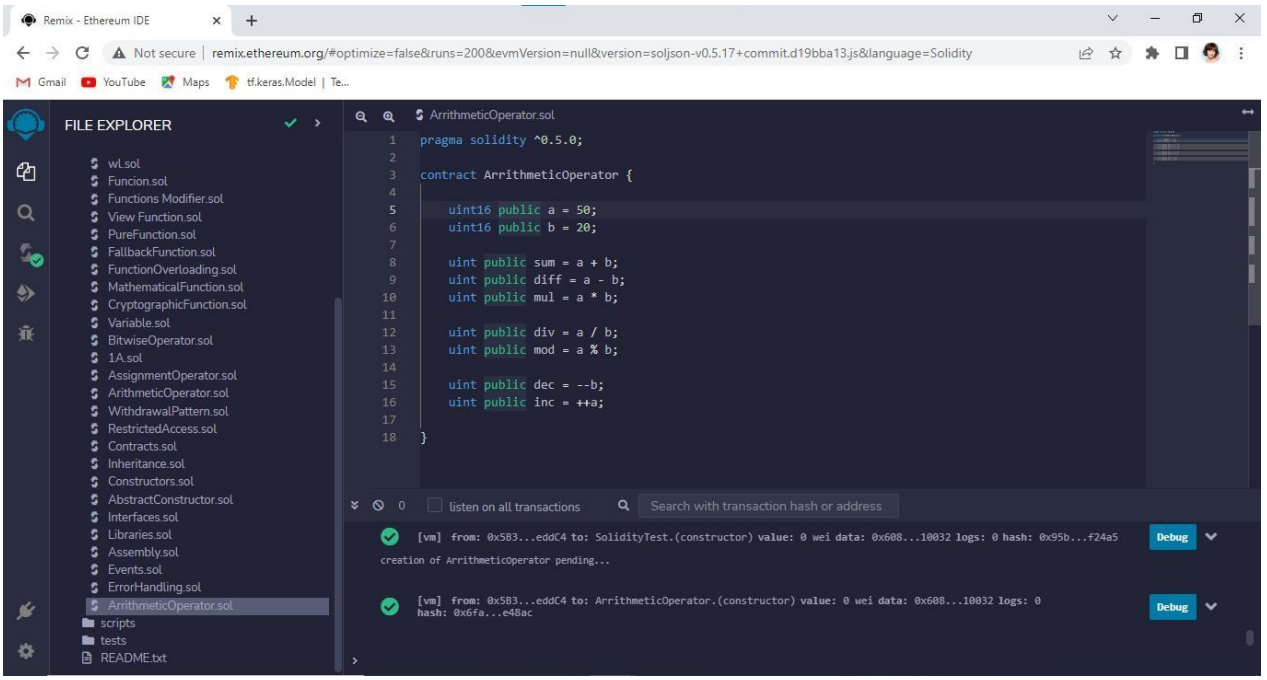


Output:

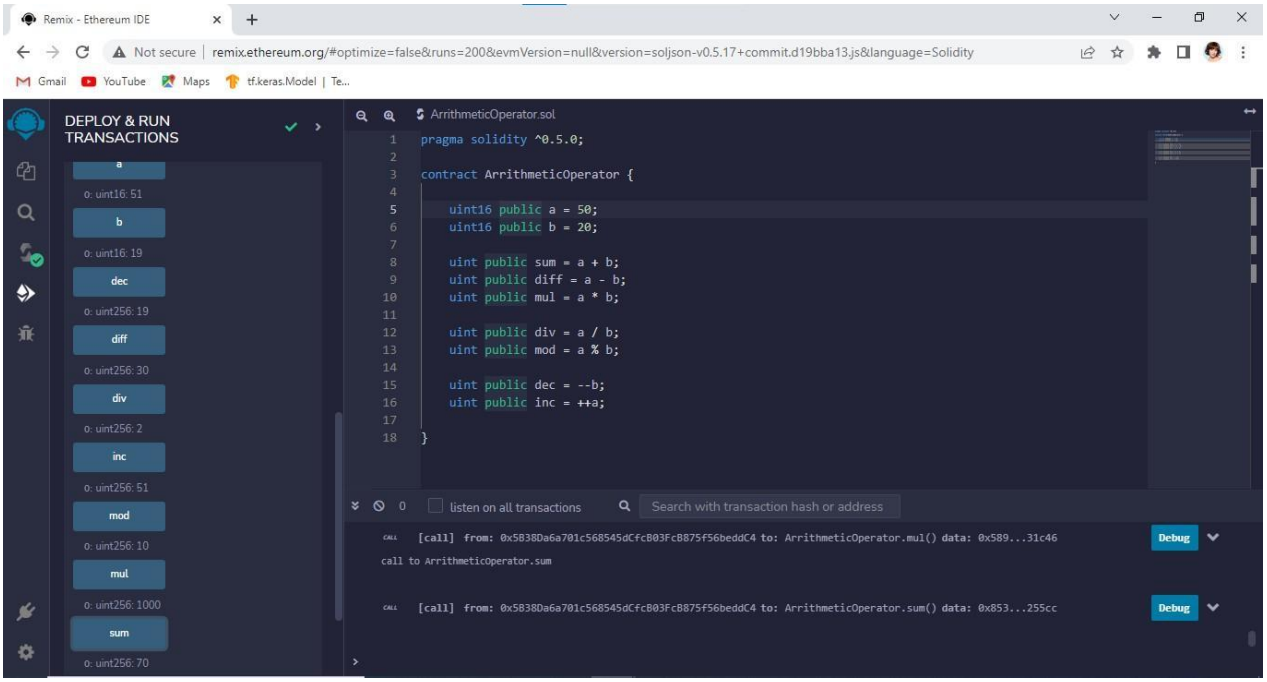
Operators:

Arithmetic Operator

Code:

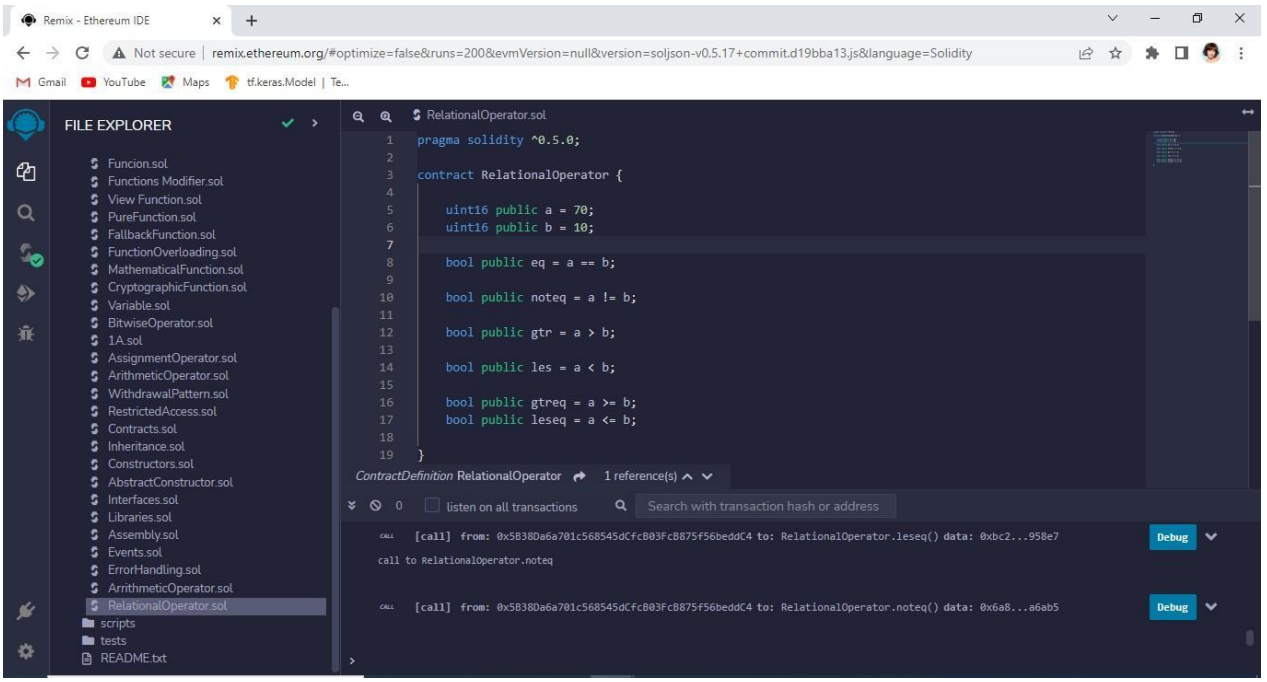


Output:

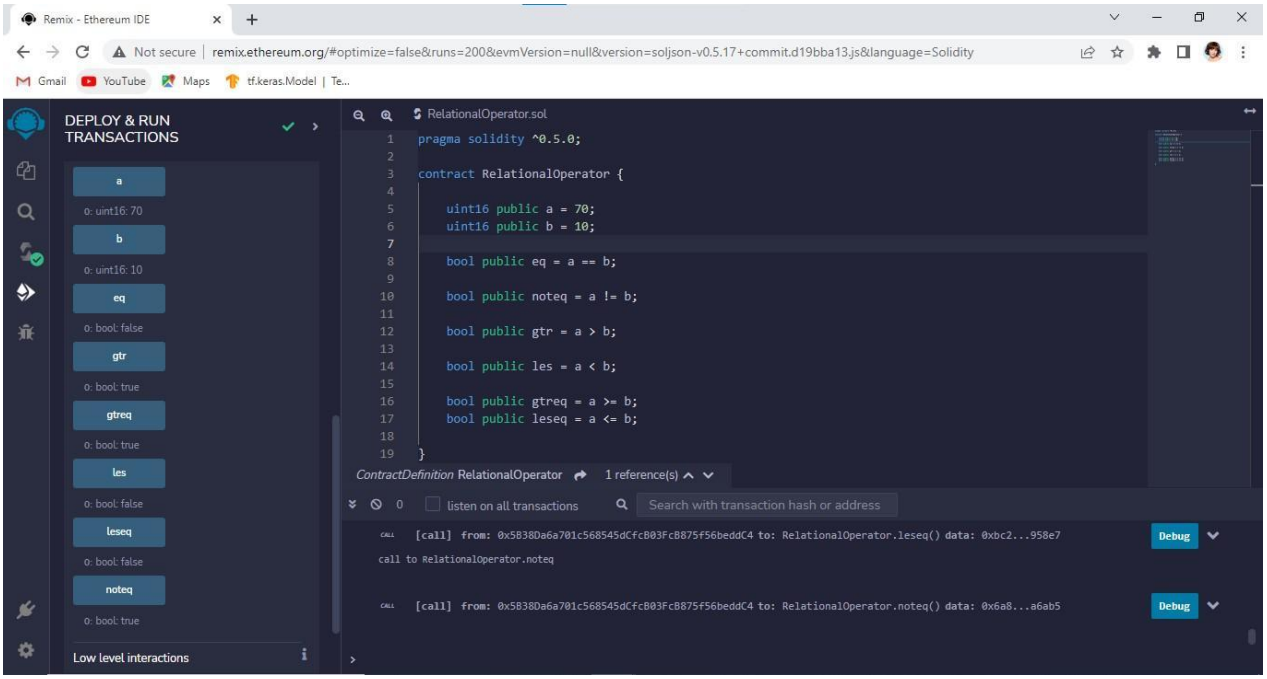


Relational

Operator Code:

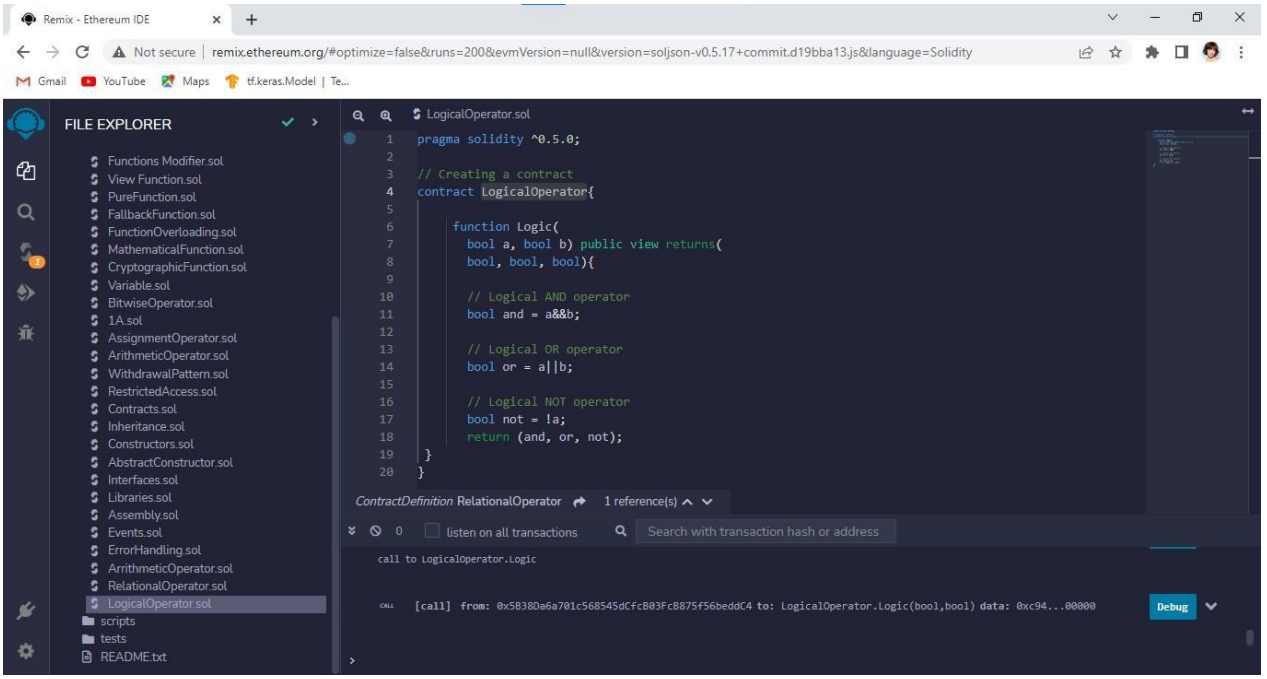


Output:

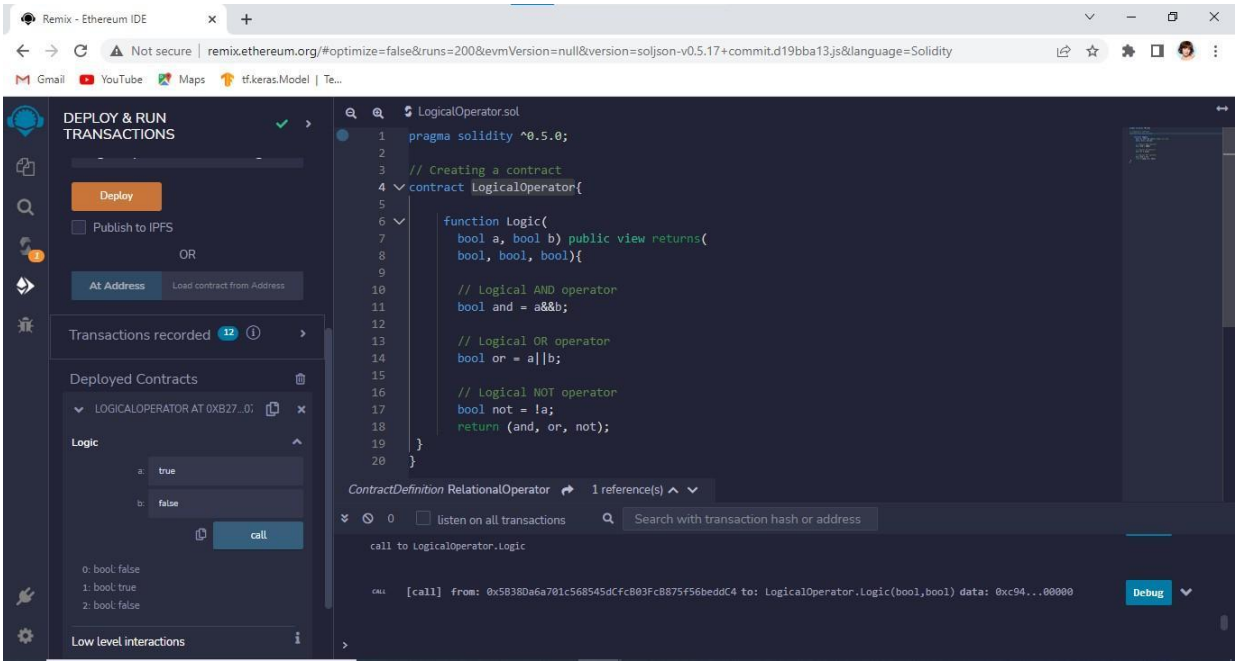


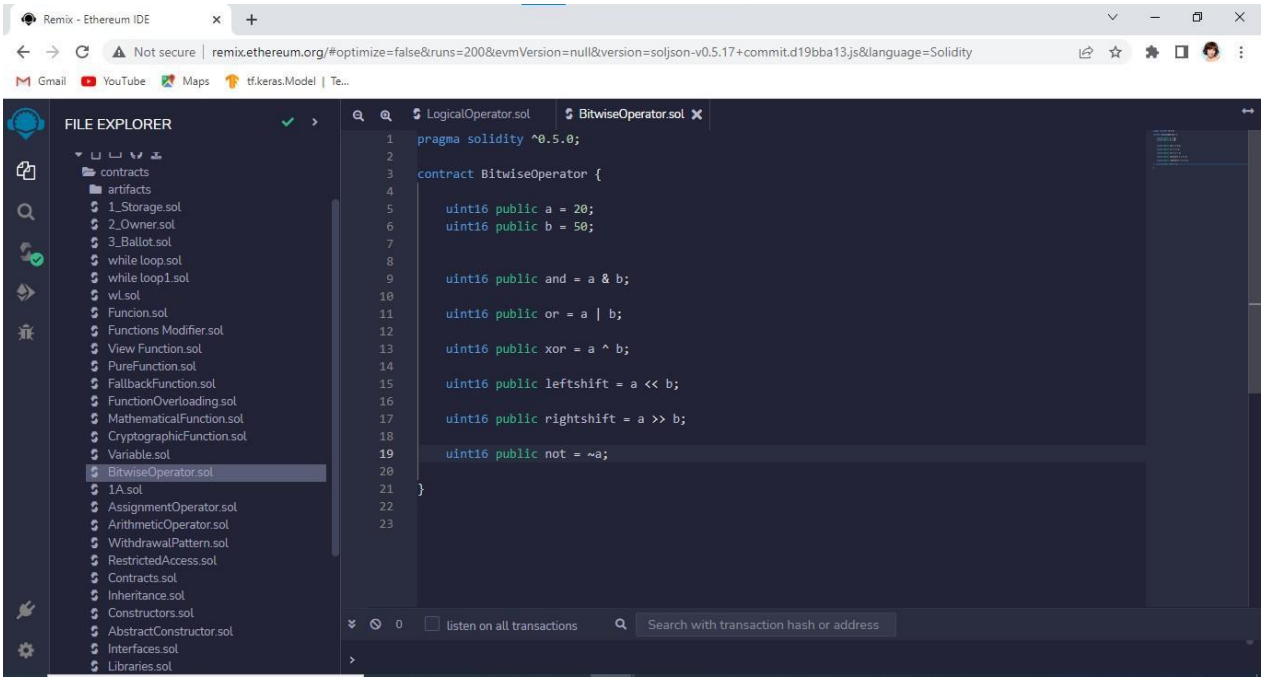
Logical Operator

Code:

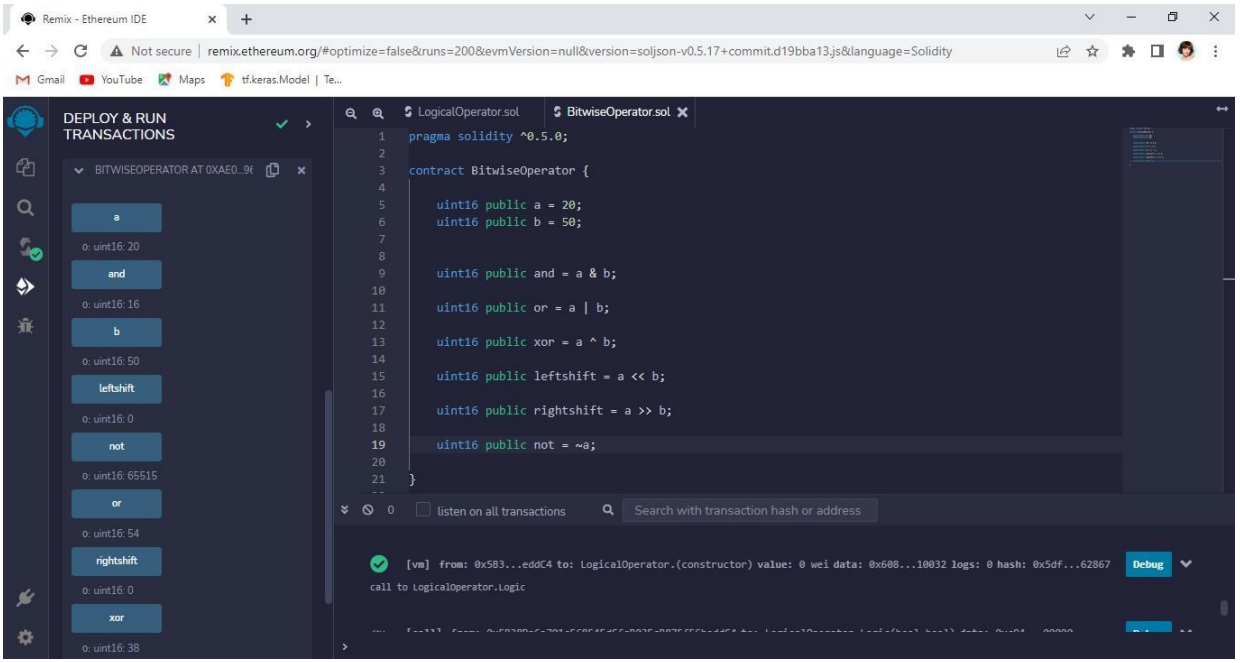

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
scripts
tests
README.txt
LogicalOperator.sol
pragma solidity ^0.5.0;
// Creating a contract
contract LogicalOperator{
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);
}
}
ContractDefinition RelationalOperator
1 reference(s)
listen on all transactions
Search with transaction hash or address
call to LogicalOperator.Logic
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: LogicalOperator.Logic(bool,bool) data: 0xc94...00000
Debug

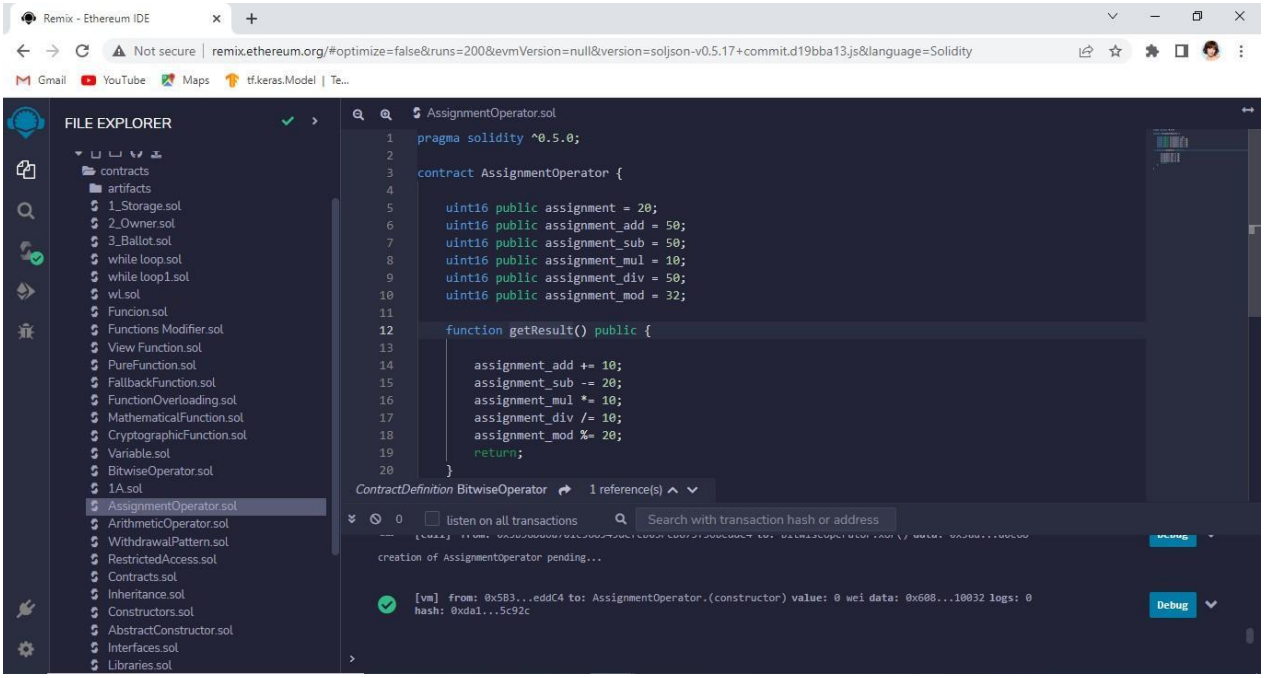
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded
12
Deployed Contracts
LOGICALOPERATOR AT 0XB27...07
Logic
a: true
b: false
call
0: bool: false
1: bool: true
2: bool: false
Low level interactions
LogicalOperator.sol
pragma solidity ^0.5.0;
// Creating a contract
contract LogicalOperator{
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);
}
}
ContractDefinition RelationalOperator
1 reference(s)
listen on all transactions
Search with transaction hash or address
call to LogicalOperator.Logic
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: LogicalOperator.Logic(bool,bool) data: 0xc94...00000
Debug

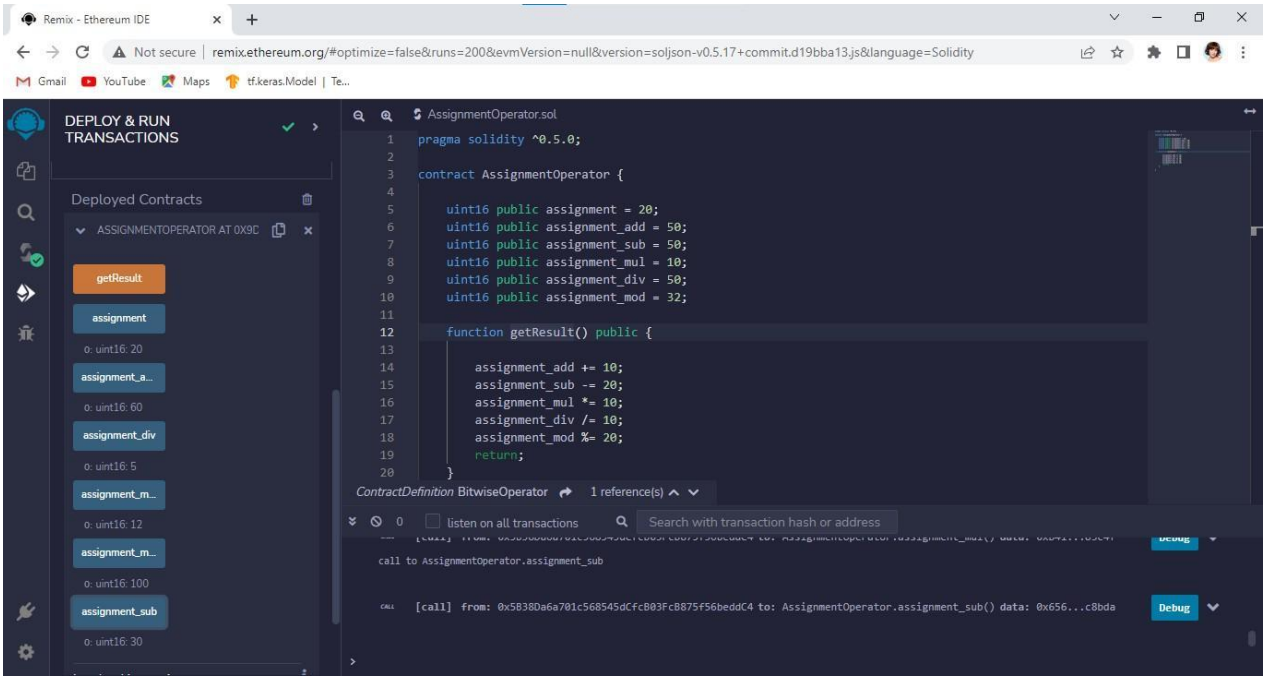

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
LogicalOperator.sol
BitwiseOperator.sol
pragma solidity ^0.5.0;
contract BitwiseOperator {
uint16 public a = 20;
uint16 public b = 50;
uint16 public and = a & b;
uint16 public or = a | b;
uint16 public xor = a ^ b;
uint16 public leftshift = a << b;
uint16 public rightshift = a >> b;
uint16 public not = ~a;
}
listen on all transactions
Search with transaction hash or address

Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
BITWISEOPERATOR AT 0XAE0...9(
a
0: uint16: 20
and
0: uint16: 16
b
0: uint16: 50
leftshift
0: uint16: 0
not
0: uint16: 65515
or
0: uint16: 54
rightshift
0: uint16: 0
xor
0: uint16: 38
LogicalOperator.sol
BitwiseOperator.sol
pragma solidity ^0.5.0;
contract BitwiseOperator {
uint16 public a = 20;
uint16 public b = 50;
uint16 public and = a & b;
uint16 public or = a | b;
uint16 public xor = a ^ b;
uint16 public leftshift = a << b;
uint16 public rightshift = a >> b;
uint16 public not = ~a;
}
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: LogicalOperator.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0x5df...62867
Debug
call to LogicalOperator.Logic



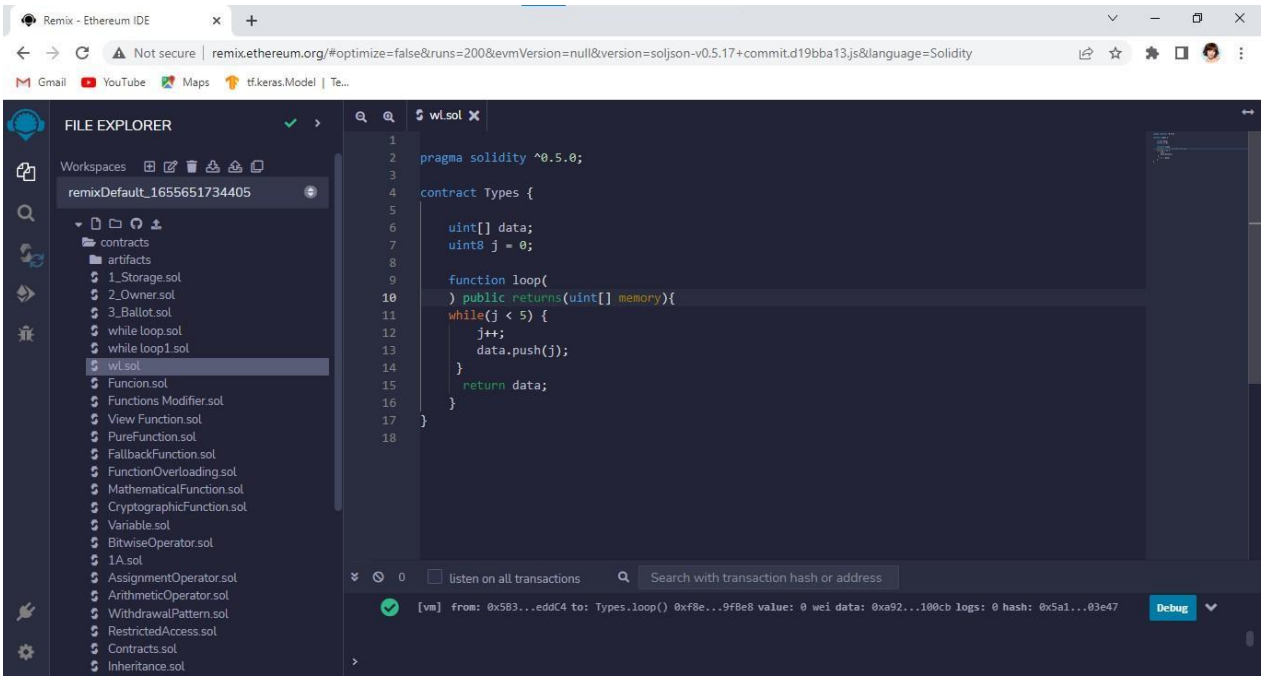
Output:



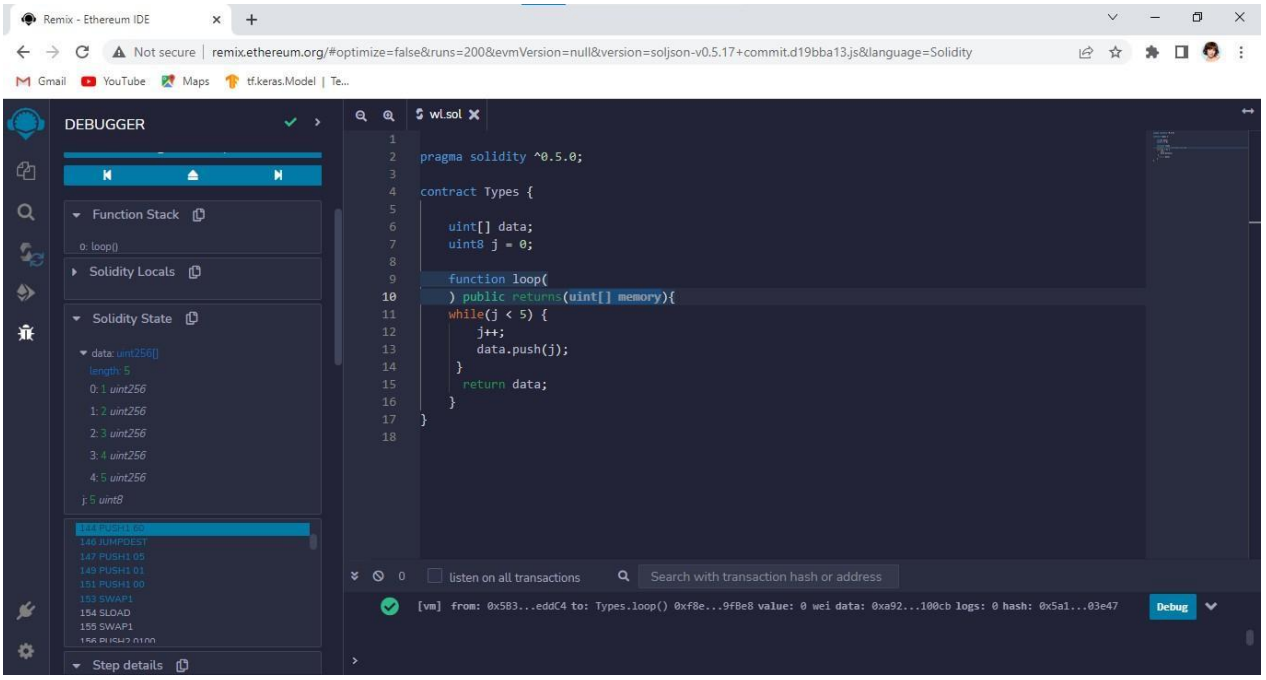
<u>Loops</u>

While Loop

Code:

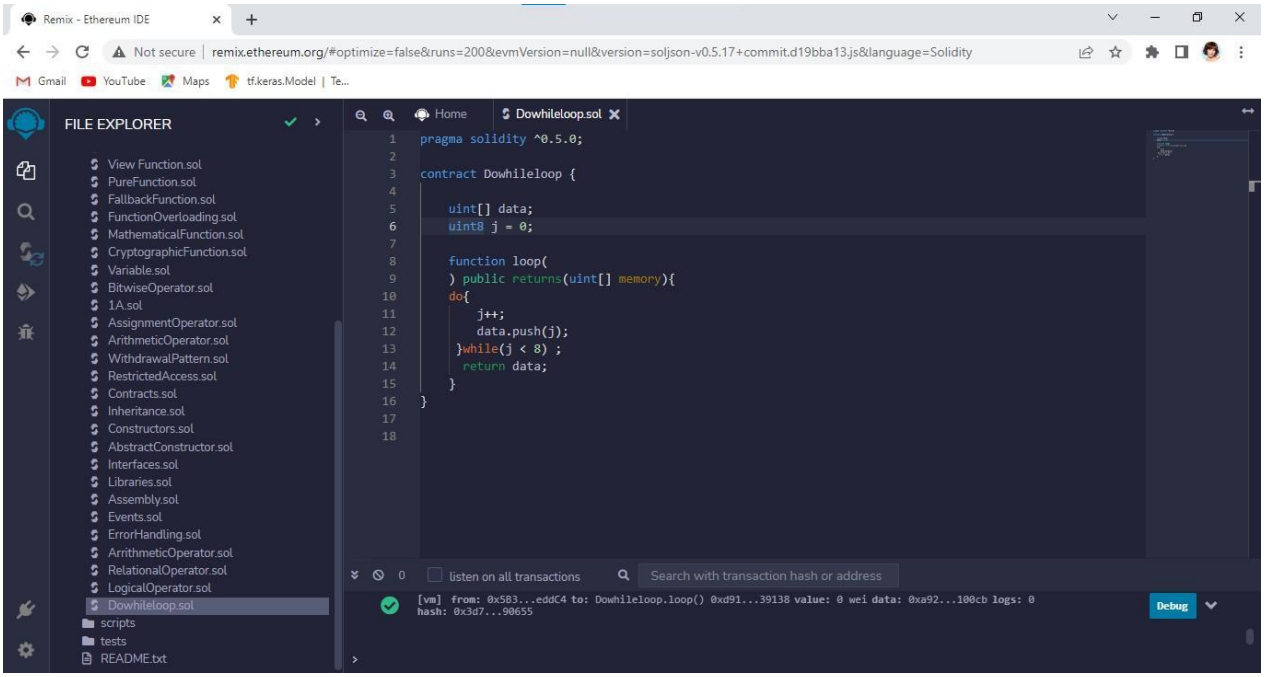


Output:

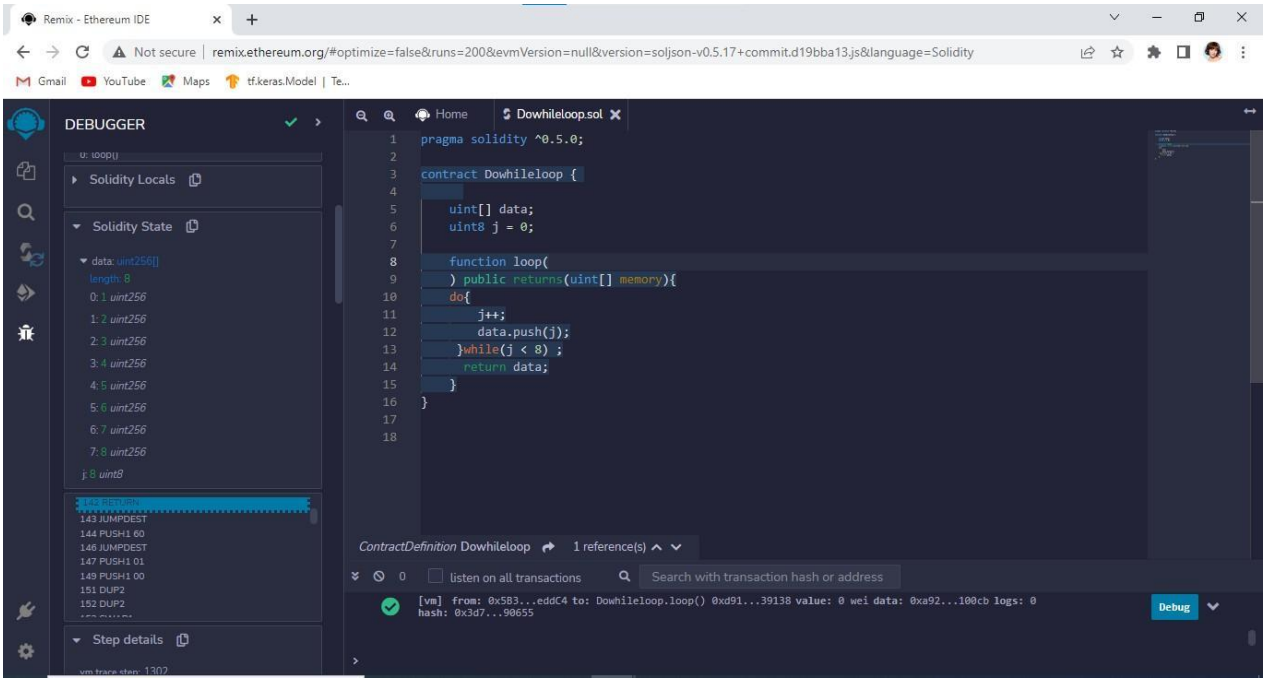


Dowhile loop

Code:

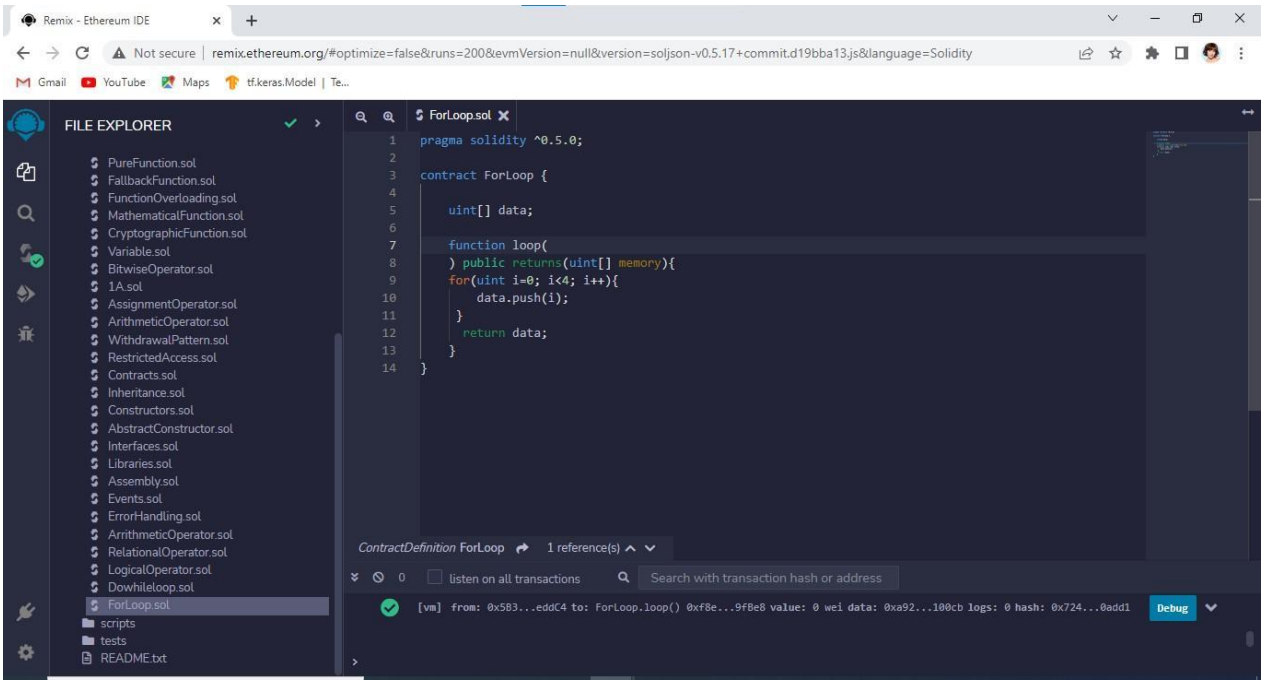


Output:

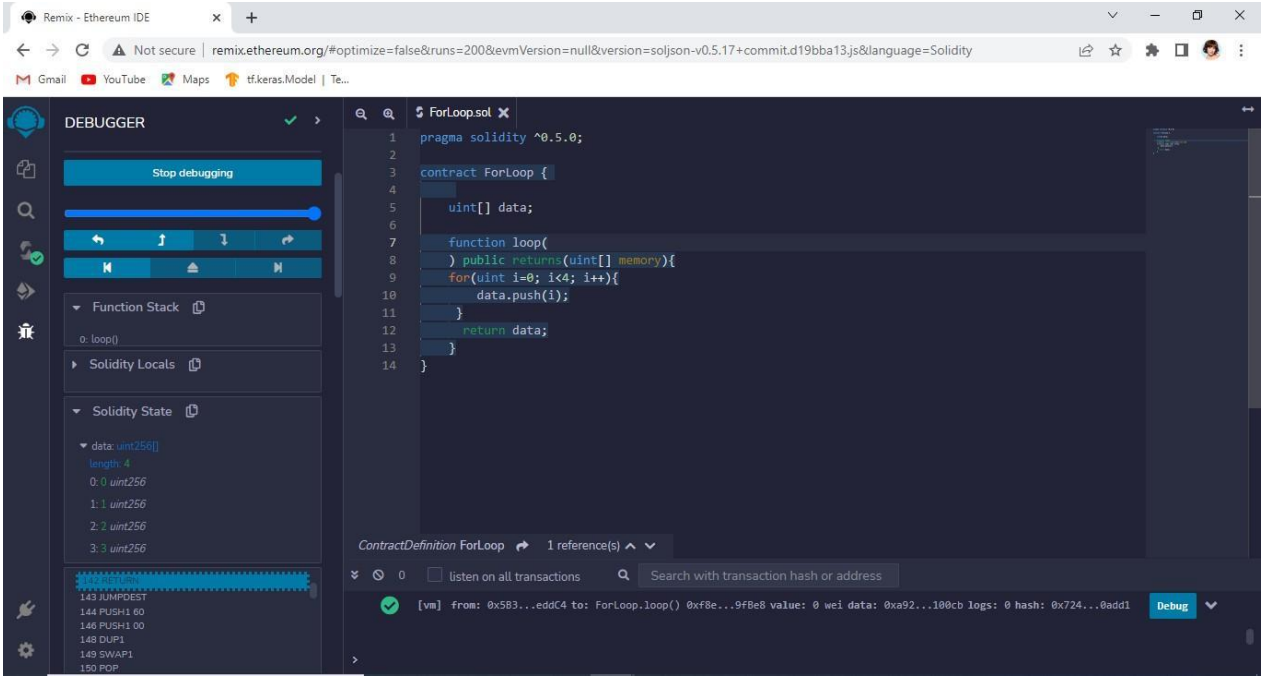


For loop

Code:



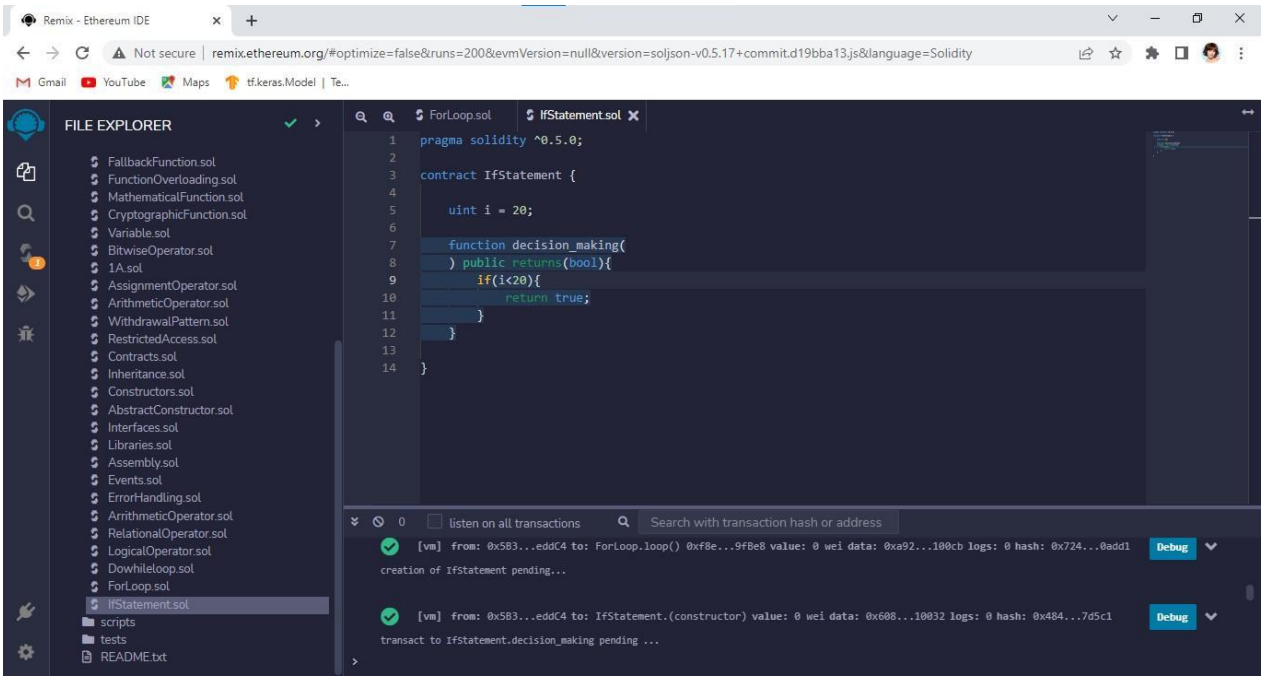
Output:



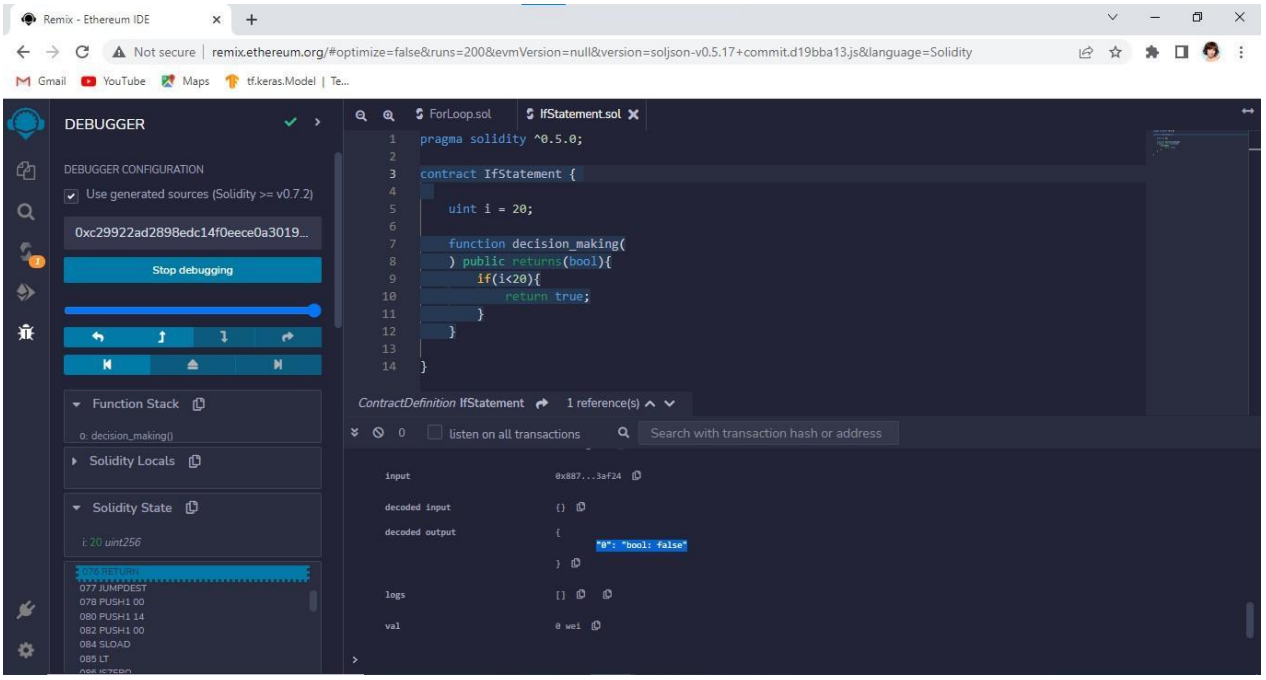
Decision making

If statement

Code:

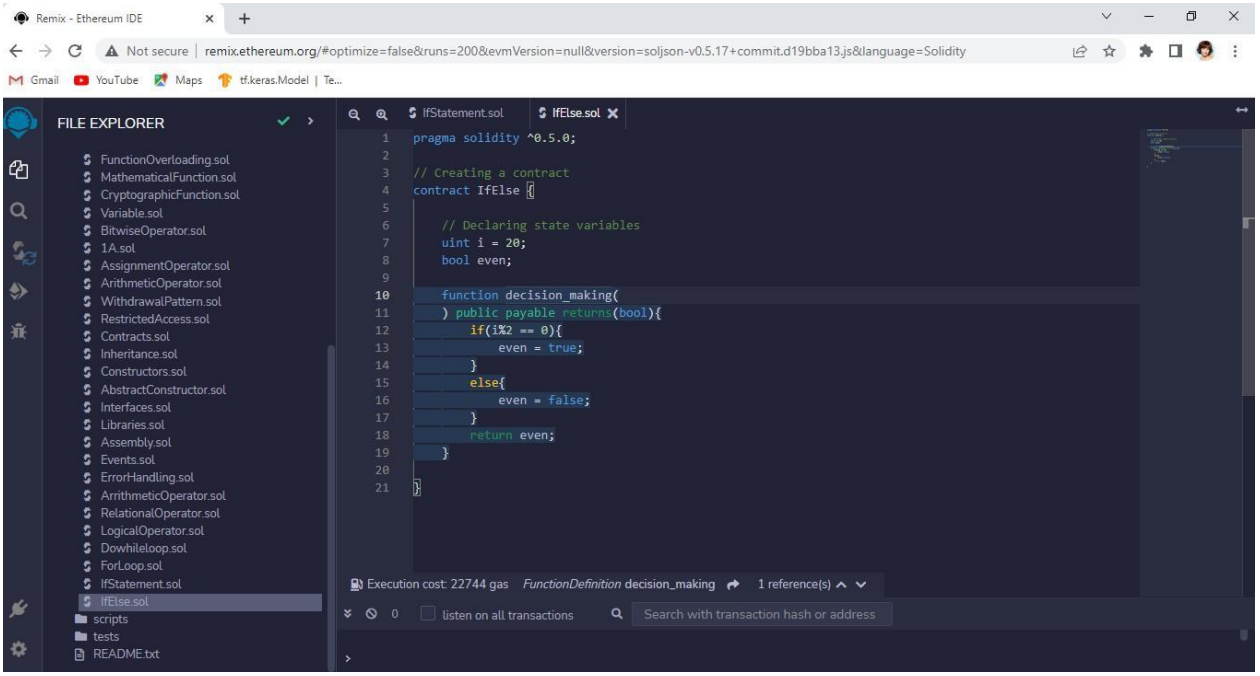


Output:

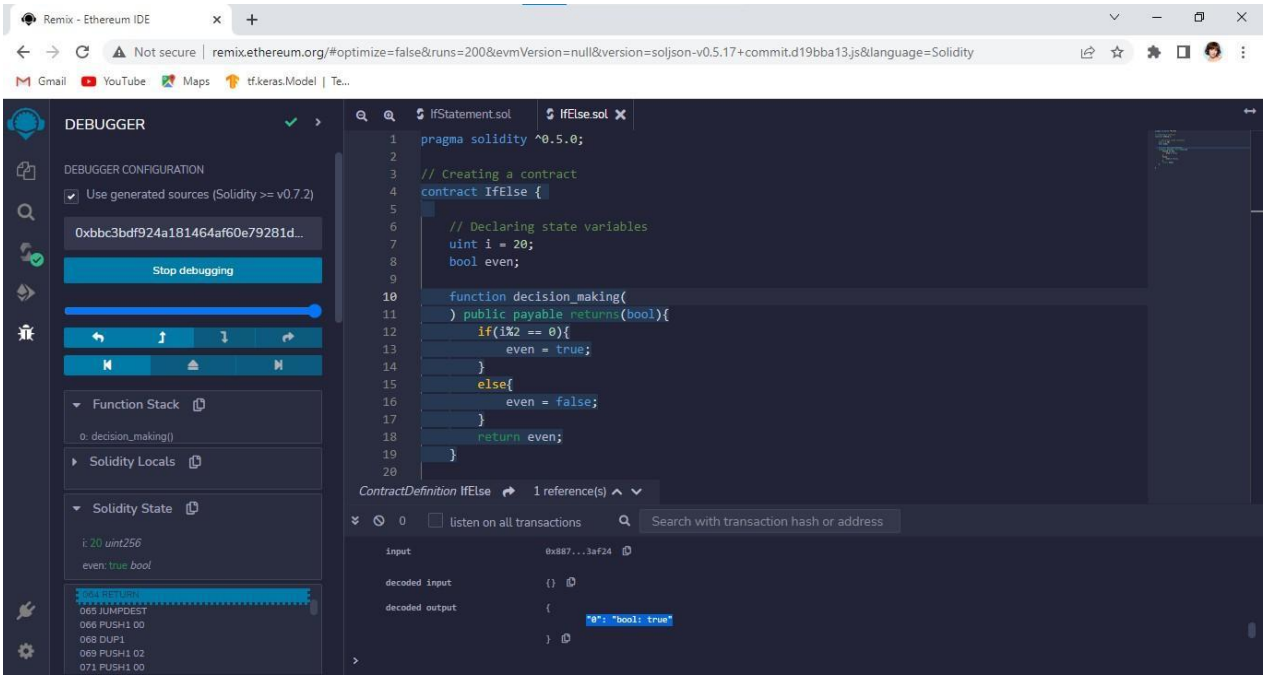


If…else

statement Code:

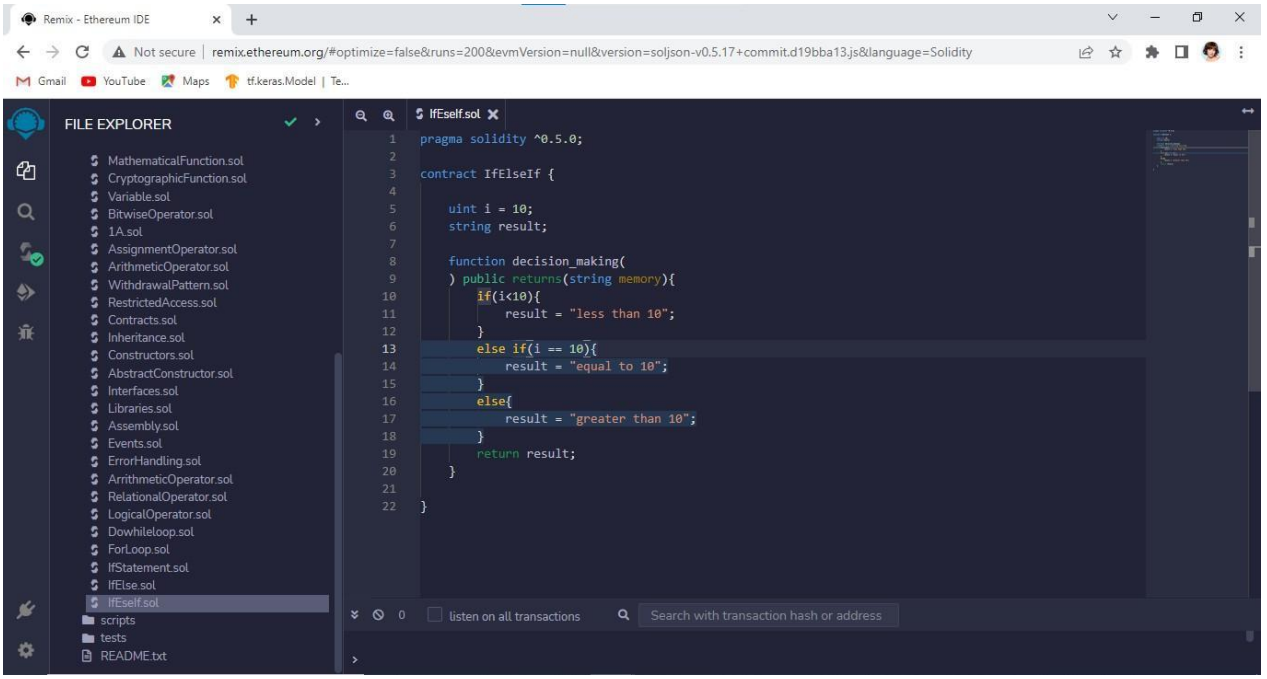


Output:

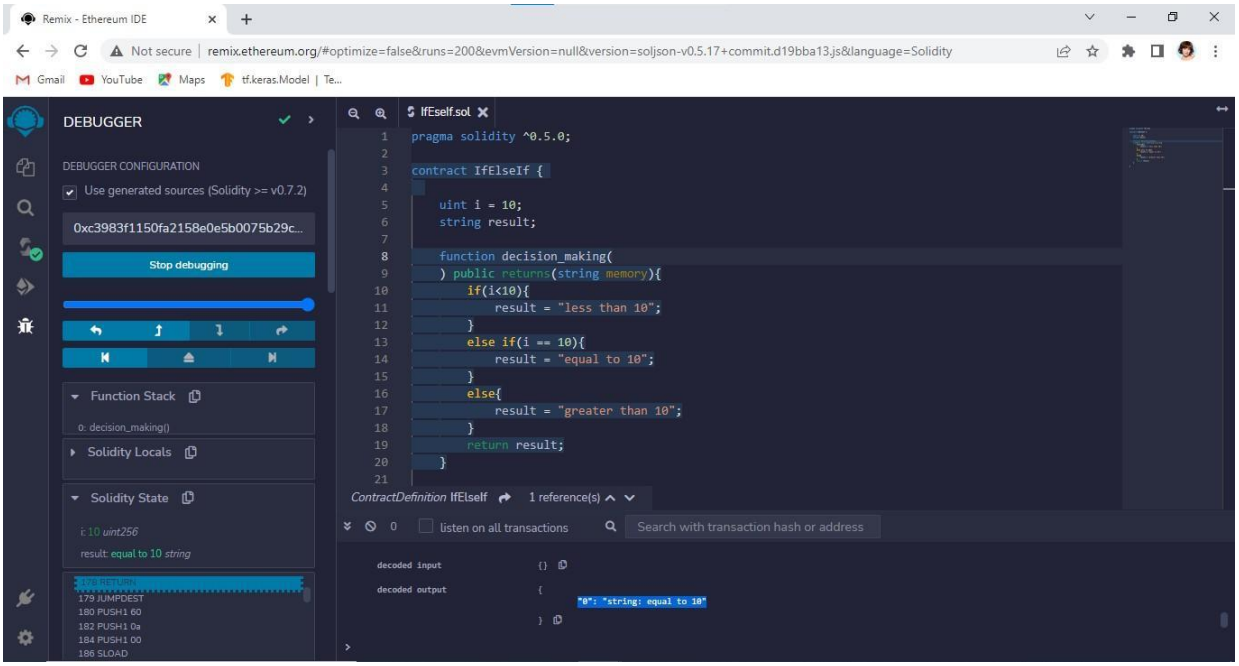


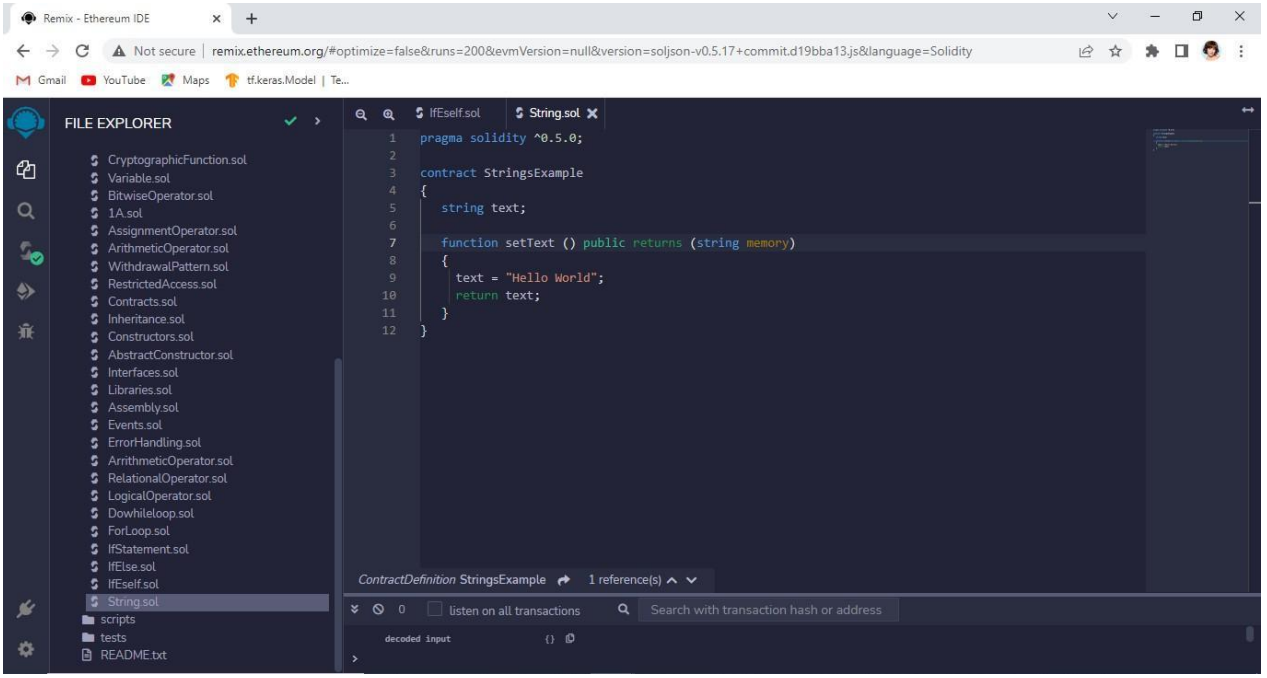
If…else if…else statement

Code:

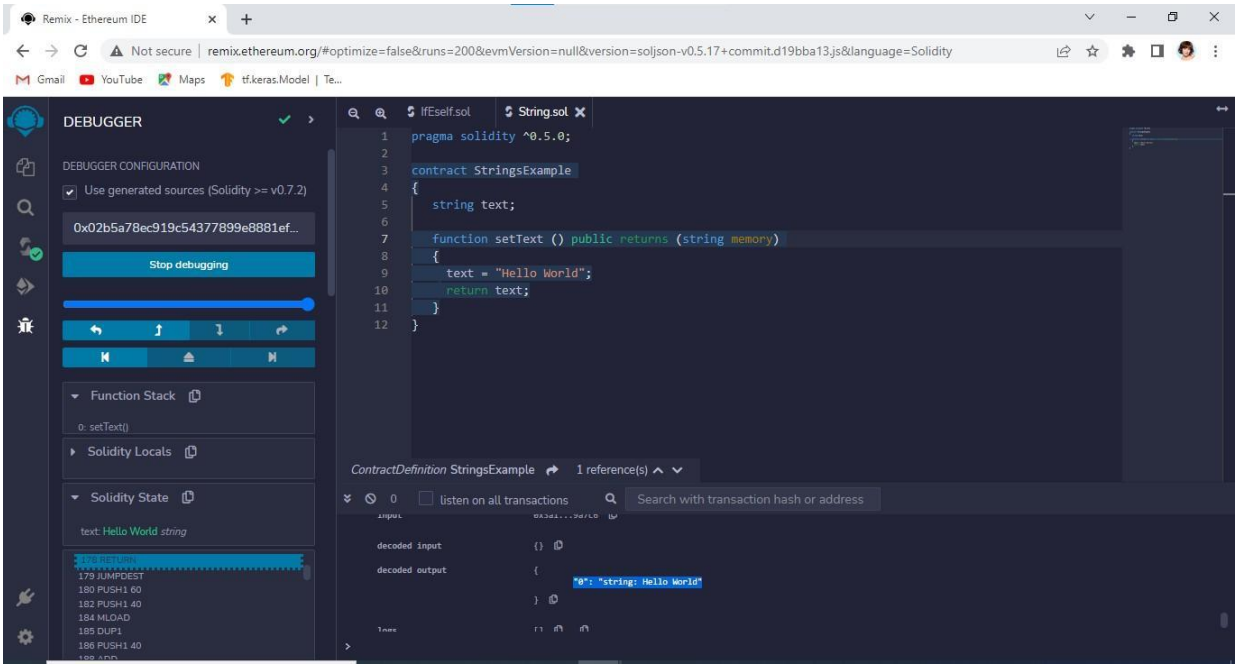

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
scripts
tests
README.txt
IfEself.sol
pragma solidity ^0.5.0;
contract IfElseIf {
uint i = 10;
string result;
function decision_making(
) public returns(string memory){
if(i<10){
result = "less than 10";
}
else if(i == 10){
result = "equal to 10";
}
else{
result = "greater than 10";
}
return result;
}
}
listen on all transactions
Search with transaction hash or address

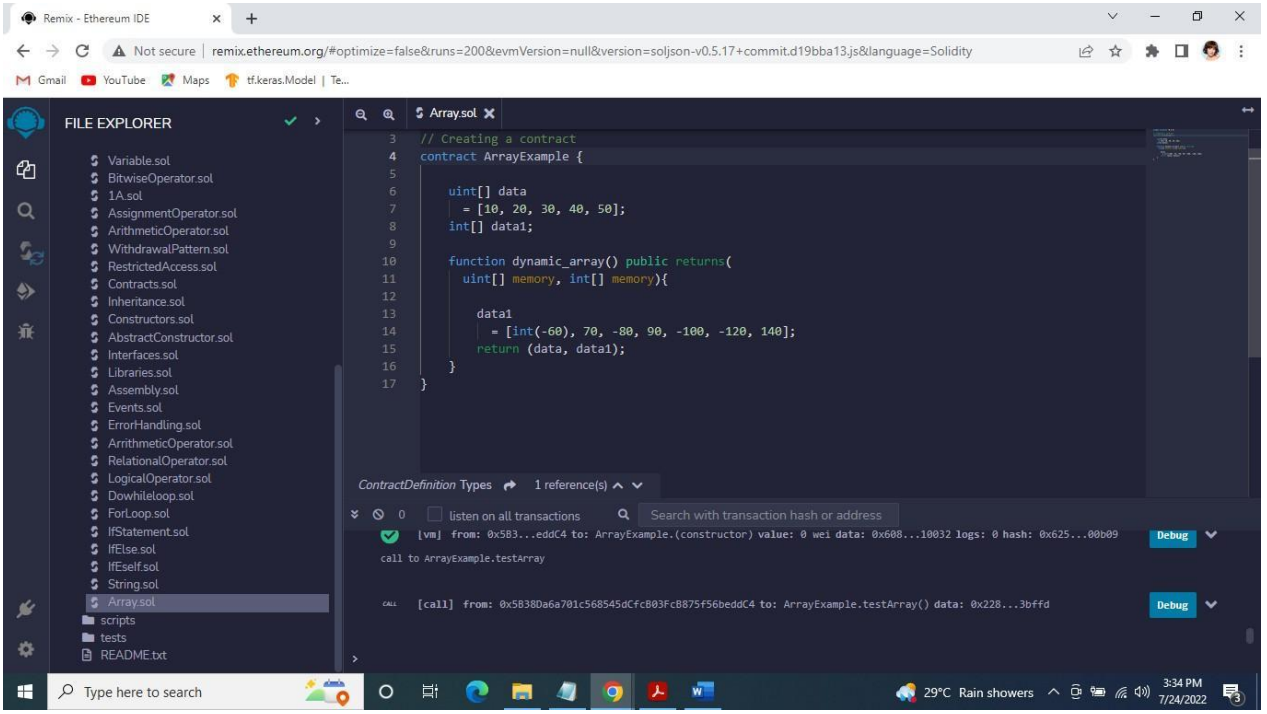
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEBUGGER
DEBUGGER CONFIGURATION
Use generated sources (Solidity >= v0.7.2)
0xc3983f1150fa2158e0e5b0075b29c...
Stop debugging
Function Stack
0: decision_making()
Solidity Locals
Solidity State
i: 10 uint256
result: equal to 10 string
179 JUMPDEST
180 PUSH1 60
182 PUSH1 0a
184 PUSH1 00
186 SLOAD
IfEself.sol
pragma solidity ^0.5.0;
contract IfElseIf {
uint i = 10;
string result;
function decision_making(
) public returns(string memory){
if(i<10){
result = "less than 10";
}
else if(i == 10){
result = "equal to 10";
}
else{
result = "greater than 10";
}
return result;
}
ContractDefinition IfElseIf
1 reference(s)
listen on all transactions
Search with transaction hash or address
decoded input
decoded output
"0": "string: equal to 10"

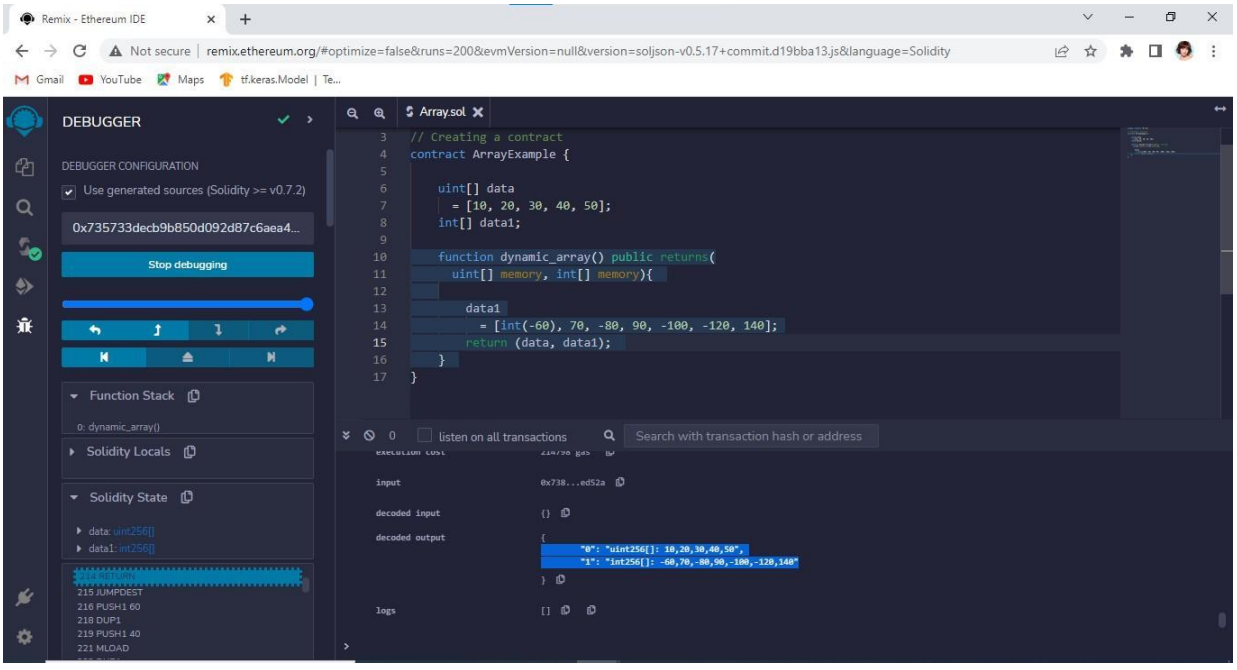

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
scripts
tests
README.txt
IfEself.sol
String.sol
pragma solidity ^0.5.0;
contract StringsExample
{
string text;
function setText () public returns (string memory)
{
text = "Hello World";
return text;
}
}
ContractDefinition StringsExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
decoded input

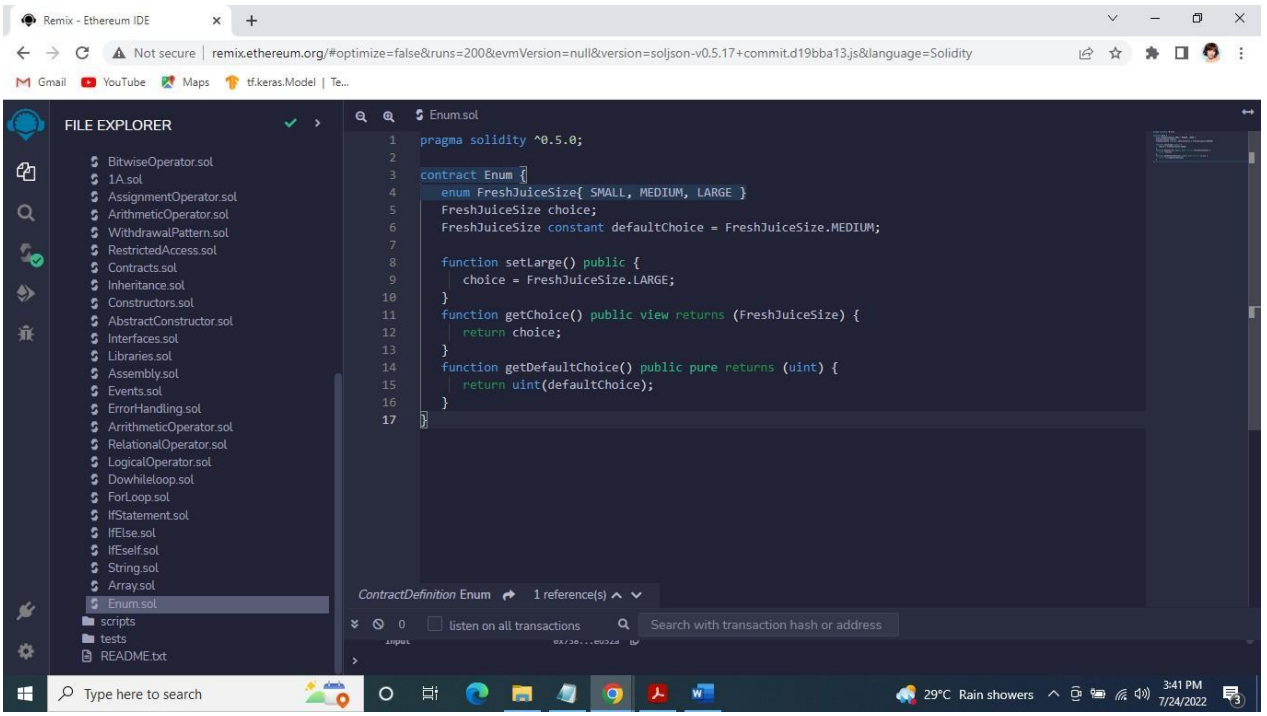
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEBUGGER
DEBUGGER CONFIGURATION
Use generated sources (Solidity >= v0.7.2)
0x02b5a78ec919c54377899e8881ef...
Stop debugging
Function Stack
0: setText()
Solidity Locals
Solidity State
text: Hello World string
179 JUMPDEST
180 PUSH1 60
182 PUSH1 40
184 MLOAD
185 DUP1
186 PUSH1 40
IfEself.sol
String.sol
pragma solidity ^0.5.0;
contract StringsExample
{
string text;
function setText () public returns (string memory)
{
text = "Hello World";
return text;
}
}
ContractDefinition StringsExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
decoded input
decoded output
"0": "string: Hello World"

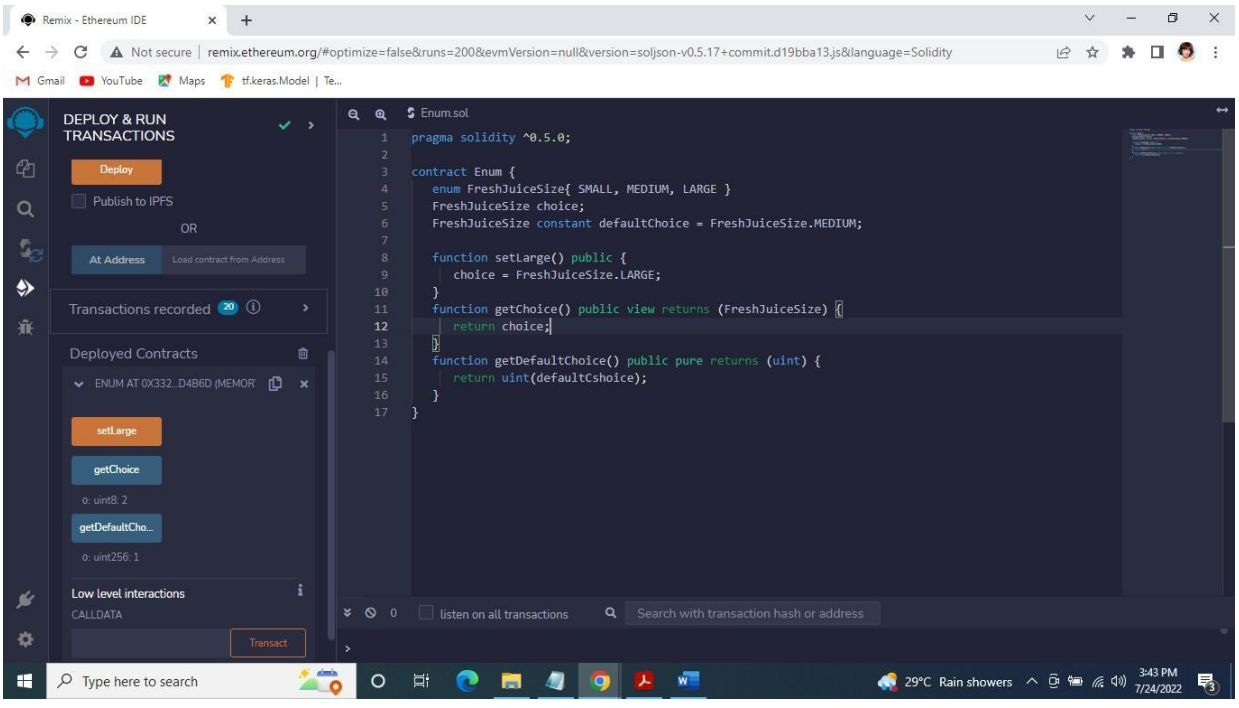

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail YouTube Maps tf.keras.Model | Te...
FILE EXPLORER
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
scripts
tests
README.txt
Array.sol
// Creating a contract
contract ArrayExample {
uint[] data
= [10, 20, 30, 40, 50];
int[] data1;
function dynamic_array() public returns(
uint[] memory, int[] memory){
data1
= [int(-60), 70, -80, 90, -100, -120, 140];
return (data, data1);
}
}
ContractDefinition Types 1 reference(s)
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: ArrayExample.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0x625...00b09
Debug
call to ArrayExample.testArray
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ArrayExample.testArray() data: 0x228...3bffd
Debug
Type here to search
29°C Rain showers
3:34 PM
7/24/2022

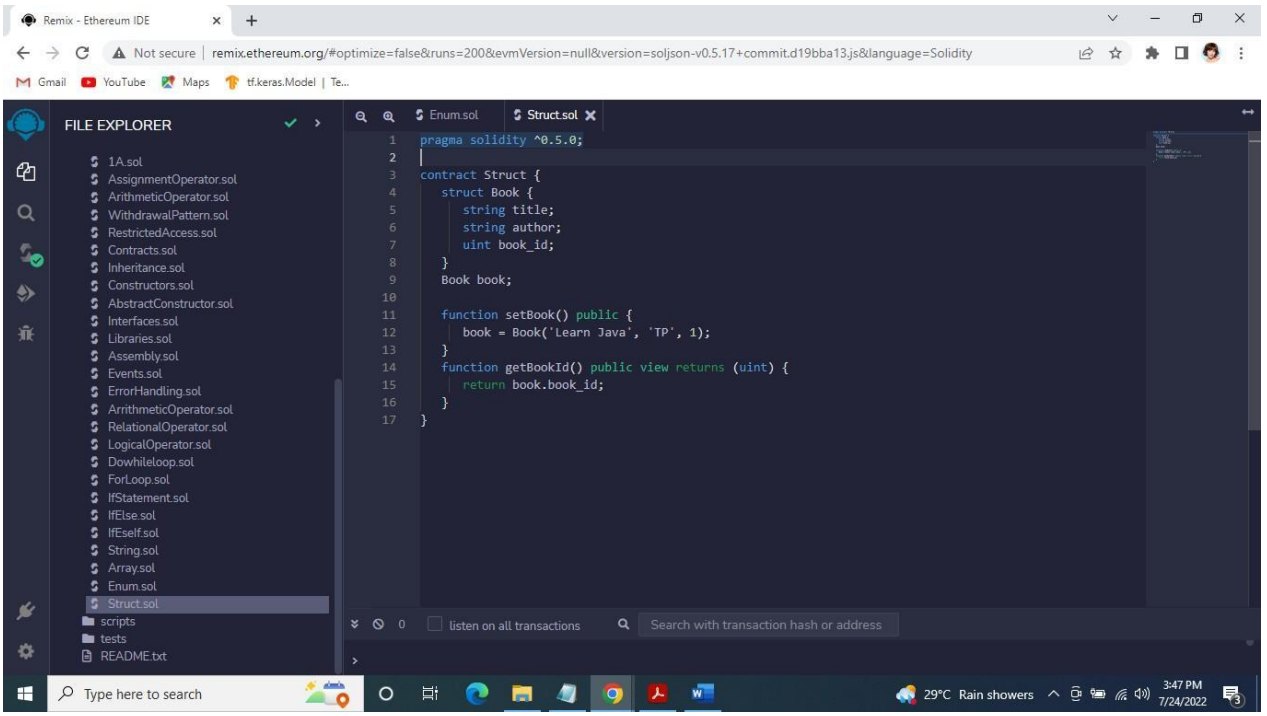
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail YouTube Maps tf.keras.Model | Te...
DEBUGGER
DEBUGGER CONFIGURATION
Use generated sources (Solidity >= v0.7.2)
0x735733decb9b850d092d87c6aea4...
Stop debugging
Function Stack
0: dynamic_array()
Solidity Locals
Solidity State
data: uint256[]
data1: int256[]
215 JUMPDEST
216 PUSH1 60
218 DUP1
219 PUSH1 40
221 MLOAD
Array.sol
// Creating a contract
contract ArrayExample {
uint[] data
= [10, 20, 30, 40, 50];
int[] data1;
function dynamic_array() public returns(
uint[] memory, int[] memory){
data1
= [int(-60), 70, -80, 90, -100, -120, 140];
return (data, data1);
}
}
listen on all transactions
Search with transaction hash or address
input 0x738...ed52a
decoded input {}
decoded output
{
"0": "uint256[]: 10,20,30,40,50",
"1": "int256[]: -60,70,-80,90,-100,-120,140"
}
logs []

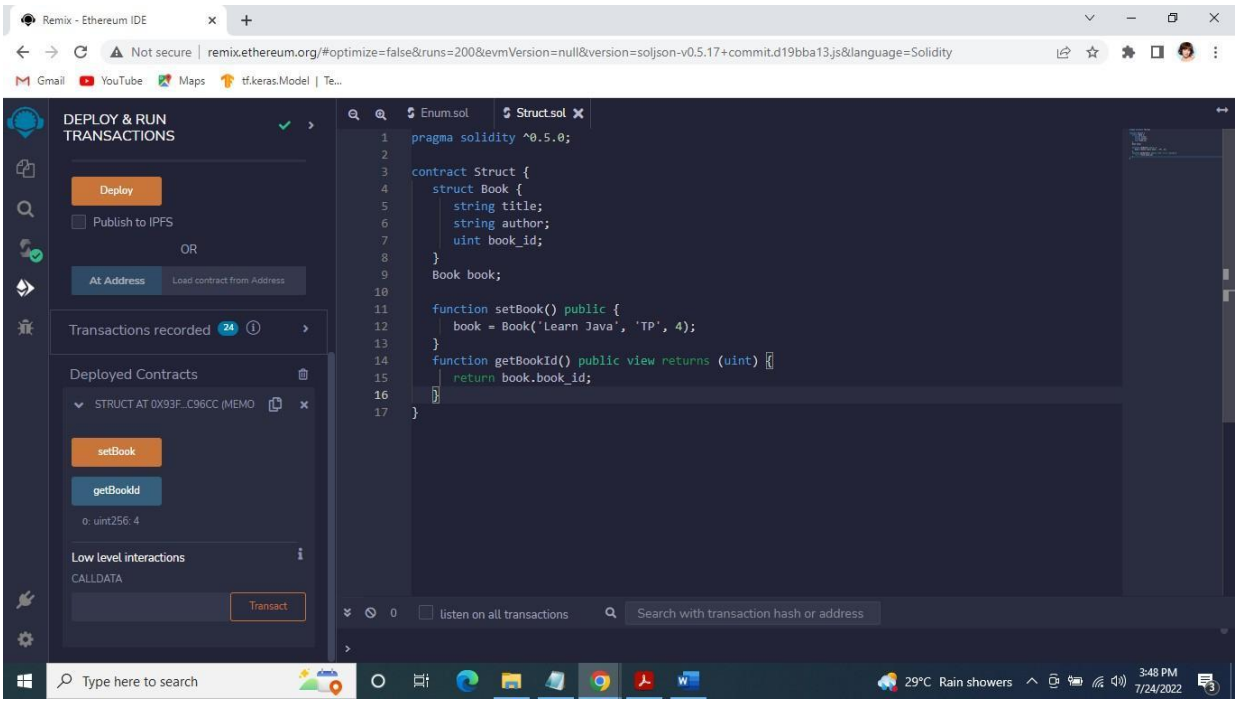

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
Enum.sol
scripts
tests
README.txt
Enum.sol
pragma solidity ^0.5.0;
contract Enum {
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);
}
}
ContractDefinition Enum
1 reference(s)
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:41 PM
7/24/2022

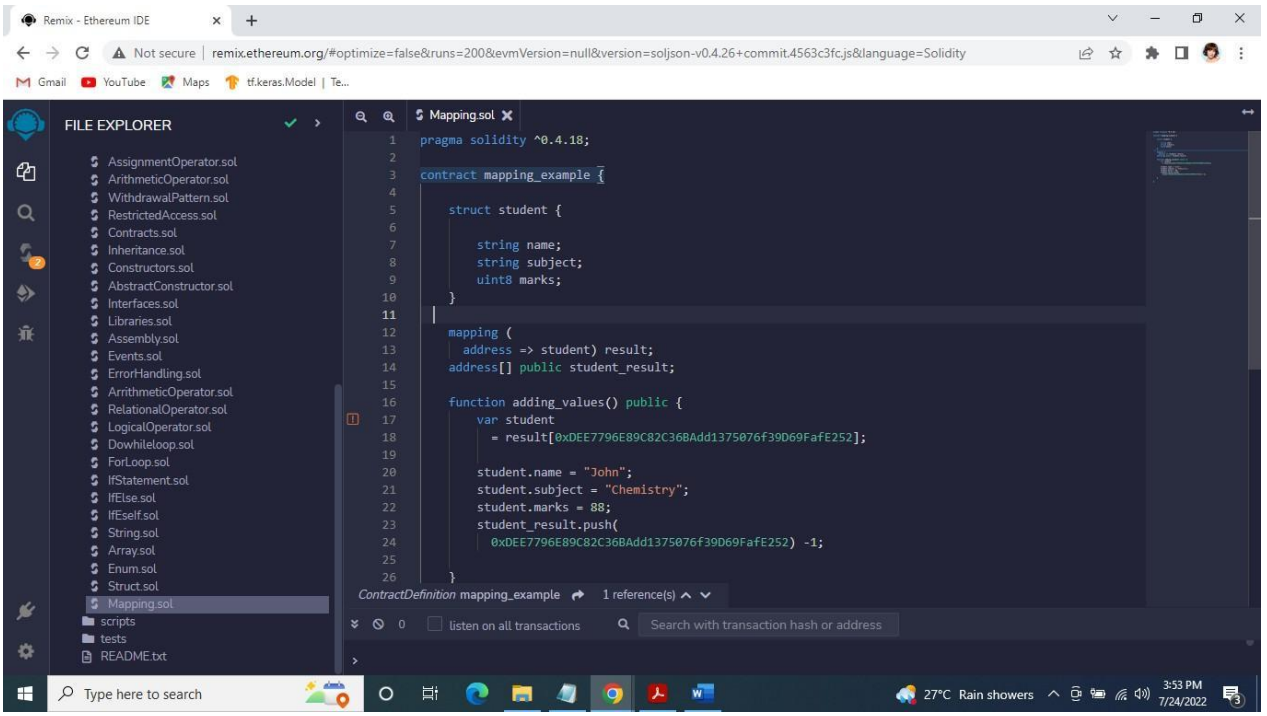
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 20
Deployed Contracts
ENUM AT 0X332...D4B6D (MEMOR
setLarge
getChoice
0: uint8: 2
getDefaultCho...
0: uint256: 1
Low level interactions
CALLDATA
Transact
Enum.sol
pragma solidity ^0.5.0;
contract Enum {
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(defaultCshoice);
}
}
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:43 PM
7/24/2022

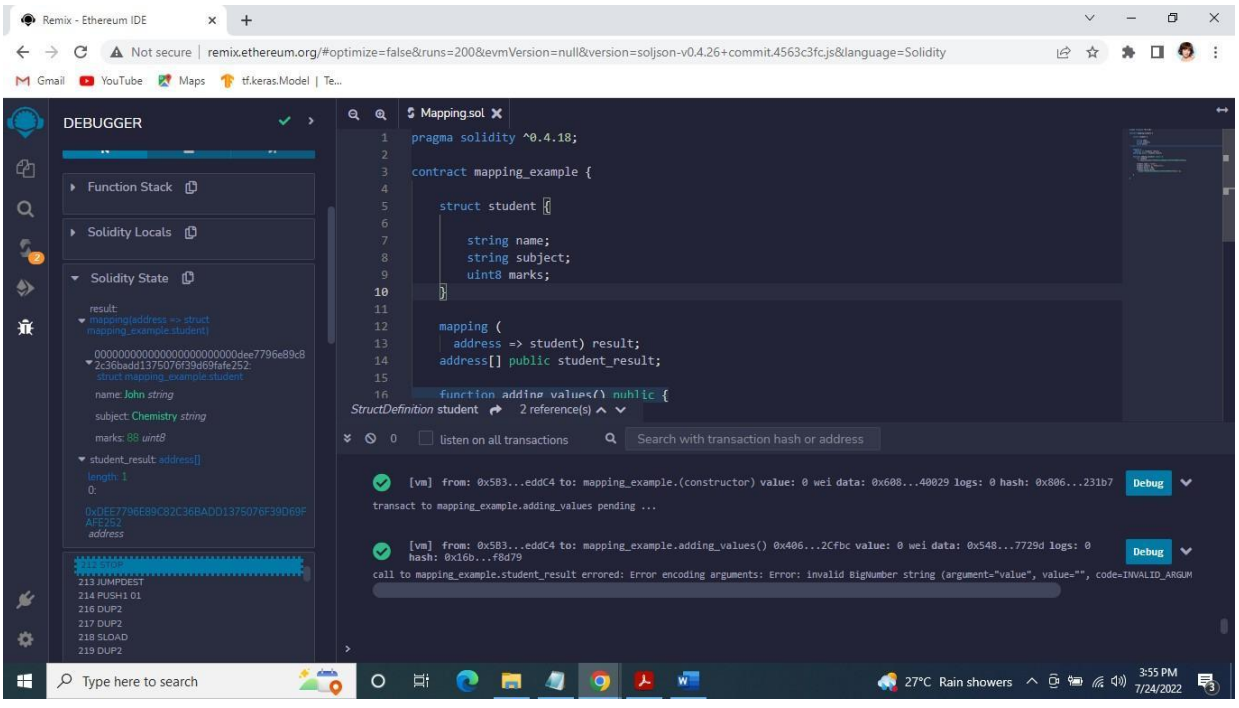

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
Enum.sol
Struct.sol
scripts
tests
README.txt
Enum.sol
Struct.sol
pragma solidity ^0.5.0;
contract Struct {
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;
}
}
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:47 PM
7/24/2022

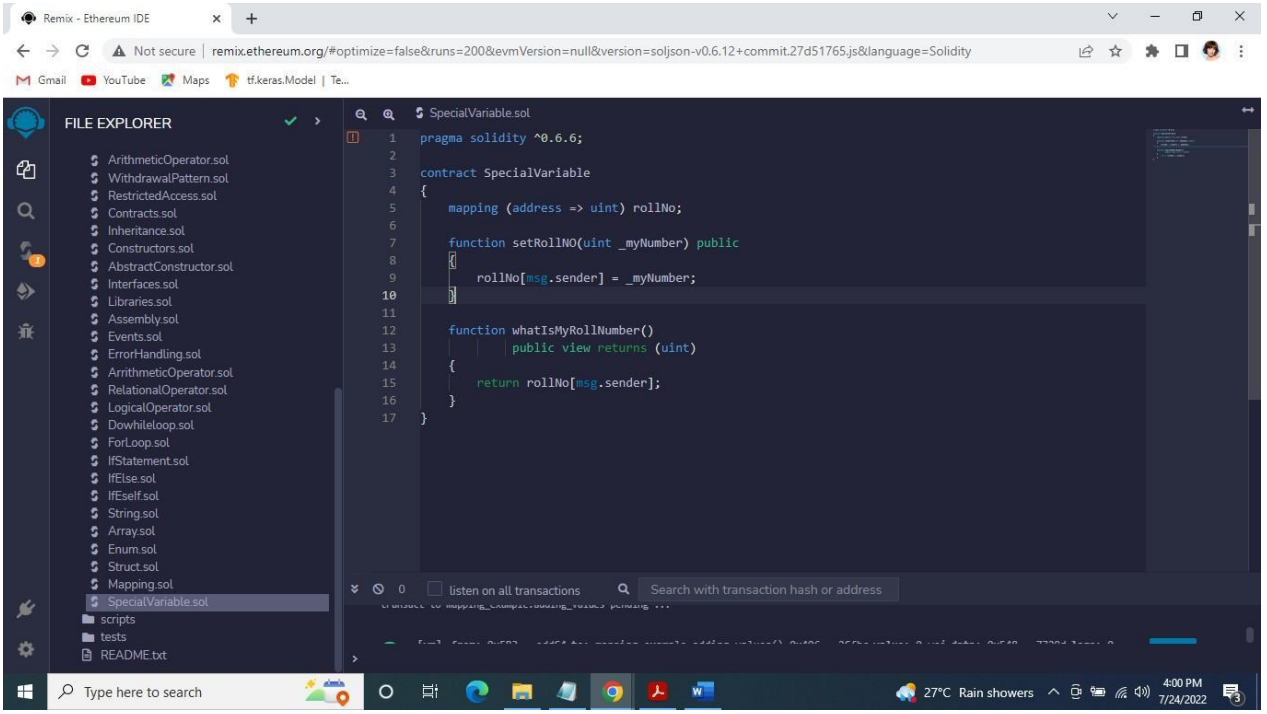
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 24
Deployed Contracts
STRUCT AT 0X93F...C96CC (MEMO
setBook
getBookId
0: uint256: 4
Low level interactions
CALLDATA
Transact
Enum.sol
Struct.sol
pragma solidity ^0.5.0;
contract Struct {
struct Book {
string title;
string author;
uint book_id;
}
Book book;
function setBook() public {
book = Book('Learn Java', 'TP', 4);
}
function getBookId() public view returns (uint) {
return book.book_id;
}
}
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:48 PM
7/24/2022

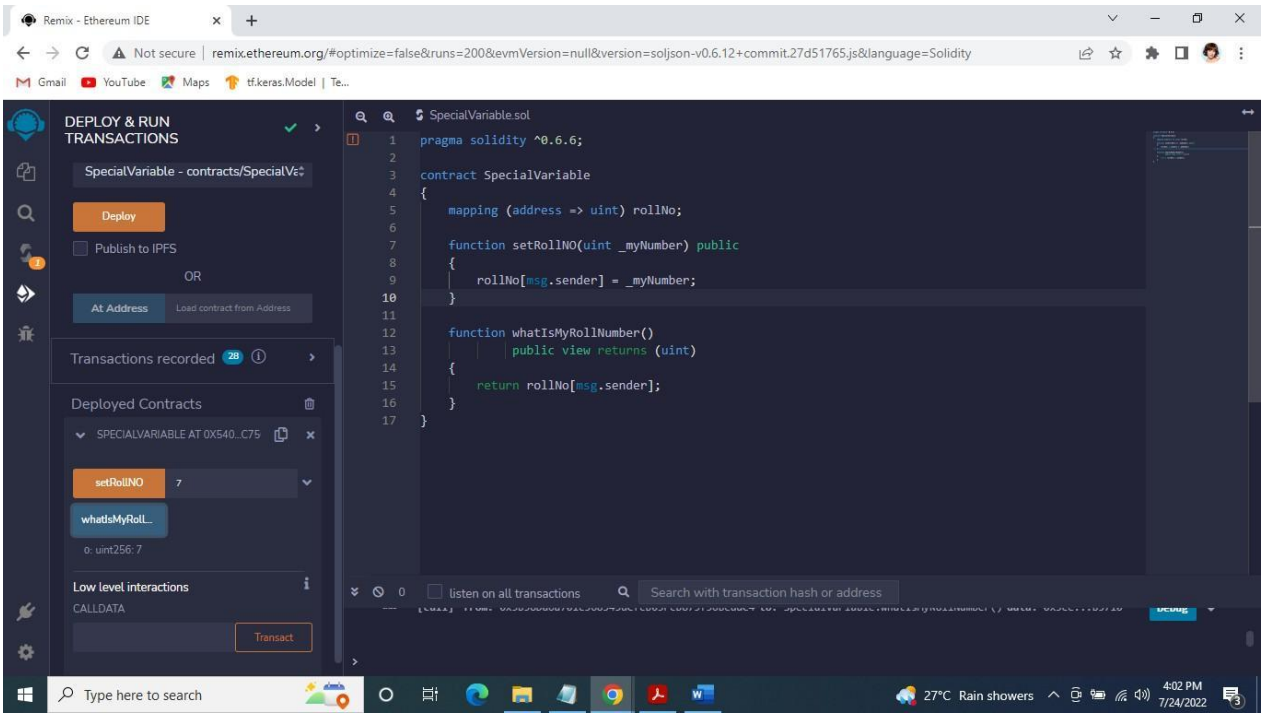

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
Enum.sol
Struct.sol
Mapping.sol
scripts
tests
README.txt
Mapping.sol
pragma solidity ^0.4.18;
contract mapping_example {
struct student {
string name;
string subject;
uint8 marks;
}
mapping (
address => student) result;
address[] public student_result;
function adding_values() public {
var student
= result[0xDEE7796E89C82C36BAdd1375076f39D69FafE252];
student.name = "John";
student.subject = "Chemistry";
student.marks = 88;
student_result.push(
0xDEE7796E89C82C36BAdd1375076f39D69FafE252) -1;
}
ContractDefinition mapping_example
1 reference(s)
listen on all transactions
Search with transaction hash or address
Type here to search
27°C Rain showers
3:53 PM
7/24/2022

Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEBUGGER
Function Stack
Solidity Locals
Solidity State
result:
mapping(address => struct mapping_example.student)
0000000000000000000000dee7796e89c82c36badd1375076f39d69fafe252:
struct mapping_example.student
name: John string
subject: Chemistry string
marks: 88 uint8
student_result: address[]
length: 1
0:
0xDEE7796E89C82C36BADD1375076F39D69FAFE252
address
212 STOP
213 JUMPDEST
214 PUSH1 01
216 DUP2
217 DUP2
218 SLOAD
219 DUP2
Mapping.sol
pragma solidity ^0.4.18;
contract mapping_example {
struct student {
string name;
string subject;
uint8 marks;
}
mapping (
address => student) result;
address[] public student_result;
function adding_values() public {
StructDefinition student
2 reference(s)
listen on all transactions
Search with transaction hash or address
[vm] from: 0x583...eddC4 to: mapping_example.(constructor) value: 0 wei data: 0x608...40029 logs: 0 hash: 0x806...231b7
Debug
transact to mapping_example.adding_values pending ...
[vm] from: 0x583...eddC4 to: mapping_example.adding_values() 0x406...2Cfbc value: 0 wei data: 0x548...7729d logs: 0 hash: 0x16b...f8d79
Debug
call to mapping_example.student_result errored: Error encoding arguments: Error: invalid BigNumber string (argument="value", value="", code=INVALID_ARGUM
Type here to search
27°C Rain showers
3:55 PM
7/24/2022



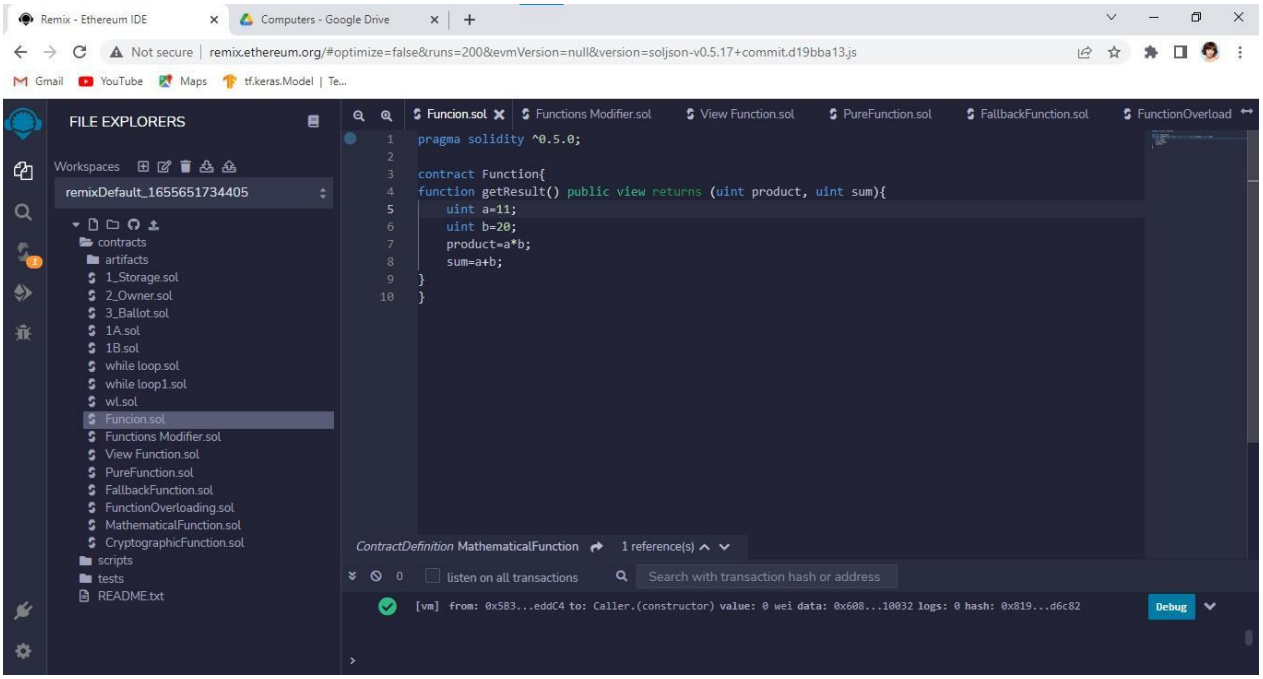
Output:



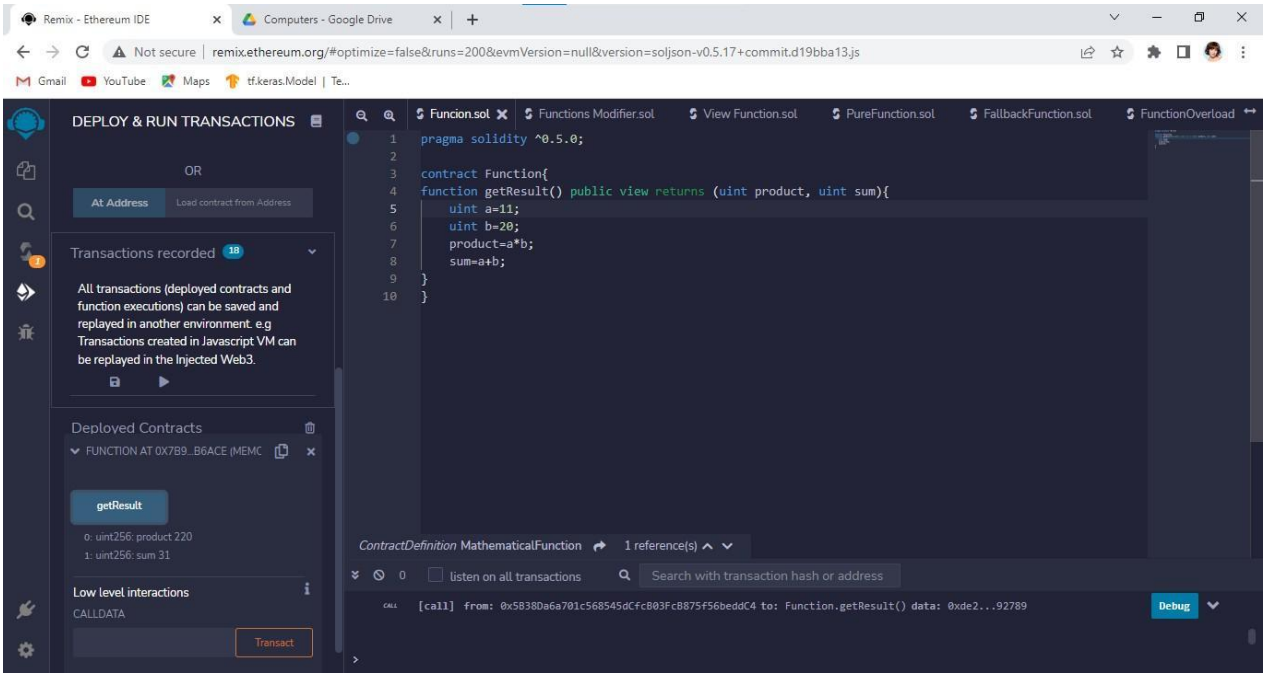
2b) Functions, Function Modifiers, View functions, Pure Functions, Fallback Function, Function Overloading, Mathematical functions, Cryptographic functions.

<u>Function:</u>

Code:

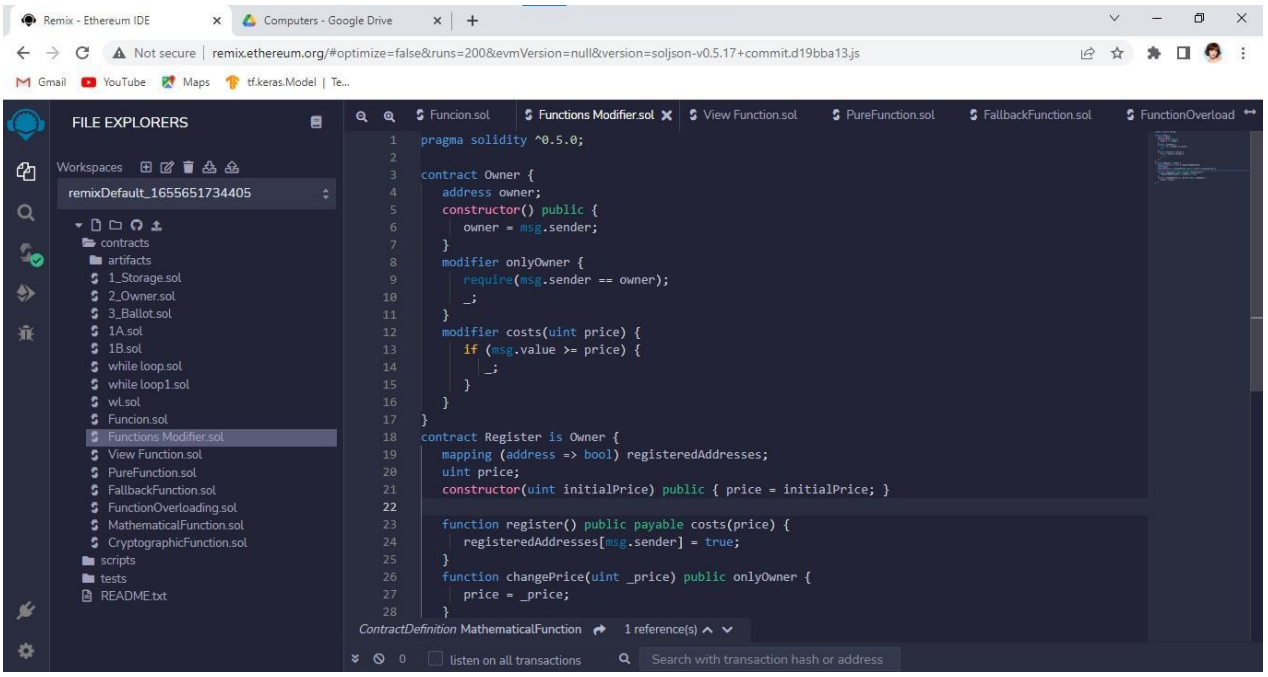


Output:

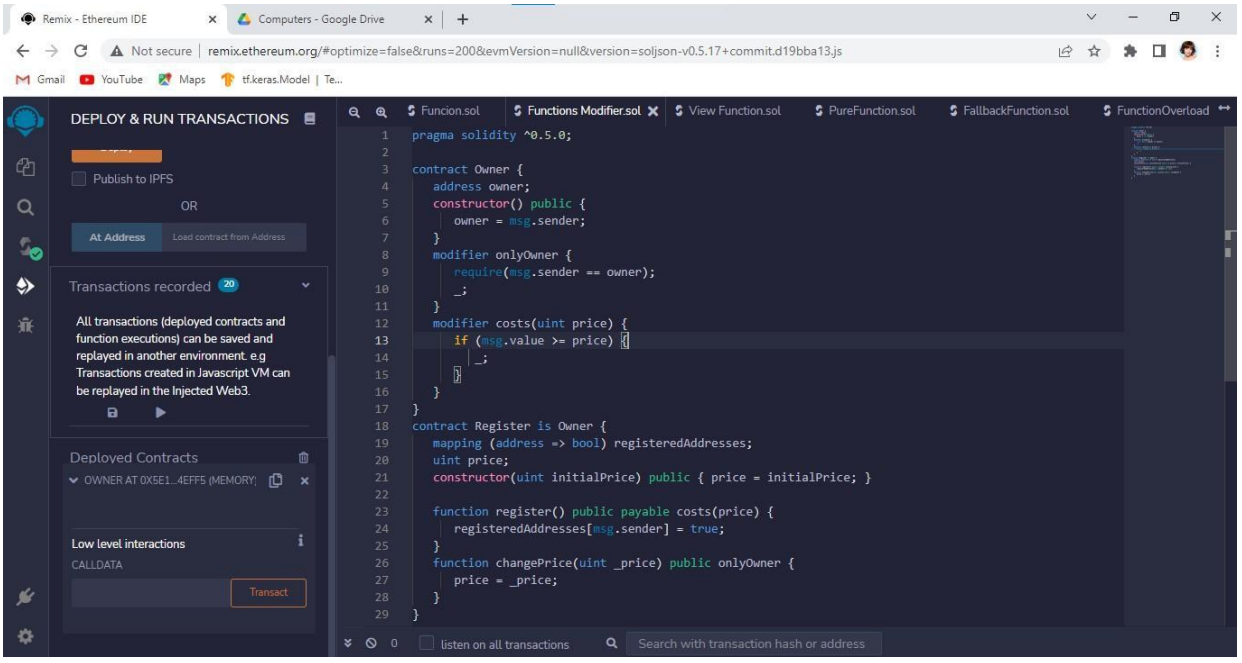


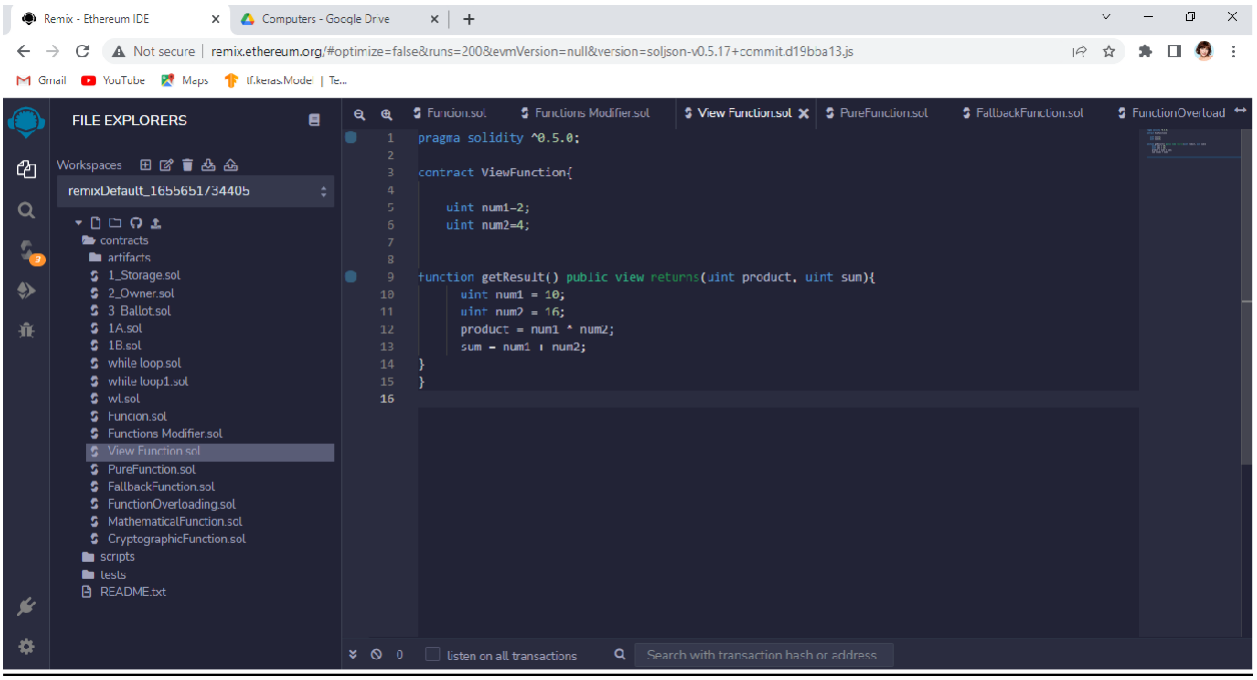
Functions Modifiers

Code:

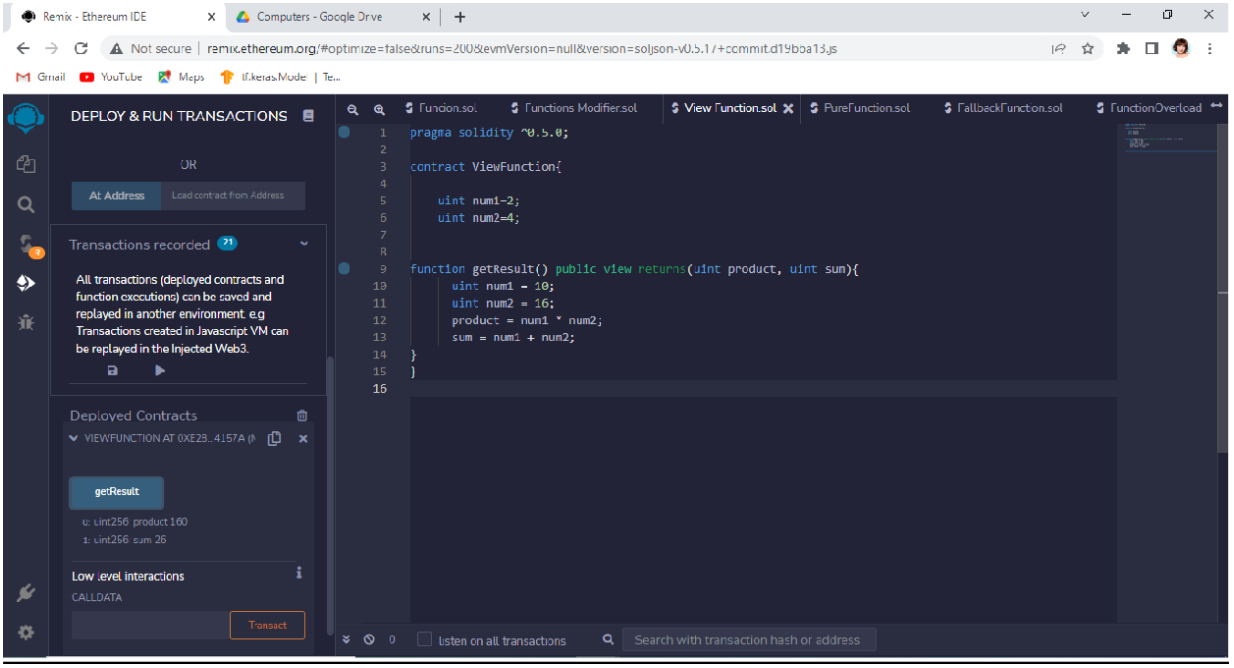

Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORERS
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
1A.sol
1B.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
scripts
tests
README.txt
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverload
pragma solidity ^0.5.0;
contract Owner {
address owner;
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;
}
ContractDefinition MathematicalFunction
1 reference(s)
listen on all transactions
Search with transaction hash or address

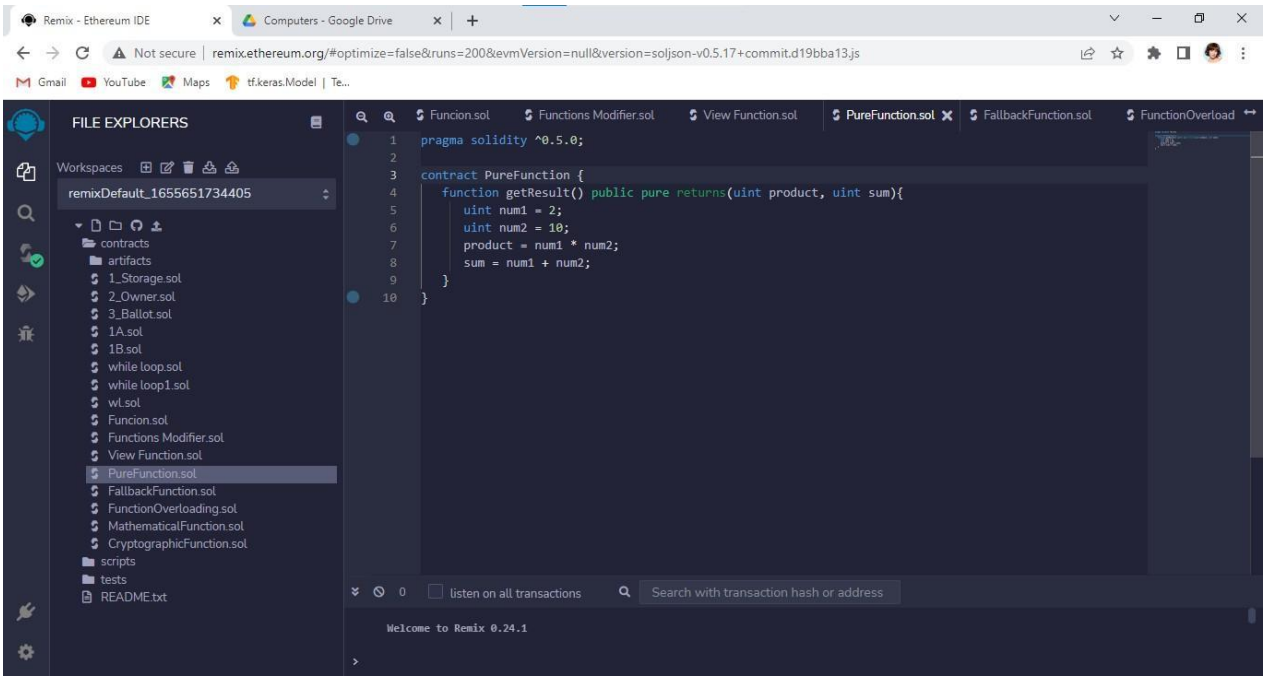
Output:


Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded
20
All transactions (deployed contracts and function executions) can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.
Deployed Contracts
OWNER AT 0X5E1...4EFF5 (MEMORY)
Low level interactions
CALLDATA
Transact
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverload
pragma solidity ^0.5.0;
contract Owner {
address owner;
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;
}
}
listen on all transactions
Search with transaction hash or address

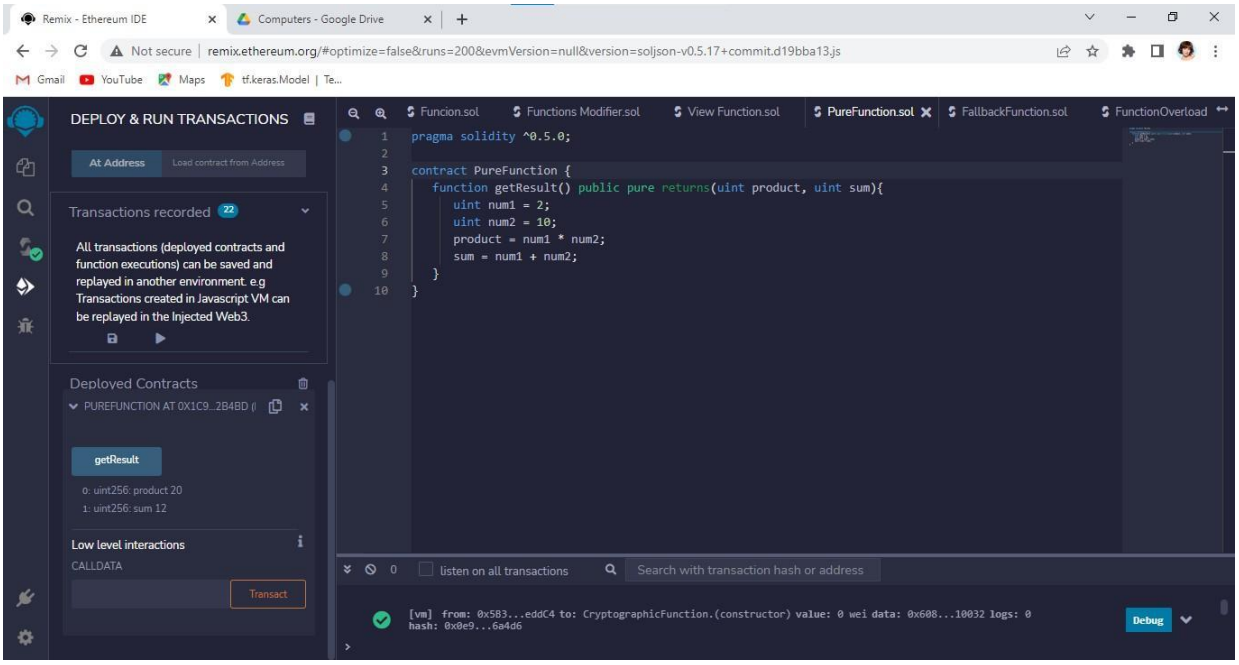

Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
FILE EXPLORERS
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
1A.sol
1B.sol
while loop.sol
while loop1.sol
wl.sol
Function.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
scripts
tests
README.txt
pragma solidity ^0.5.0;
contract ViewFunction{
uint num1=2;
uint num2=4;
function getResult() public view returns(uint product, uint sum){
uint num1 = 10;
uint num2 = 16;
product = num1 * num2;
sum = num1 + num2;
}
}
listen on all transactions
Search with transaction hash or address

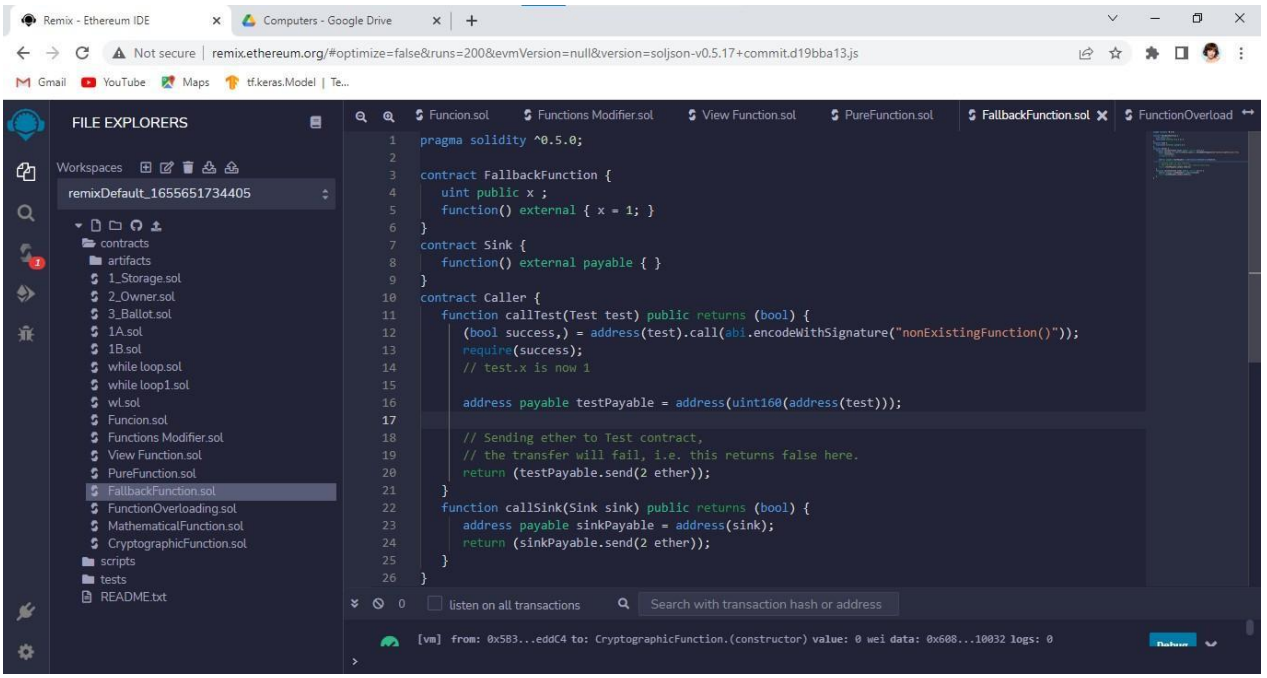
Output:


Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
DEPLOY & RUN TRANSACTIONS
OR
At Address
Load contract from Address
Transactions recorded
All transactions (deployed contracts and function executions) can be saved and replayed in another environment e.g Transactions created in Javascript VM can be replayed in the Injected Web3.
Deployed Contracts
VIEWFUNCTION AT 0XE28...4157A
getResult
0: uint256: product 160
1: uint256: sum 26
Low level interactions
CALLDATA
Transact
pragma solidity ^0.5.0;
contract ViewFunction{
uint num1=2;
uint num2=4;
function getResult() public view returns(uint product, uint sum){
uint num1 = 10;
uint num2 = 16;
product = num1 * num2;
sum = num1 + num2;
}
}
listen on all transactions
Search with transaction hash or address

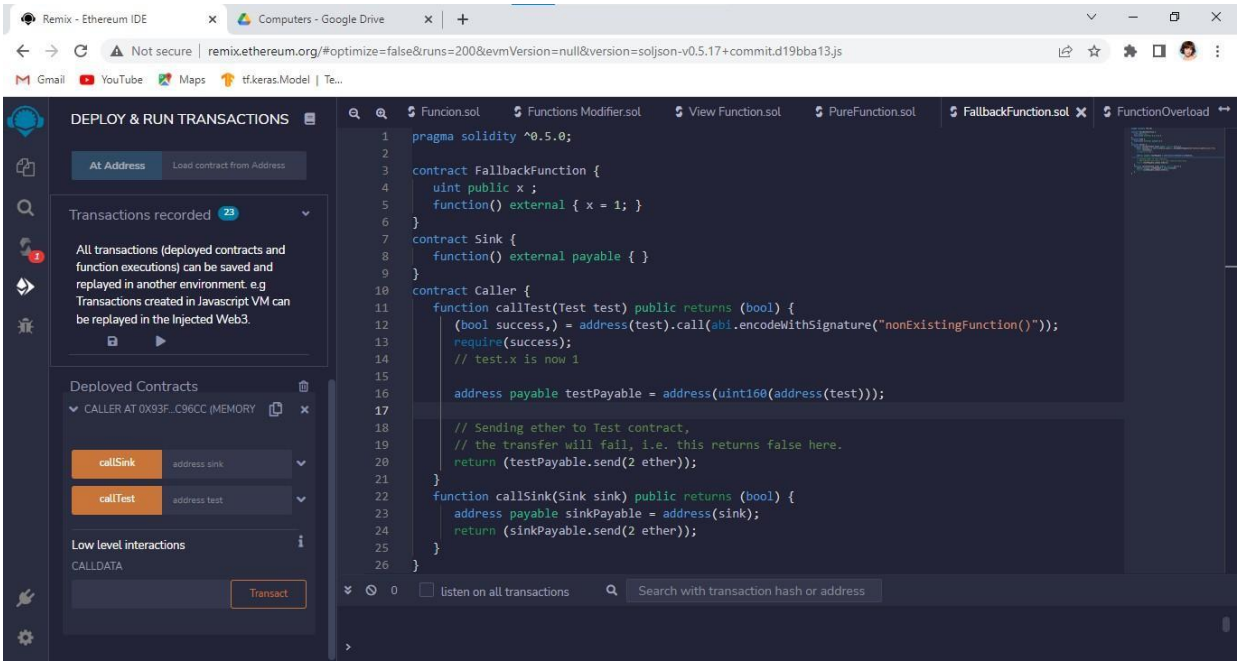

Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORERS
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
1A.sol
1B.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
scripts
tests
README.txt
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverload
pragma solidity ^0.5.0;
contract PureFunction {
function getResult() public pure returns(uint product, uint sum){
uint num1 = 2;
uint num2 = 10;
product = num1 * num2;
sum = num1 + num2;
}
}
listen on all transactions
Search with transaction hash or address
Welcome to Remix 0.24.1

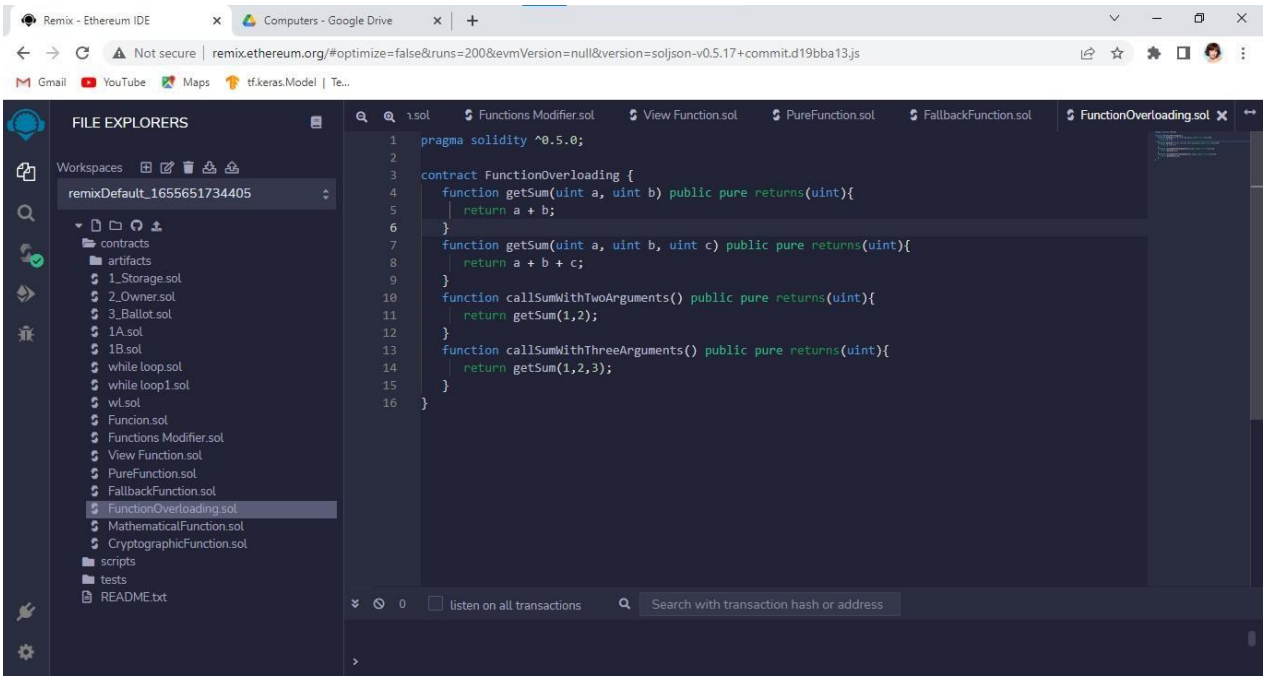
Output:


Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
At Address
Load contract from Address
Transactions recorded 22
All transactions (deployed contracts and function executions) can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.
Deployed Contracts
PUREFUNCTION AT 0X1C9...2B4BD
getResult
0: uint256: product 20
1: uint256: sum 12
Low level interactions
CALLDATA
Transact
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverload
pragma solidity ^0.5.0;
contract PureFunction {
function getResult() public pure returns(uint product, uint sum){
uint num1 = 2;
uint num2 = 10;
product = num1 * num2;
sum = num1 + num2;
}
}
listen on all transactions
Search with transaction hash or address
[vm] from: 0x583...eddC4 to: CryptographicFunction.(constructor) value: 0 wei data: 0x608...10032 logs: 0
hash: 0x0e9...6a4d6
Debug

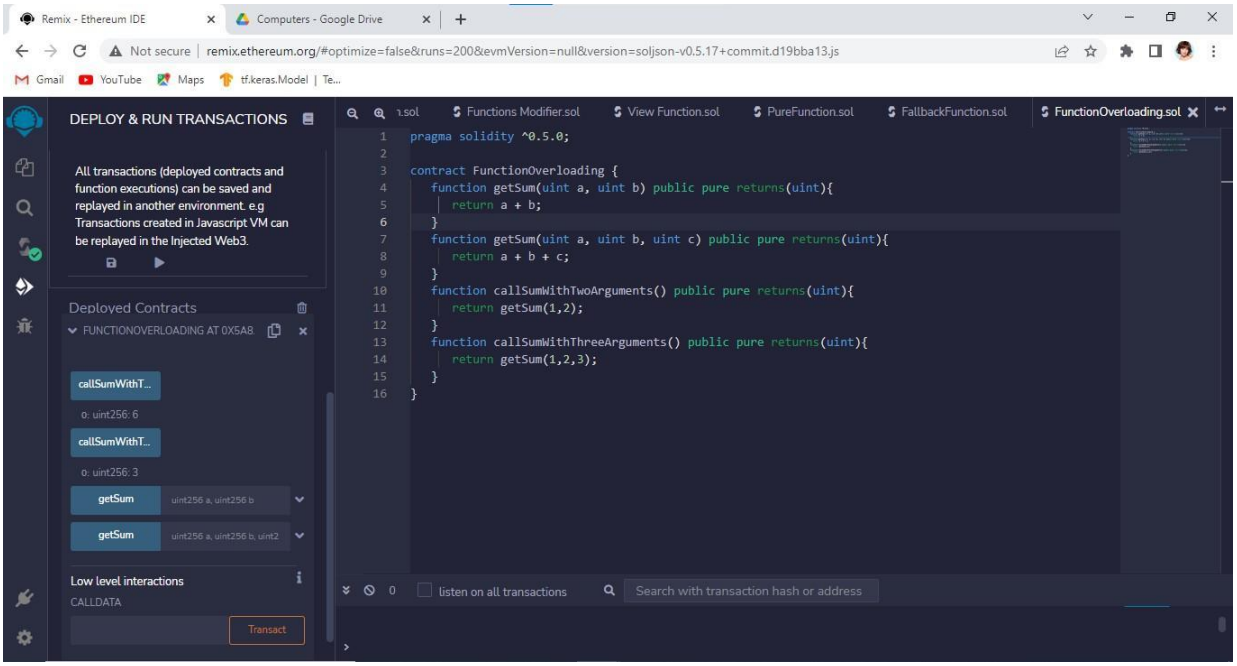

Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORERS
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
1A.sol
1B.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
scripts
tests
README.txt
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverload
pragma solidity ^0.5.0;
contract FallbackFunction {
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));
}
}
listen on all transactions
Search with transaction hash or address
[vm] from: 0x583...eddC4 to: CryptographicFunction.(constructor) value: 0 wei data: 0x608...10032 logs: 0

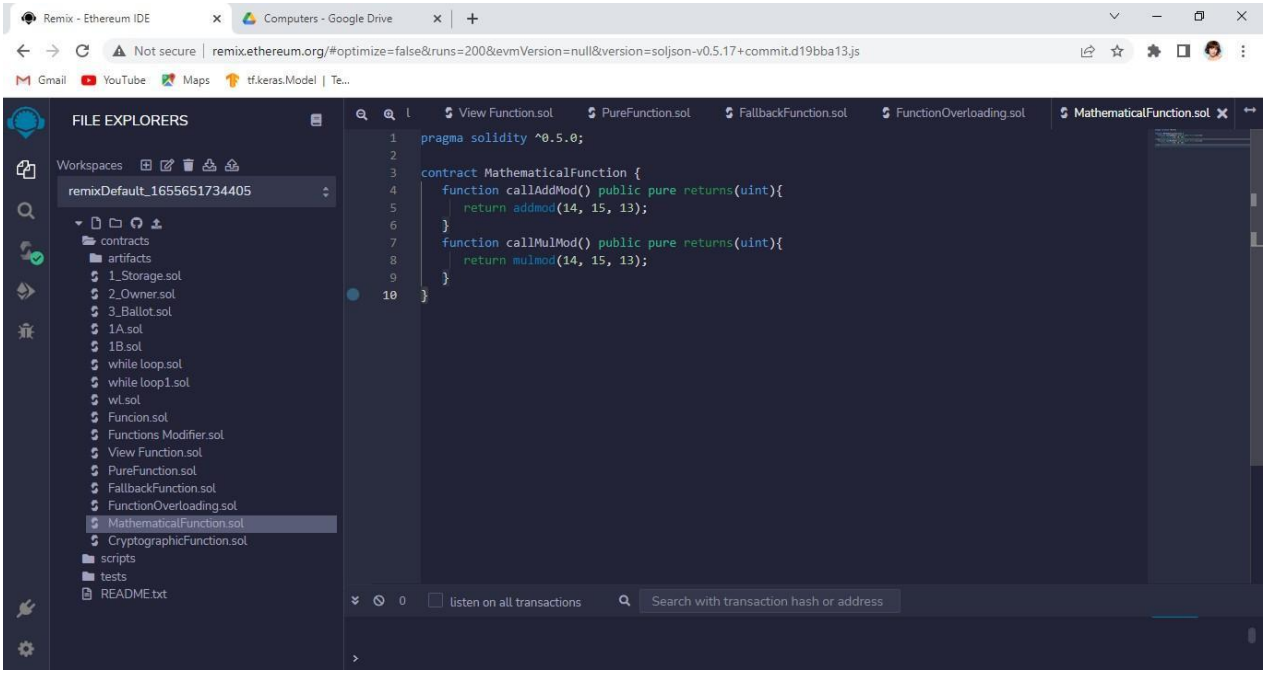
Output:


Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
At Address
Load contract from Address
Transactions recorded 23
All transactions (deployed contracts and function executions) can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.
Deployed Contracts
CALLER AT 0X93F...C96CC (MEMORY
callSink
address sink
callTest
address test
Low level interactions
CALLDATA
Transact
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverload
pragma solidity ^0.5.0;
contract FallbackFunction {
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));
}
}
listen on all transactions
Search with transaction hash or address

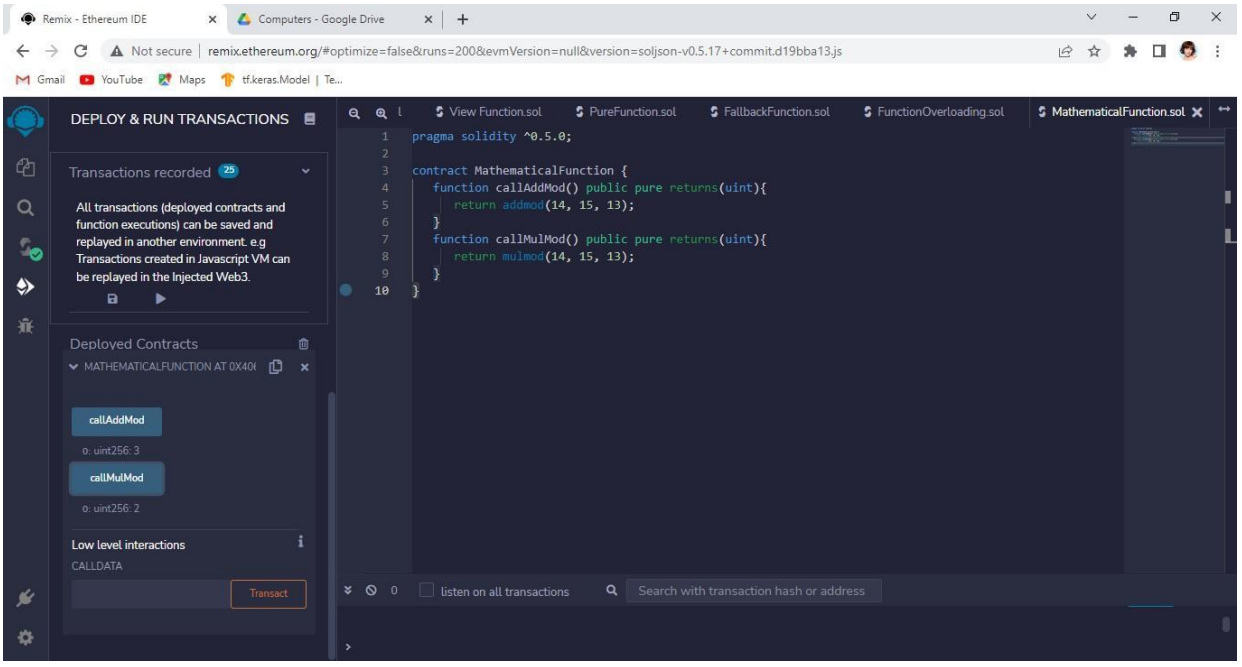

Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORERS
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
1A.sol
1B.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
scripts
tests
README.txt
1.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
pragma solidity ^0.5.0;
contract FunctionOverloading {
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);
}
}
0
listen on all transactions
Search with transaction hash or address

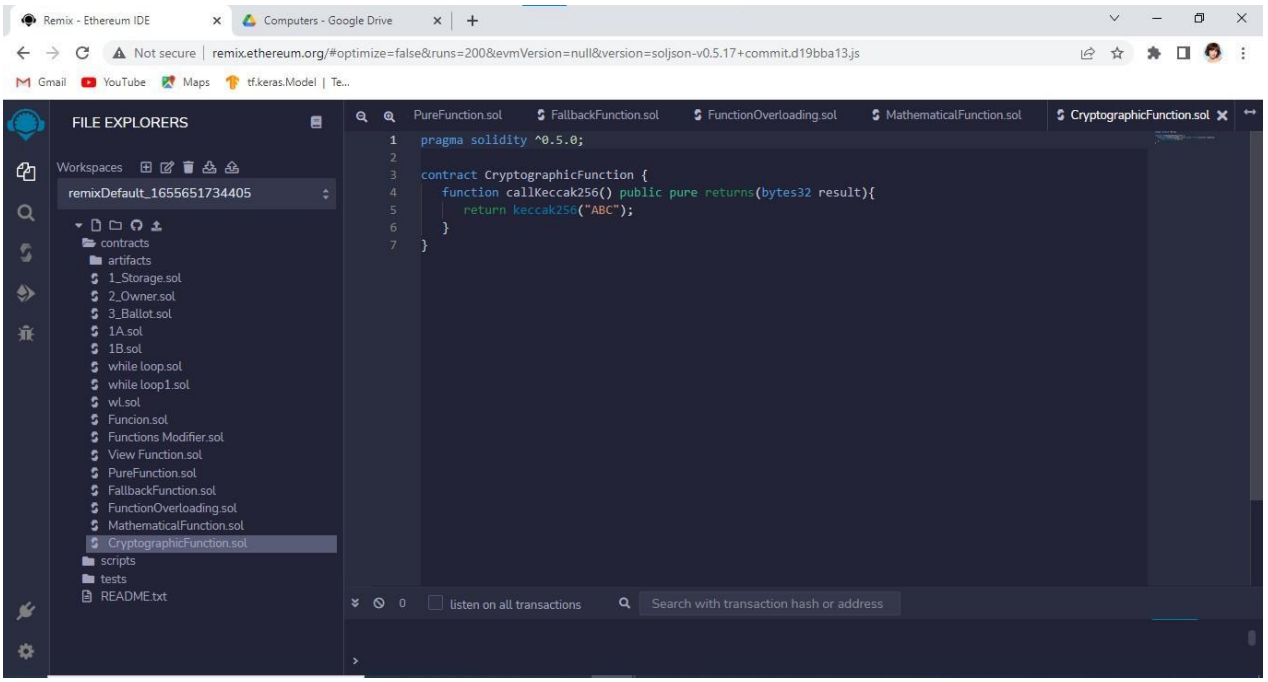
Output:


Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
All transactions (deployed contracts and function executions) can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.
Deployed Contracts
FUNCTIONOVERLOADING AT 0X5A8...
callSumWithT...
0: uint256: 6
callSumWithT...
0: uint256: 3
getSum
uint256 a, uint256 b
getSum
uint256 a, uint256 b, uint2
Low level interactions
CALLDATA
Transact
1.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
pragma solidity ^0.5.0;
contract FunctionOverloading {
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);
}
}
0
listen on all transactions
Search with transaction hash or address

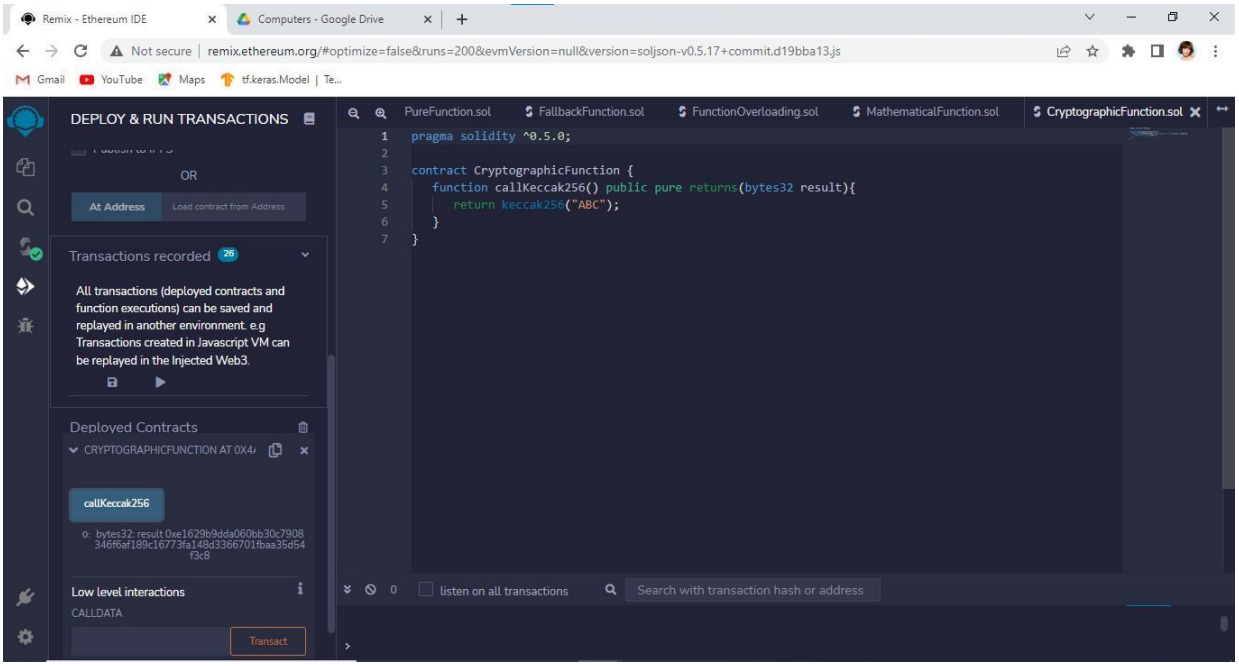

Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORERS
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
1A.sol
1B.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
scripts
tests
README.txt
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
pragma solidity ^0.5.0;
contract MathematicalFunction {
function callAddMod() public pure returns(uint){
return addmod(14, 15, 13);
}
function callMulMod() public pure returns(uint){
return mulmod(14, 15, 13);
}
}
listen on all transactions
Search with transaction hash or address

Output:


Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Transactions recorded 25
All transactions (deployed contracts and function executions) can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.
Deployed Contracts
MATHEMATICALFUNCTION AT 0X40(
callAddMod
0: uint256: 3
callMulMod
0: uint256: 2
Low level interactions
CALLDATA
Transact
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
pragma solidity ^0.5.0;
contract MathematicalFunction {
function callAddMod() public pure returns(uint){
return addmod(14, 15, 13);
}
function callMulMod() public pure returns(uint){
return mulmod(14, 15, 13);
}
}
listen on all transactions
Search with transaction hash or address


Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORERS
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
1A.sol
1B.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
scripts
tests
README.txt
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
pragma solidity ^0.5.0;
contract CryptographicFunction {
function callKeccak256() public pure returns(bytes32 result){
return keccak256("ABC");
}
}
listen on all transactions
Search with transaction hash or address

Output:


Remix - Ethereum IDE
Computers - Google Drive
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
OR
At Address
Load contract from Address
Transactions recorded 26
All transactions (deployed contracts and function executions) can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.
Deployed Contracts
CRYPTOGRAPHICFUNCTION AT 0X4
callKeccak256
0: bytes32: result 0xe1629b9dda060bb30c7908346f6af189c16773fa148d3366701fbaa35d54f3c8
Low level interactions
CALLDATA
Transact
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
pragma solidity ^0.5.0;
contract CryptographicFunction {
function callKeccak256() public pure returns(bytes32 result){
return keccak256("ABC");
}
}
listen on all transactions
Search with transaction hash or address

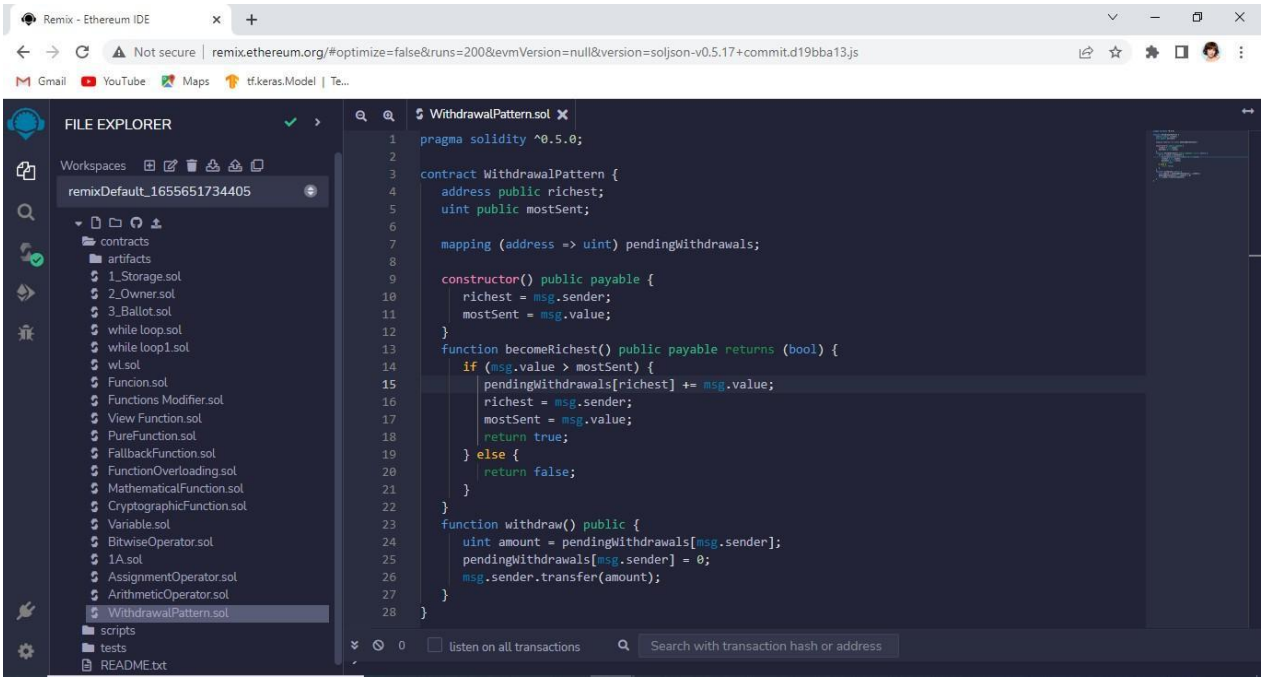
Practical 3

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

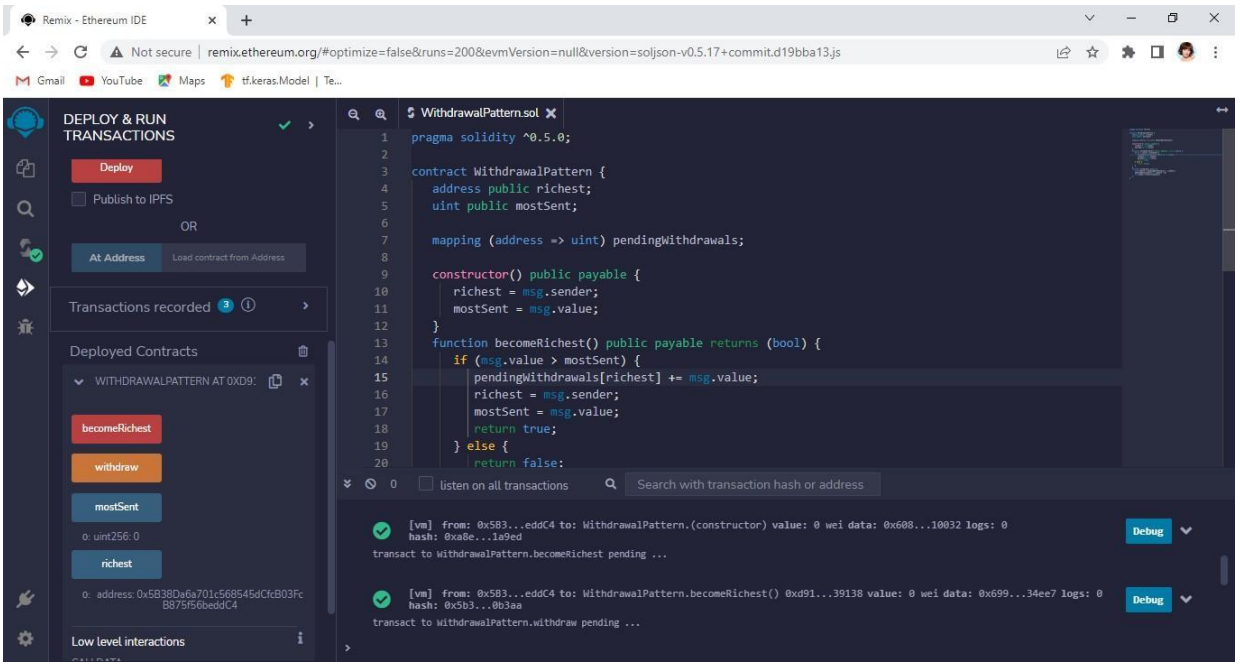
3a) Withdrawal Pattern, Restricted Access.

Withdrawal Pattern

Code:

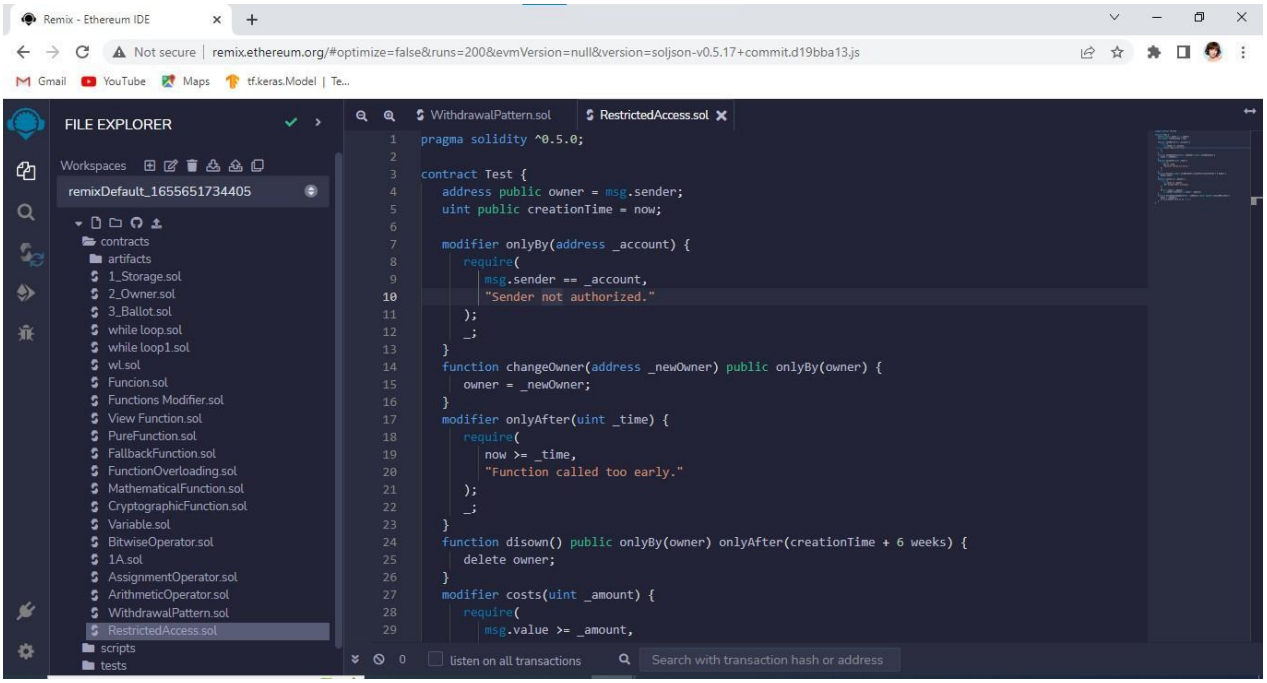


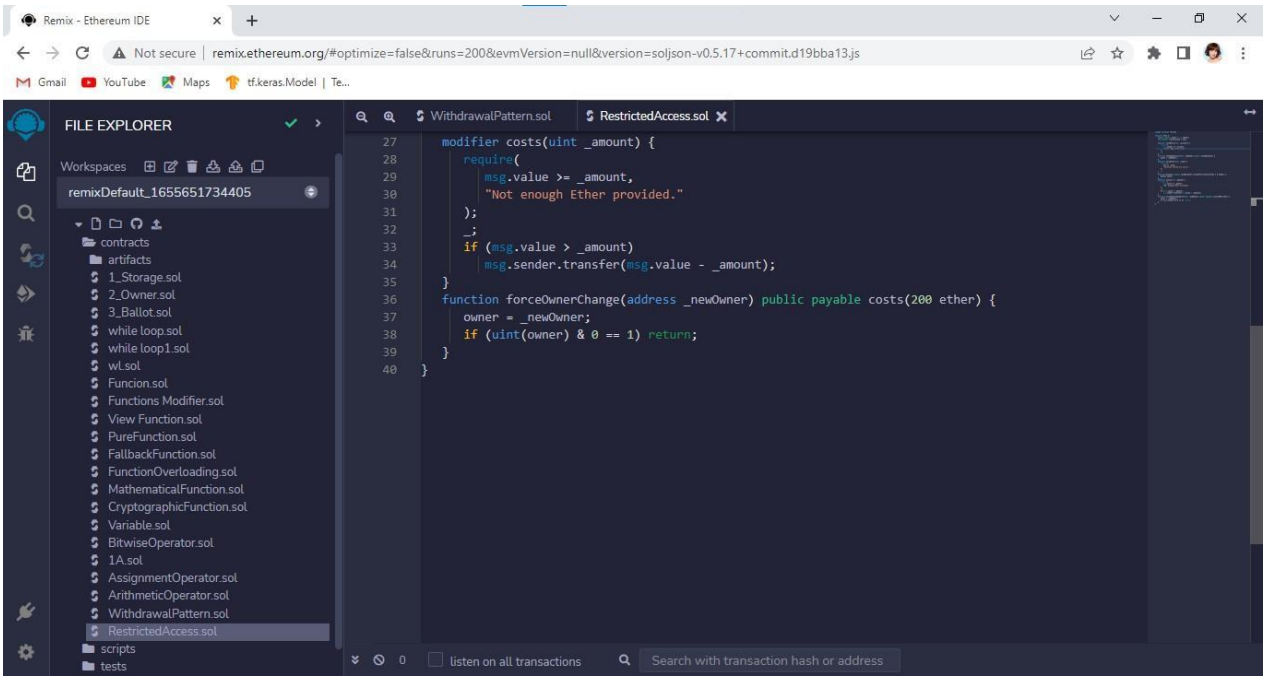
Output:



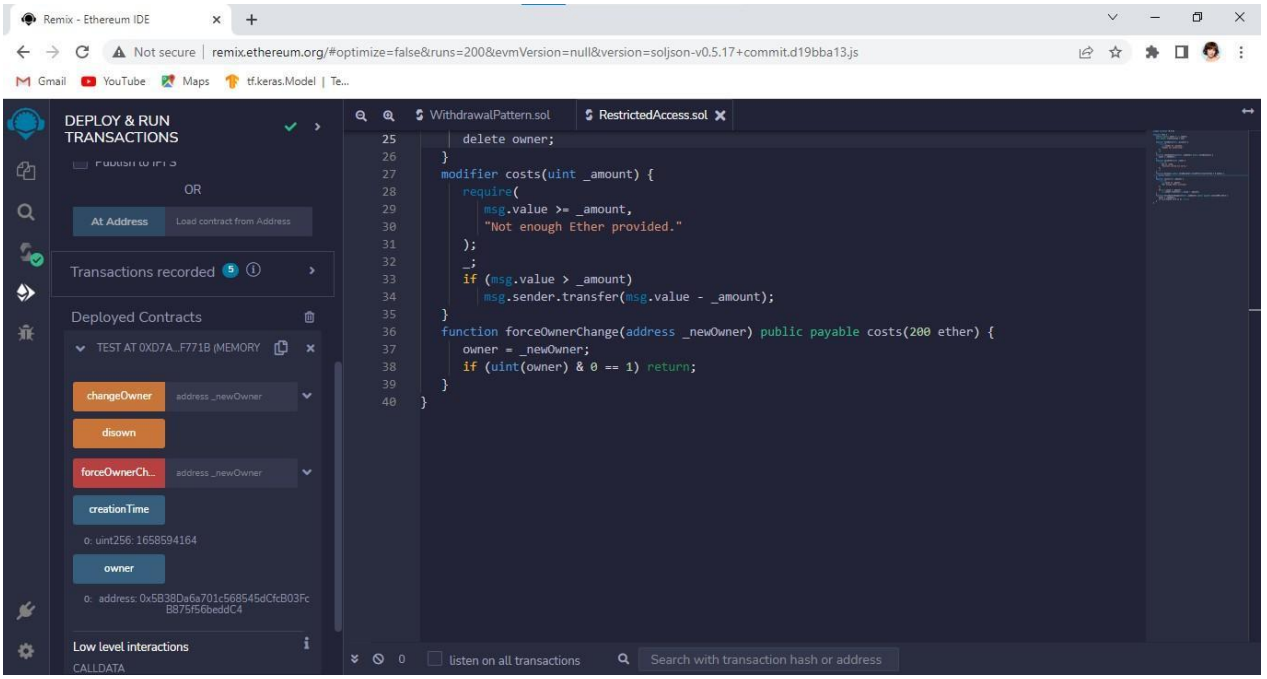
Restricted Access

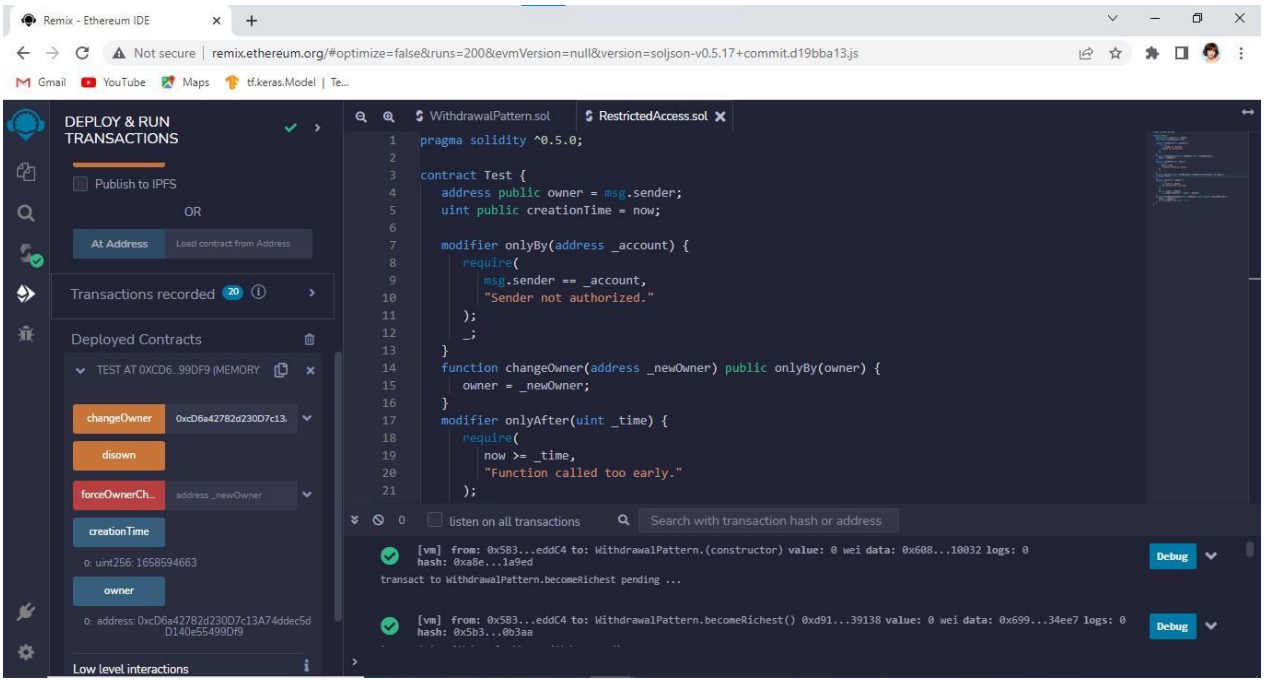
Code:





Output:

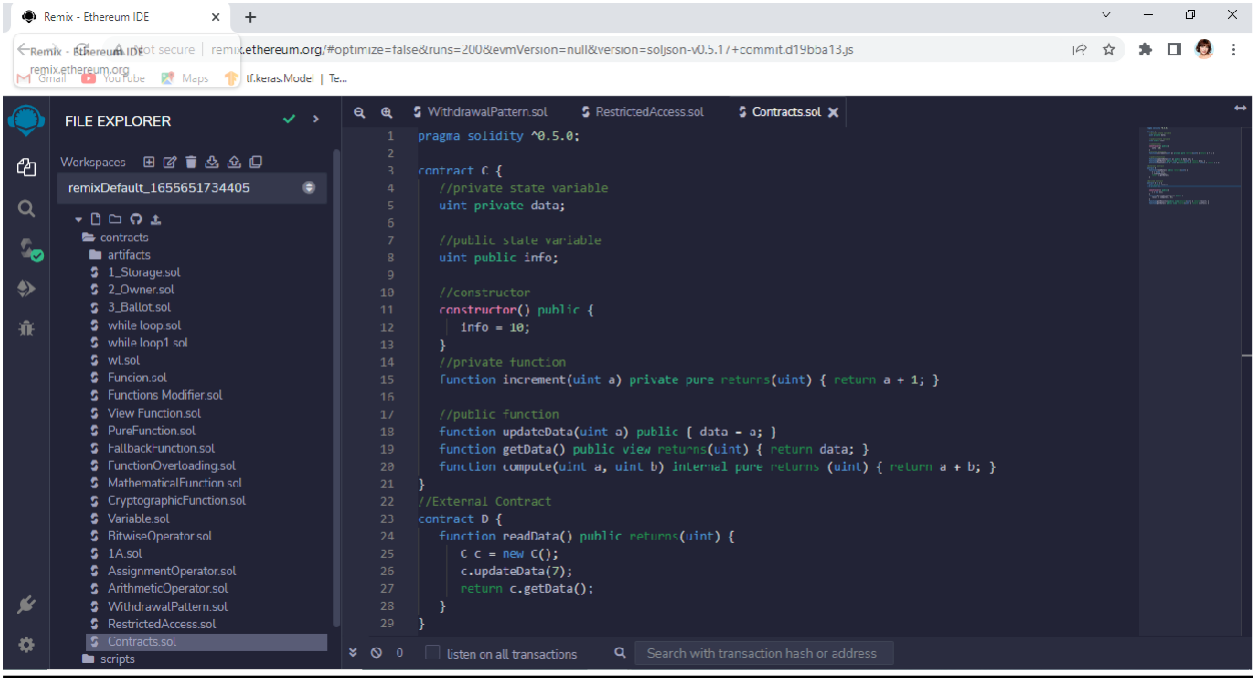


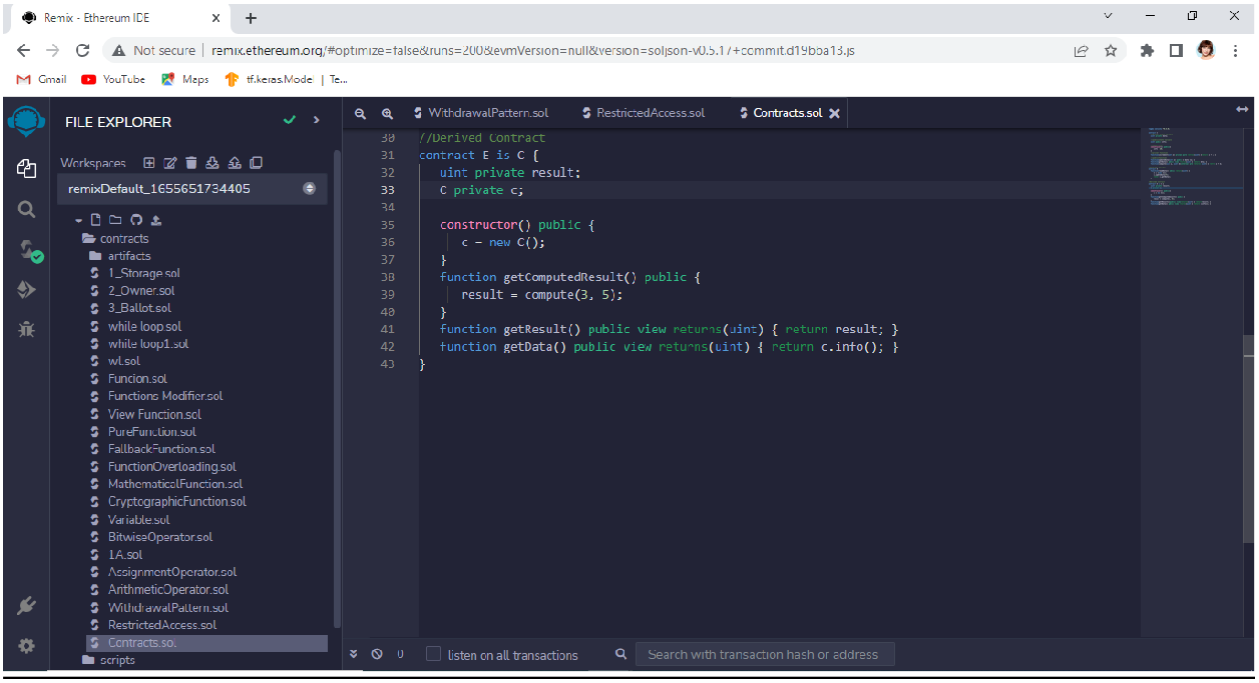


3b) Contracts, Inheritance, Constructors, Abstract Contracts, Interfaces.

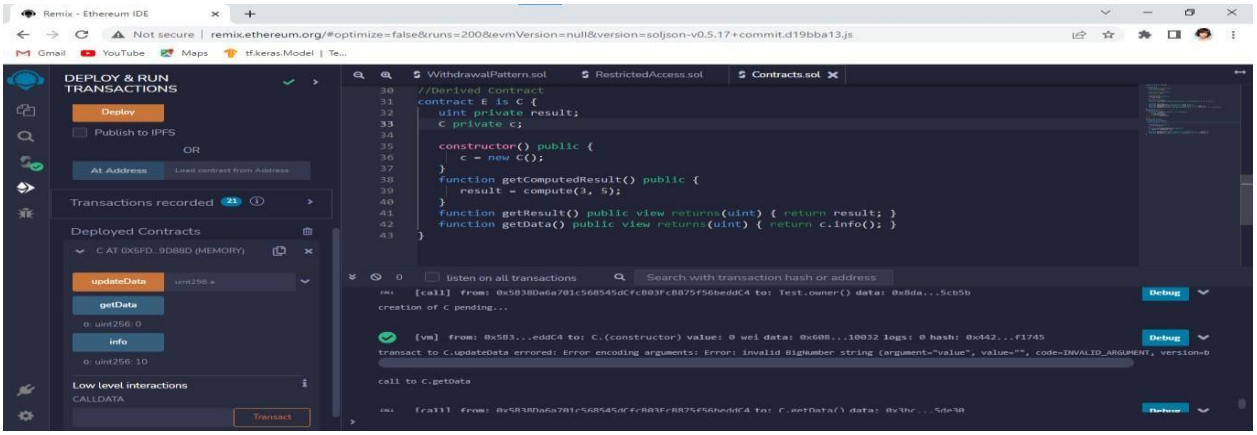
Contracts

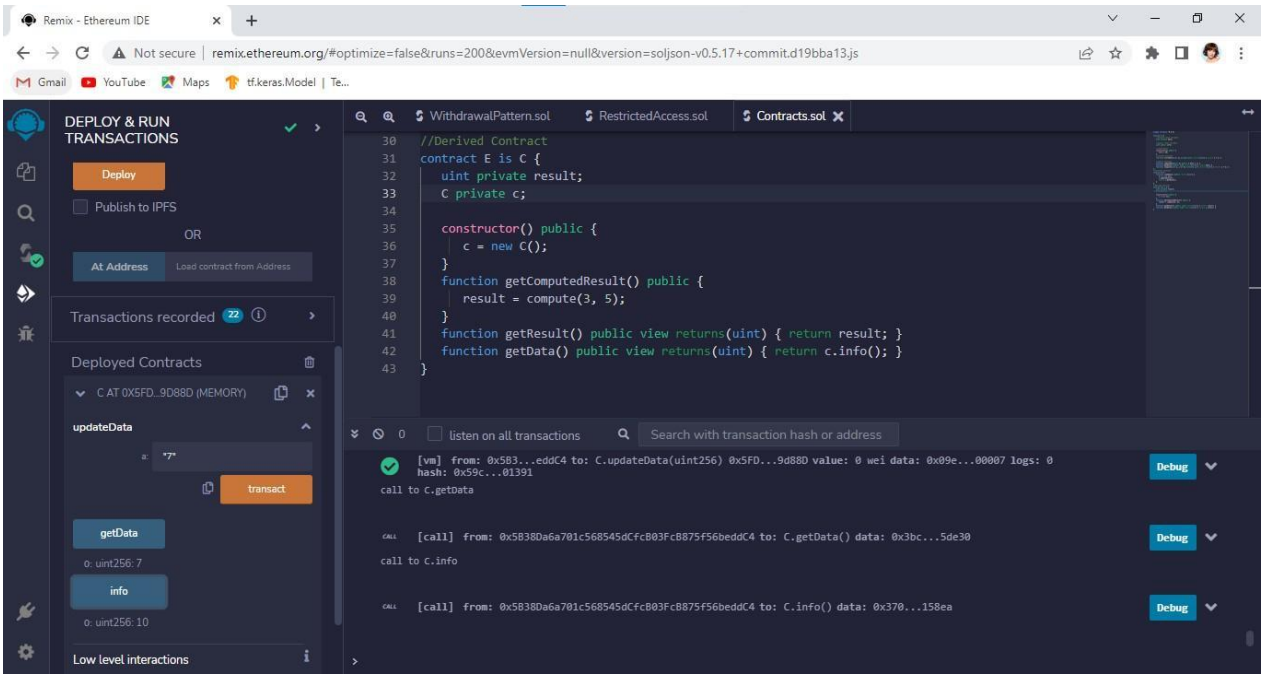
Code:


Remix - Ethereum IDE
FILE EXPLORER
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
Fallback-unction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.so.
scripts
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
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();
}
}
Listen on all transactions
Search with transaction hash or address


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.1+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Mode
FILE EXPLORER
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
Fallback-function.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.so.
scripts
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
//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(); }
}
Listen on all transactions
Search with transaction hash or address

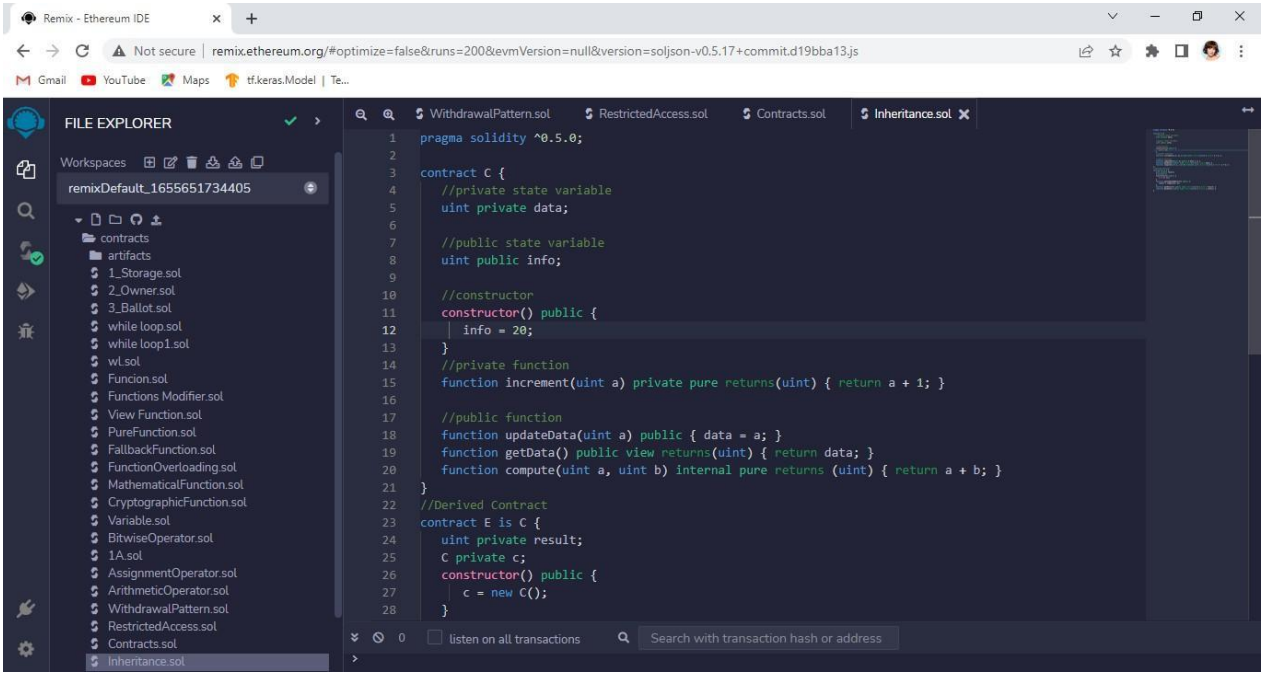
Output:

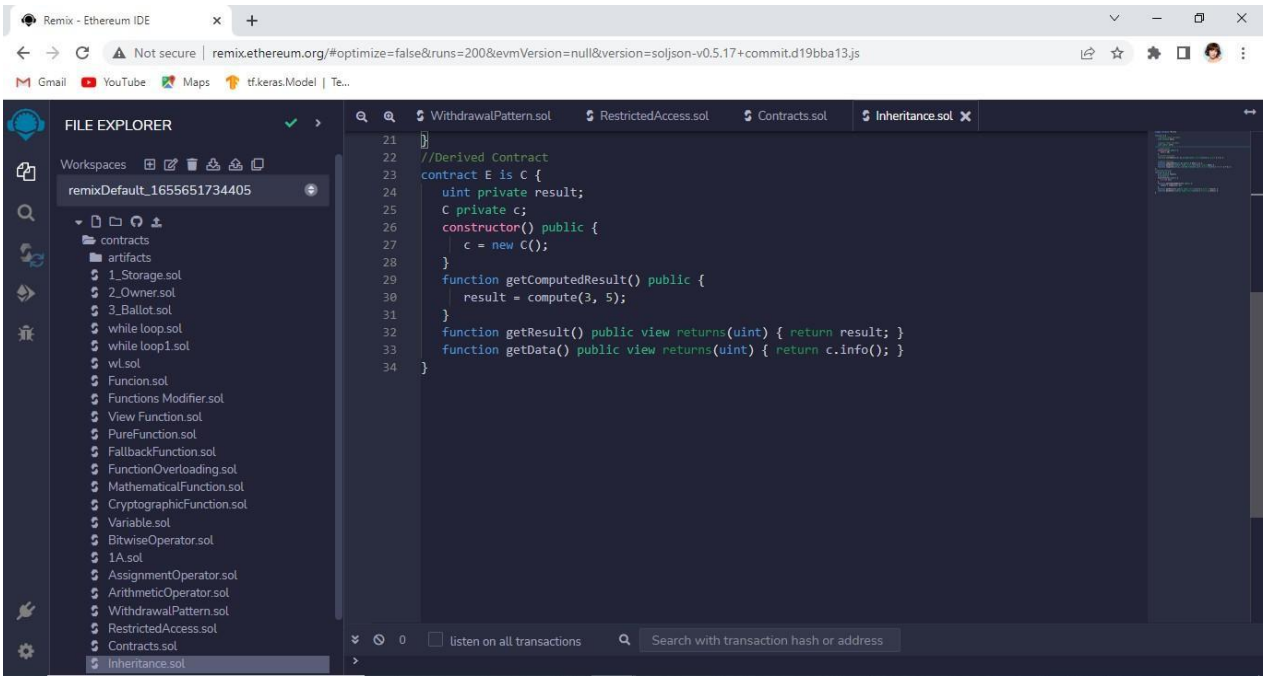




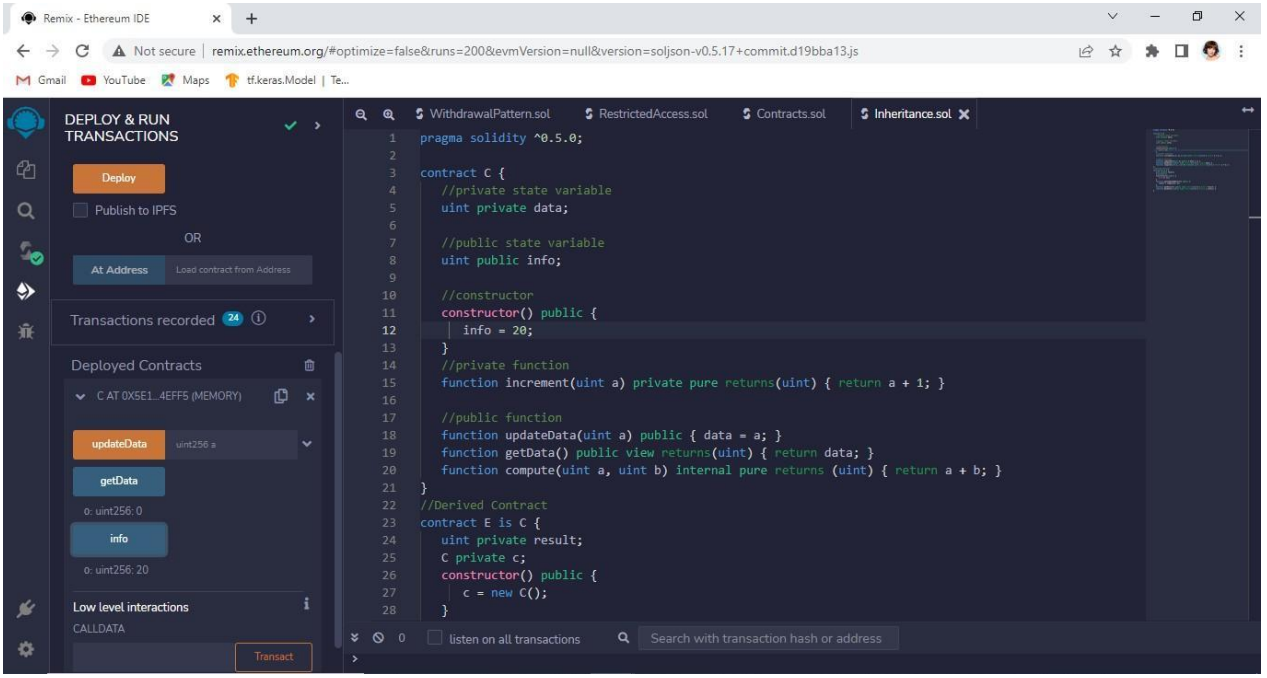
Inheritance

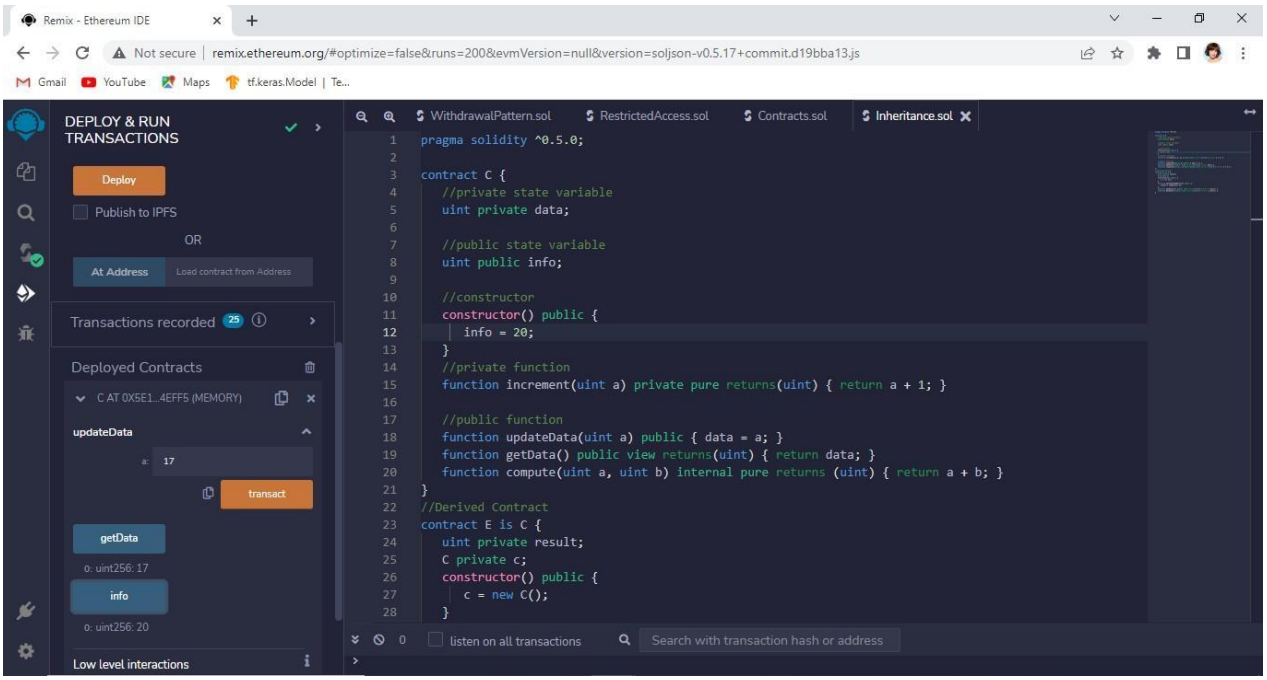
Code:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail YouTube Maps tf.keras.Model | Te...
FILE EXPLORER
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
WithdrawalPattern.sol RestrictedAccess.sol Contracts.sol Inheritance.sol
1 pragma solidity ^0.5.0;
2
3 contract C {
4 //private state variable
5 uint private data;
6
7 //public state variable
8 uint public info;
9
10 //constructor
11 constructor() public {
12 info = 20;
13 }
14 //private function
15 function increment(uint a) private pure returns(uint) { return a + 1; }
16
17 //public function
18 function updateData(uint a) public { data = a; }
19 function getData() public view returns(uint) { return data; }
20 function compute(uint a, uint b) internal pure returns (uint) { return a + b; }
21 }
22 //Derived Contract
23 contract E is C {
24 uint private result;
25 C private c;
26 constructor() public {
27 c = new C();
28 }
0
listen on all transactions
Search with transaction hash or address


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail YouTube Maps tf.keras.Model | Te...
FILE EXPLORER
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
WithdrawalPattern.sol RestrictedAccess.sol Contracts.sol Inheritance.sol
21 }
22 //Derived Contract
23 contract E is C {
24 uint private result;
25 C private c;
26 constructor() public {
27 c = new C();
28 }
29 function getComputedResult() public {
30 result = compute(3, 5);
31 }
32 function getResult() public view returns(uint) { return result; }
33 function getData() public view returns(uint) { return c.info(); }
34 }
0
listen on all transactions
Search with transaction hash or address

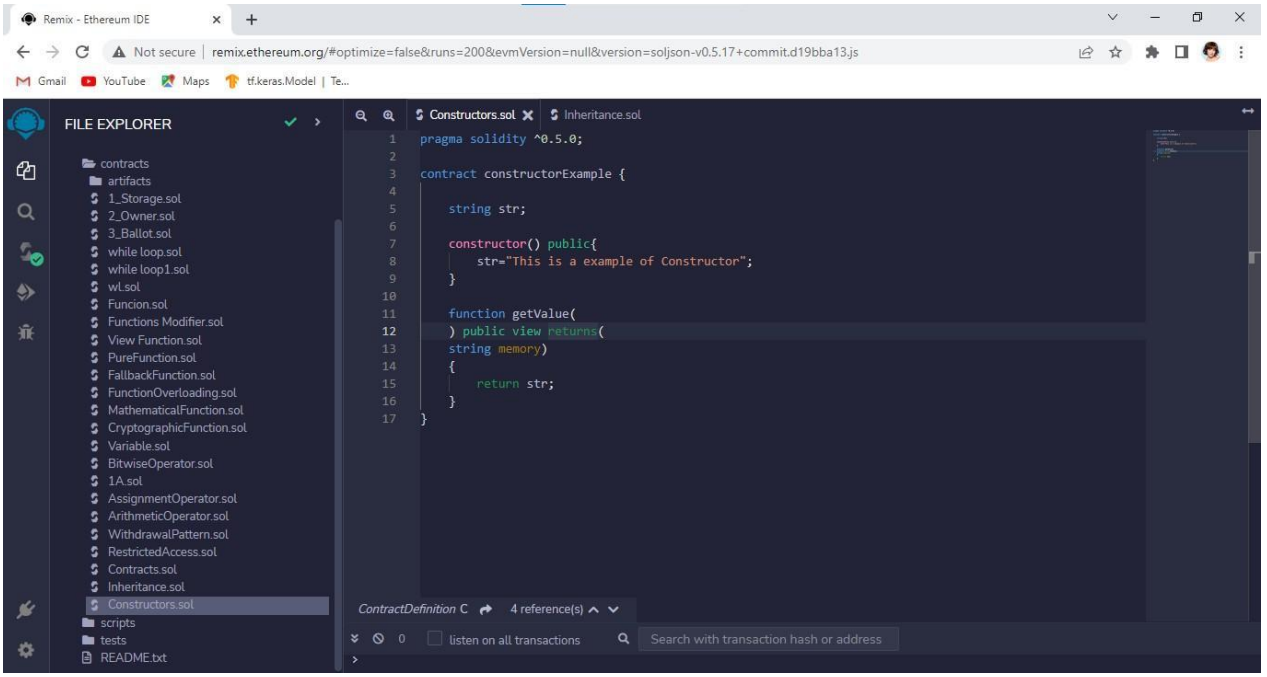
Output:



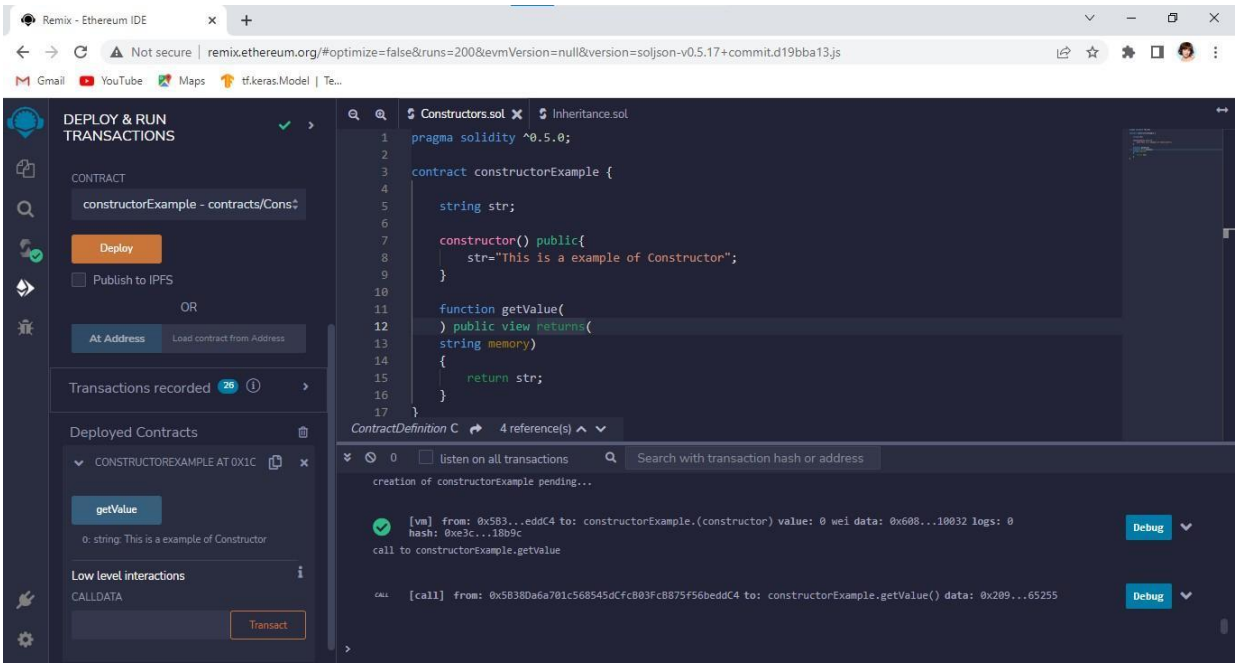


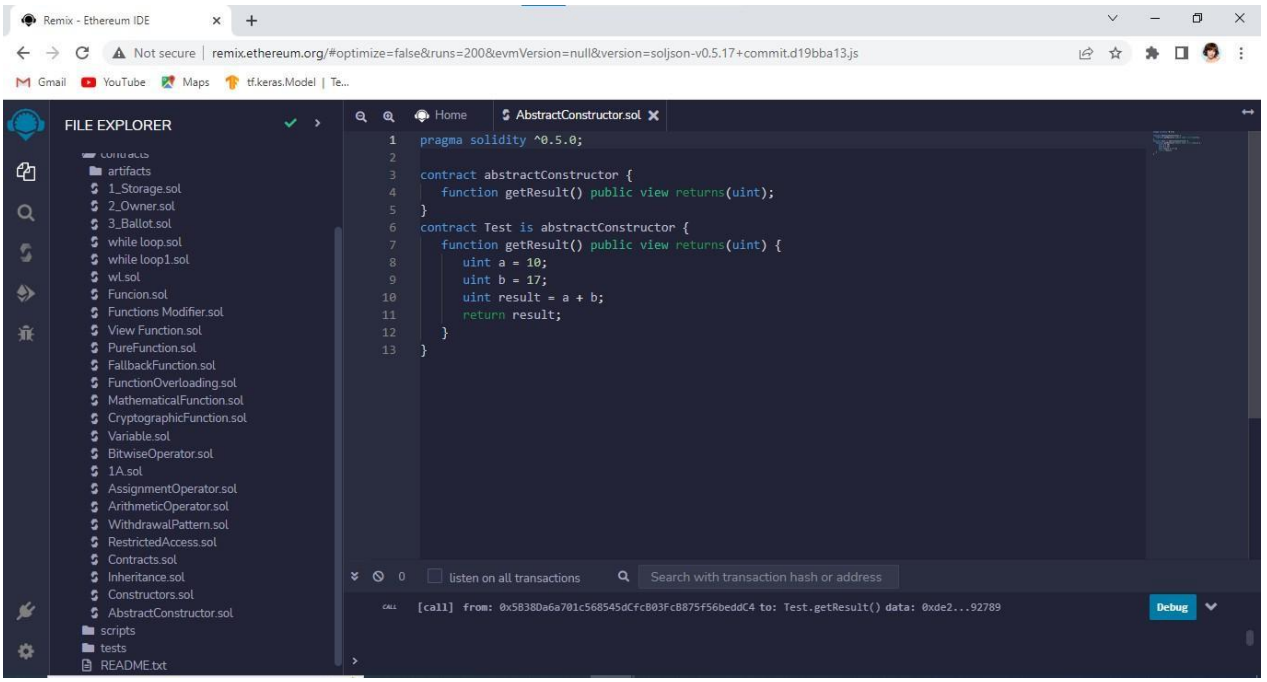
Constructors

Code:

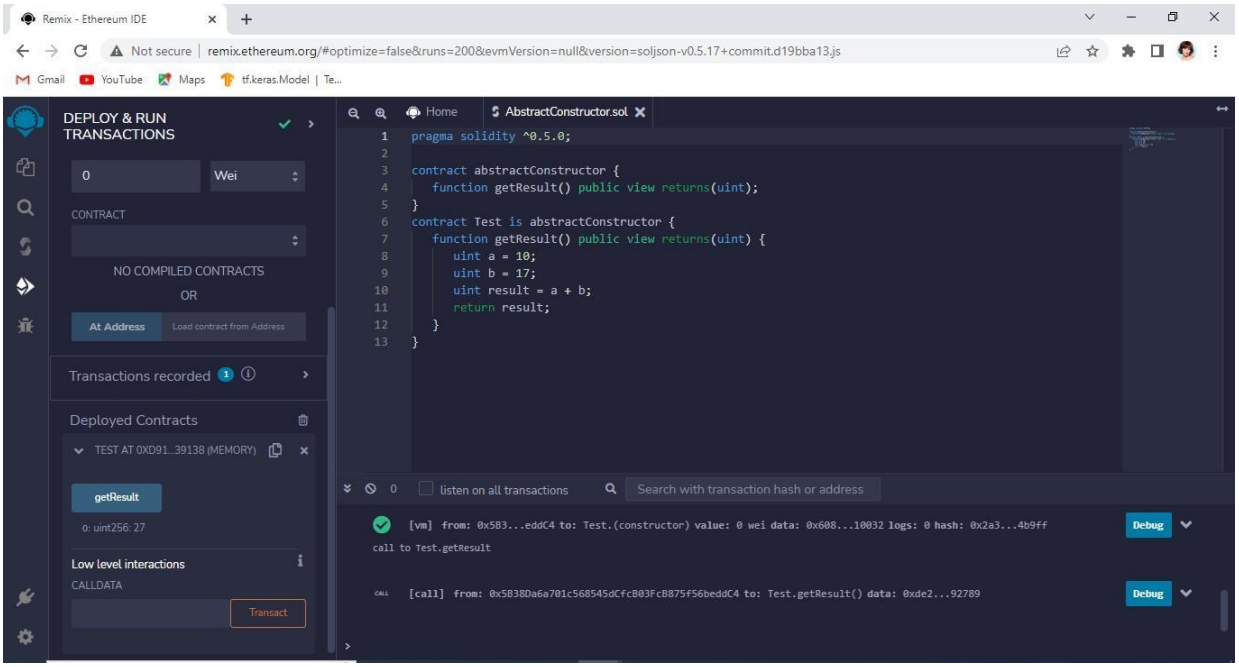

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail YouTube Maps tf.keras.Model | Te...
FILE EXPLORER
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
scripts
tests
README.txt
Constructors.sol
Inheritance.sol
pragma solidity ^0.5.0;
contract constructorExample {
string str;
constructor() public{
str="This is a example of Constructor";
}
function getValue(
) public view returns(
string memory)
{
return str;
}
}
ContractDefinition C
4 reference(s)
listen on all transactions
Search with transaction hash or address

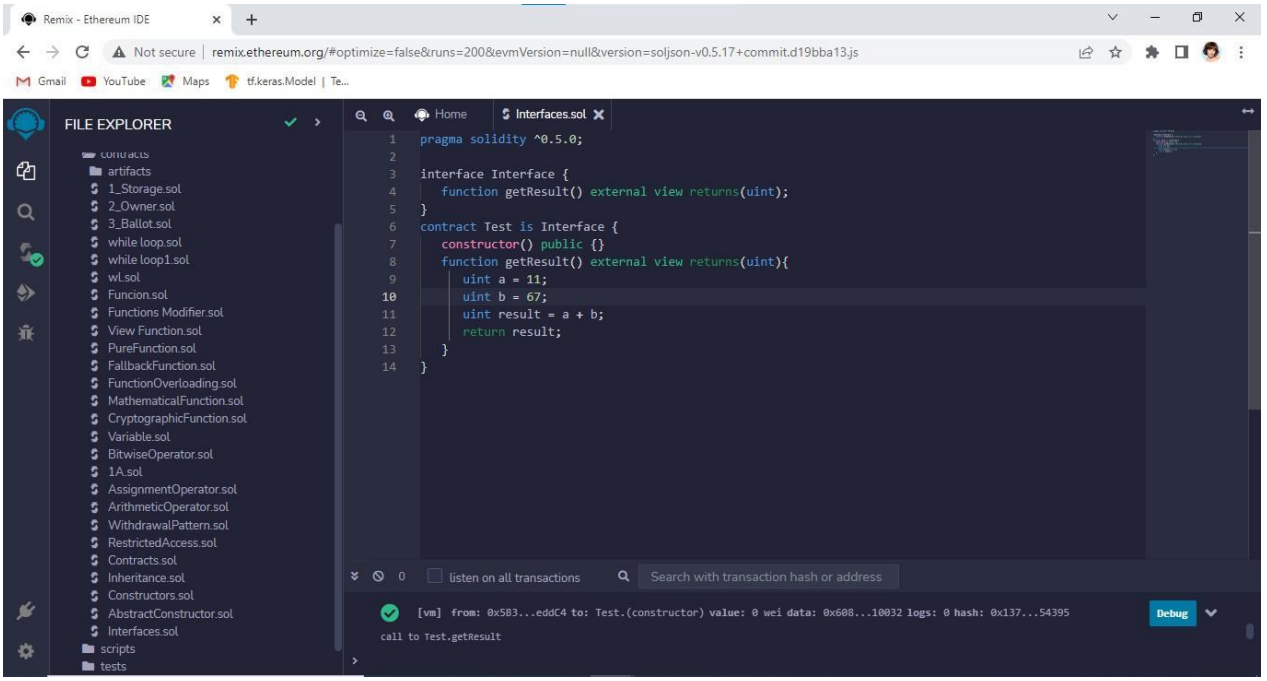
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail YouTube Maps tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
CONTRACT
constructorExample - contracts/Cons
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 26
Deployed Contracts
CONSTRUCTOREXAMPLE AT 0X1C
getValue
0: string: This is a example of Constructor
Low level interactions
CALLDATA
Transact
Constructors.sol
Inheritance.sol
pragma solidity ^0.5.0;
contract constructorExample {
string str;
constructor() public{
str="This is a example of Constructor";
}
function getValue(
) public view returns(
string memory)
{
return str;
}
}
ContractDefinition C
4 reference(s)
listen on all transactions
Search with transaction hash or address
creation of constructorExample pending...
[vm] from: 0x583...eddC4 to: constructorExample.(constructor) value: 0 wei data: 0x608...10032 logs: 0
hash: 0xe3c...18b9c
Debug
call to constructorExample.getValue
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: constructorExample.getValue() data: 0x209...65255
Debug

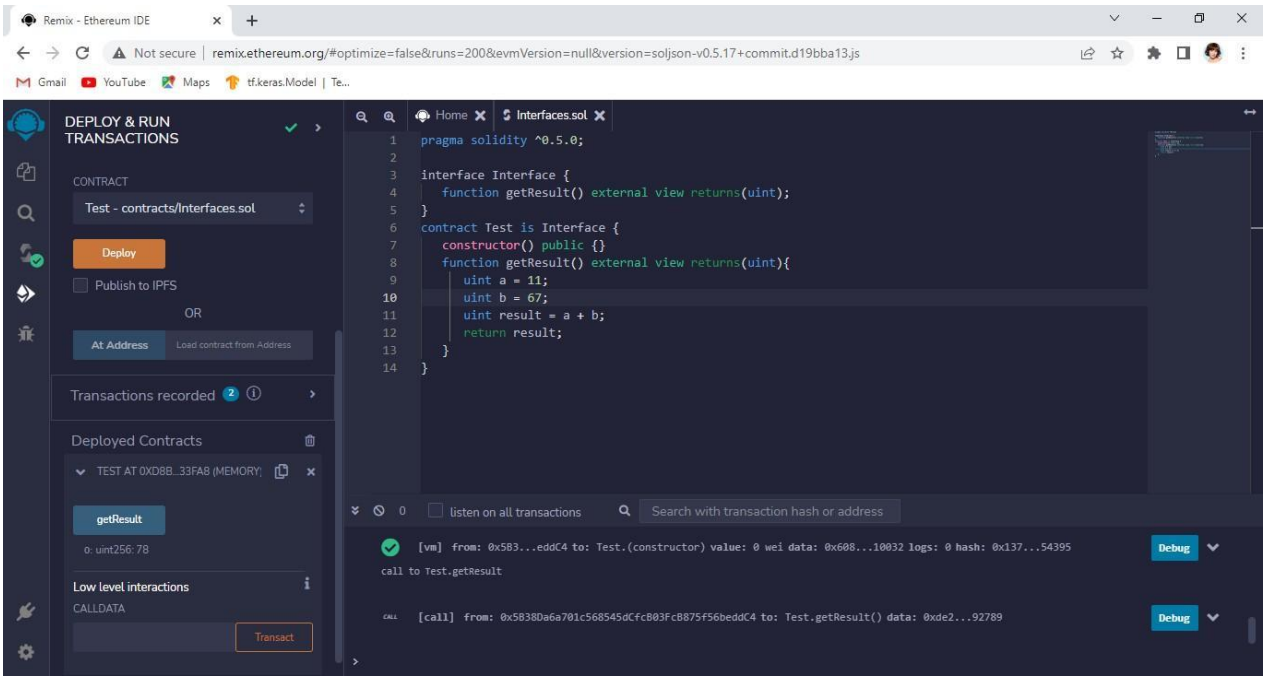

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
scripts
tests
README.txt
Home
AbstractConstructor.sol
pragma solidity ^0.5.0;
contract abstractConstructor {
function getResult() public view returns(uint);
}
contract Test is abstractConstructor {
function getResult() public view returns(uint) {
uint a = 10;
uint b = 17;
uint result = a + b;
return result;
}
}
listen on all transactions
Search with transaction hash or address
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Test.getResult() data: 0xde2...92789
Debug

Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
0
Wei
CONTRACT
NO COMPILED CONTRACTS
OR
At Address
Load contract from Address
Transactions recorded
Deployed Contracts
TEST AT 0XD91...39138 (MEMORY)
getResult
0: uint256: 27
Low level interactions
CALLDATA
Transact
Home
AbstractConstructor.sol
pragma solidity ^0.5.0;
contract abstractConstructor {
function getResult() public view returns(uint);
}
contract Test is abstractConstructor {
function getResult() public view returns(uint) {
uint a = 10;
uint b = 17;
uint result = a + b;
return result;
}
}
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: Test.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0x2a3...4b9ff
Debug
call to Test.getResult
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Test.getResult() data: 0xde2...92789
Debug



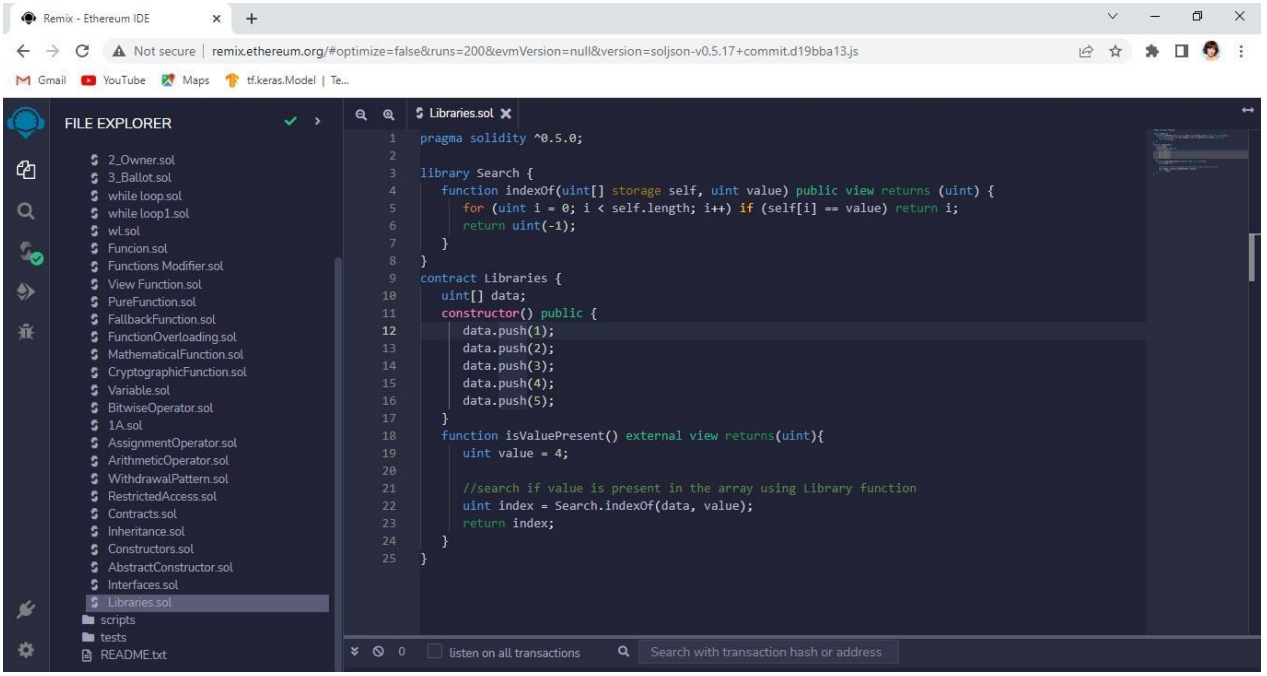
Output:



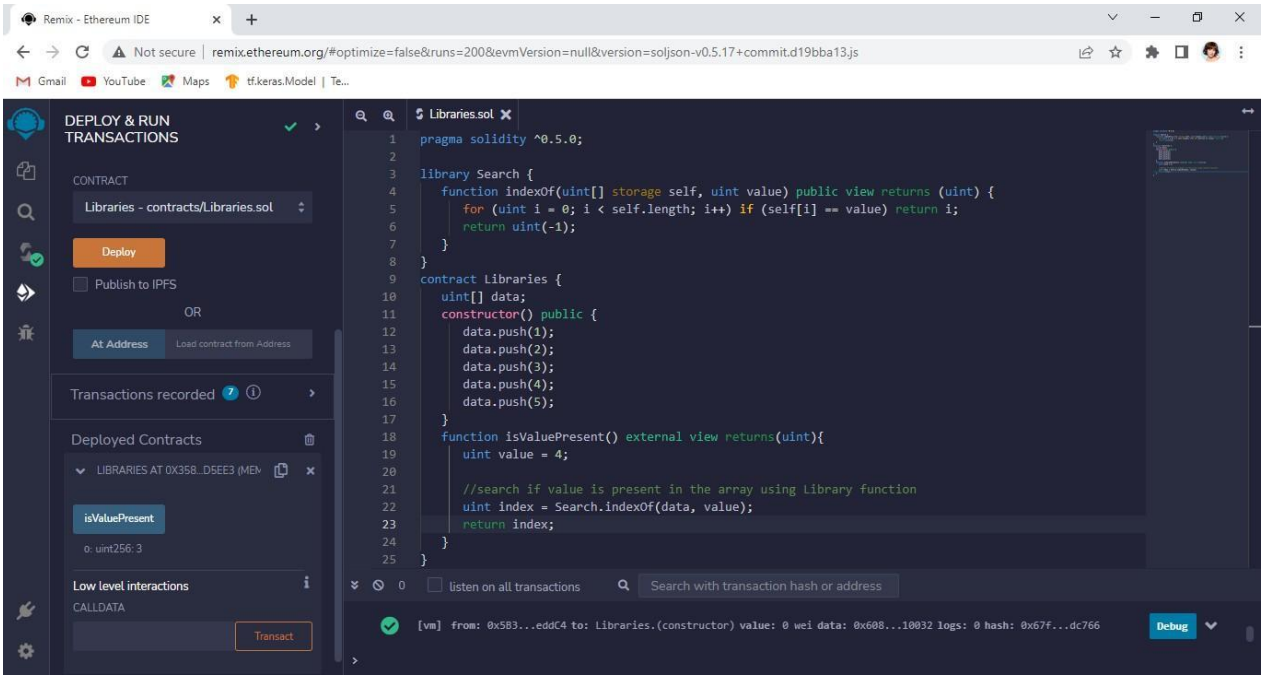
3c) Libraries, Assembly, Events, Error handling.

<u>Libraries</u>

Code:

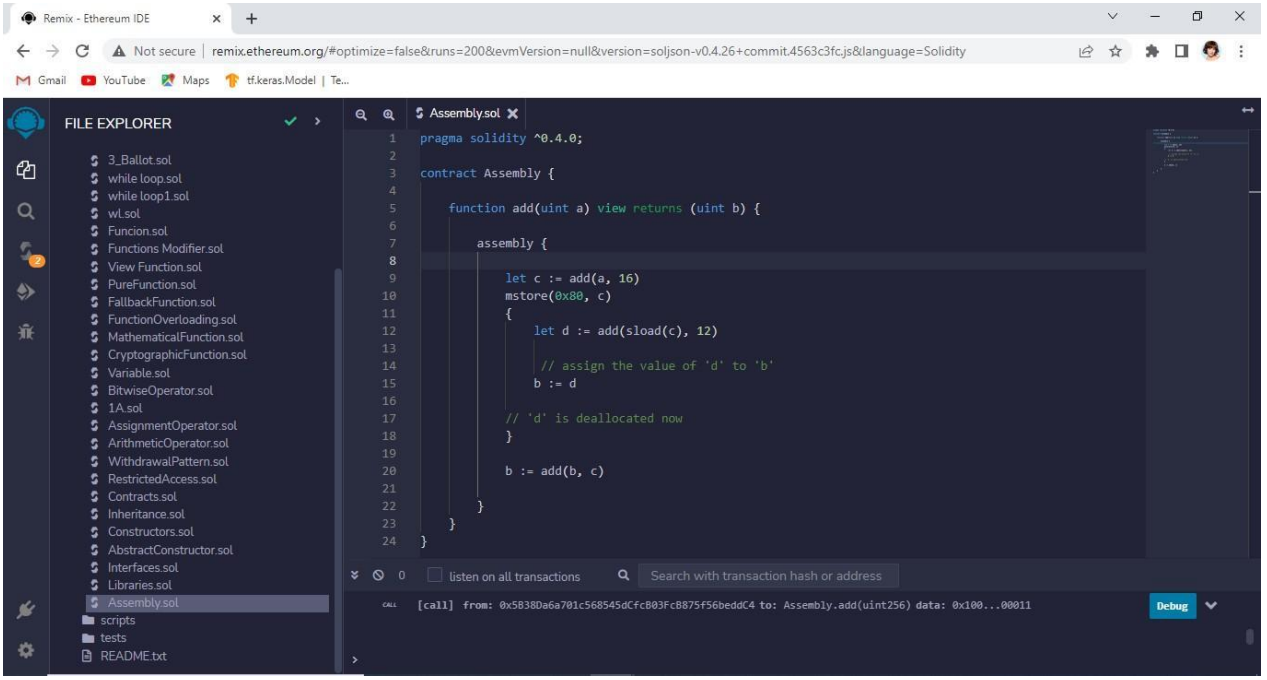


Output:

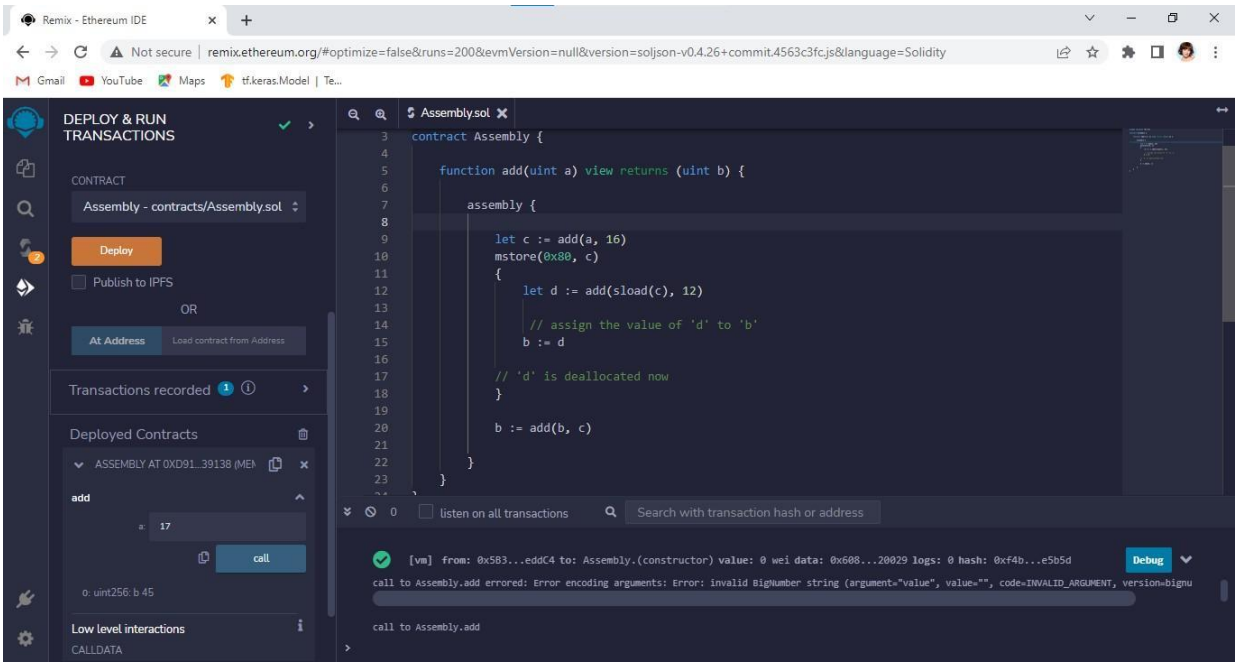


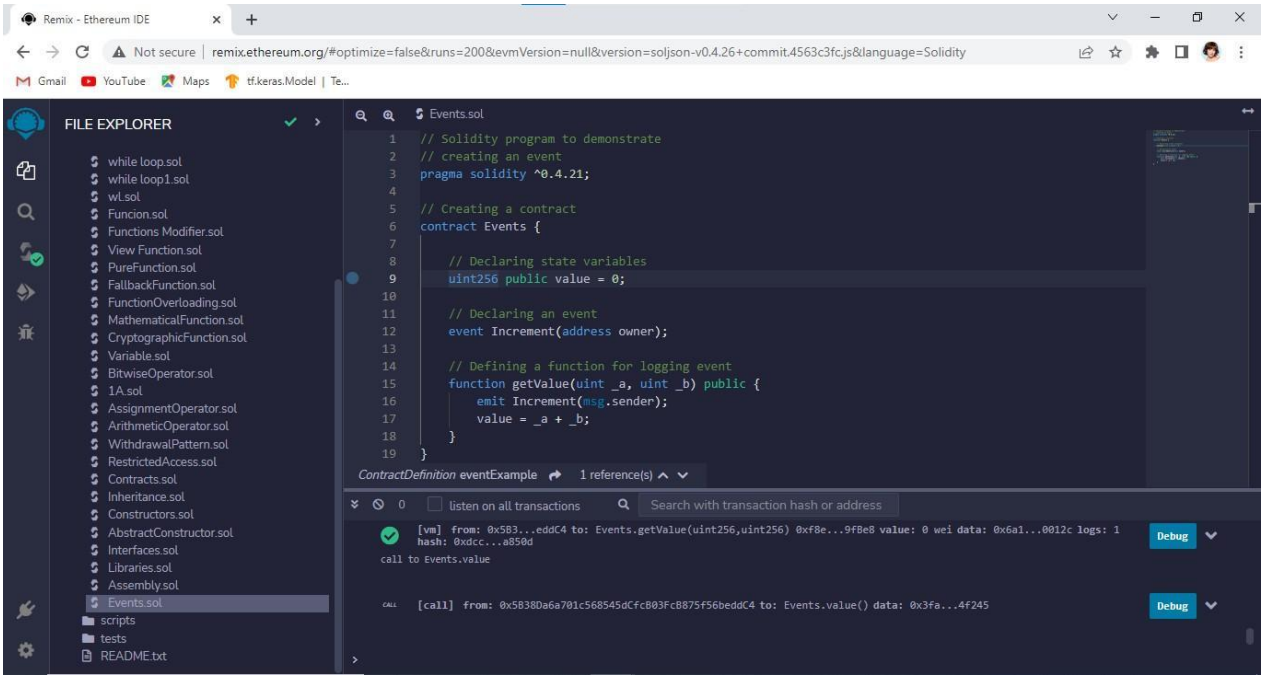
<u>Assembly</u>

Code:

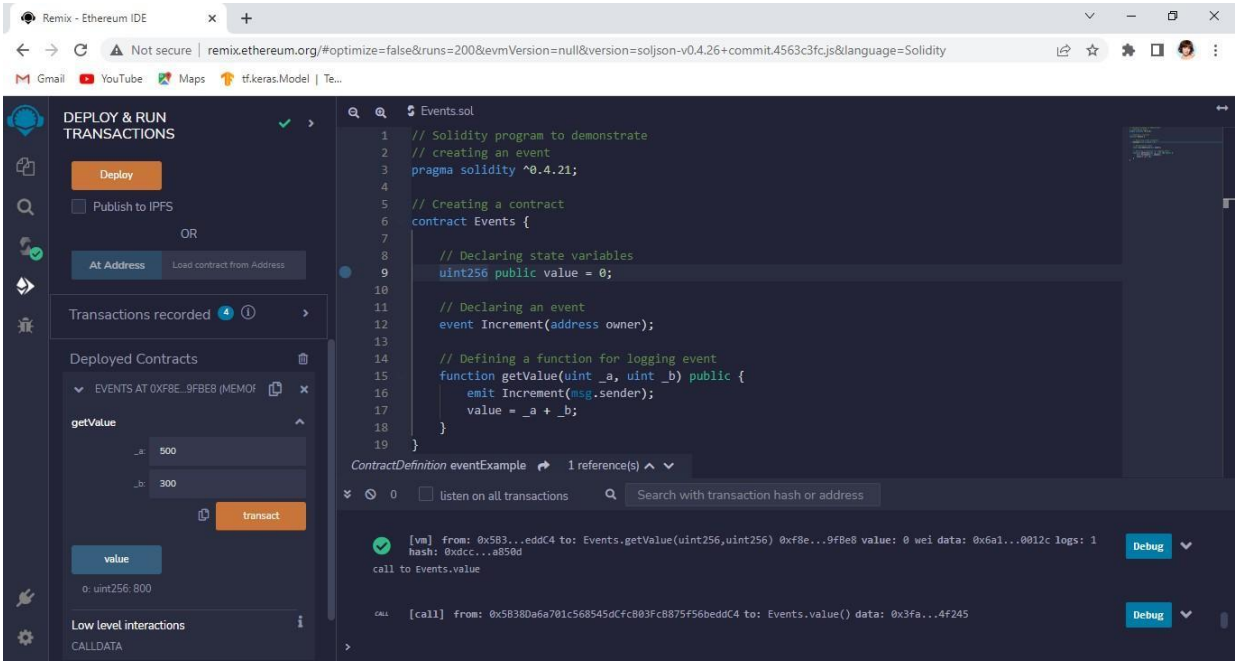

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
scripts
tests
README.txt
Assembly.sol
pragma solidity ^0.4.0;
contract Assembly {
function add(uint a) view returns (uint b) {
assembly {
let c := add(a, 16)
mstore(0x80, c)
{
let d := add(sload(c), 12)
// assign the value of 'd' to 'b'
b := d
// 'd' is deallocated now
}
b := add(b, c)
}
}
}
listen on all transactions
Search with transaction hash or address
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Assembly.add(uint256) data: 0x100...00011
Debug

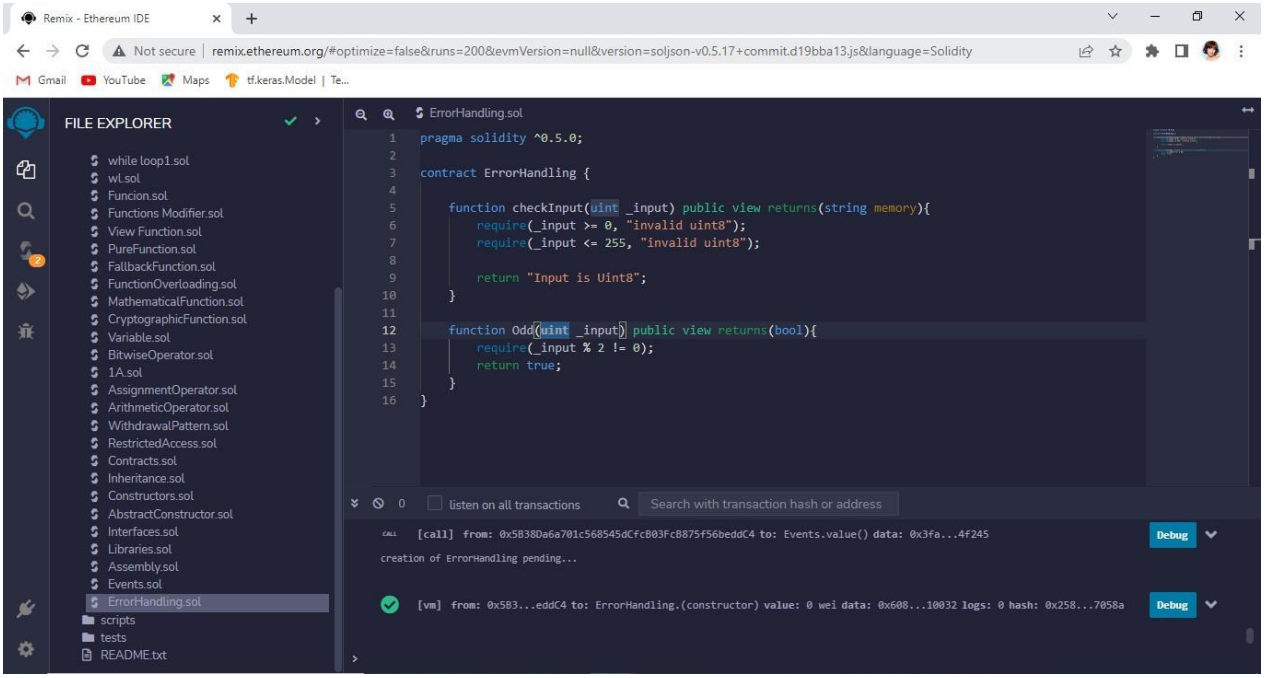
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
CONTRACT
Assembly - contracts/Assembly.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 1
Deployed Contracts
ASSEMBLY AT 0XD91...39138 (MEM
add
a: 17
call
0: uint256: b 45
Low level interactions
CALLDATA
Assembly.sol
contract Assembly {
function add(uint a) view returns (uint b) {
assembly {
let c := add(a, 16)
mstore(0x80, c)
{
let d := add(sload(c), 12)
// assign the value of 'd' to 'b'
b := d
// 'd' is deallocated now
}
b := add(b, c)
}
}
listen on all transactions
Search with transaction hash or address
[vm] from: 0x583...eddC4 to: Assembly.(constructor) value: 0 wei data: 0x608...20029 logs: 0 hash: 0xf4b...e5b5d
Debug
call to Assembly.add errored: Error encoding arguments: Error: invalid BigNumber string (argument="value", value="", code=INVALID_ARGUMENT, version=bignu
call to Assembly.add

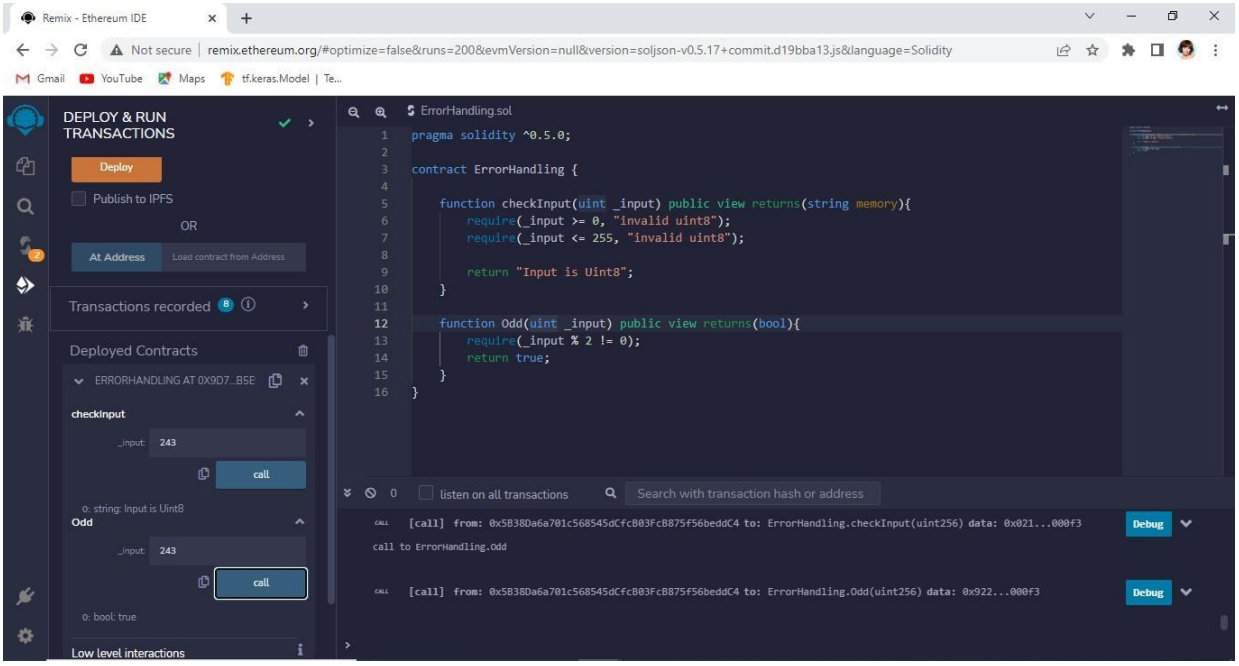

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
scripts
tests
README.txt
Events.sol
// Solidity program to demonstrate
// creating an event
pragma solidity ^0.4.21;
// Creating a contract
contract Events {
// Declaring state variables
uint256 public value = 0;
// Declaring an event
event Increment(address owner);
// Defining a function for logging event
function getValue(uint _a, uint _b) public {
emit Increment(msg.sender);
value = _a + _b;
}
}
ContractDefinition eventExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
[vm] from: 0x583...eddC4 to: Events.getValue(uint256,uint256) 0xf8e...9f8e8 value: 0 wei data: 0x6a1...0012c logs: 1 hash: 0xdcc...a850d
call to Events.value
Debug
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Events.value() data: 0x3fa...4f245
Debug

Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 4
Deployed Contracts
EVENTS AT 0XF8E...9FBE8 (MEMOI
getValue
_a: 500
_b: 300
transact
value
0: uint256: 800
Low level interactions
CALLDATA
Events.sol
// Solidity program to demonstrate
// creating an event
pragma solidity ^0.4.21;
// Creating a contract
contract Events {
// Declaring state variables
uint256 public value = 0;
// Declaring an event
event Increment(address owner);
// Defining a function for logging event
function getValue(uint _a, uint _b) public {
emit Increment(msg.sender);
value = _a + _b;
}
}
ContractDefinition eventExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
[vm] from: 0x583...eddC4 to: Events.getValue(uint256,uint256) 0xf8e...9f8e8 value: 0 wei data: 0x6a1...0012c logs: 1 hash: 0xdcc...a850d
call to Events.value
Debug
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Events.value() data: 0x3fa...4f245
Debug


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail YouTube Maps tf.keras.Model | Te...
FILE EXPLORER
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
scripts
tests
README.txt
ErrorHandling.sol
pragma solidity ^0.5.0;
contract ErrorHandling {
function checkInput(uint _input) public view returns(string memory){
require(_input >= 0, "invalid uint8");
require(_input <= 255, "invalid uint8");
return "Input is Uint8";
}
function Odd(uint _input) public view returns(bool){
require(_input % 2 != 0);
return true;
}
}
listen on all transactions
Search with transaction hash or address
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Events.value() data: 0x3fa...4f245
Debug
creation of ErrorHandling pending...
[vm] from: 0x5B3...eddC4 to: ErrorHandling.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0x258...7058a
Debug

Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail YouTube Maps tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 8
Deployed Contracts
ERRORHANDLING AT 0X9D7...B5E
checkInput
_input: 243
call
0: string: Input is Uint8
Odd
_input: 243
call
0: bool: true
Low level interactions
ErrorHandling.sol
pragma solidity ^0.5.0;
contract ErrorHandling {
function checkInput(uint _input) public view returns(string memory){
require(_input >= 0, "invalid uint8");
require(_input <= 255, "invalid uint8");
return "Input is Uint8";
}
function Odd(uint _input) public view returns(bool){
require(_input % 2 != 0);
return true;
}
}
listen on all transactions
Search with transaction hash or address
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ErrorHandling.checkInput(uint256) data: 0x021...000f3
Debug
call to ErrorHandling.Odd
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ErrorHandling.Odd(uint256) data: 0x922...000f3
Debug

Practical 4

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

Program Steps:

The following are prerequisites for installing the required development tools:

- Operating Systems: Ubuntu Linux 14.04 / 16.04 LTS (both 64-bit), or Mac OS 10.12
- Docker Engine: Version 17.03 or higher
- Docker-Compose: Version 1.8 or higher
- Node: 8.9 or higher (Note: version 9 is not supported)
- npm: v5.x
- git: 2.9.x or higher
- Python: 2.7.x
- A code editor of your choice, we recommend VSCode.

If installing Hyperledger Composer using Linux, the following points need to be kept in mind:

- Login as a normal user, rather than root.
- Do not use sudo su to root.
- When installing prerequisites, use curl, then unzip using sudo.
- Run prereqs-ubuntu.sh as a normal user. It may prompt for root password as some of its actions are required to be run as root.
- Do not use npm with sudo or su to root to use it.
- Avoid installing node globally as root.

Prerequisites
To download prerequisites for Hyperledger Fabric development, run following command –
| curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh
This command will download and install -
| docker-compose , docker-engine, npm, git, python etc

- Give permissions — chmod u+x prereqs-ubuntu.sh
- Run Script — ./prereqs-ubuntu.sh (restart system after it)
- Essential CLI tools — npm install -g composer-cli@0.20

Steps

1. **Create a directory — mkdir multichain_network**

```
cd multichain_network

curl -sSL http://bit.ly/2ysbOFE | bash -s 1.2.0

export PATH=<path to download location>/multichain_network/fabric-
samples/first-network/bin:$PATH
```

2. **Generate Certificates**

```
cd first-network

export FABRIC_CFG_PATH=$PWD

cryptogen generate — config=./crypto-config.yaml
```

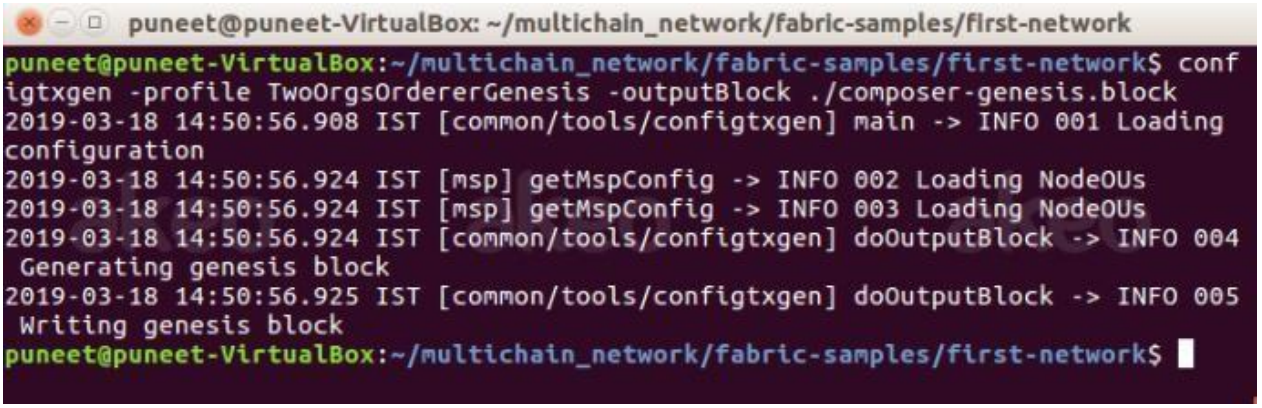
This will create all certificates for orderers and peers in crypto-config folder.

Copy certificates for all peers and orderer to temporary folder.

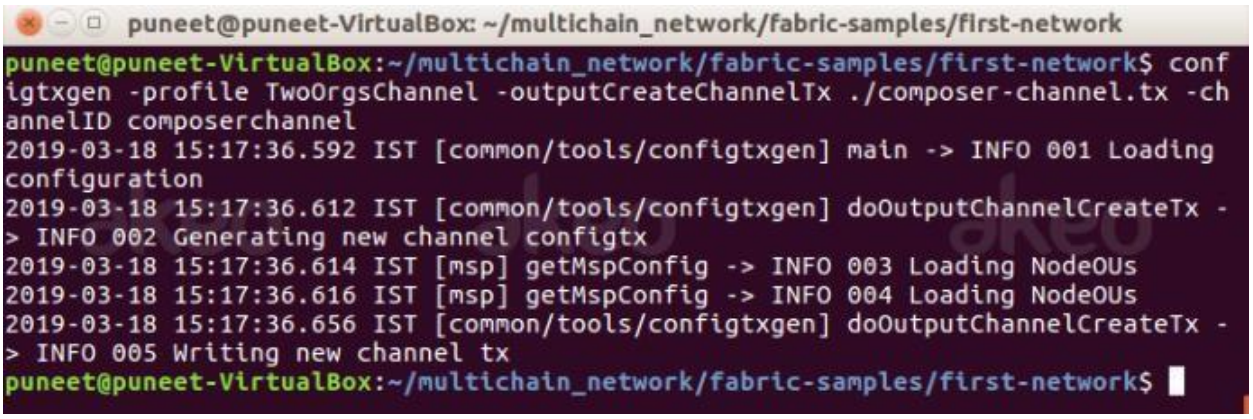
```
In first-network folder run this command — mkdir -p tmp/composer/org1

awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' ./crypto-
config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com
/tls/ca.crt > ./tmp/composer/org1/ca-org1.txt

awk 'NF {sub(/\r/, ""); printf "%s\\n",$0;}' ./crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com/t
ls/ca.crt > ./tmp/composer/ca-orderer.txt

export ORG1=./crypto-
config/peerOrganizations/org1.example.com/users/Admin@org1.example.com
/msp

cp -p $ORG1/signcerts/A*.pem ./tmp/composer/org1

cp -p $ORG1/keystore/*_sk ./tmp/composer/org1
```

3. **Create genesis block and channeltx**

```
configtxgen -profile TwoOrgsOrdererGenesis -outputBlock ./composer-
genesis.block
```




```
configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./composer-
channel.tx -channelID composerchannel
```



4. Update CA keys in docker composer file

Open project in code editor.

We have to change CA keys in “docker-compose-e2e-template.yaml” file, therefore navigate to this file in vscode.

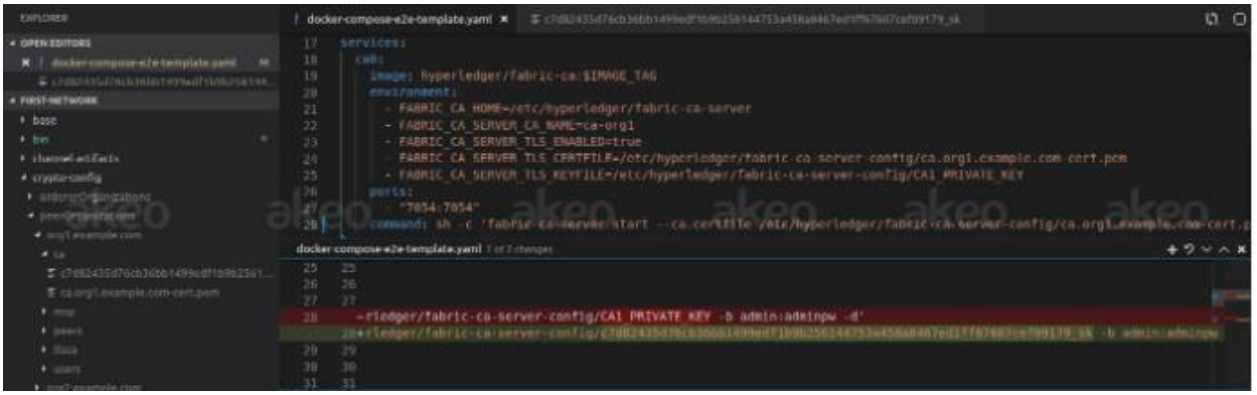
Under services section we have two certificate authorities named “ca0” and “ca1”.

For ca0 — go to command section under ca0 and find CA1_PRIVATE_KEY and replace it with private key –

```
C7d82435d76cb36bb1499edf1b9b256144753a458a8467ed1ff67607cef09179_sk
```

This key can be found in –

```
"first-network/crypto-
config/peerOrganizations/org1.example.com/ca/c7d82435d76cb36bb1499edf1
b9b256144753a458a8467ed1ff67607cef09179_sk"
```

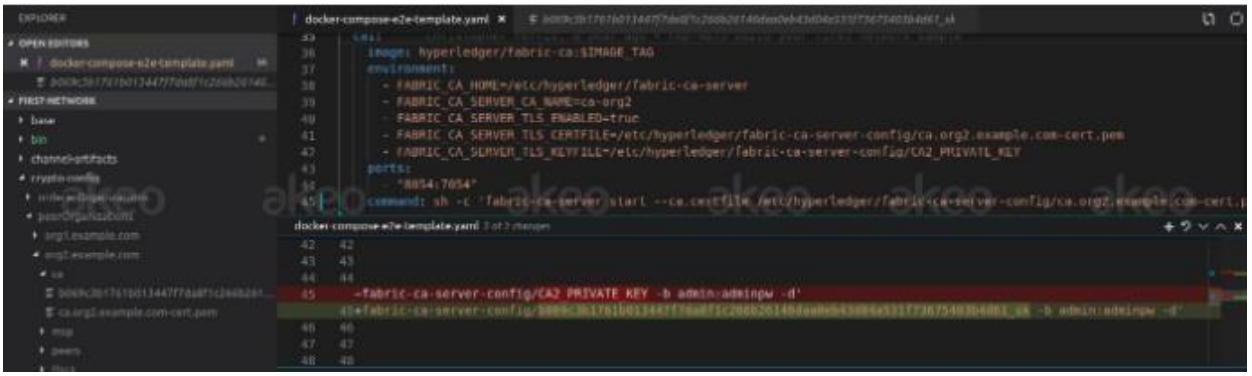


For ca1 — goto command section under ca1 and find CA2_PRIVATE_KEY and replace it with private key –

```
B069c3b1761b013447f7da8f1c266b26146daa0eb43d04a531f73675403b4d61_sk
```

This key can be found in –

```
"first-network/crypto-
config/peerOrganizations/org1.example.com/ca/b069c3b1761b013447f7da8f1
c266b26146daa0eb43d04a531f73675403b4d61_sk
```



Steps of Starting Hyperledger Fabric

1. Open docker-compose-base.yaml file which is present in the bin folder and introduce following changes.

```
Change orderer volume binding to -


../composer-
genesis.block:/var/hyperledger/orderer/orderer.genesis.block
```

As shown in the figure below –

```
orderer.example.com:
  container_name: orderer.example.com
  image: hyperledger/fabric-orderer:$IMAGE_TAG
  environment:
    - ORDERER_GENERAL_LOGLEVEL=INFO
    - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
    - ORDERER_GENERAL_GENESISMETHOD=file
    - ORDERER_GENERAL_GENESISFILE=/var/hyperledger/orderer/orderer.genesis.block
    - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
    - ORDERER_GENERAL_LOCALMSPDIR=/var/hyperledger/orderer/msp
    # enabled TLS
    - ORDERER_GENERAL_TLS_ENABLED=true
    - ORDERER_GENERAL_TLS_PRIVATEKEY=/var/hyperledger/orderer/tls/server.key
    - ORDERER_GENERAL_TLS_CERTIFICATE=/var/hyperledger/orderer/tls/server.crt
    - ORDERER_GENERAL_TLS_ROOTCAS=[/var/hyperledger/orderer/tls/ca.crt]
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric
  command: orderer
  volumes:
  - ../composer-genesis.block:/var/hyperledger/orderer/orderer.genesis.block
  - ../crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/msp:/var/hyperledger/orderer/msp
  - ../crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/tls/:/var/hyperledger/orderer/tls
  - orderer.example.com:/var/hyperledger/production/orderer
  ports:
    - 7050:7050
```

Change peer volume binding to (for all 4 peers)-

```
- ../:/etc/configtx
```

```
peer0.org1.example.com:
  extends:
    file: peer-base.yaml
    service: peer-base
  environment:
    - CORE_PEER_ID=peer0.org1.example.com
    - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
    - CORE_PEER_GOSSIP_BOOTSTRAP=peer1.org1.example.com:7051
    - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org1.example.com:7051
    - CORE_PEER_LOCALMSPID=Org1MSP
  volumes:
      - ../:/etc/configtx
      - /var/run/:/host/var/run/
      - ../crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/msp:/etc/hyperledger/fabric/msp
      - ../crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls:/etc/hyperledger/fabric/tls
      - peer0.org1.example.com:/var/hyperledger/production
  ports:
    - 7051:7051
    - 7053:7053
```

2. To start fabric, run the following command –

```
docker-compose -f docker-compose-cli.yaml -f docker-compose-couch.yaml
up -d 2>&1
```

The output of the above command is shown below –

```
puneet@puneet-VirtualBox:~/multichain_network/fabric-samples/first-network$ docker-compose -f docker-compose-cli.yaml -f docker-compose-couch.yaml up -
d 2>&1
Creating orderer.example.com ...
Creating couchdb1 ...
Creating couchdb3 ...
Creating couchdb2 ...
Creating orderer.example.com
Creating couchdb0 ...
Creating couchdb2
Creating couchdb1
Creating couchdb3
Creating couchdb0 ... done
Creating peer0.org1.example.com ...
Creating couchdb3 ... done
Creating peer0.org2.example.com ...
Creating peer1.org1.example.com ...
Creating peer1.org2.example.com ...
Creating peer1.org1.example.com
Creating peer0.org2.example.com
Creating peer1.org2.example.com ... done
Creating cli ...
Creating cli ... done
```

3. After completing all these steps, you can run command –

```
docker ps -a
```

This will list all running containers regarding our network setup, in our case it will list 10 containers.

```
CONTAINER ID        IMAGE                               COMMAND                  CREATED             STATUS
    PORTS                                       NAMES
badfe5ea8a92        hyperledger/fabric-peer:latest      "peer node start"        4 hours ago         Up 3 hours
                                                peer1.org1.example.com
50d9fa8f5dfb        hyperledger/fabric-peer:latest      "peer node start"        4 hours ago         Up 3 hours
                                                peer0.org2.example.com
ee4b501b7d00        hyperledger/fabric-peer:latest      "peer node start"        4 hours ago         Up 3 hours
                                                peer0.org1.example.com
32f0f4426a22        hyperledger/fabric-peer:latest      "peer node start"        4 hours ago         Up 3 hours
                                                peer1.org2.example.com
ec8f9df17258        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   4 hours ago         Up 4 hours
    4369/tcp, 9100/tcp, 0.0.0.0:8984->5984/tcp  couchdb3
2b4be64efcfe        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   4 hours ago         Up 4 hours
    4369/tcp, 9100/tcp, 0.0.0.0:7984->5984/tcp  couchdb2
f922625a027e        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   4 hours ago         Up 4 hours
    4369/tcp, 9100/tcp, 0.0.0.0:5984->5984/tcp  couchdb0
8de729e174b0        hyperledger/fabric-couchdb          "tini -- /docker-ent…"   4 hours ago         Up 4 hours
    4369/tcp, 9100/tcp, 0.0.0.0:6984->5984/tcp  couchdb1
18830d7ccf5e        hyperledger/fabric-tools:latest     "/bin/bash"              11 hours ago        Up 11 hours
                                                cli
c7ef15868cd5        hyperledger/fabric-orderer:latest   "orderer"                11 hours ago        Up 11 hours
    0.0.0.0:7050->7050/tcp                      orderer.example.com
```

Proposed network setup is complete, our network have -

- One orderer
- Two Organizations
- Four peers (two peers on each organization)
- Couchdb for all peers

4. Creating channel

```
docker exec peer0.org1.example.com peer channel create -o
orderer.example.com:7050 -c composerchannel -f /etc/configtx/composer-
channel.tx — tls true — cafile /etc/configtx/crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com/m
sp/tlscacerts/tlsca.example.com-cert.pem
```

5. Joining first peer where channel is created —

```
docker exec -e


"CORE_PEER_MSPCONFIGPATH=/etc/configtx/tmp/composer/org1/Admin@org1.ex
ample.com/msp" peer0.org1.example.com peer channel join -b composer-
genesis.block
```

6. For other peers, we have to first fetch the config of block from

   orderer For Fetching –

```
docker exec -e

"CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org1.example
.com/msp"

peer1.org1.example.com peer channel fetch config -o

orderer.example.com:7050 -c TwoOrgsChannel
```

For joining channel –

```
docker exec -e

"CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/users/Admin@org1.example
.com/msp"

peer1.org1.example.com peer channel join -b
composerchannel_config.block
```

Install chainCode on every machine

```
composer network install -a device-network.bna -c

PeerAdmin@multi_org1
```

Start chaincode on one machine

```
composer network start -n device-network -V 0.0.2-
deploy.${bna_deployment_version} -A admin -S adminpw -c
PeerAdmin@multi_org1
```

Conclusion

We learnt about installing Hyperledger Fabric development tools and Hyperledger Composer using Linux. We also successfully deployed a Hyperledger Fabric network having one orderer, two organizations and two peers in each organization.

Practical 5

Aim: Demonstrate the running of the blockchain node.

1a) A simple client class that generates the private and public keys by using the built in Python RSA algorithm 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.PublicKey import RSA
from Crypto.Signature import PKCS1_v1_5
import binascii

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(se
                                        lf._private_key)
    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER'
)).decode('ascii')
Dinesh = Client()
print (Dinesh.identity)
```

Output:

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

Code:

```
class Transaction:
                                    def __init_(self,
                                        sender,
                                        recipient,
                                        value):
                                        self.sender =
                                        sender
                                        s
                                        e
                                        l
                                        f
                                        .
                                        r
                                        e
                                        c
                                        i
                                        p
                                        i
                                        e
                                        n
                                        t
                                        =
                                        r
                                        e
                                        c
                                        i
                                        p
                                        i
                                        e
                                        n
                                        t
                                        s
                                        e
                                        l
                                        f
                                        .
                                        v
                                        a
                                        l
                                        u
                                        e
                                        =
                                        v
                                        a
                                        l
                                        u
                                        e
                                        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')
Dinesh = Client()
Ramesh = Client()
t = Transaction(Dinesh,Ramesh.identity,5.0)
signature = t.sign_transaction()
print (signature)
```

Output:

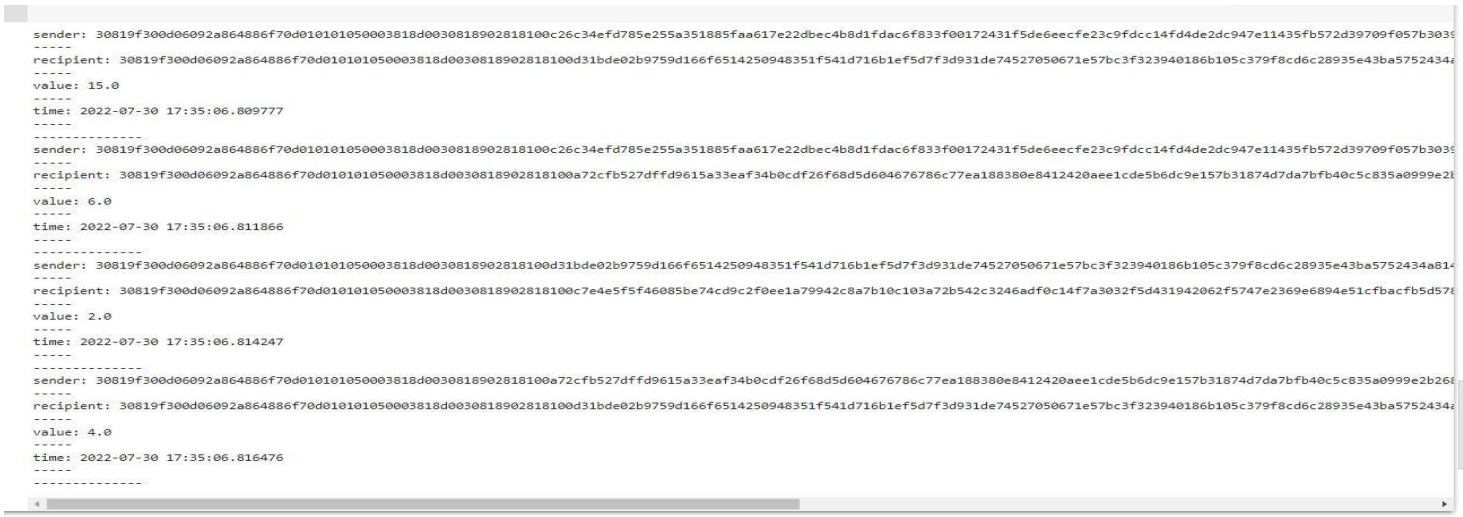
1c) Create Multiple Transaction and display them.

Code:

```
def display_transaction(transaction):
  dict = transaction.to_dict()
  print ("sender: " + dict['sender'])
```

```
    print ('      ')
    print ("recipient: " + dict['recipient'])
    print ('      ')
    print ("value: " + str(dict['value']))
    print ('      ')
    print ("time: " + str(dict['time']))
    print ('      ')

    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)

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

Output:



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

Code:

```
class Block:
                                    def __init(self):self.verified_transa
```

```
ctions=[]self.previous_block
                                    ock_hash=""self.Nonce=""
```

```
last_block_hash = ""

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 = []

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

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

Output:

```
Number of blocks in the chain: 1
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a4a79ac0645d8483c7c97fb6b67e1bf90f17d87d1f
-----
value: 500.0
-----
time: 2022-07-30 17:50:16.288802
-----
--------------
====================================
```

1e) Create a mining function and test it.

Code:

```
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: "+ dig
est)
    return digest

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

Output:

```
after 238 iterations found nonce: 1124711db6185591392c6a06c24a3c2ebbaeca647fb8374824f939ada27c09e9
after 353 iterations found nonce: 11e0a4b57bb76496ecc6ab5a3c126165bc9dbbaf80f664e7e192a422360c3884
after 419 iterations found nonce: 11280dfd9ab05b3dbfa8699901537329414O8faa4a3b0832819b161f52321c08
after 511 iterations found nonce: 1150944bd7ea429acd052da390b148f27eb881686e0f8bfcdf77f833af878e2e
after 822 iterations found nonce: 110e61d41d94b48f6dfcb6f43f9ceafe2ab7168456dab088e05126d43d0366ef
after 924 iterations found nonce: 11147bf32bafeee4505b5cc40c3b227366d8a41ab3bd740e437c74400b7a8127
'6d80c4cf3cd4fb49aad25deced696c3a48e17974ecbfb1441e2128517f0152b2'
```

1f) Add blocks to the miner and dump the blockchain

Code:

```
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

    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)
```

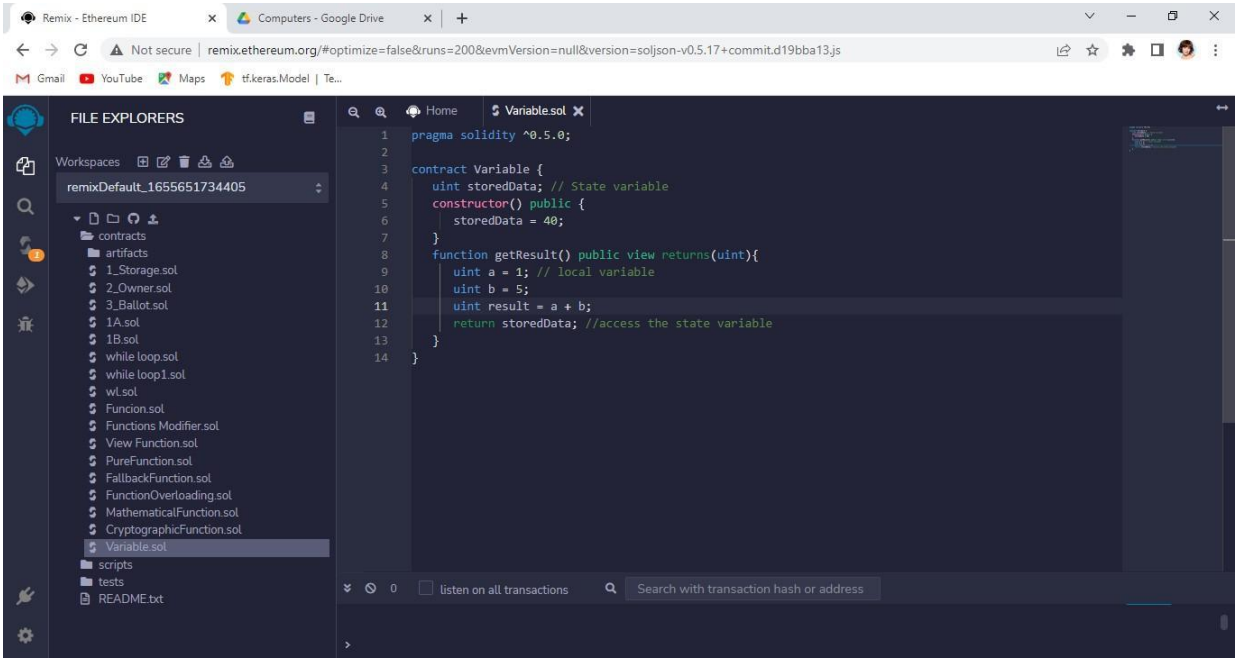
Output:

```
after 308 iterations found nonce: 111b1715b007a9920e76b241e9a4dde15a50b00faeed353bf9279311deb59804
after 398 iterations found nonce: 11a65418aca0201df85c2eeacabbcf0624092abb64bf6e159ccf5539cbb1cfe1
after 527 iterations found nonce: 11310267f14590e45be2375fe94e881c65f86020e3fcf7844a849241783680a7
after 718 iterations found nonce: 117ba8286d575afe73007e6f588b6f541580bec444d5165a3f71d9e4b623f3fa
after 308 iterations found nonce: 111b1715b007a9920e76b241e9a4dde15a50b00faeed353bf9279311deb59804
after 398 iterations found nonce: 11a65418aca0201df85c2eeacabbcf0624092abb64bf6e159ccf5539cbb1cfe1
after 527 iterations found nonce: 11310267f14590e45be2375fe94e881c65f86020e3fcf7844a849241783680a7
after 718 iterations found nonce: 117ba8286d575afe73007e6f588b6f541580bec444d5165a3f71d9e4b623f3fa
after 308 iterations found nonce: 111b1715b007a9920e76b241e9a4dde15a50b00faeed353bf9279311deb59804
after 398 iterations found nonce: 11a65418aca0201df85c2eeacabbcf0624092abb64bf6e159ccf5539cbb1cfe1
after 527 iterations found nonce: 11310267f14590e45be2375fe94e881c65f86020e3fcf7844a849241783680a7
after 718 iterations found nonce: 117ba8286d575afe73007e6f588b6f541580bec444d5165a3f71d9e4b623f3fa
```

2a) Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs, Mappings, Conversions, Ether Units, Special Variables. Variable

<u>Variable</u>

Code:

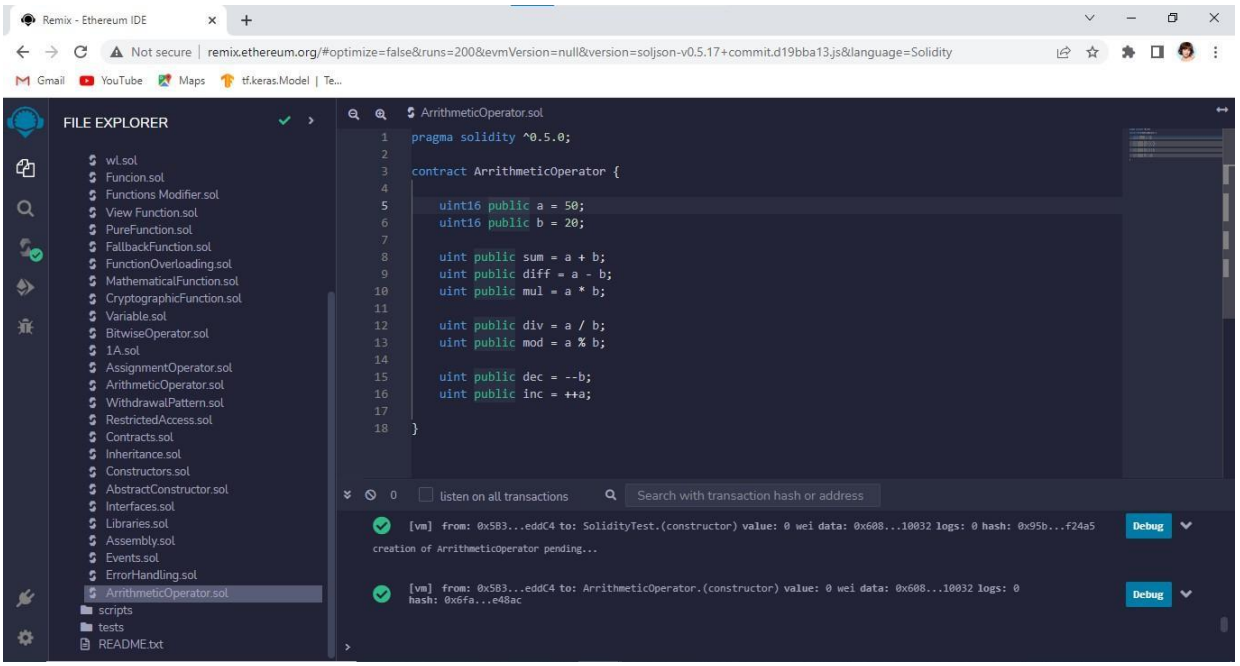


Output:

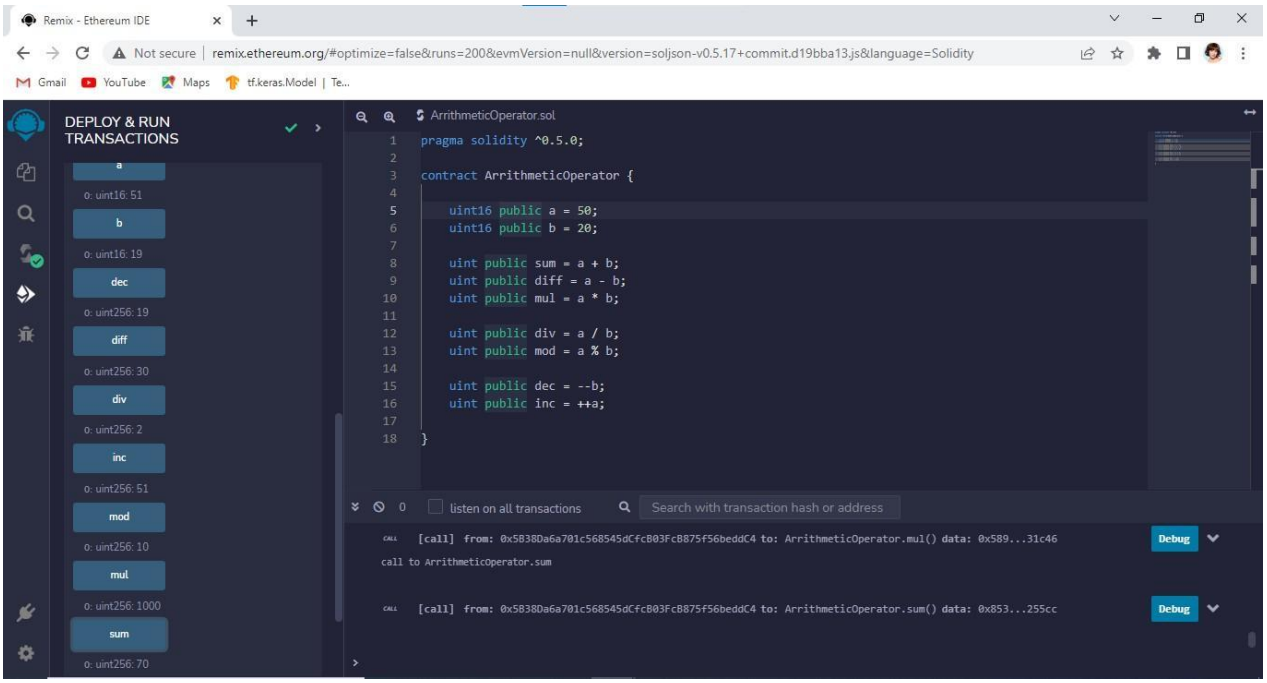
Operators:

Arithmetic Operator

Code:

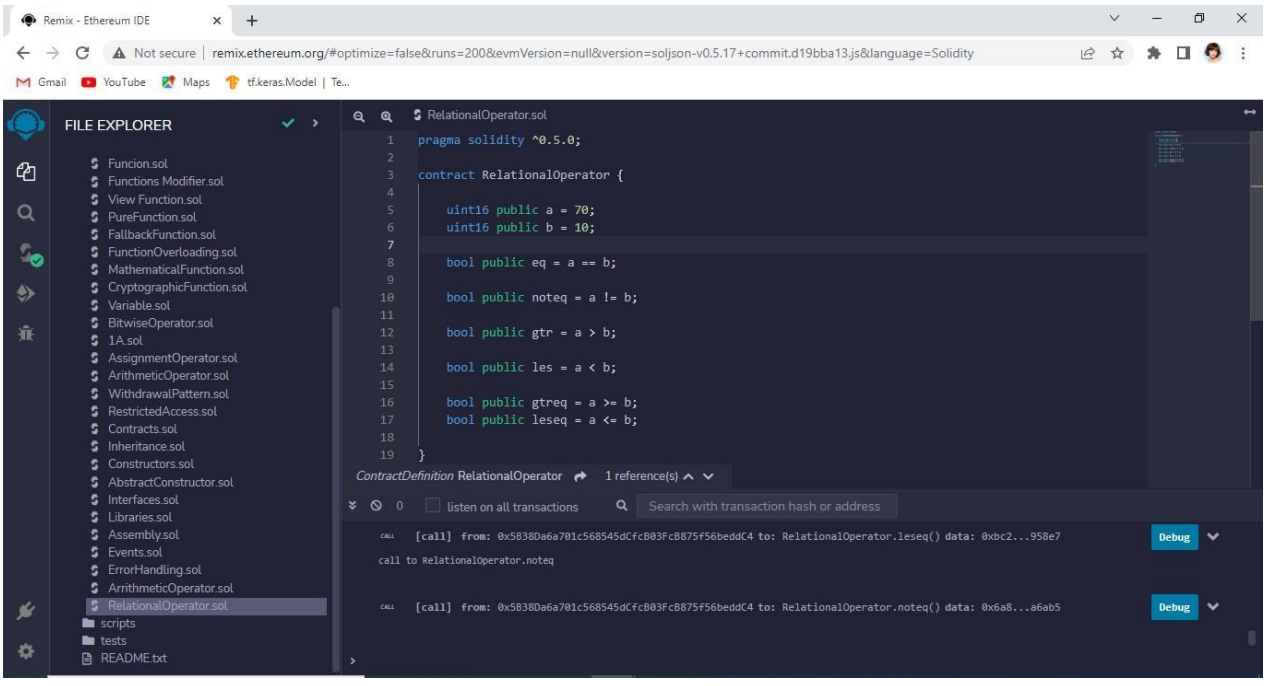


Output:

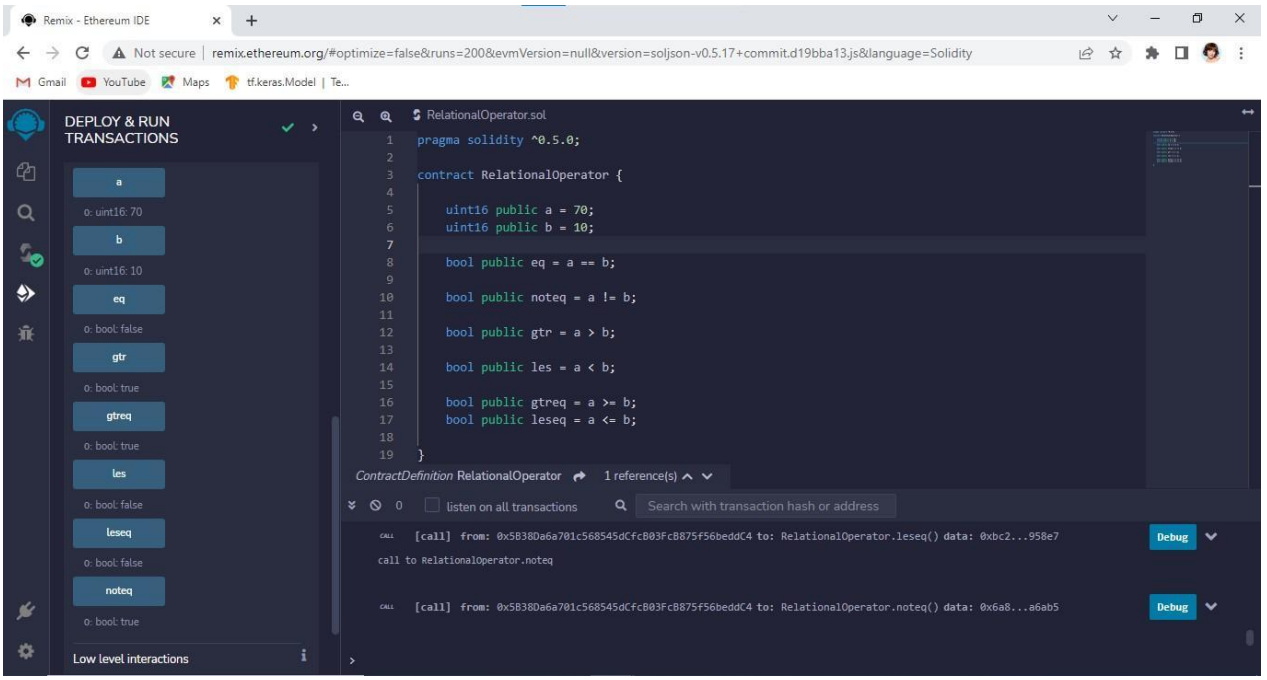


Relational

Operator Code:

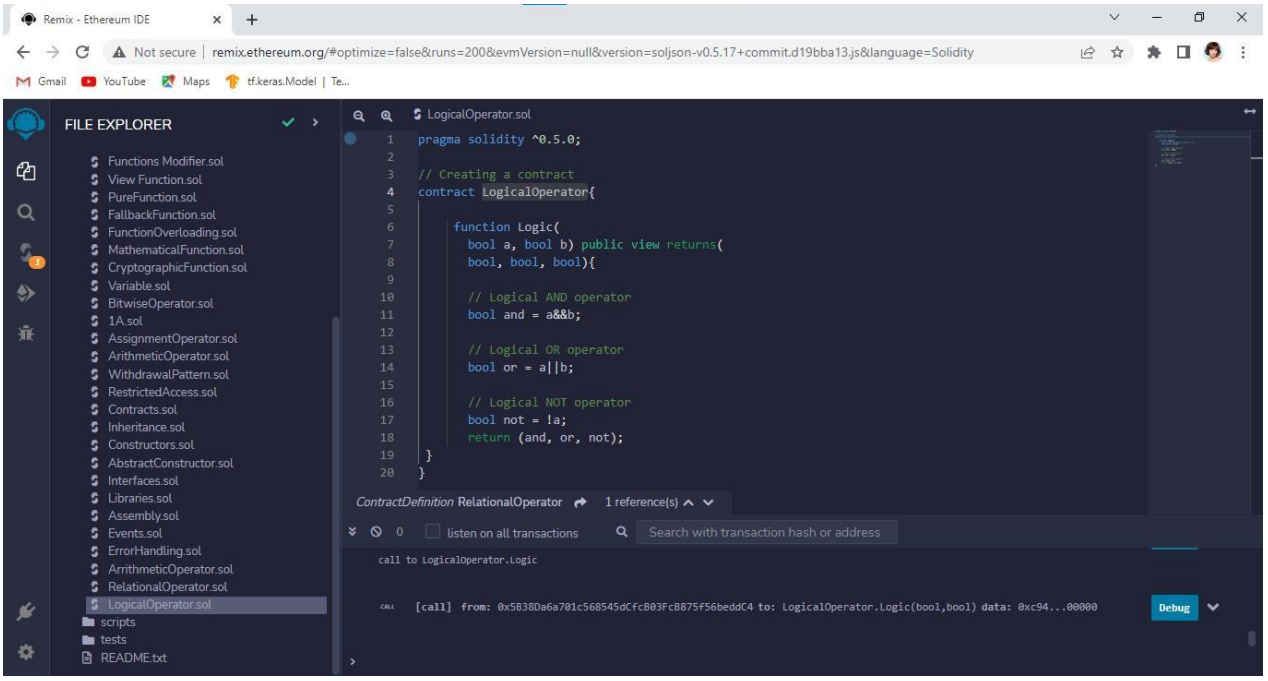


Output:

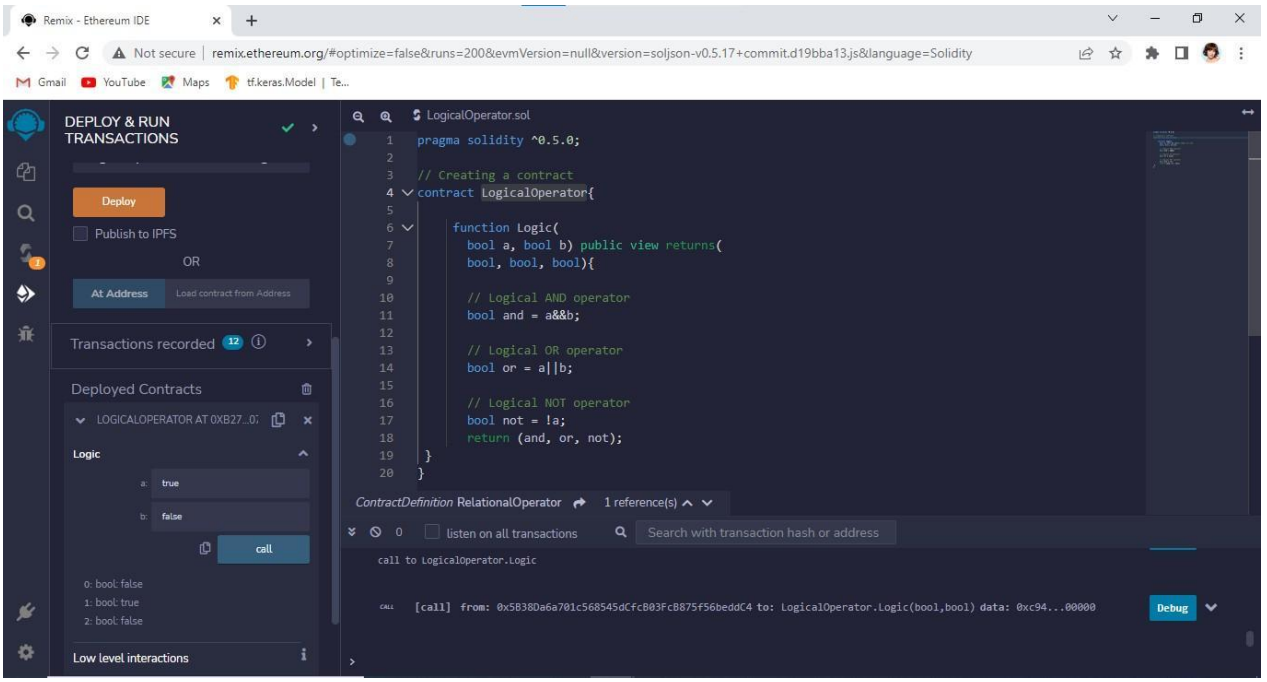


Logical Operator

Code:

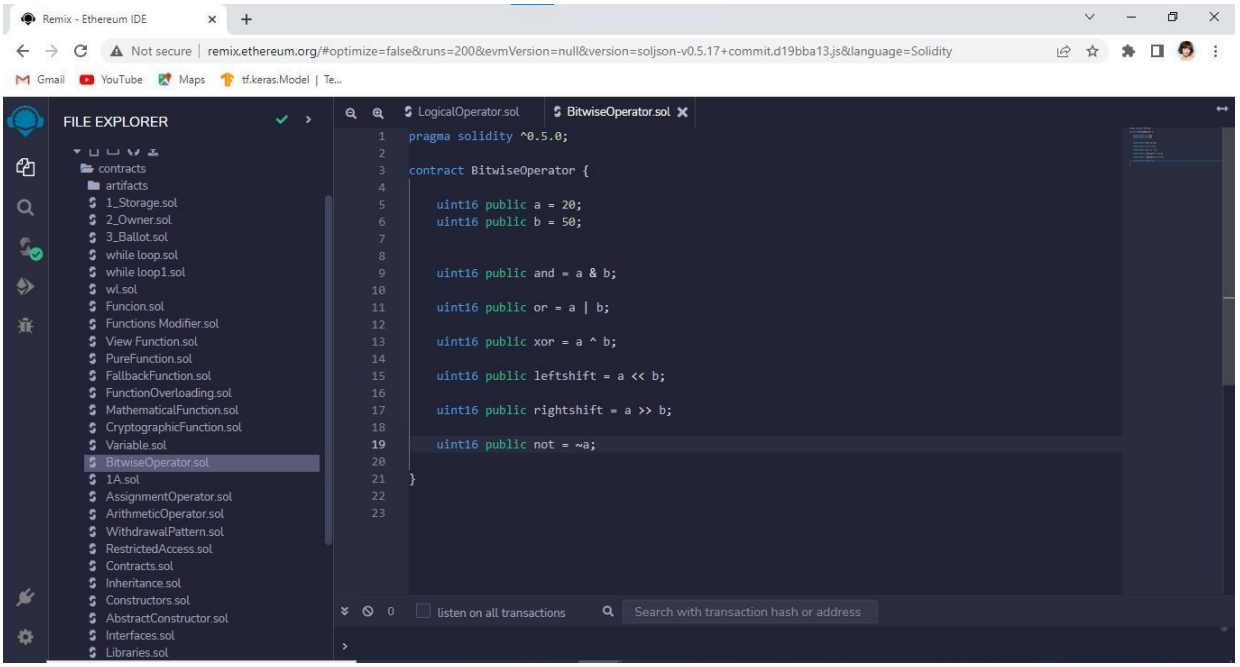


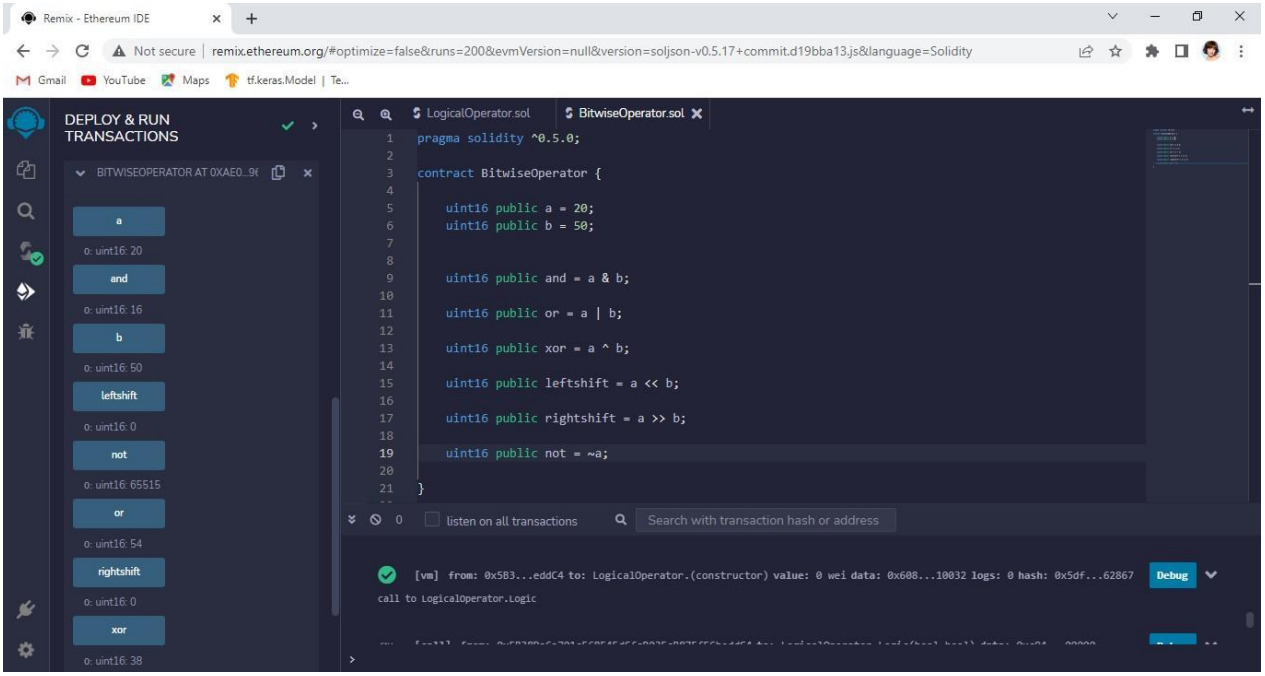
Output:



Bitwise Operator

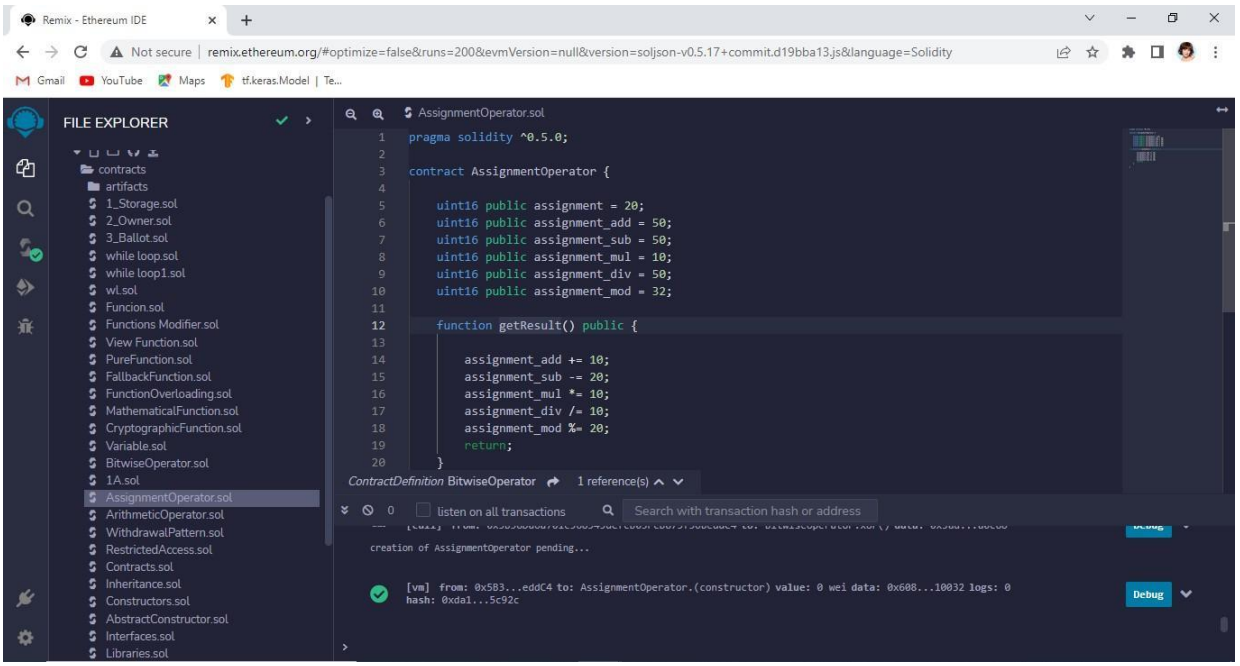
Code:

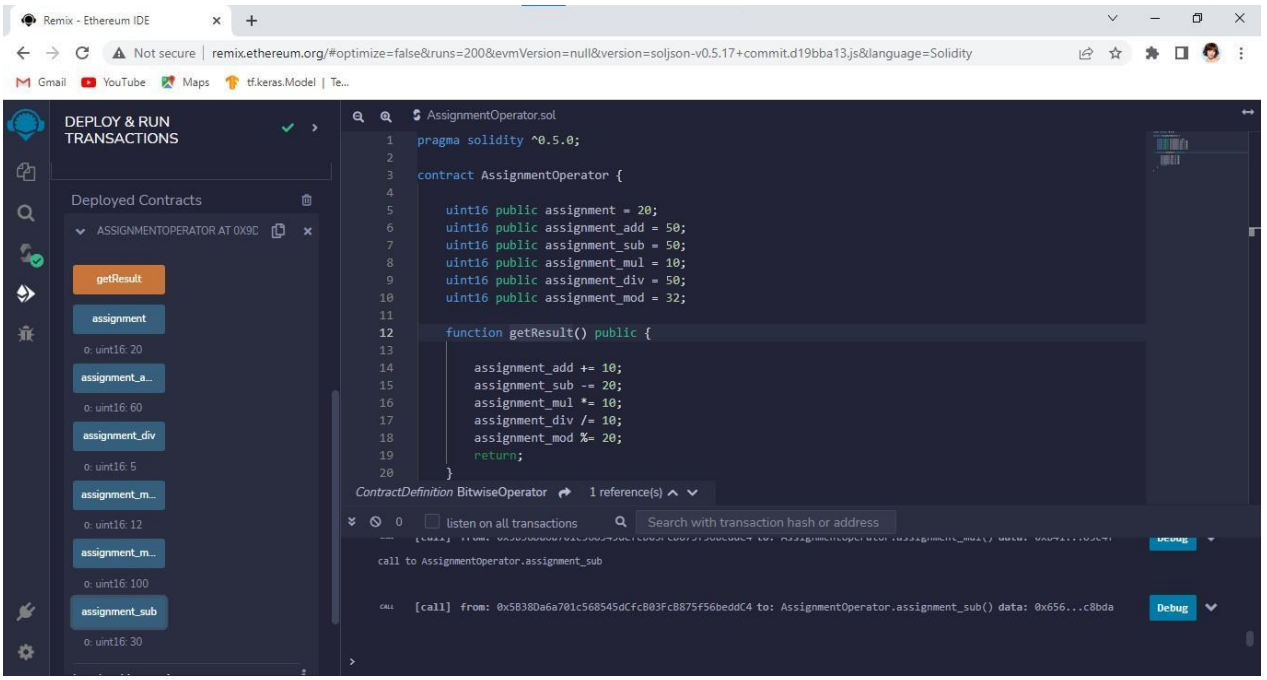




Assignment Operator

Code:

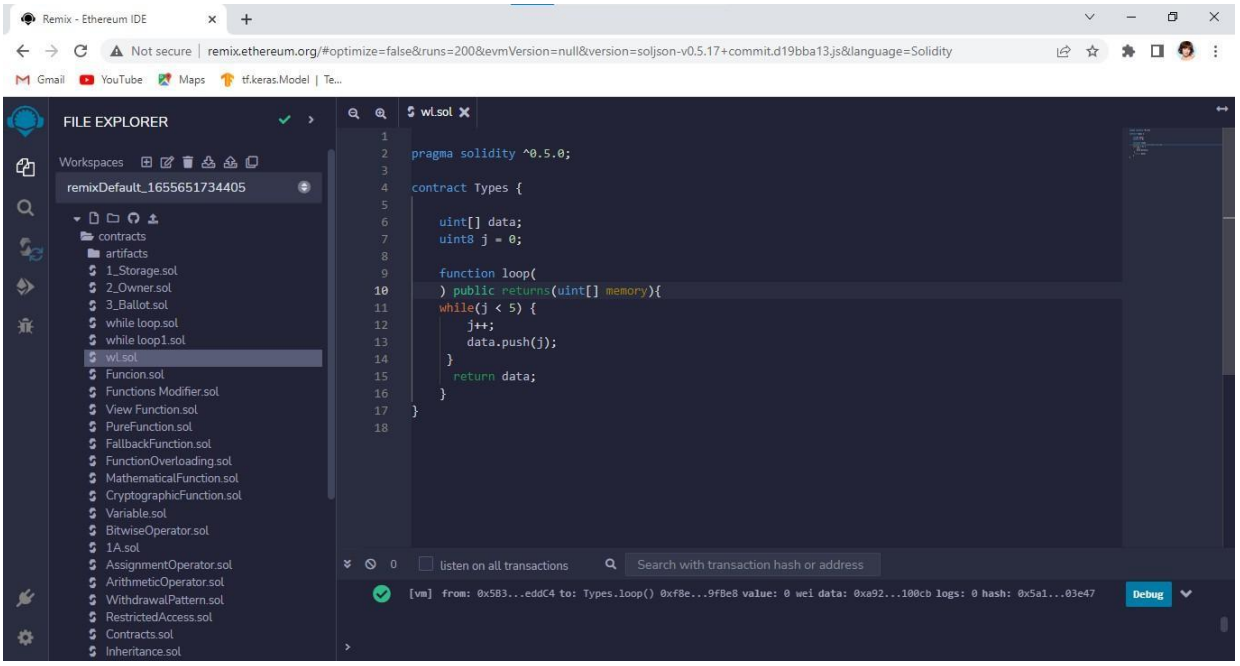


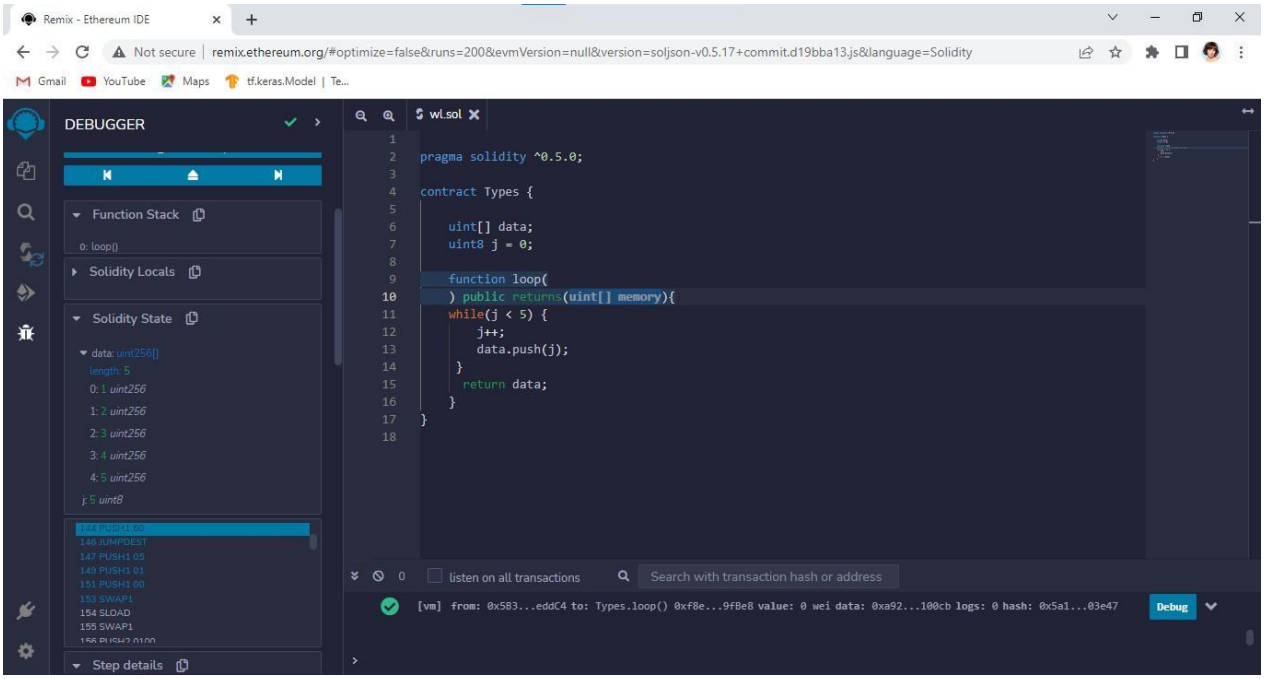


Loops

While Loop

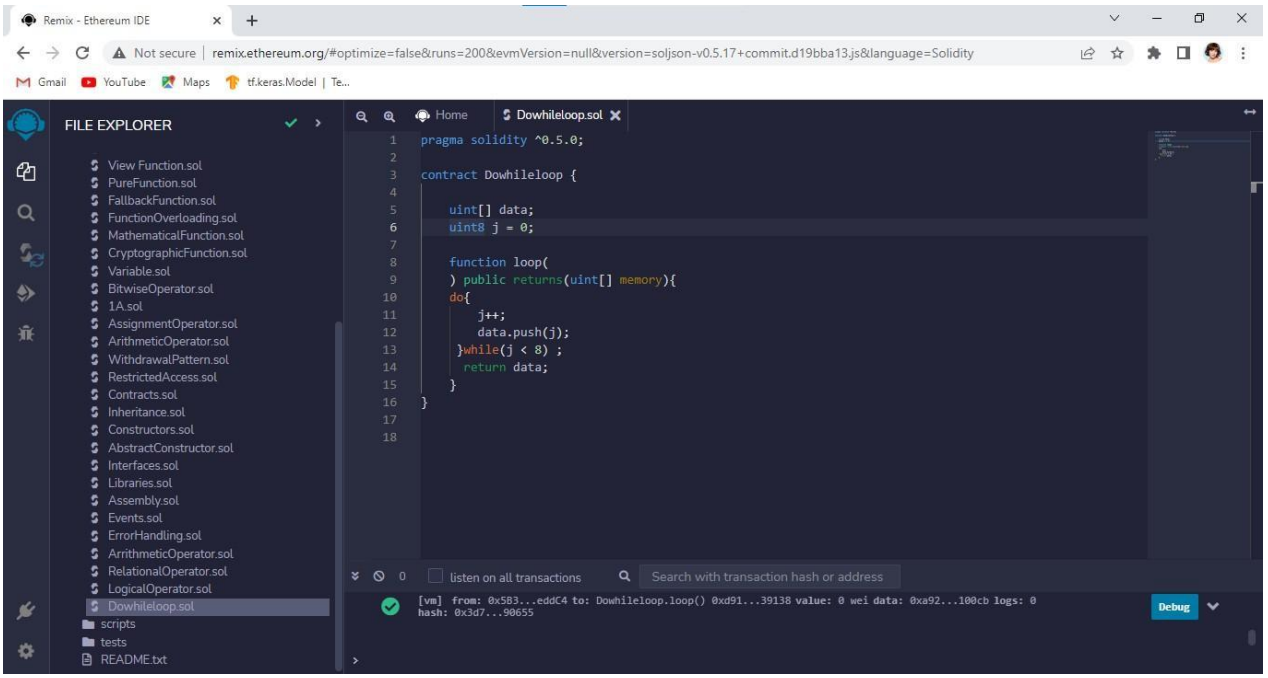
Code:



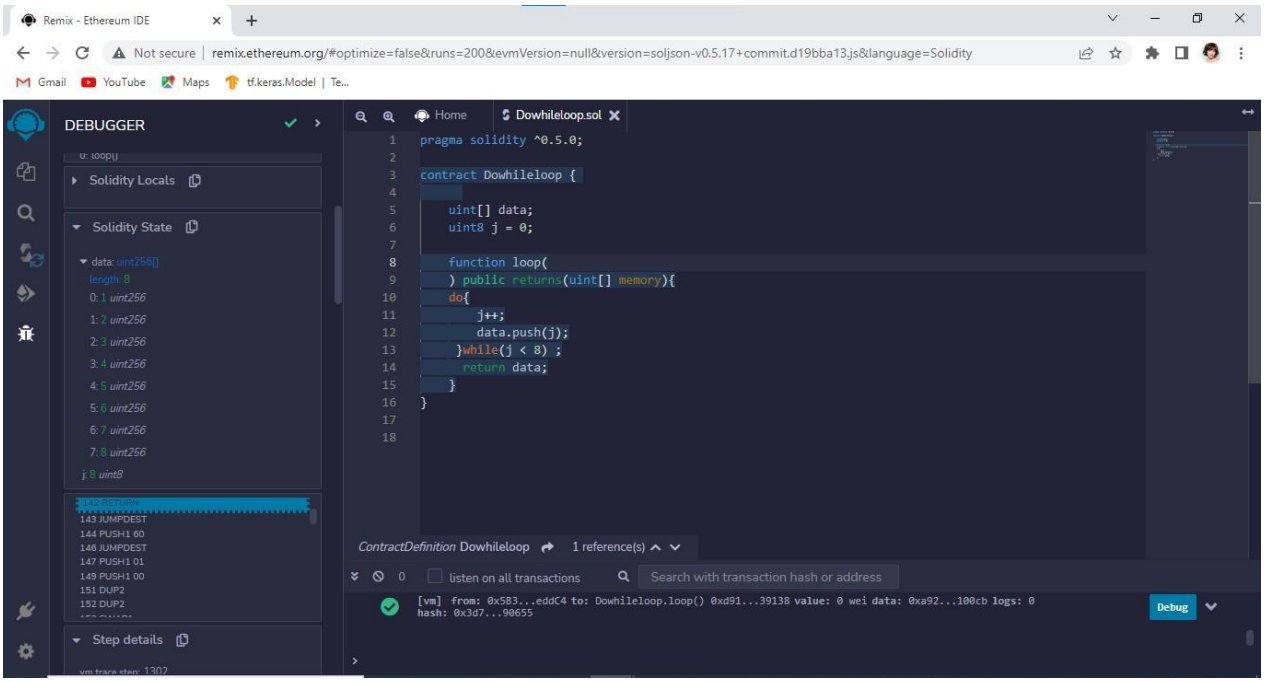


Dowhile loop

Code:

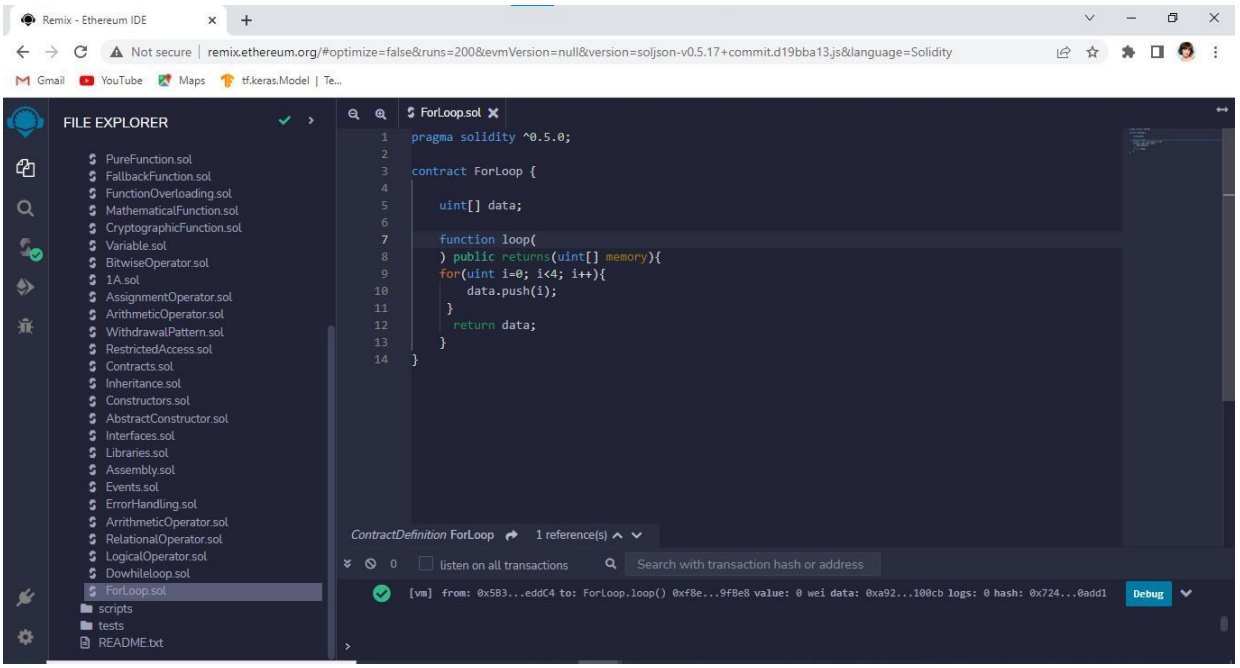


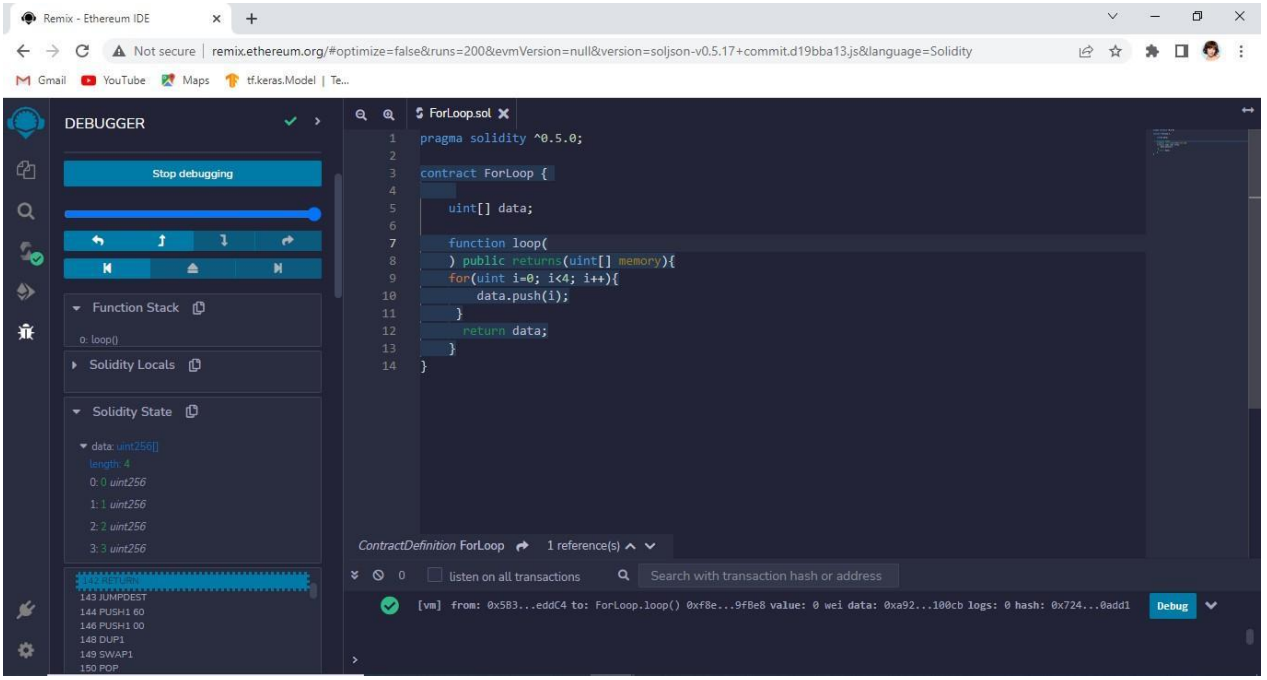
Output:



For loop

Code:

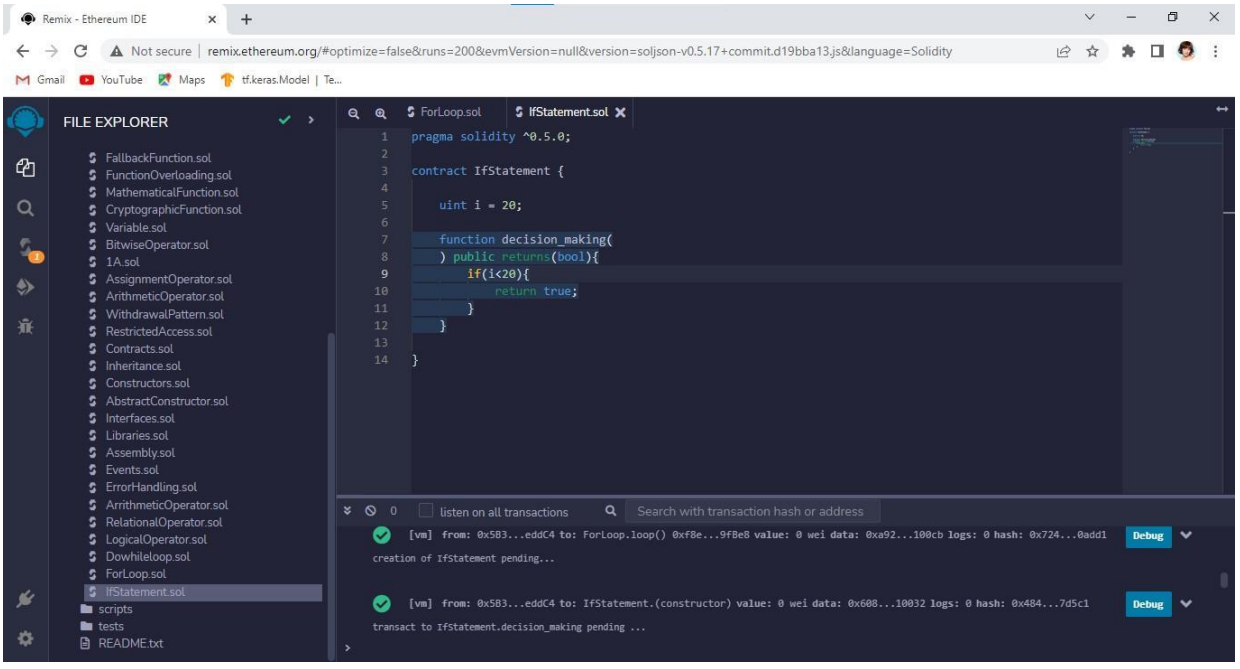


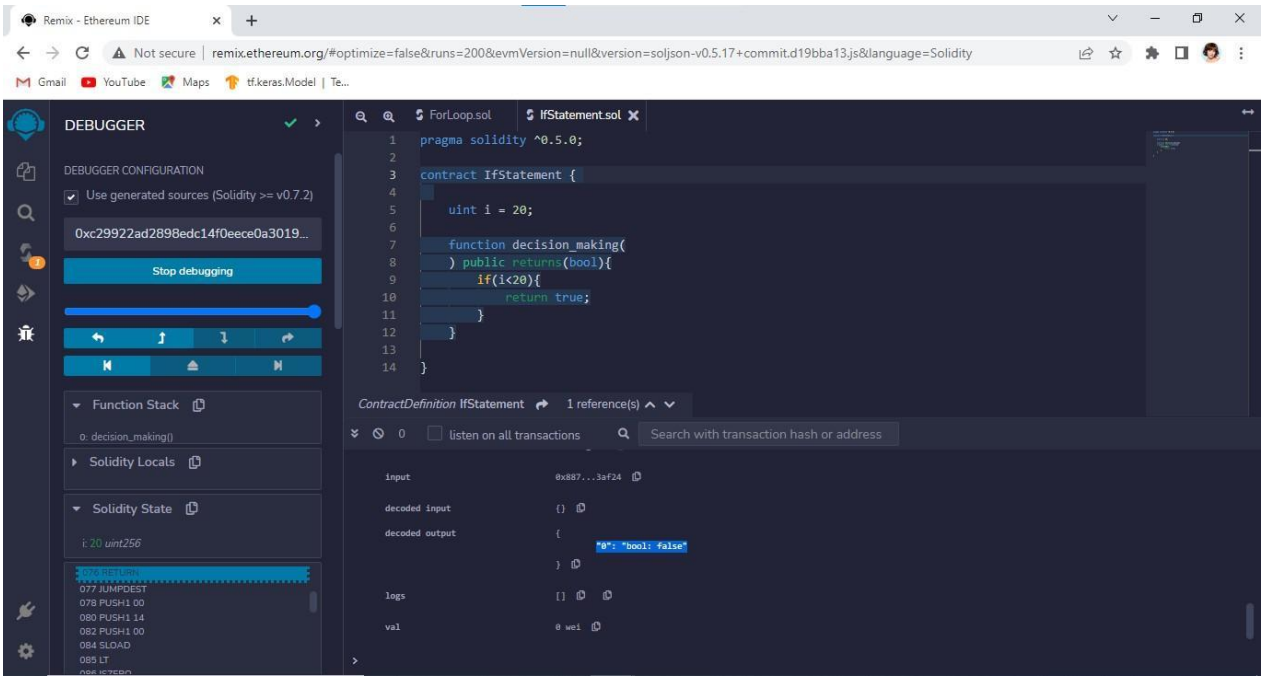


Decision making

If statement

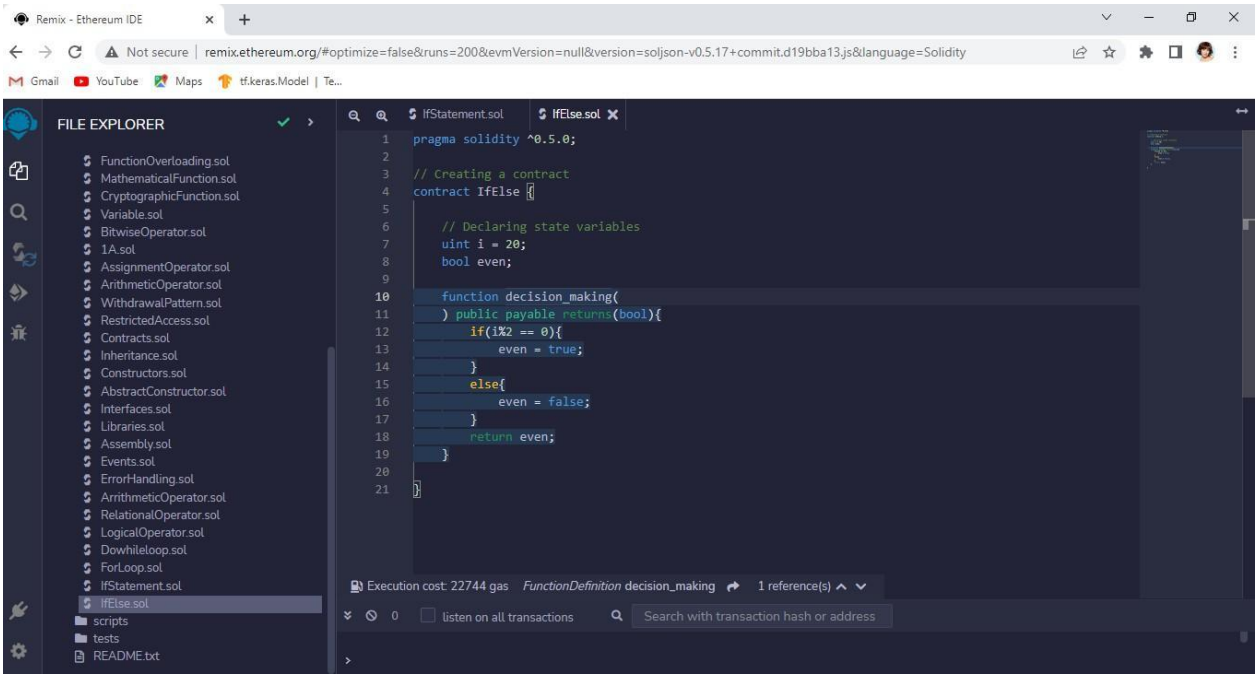
Code:



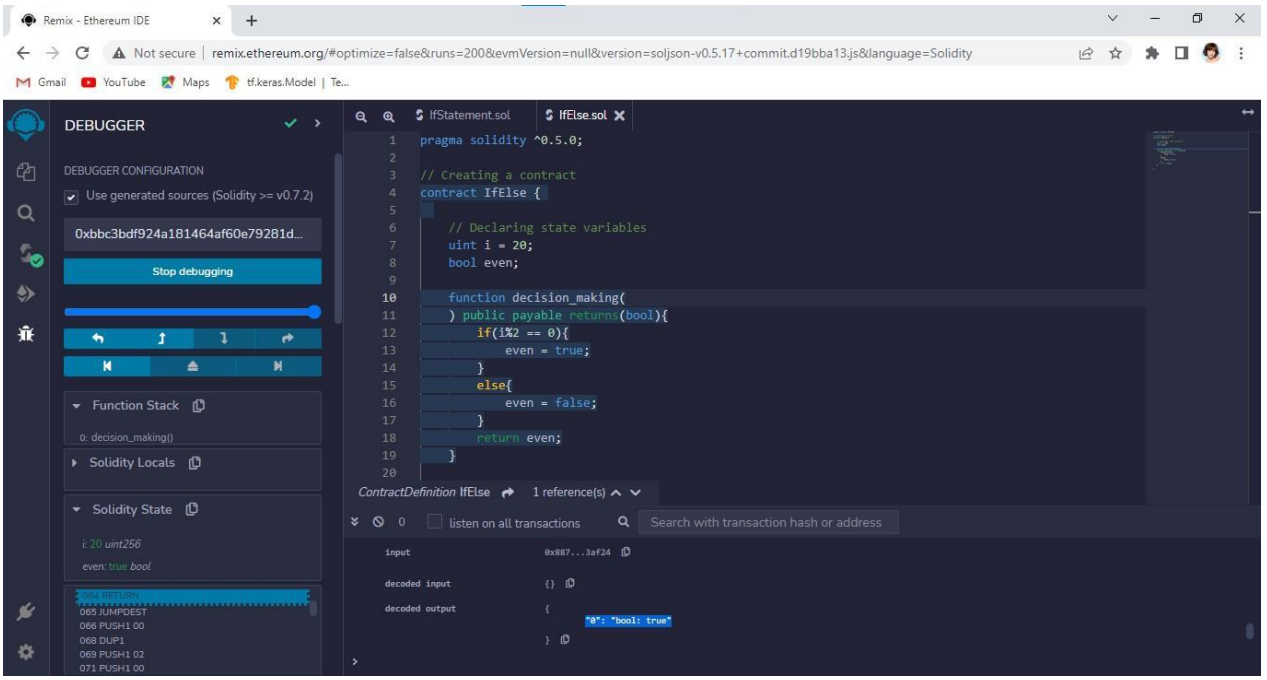


If…else

statement Code:

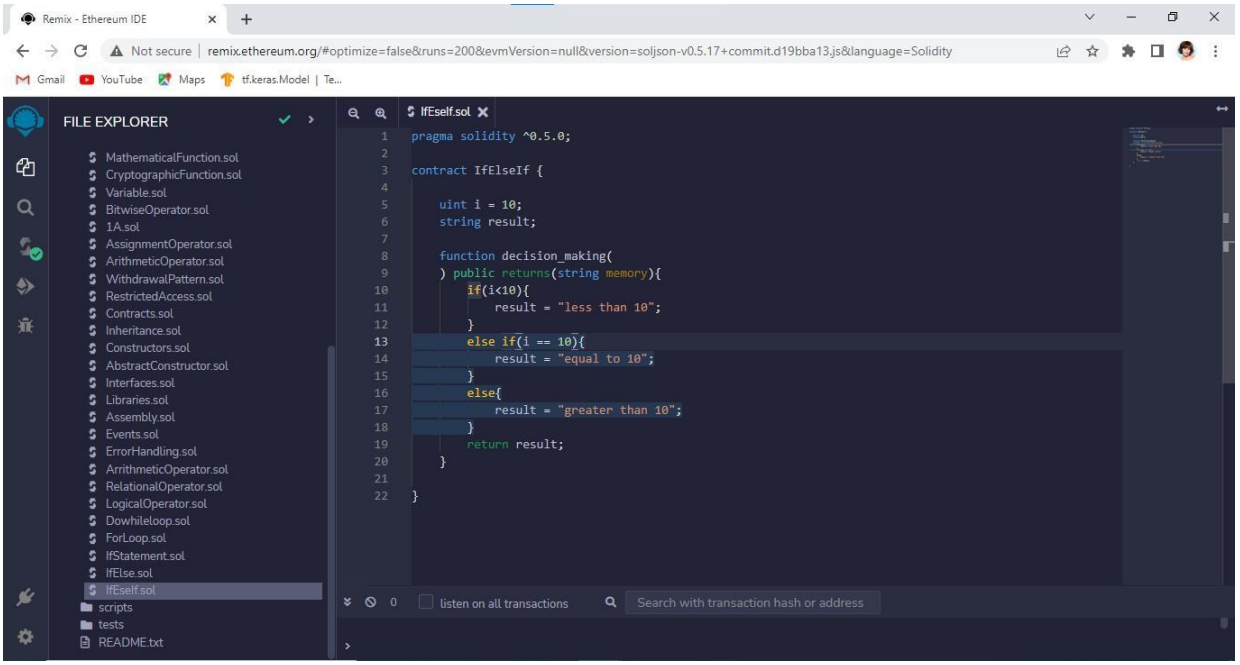


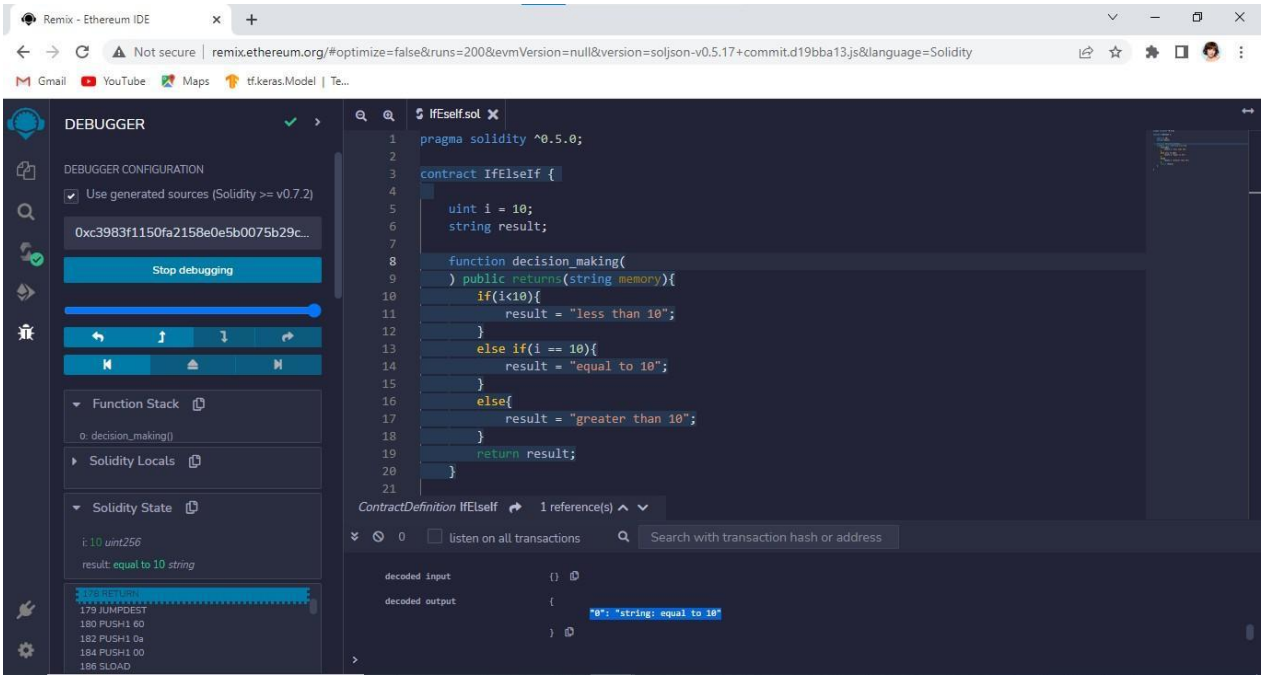
Output:



If…else if…else statement

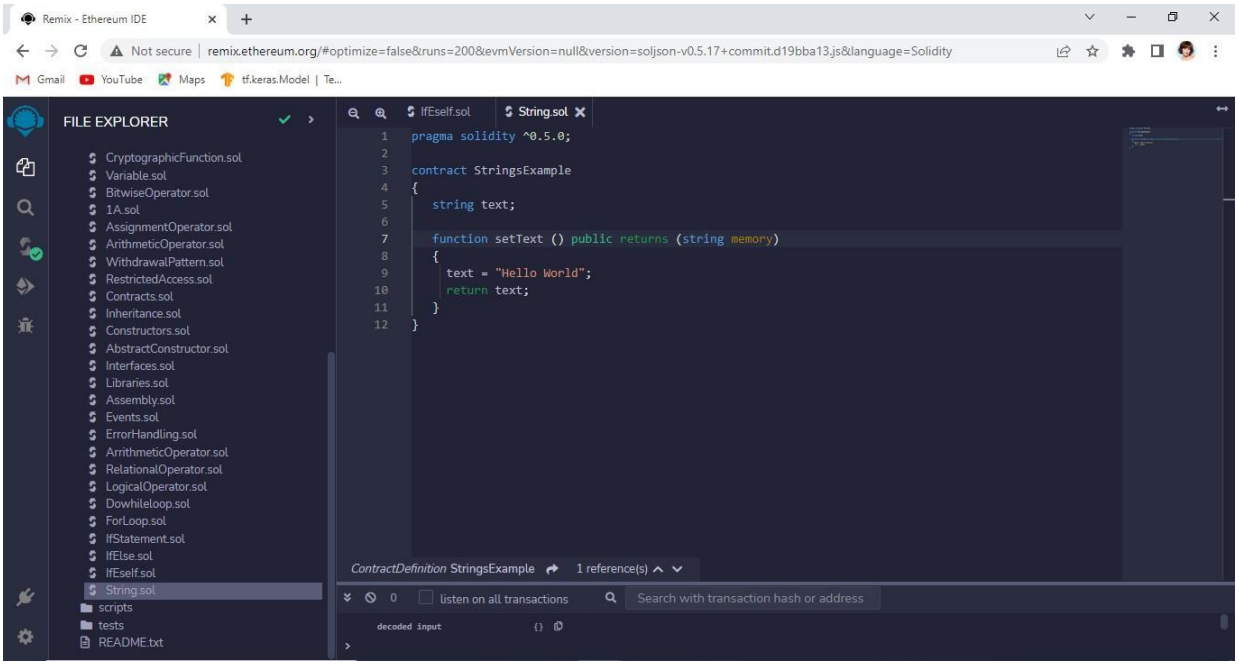
Code:

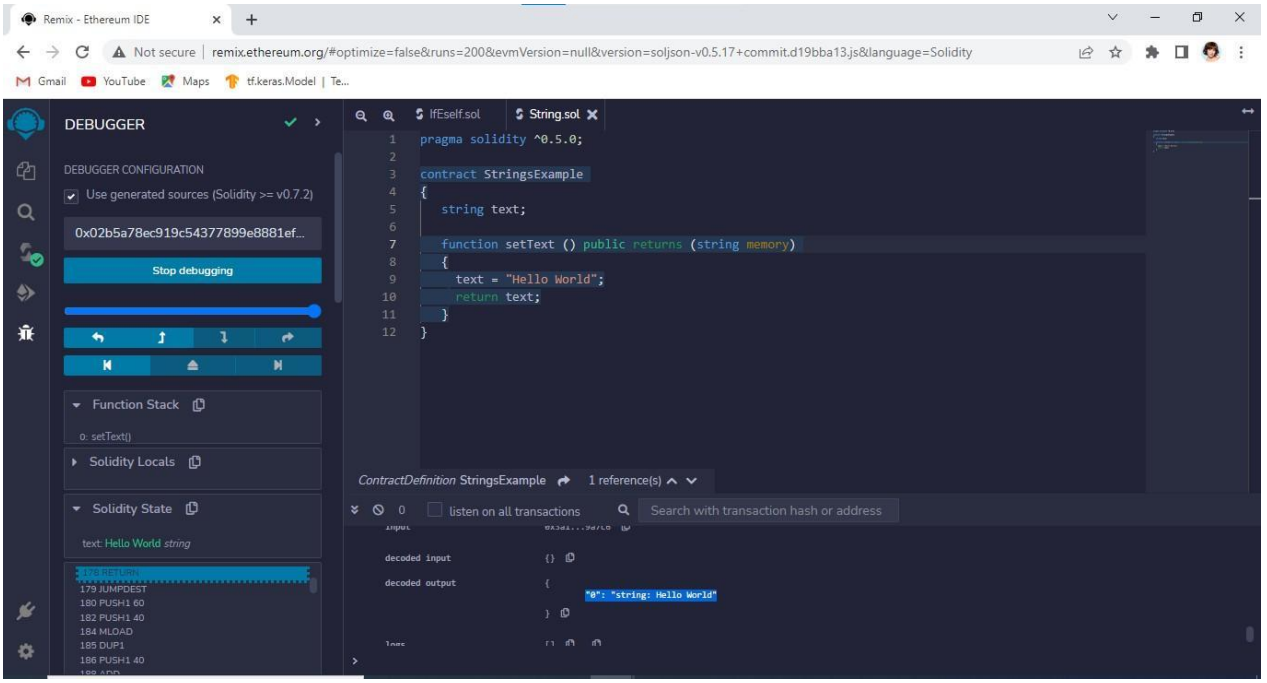



Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEBUGGER
DEBUGGER CONFIGURATION
Use generated sources (Solidity >= v0.7.2)
0xc3983f1150fa2158e0e5b0075b29c...
Stop debugging
Function Stack
0: decision_making()
Solidity Locals
Solidity State
i: 10 uint256
result: equal to 10 string
IfEself.sol
pragma solidity ^0.5.0;
contract IfElseIf {
uint i = 10;
string result;
function decision_making(
) public returns(string memory){
if(i<10){
result = "less than 10";
}
else if(i == 10){
result = "equal to 10";
}
else{
result = "greater than 10";
}
return result;
}
ContractDefinition IfElseIf
1 reference(s)
listen on all transactions
Search with transaction hash or address
decoded input
decoded output
"0": "string: equal to 10"

String

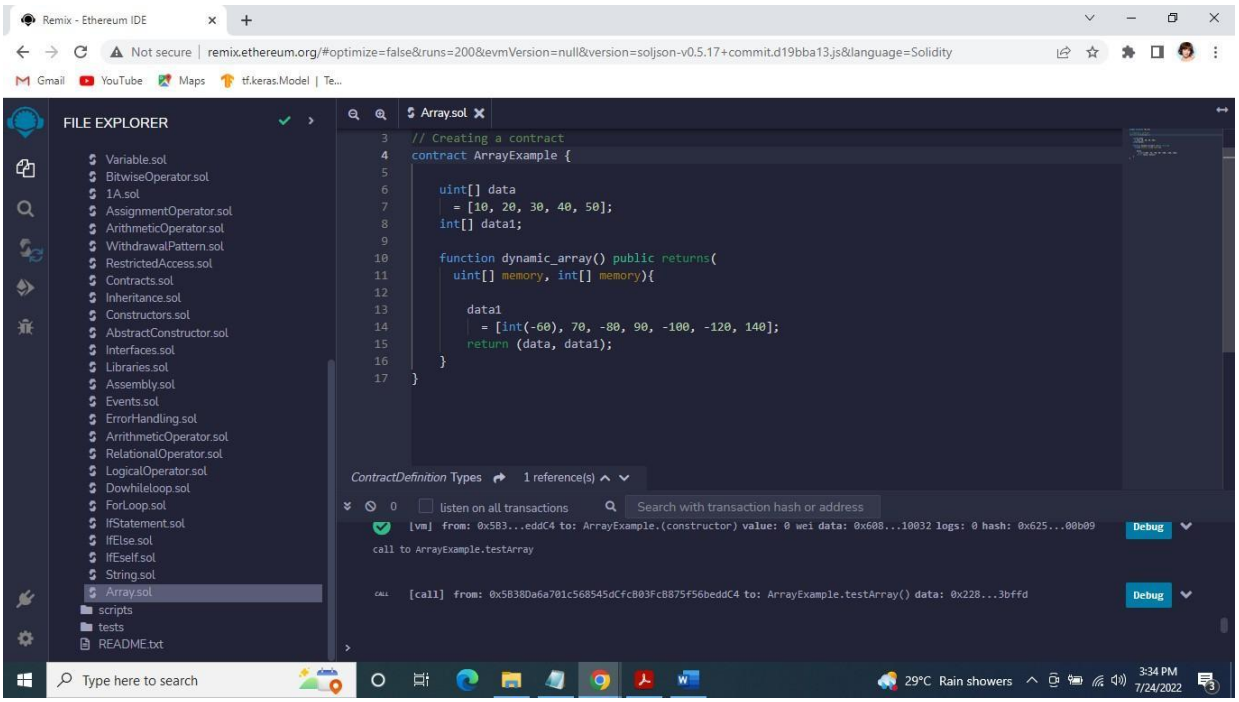
Code:

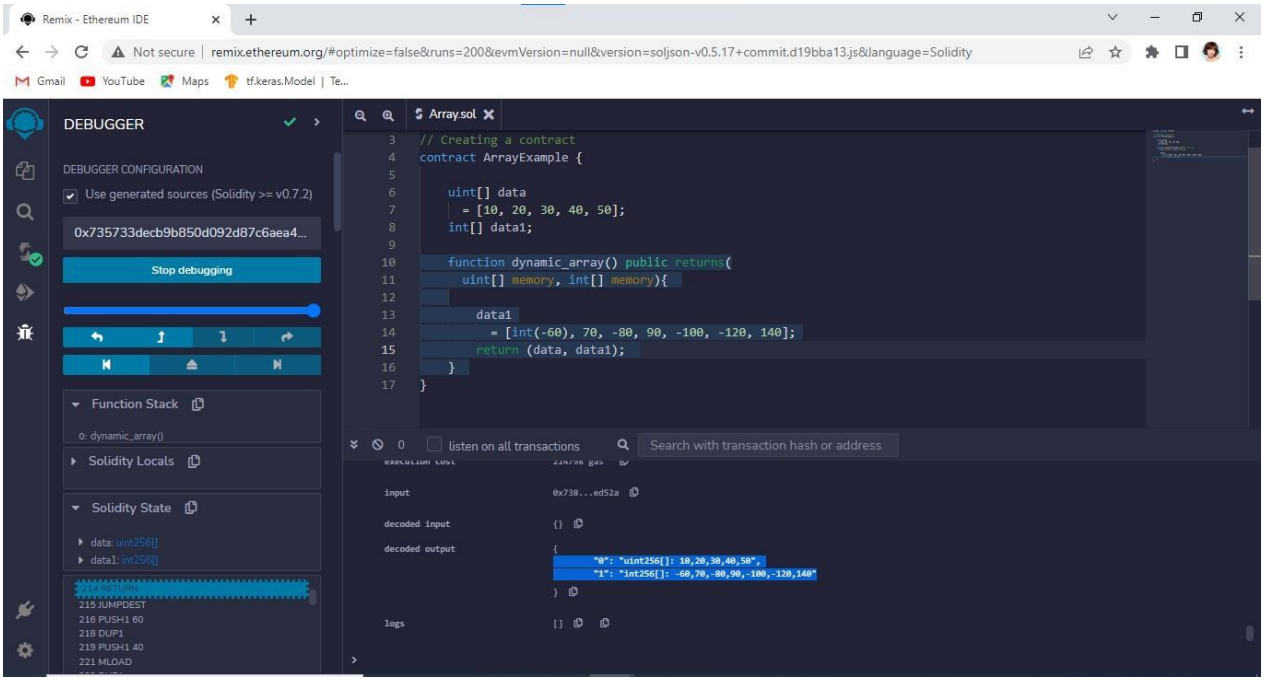

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
scripts
tests
README.txt
IfEself.sol
String.sol
pragma solidity ^0.5.0;
contract StringsExample
{
string text;
function setText () public returns (string memory)
{
text = "Hello World";
return text;
}
}
ContractDefinition StringsExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
decoded input


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEBUGGER
DEBUGGER CONFIGURATION
Use generated sources (Solidity >= v0.7.2)
0x02b5a78ec919c54377899e8881ef...
Stop debugging
Function Stack
0: setText()
Solidity Locals
Solidity State
text: Hello World string
178 RETURN
179 JUMPDEST
180 PUSH1 60
182 PUSH1 40
184 MLOAD
185 DUP1
186 PUSH1 40
IfEself.sol
String.sol
1 pragma solidity ^0.5.0;
2
3 contract StringsExample
4 {
5 string text;
6
7 function setText () public returns (string memory)
8 {
9 text = "Hello World";
10 return text;
11 }
12 }
ContractDefinition StringsExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
decoded input
decoded output
"0": "string: Hello World"

Array

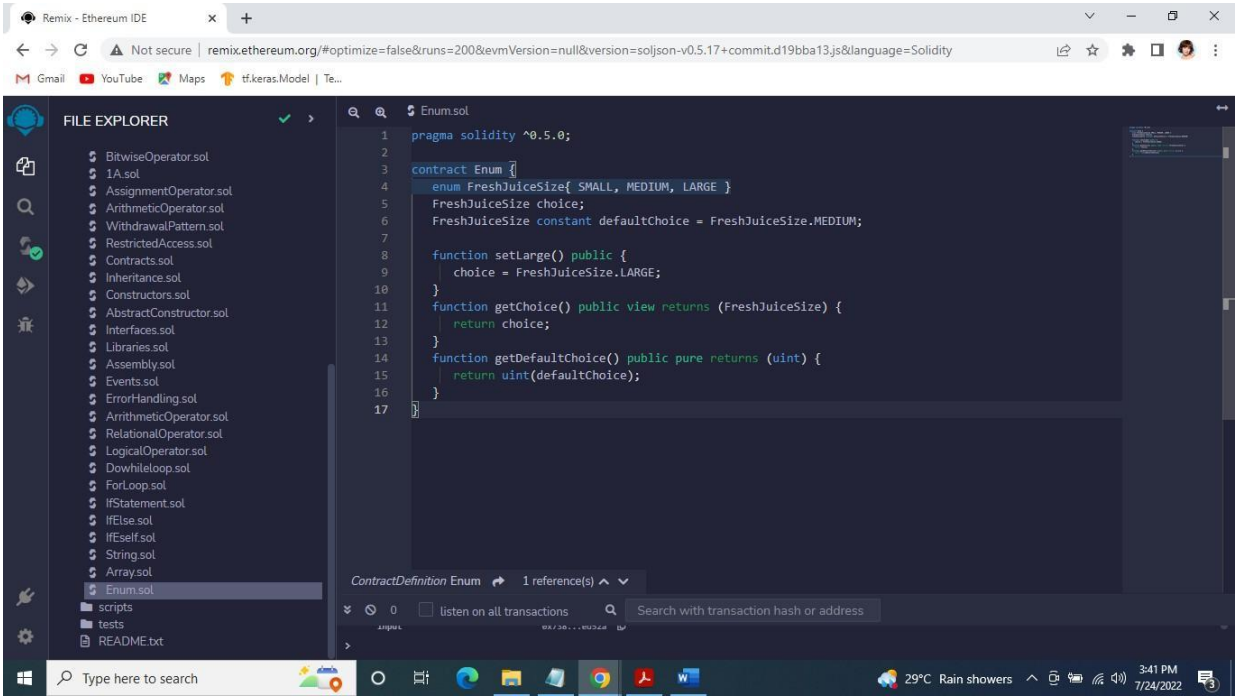
Code:

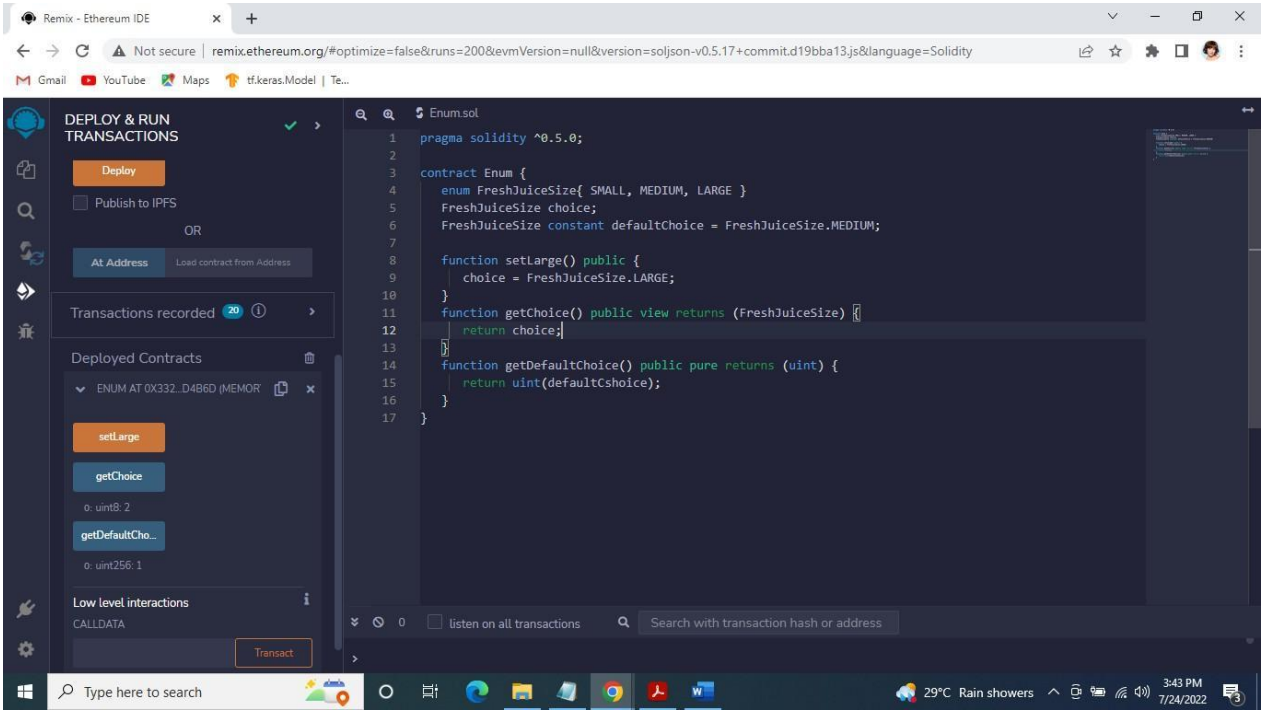

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
scripts
tests
README.txt
Array.sol
3 // Creating a contract
4 contract ArrayExample {
5
6 uint[] data
7 = [10, 20, 30, 40, 50];
8 int[] data1;
9
10 function dynamic_array() public returns(
11 uint[] memory, int[] memory){
12
13 data1
14 = [int(-60), 70, -80, 90, -100, -120, 140];
15 return (data, data1);
16 }
17 }
ContractDefinition Types
1 reference(s)
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: ArrayExample.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0x625...00b09
Debug
call to ArrayExample.testArray
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ArrayExample.testArray() data: 0x228...3bffd
Debug
Type here to search
29°C Rain showers
3:34 PM
7/24/2022


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEBUGGER
DEBUGGER CONFIGURATION
Use generated sources (Solidity >= v0.7.2)
0x735733decb9b850d092d87c6aea4...
Stop debugging
Function Stack
0: dynamic_array()
Solidity Locals
Solidity State
data: uint256[]
data1: int256[]
214 RETURN
215 JUMPDEST
216 PUSH1 60
218 DUP1
219 PUSH1 40
221 MLOAD
Array.sol
// Creating a contract
contract ArrayExample {
uint[] data
= [10, 20, 30, 40, 50];
int[] data1;
function dynamic_array() public returns(
uint[] memory, int[] memory){
data1
= [int(-60), 70, -80, 90, -100, -120, 140];
return (data, data1);
}
}
listen on all transactions
Search with transaction hash or address
input
0x738...ed52a
decoded input
{}
decoded output
{
"0": "uint256[]: 10,20,30,40,50",
"1": "int256[]: -60,70,-80,90,-100,-120,140"
}
logs
[]

Enums

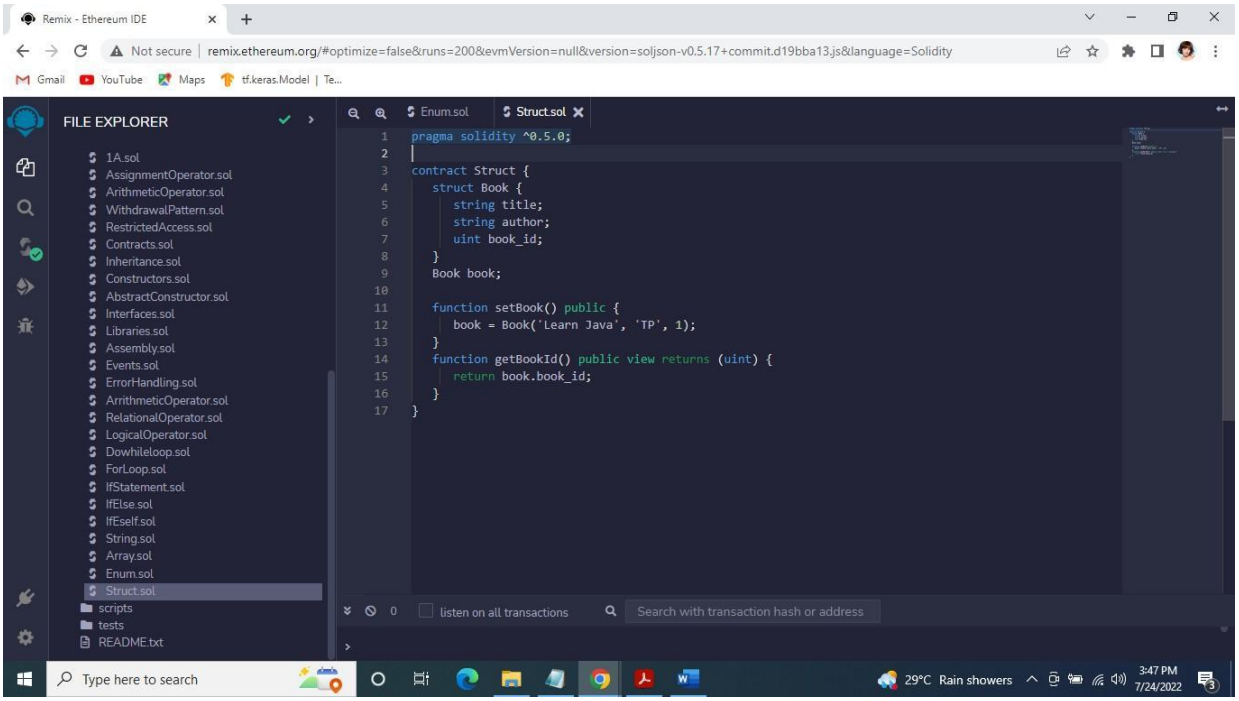
Code:

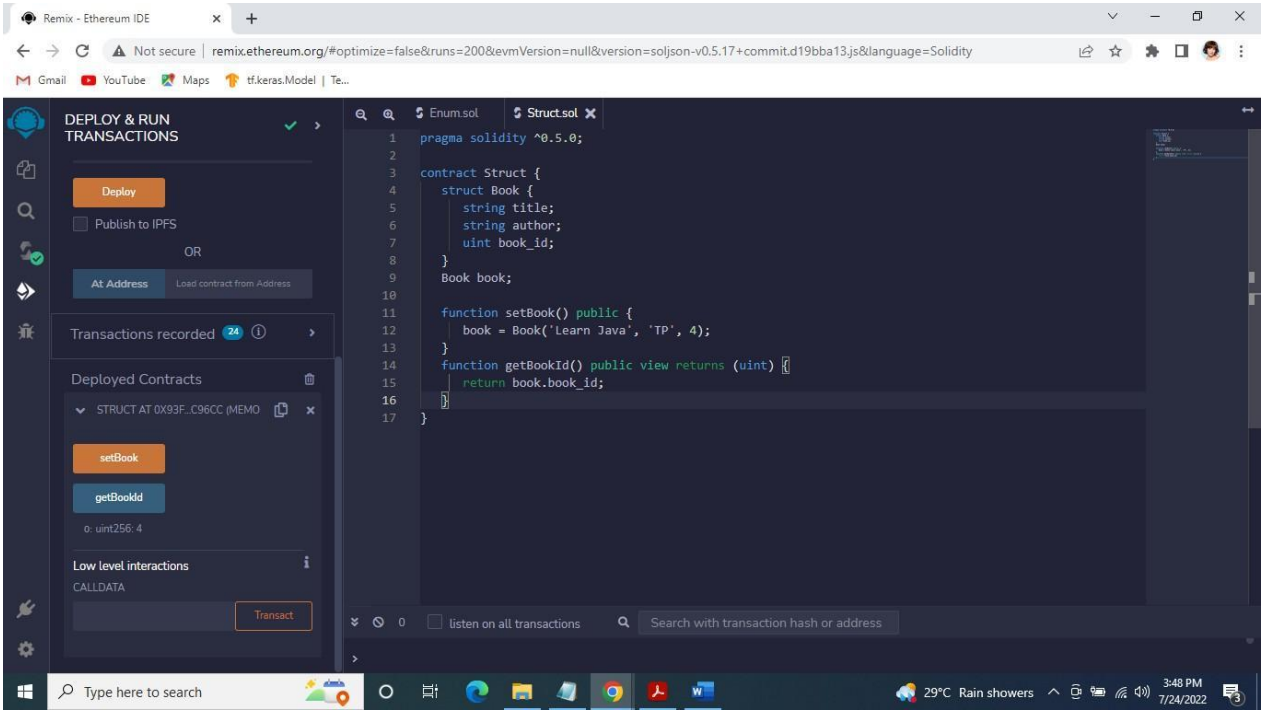

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
Enum.sol
scripts
tests
README.txt
Enum.sol
pragma solidity ^0.5.0;
contract Enum {
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);
}
}
ContractDefinition Enum
1 reference(s)
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:41 PM
7/24/2022


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 20
Deployed Contracts
ENUM AT 0X332...D4B6D (MEMOR
setLarge
getChoice
0: uint8: 2
getDefaultCho...
0: uint256: 1
Low level interactions
CALLDATA
Transact
Enum.sol
pragma solidity ^0.5.0;
contract Enum {
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(defaultCshoice);
}
}
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:43 PM
7/24/2022

Struct

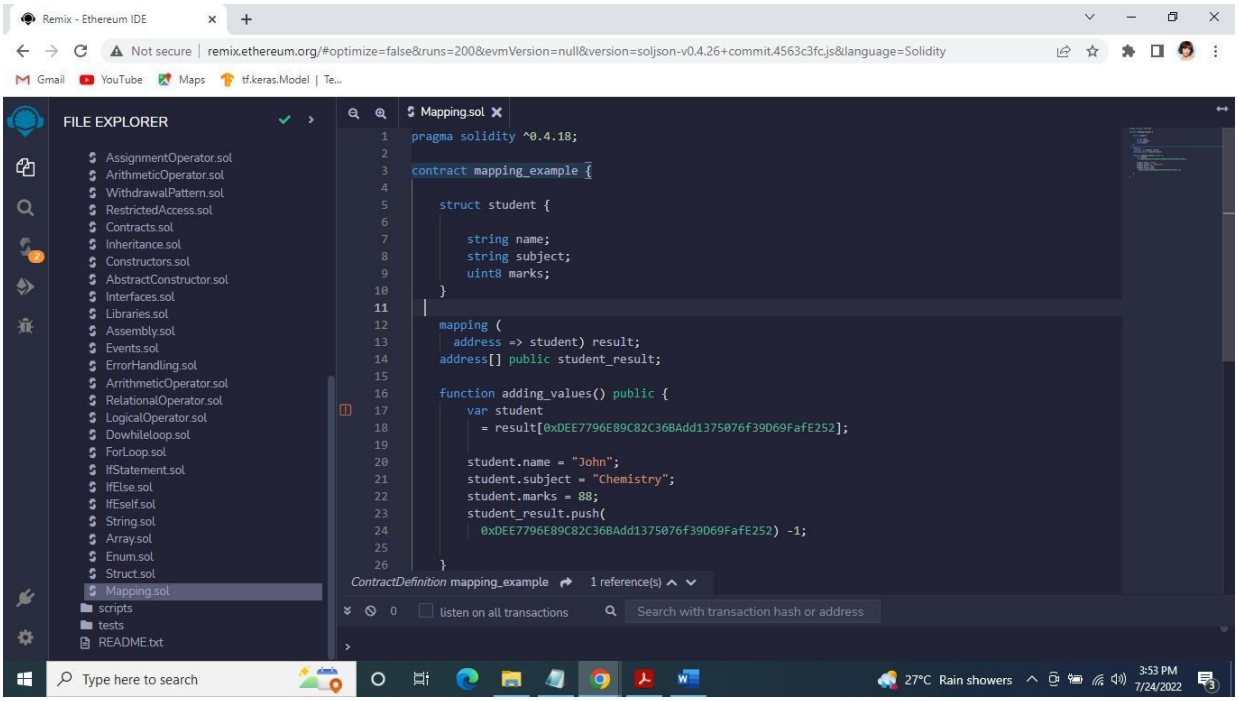
Code:

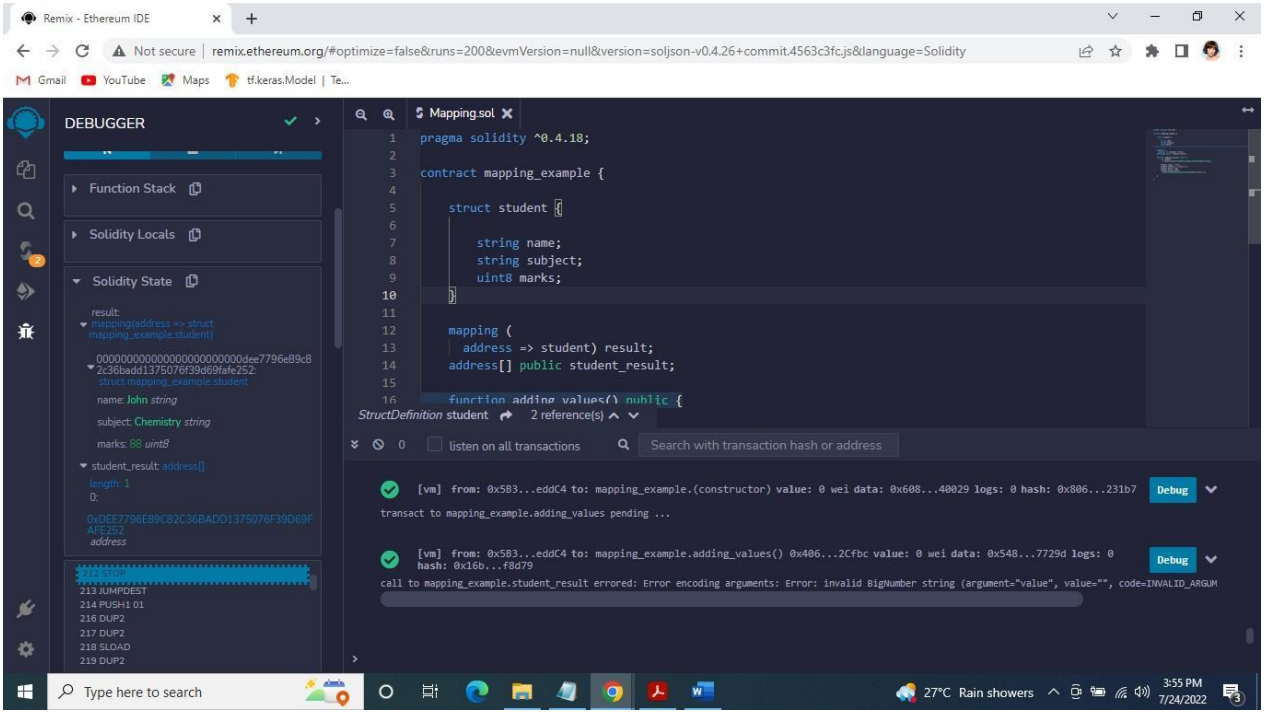

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
Enum.sol
Struct.sol
scripts
tests
README.txt
Enum.sol
Struct.sol
pragma solidity ^0.5.0;
contract Struct {
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;
}
}
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:47 PM
7/24/2022



Mapping

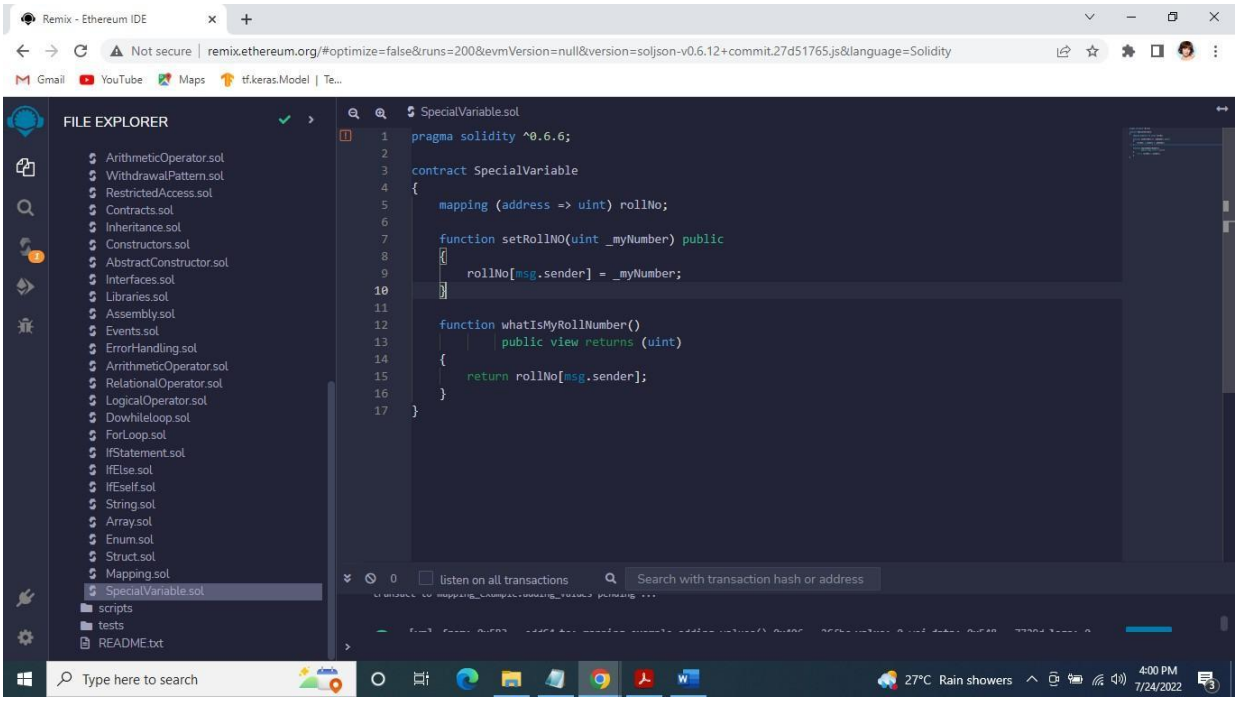
Code:

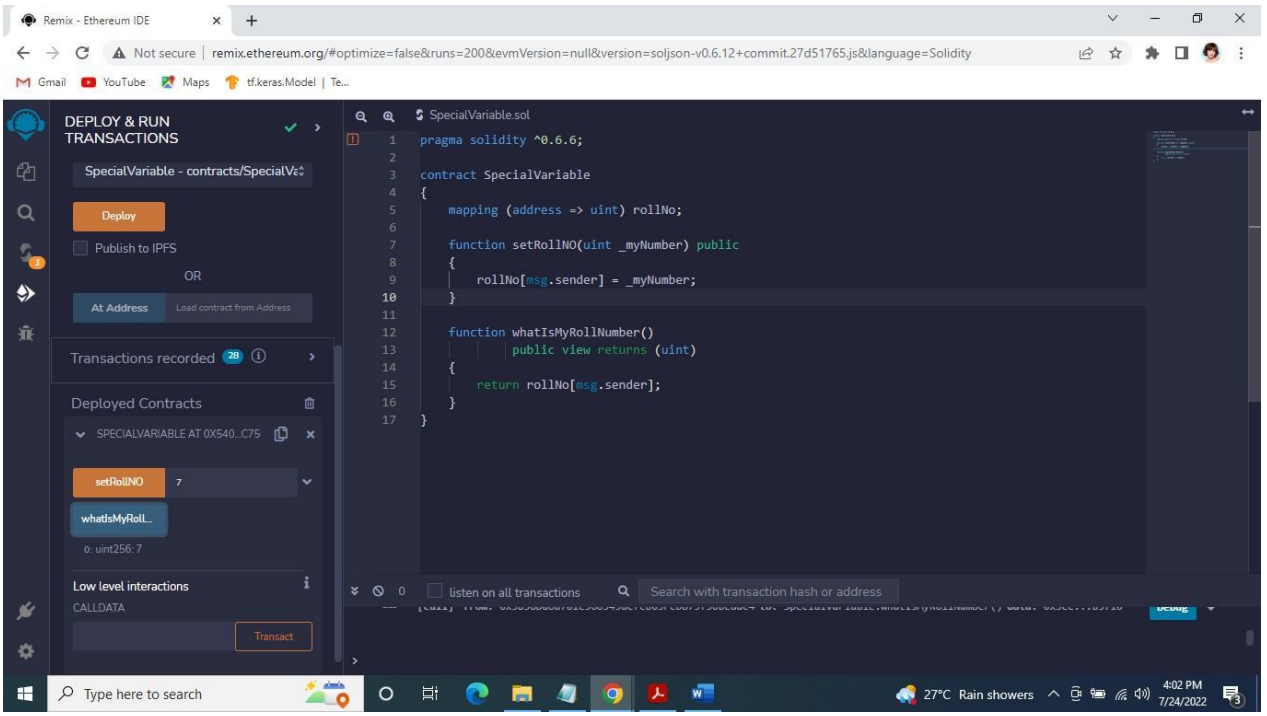




Special Variable

Code:

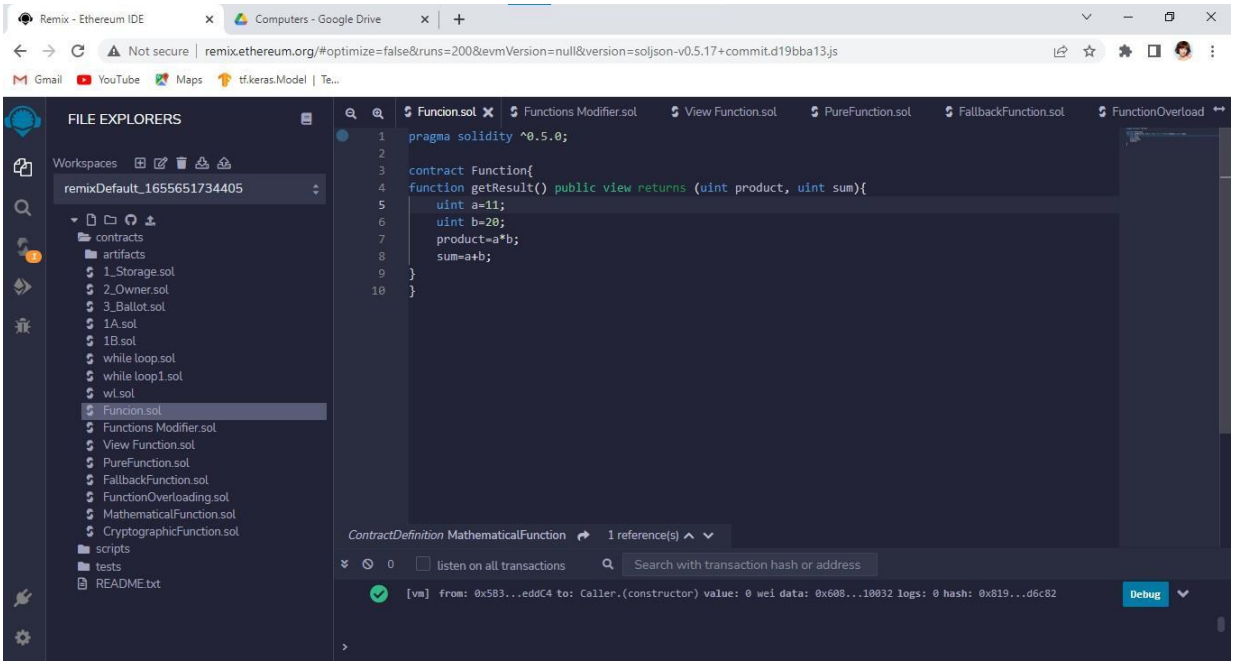


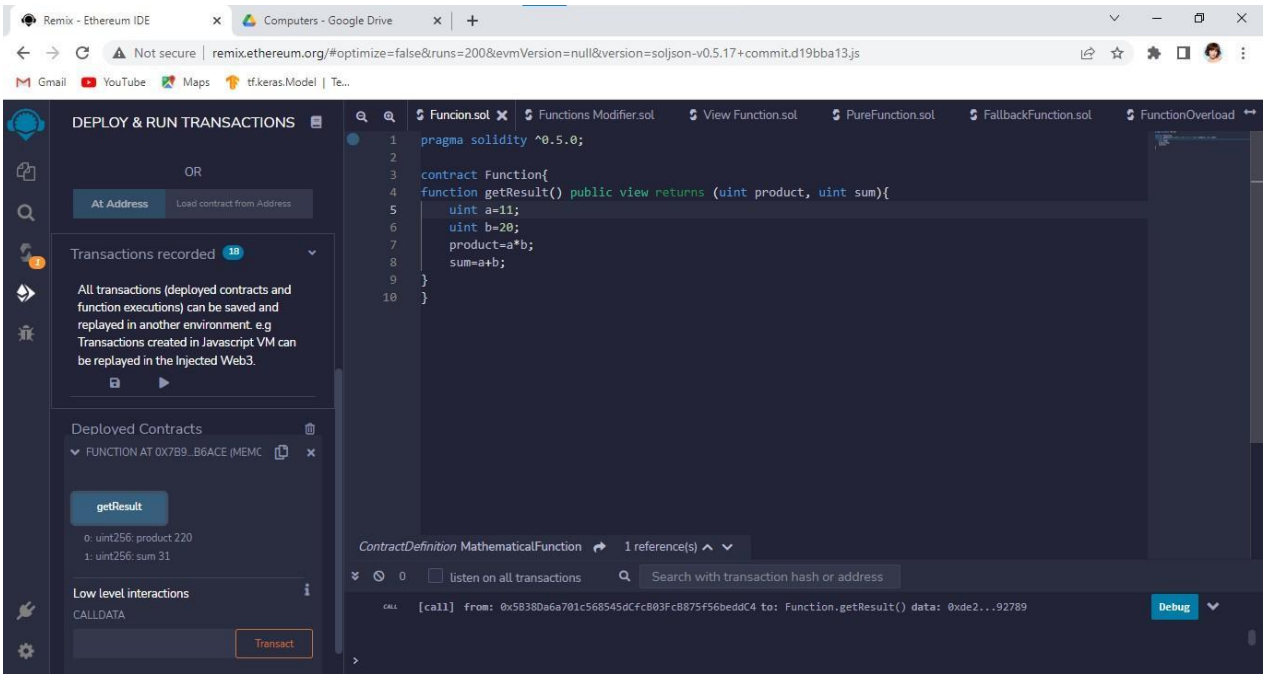


2b) Functions, Function Modifiers, View functions, Pure Functions, Fallback Function, Function Overloading, Mathematical functions, Cryptographic functions.

Function:

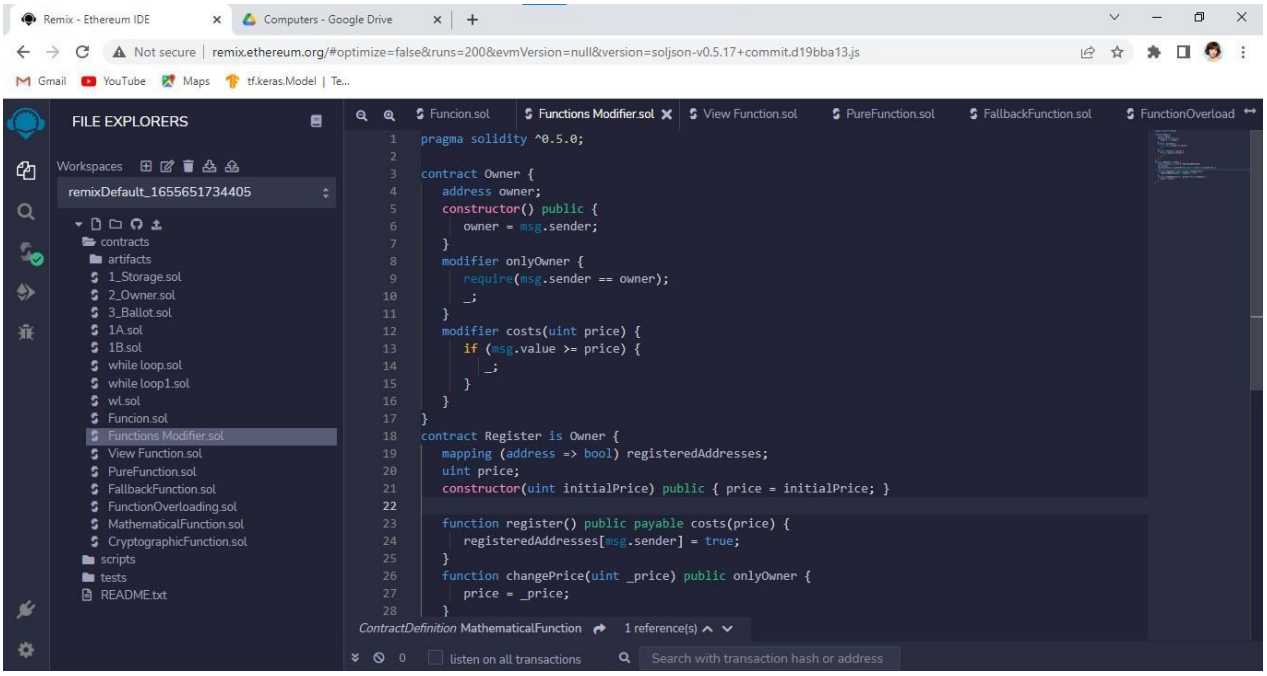
Code:



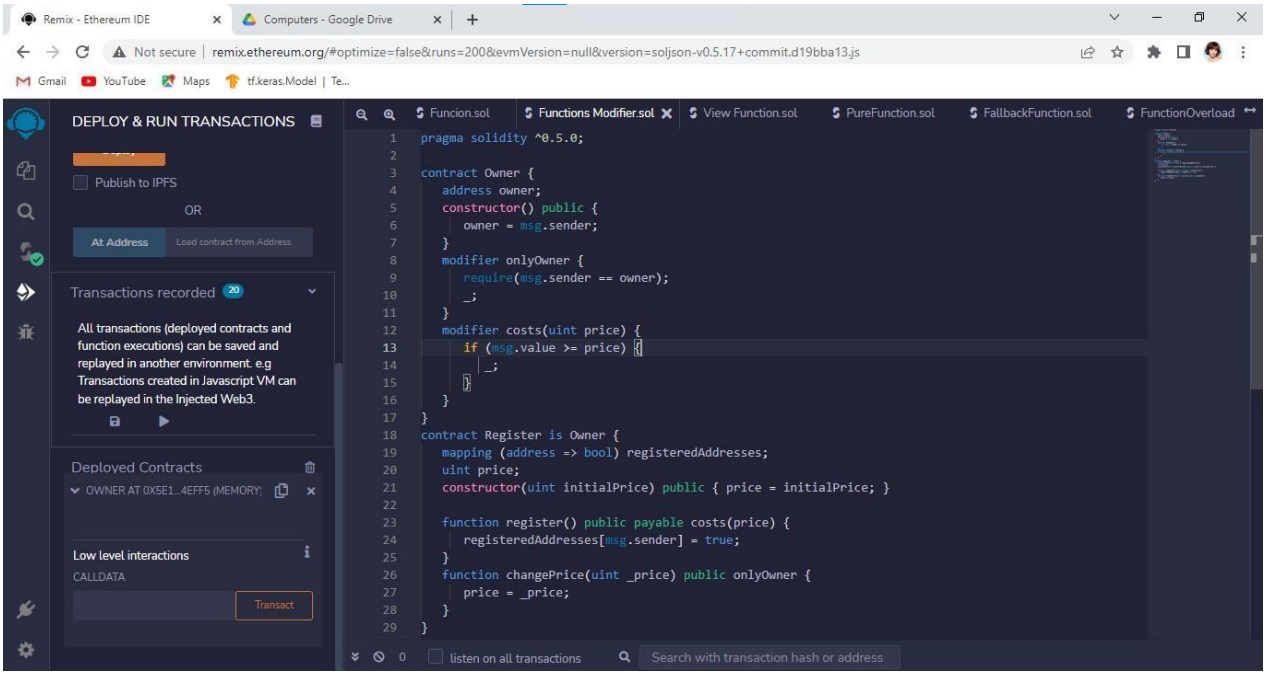


Functions Modifiers

Code:

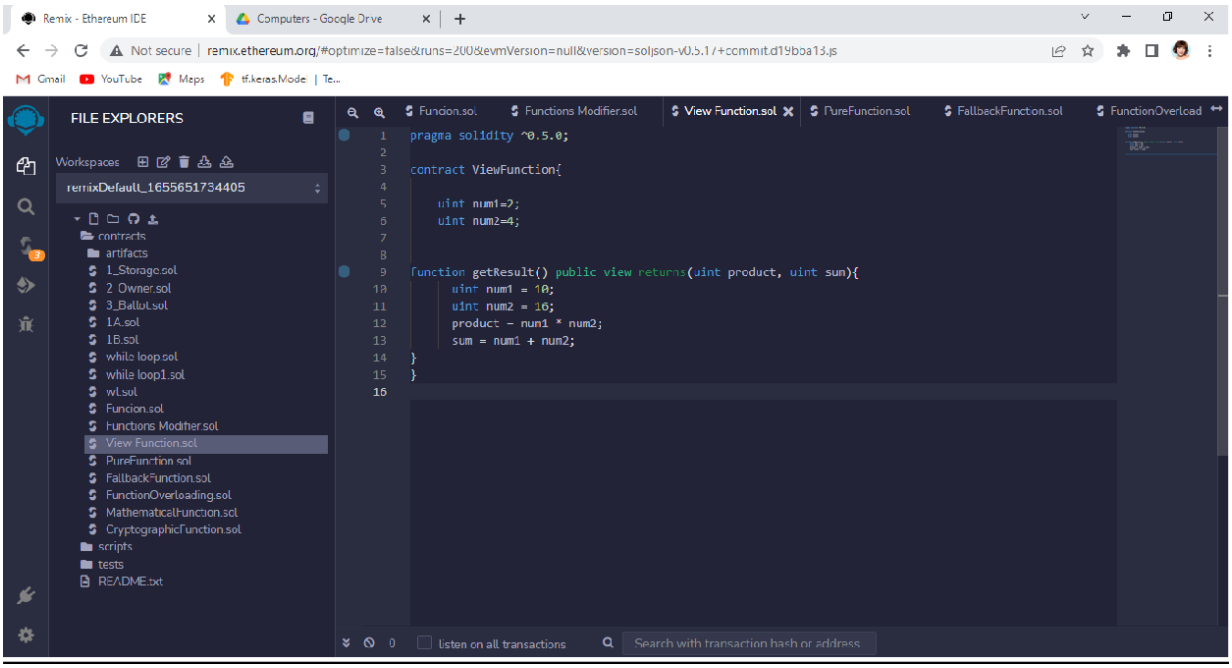


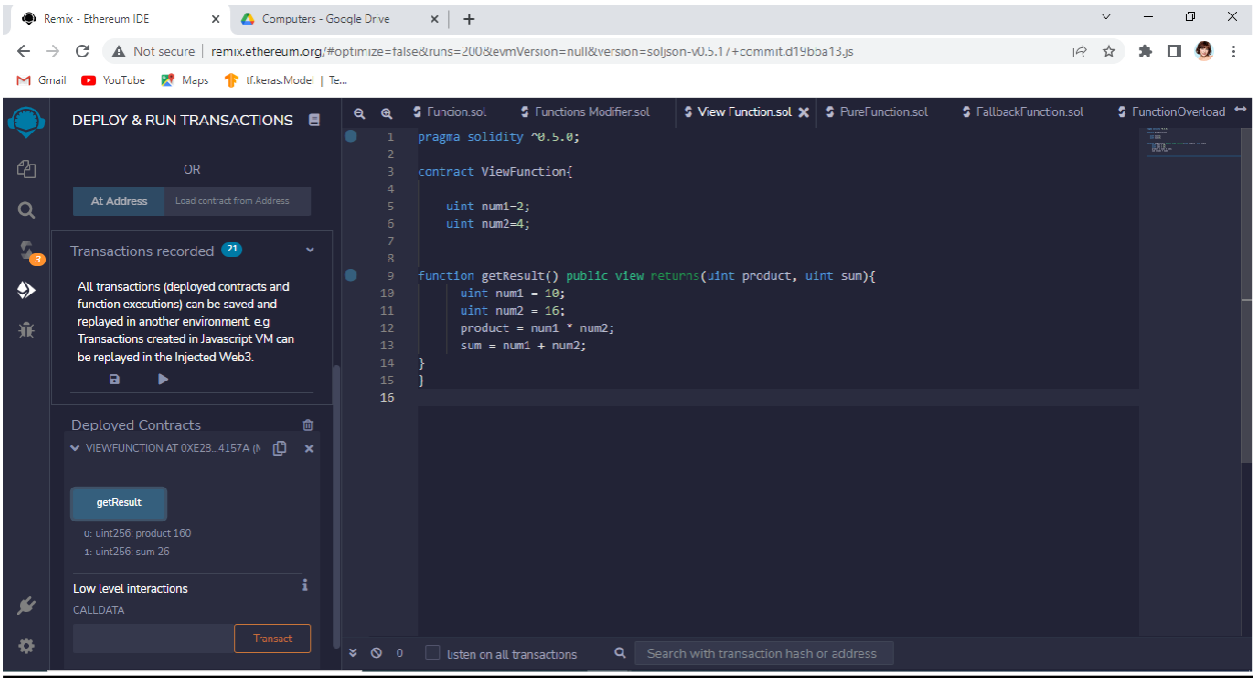
Output:



View function

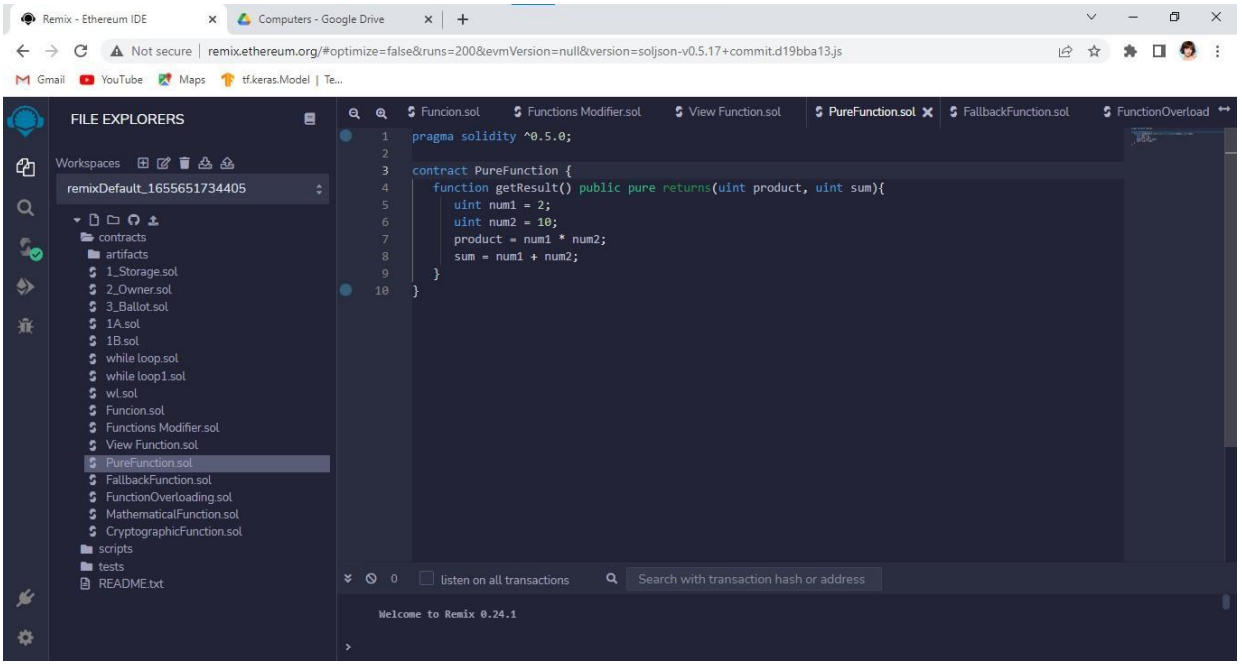
Code:

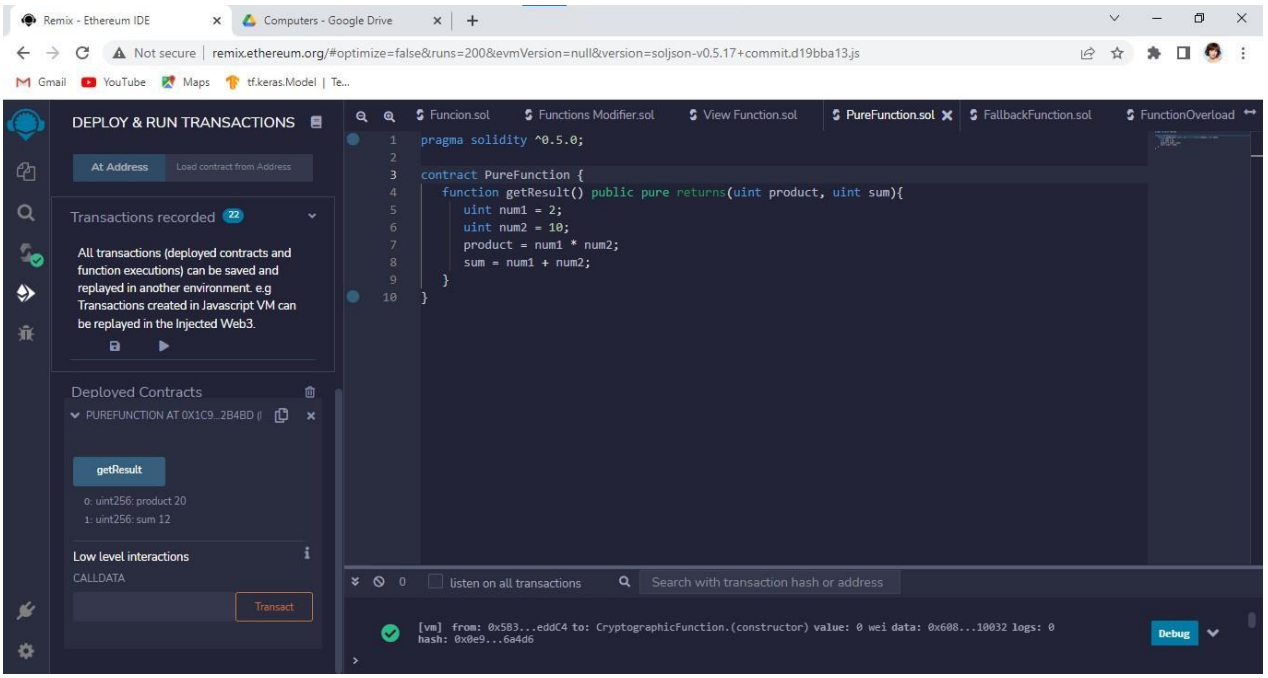




Pure function

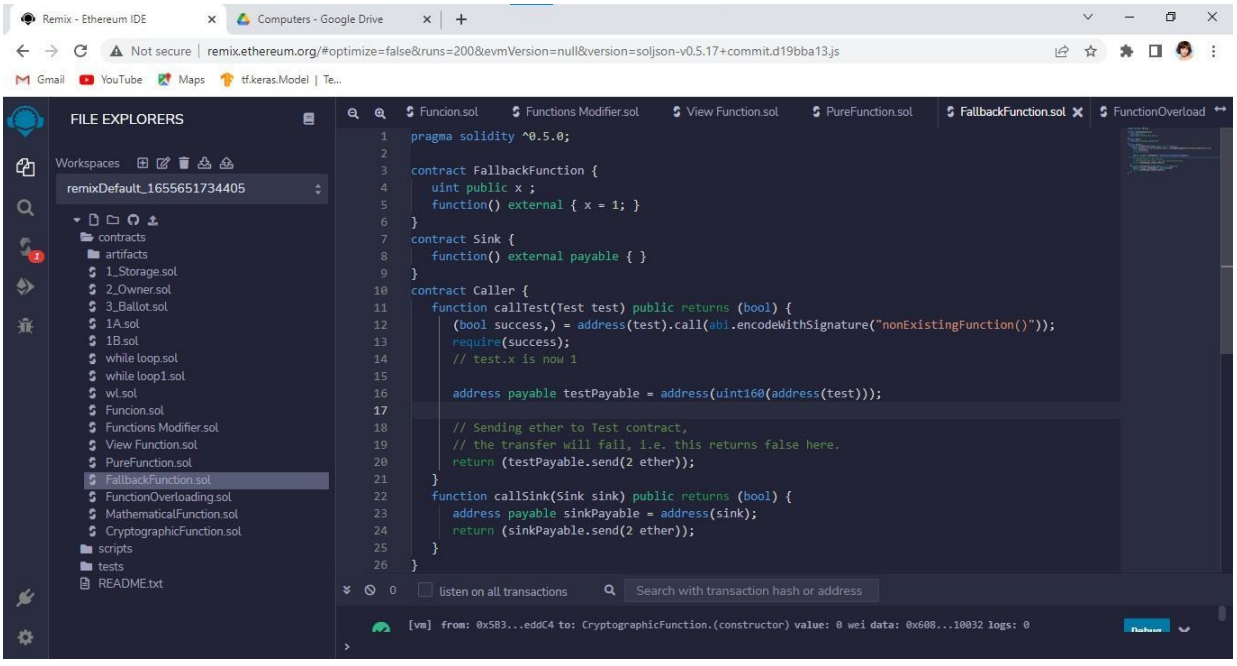
Code:

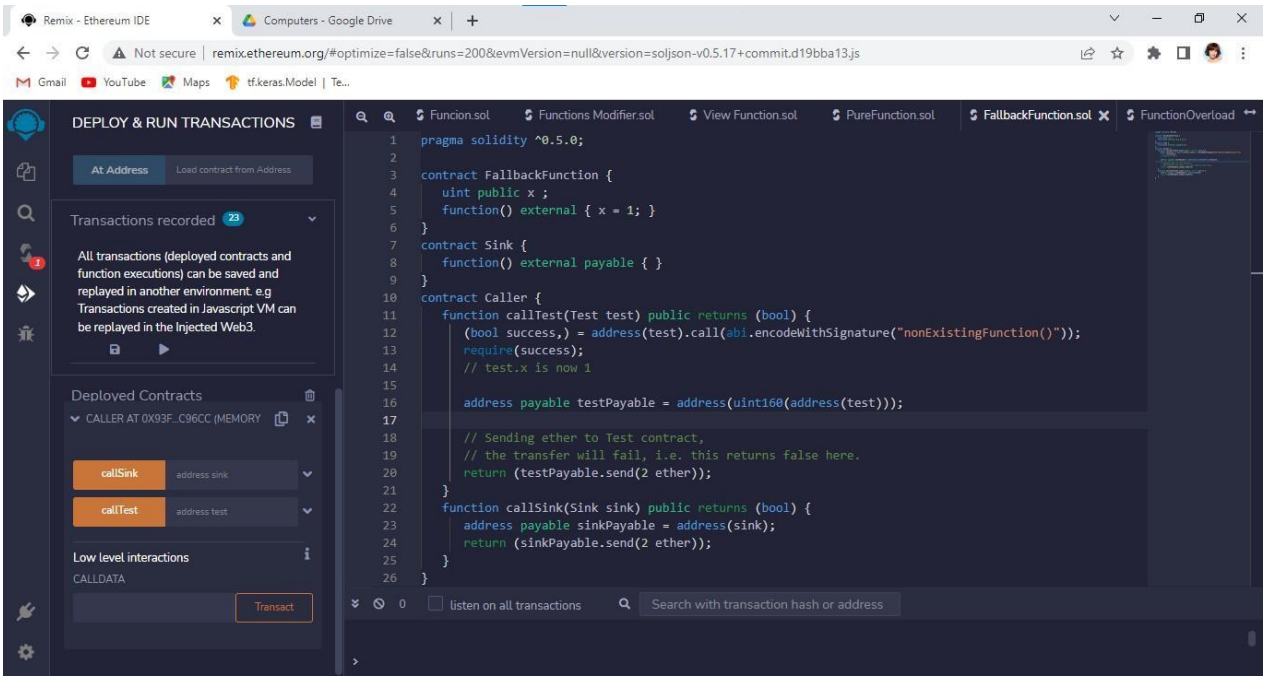




Fallback function

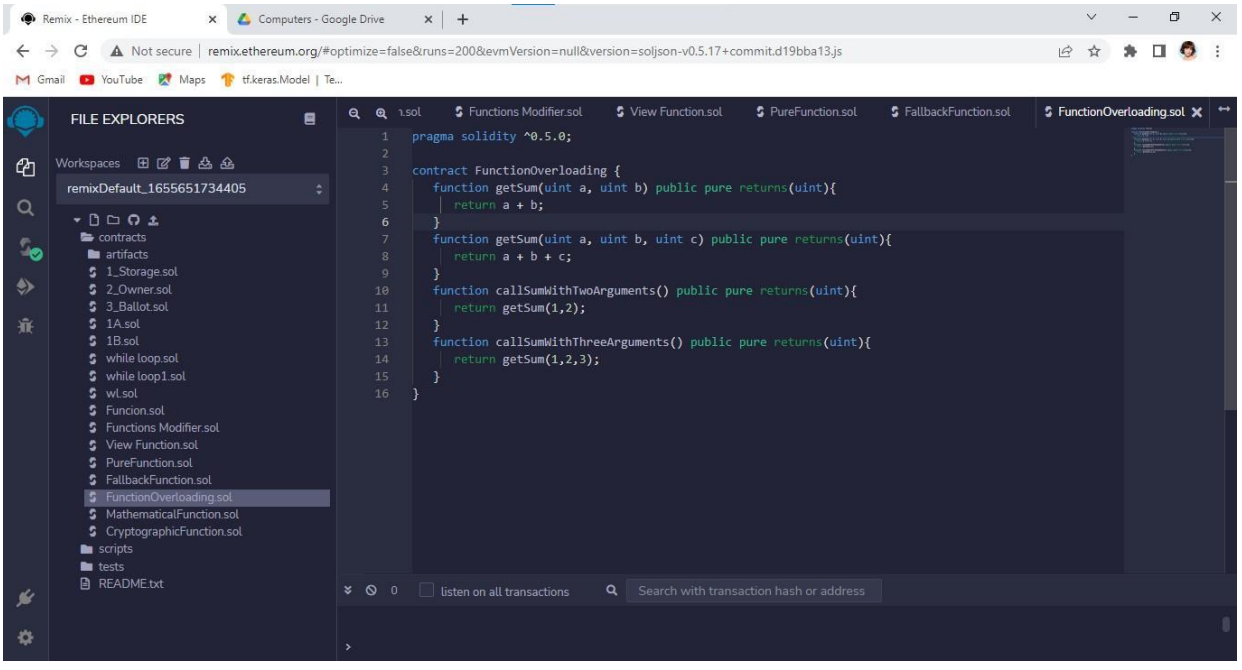
Code:

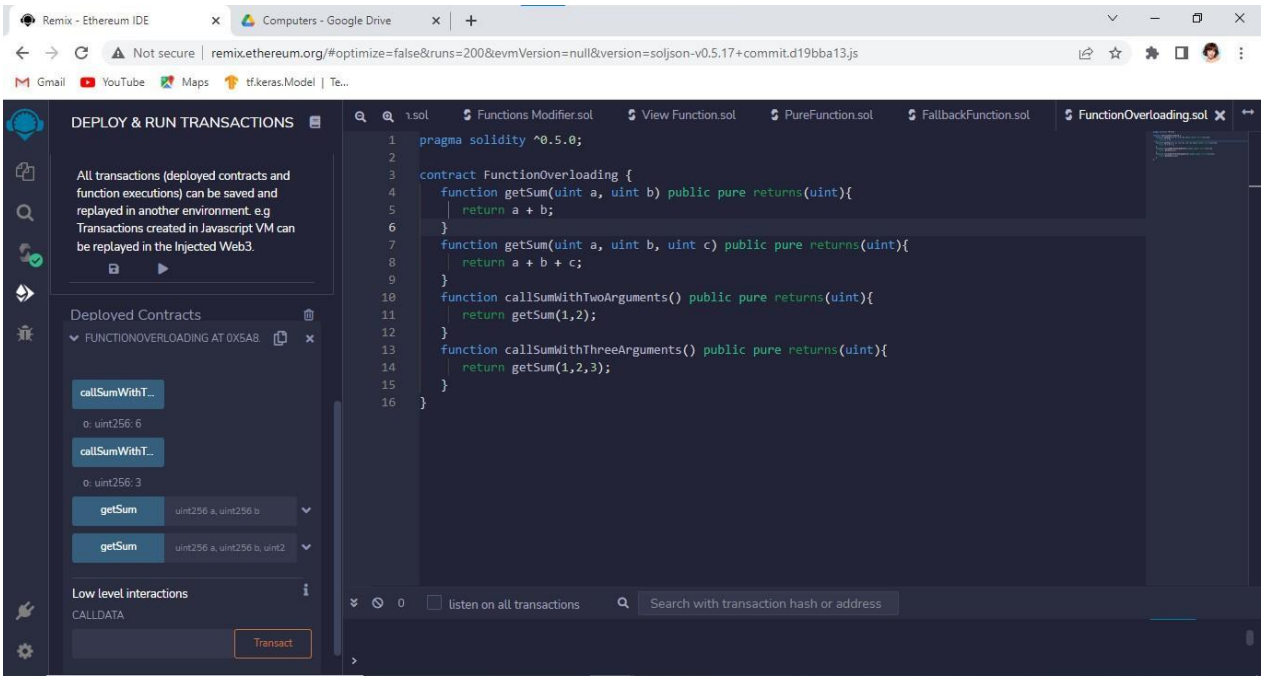




Function Overloading

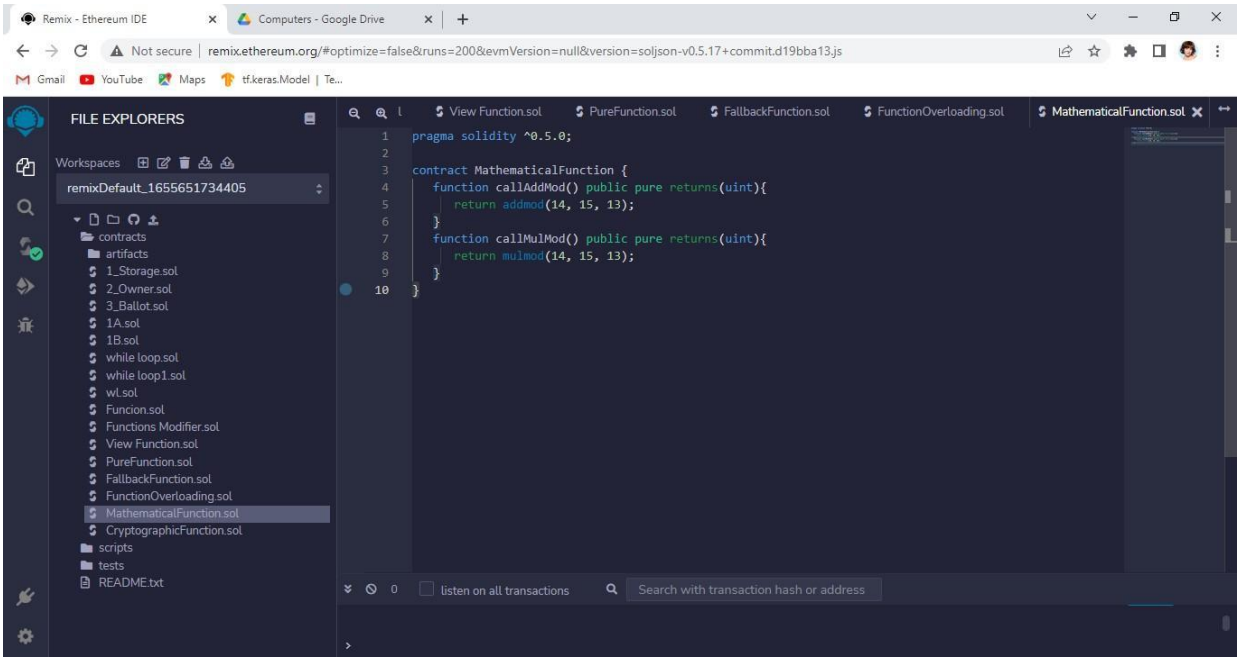
Code:

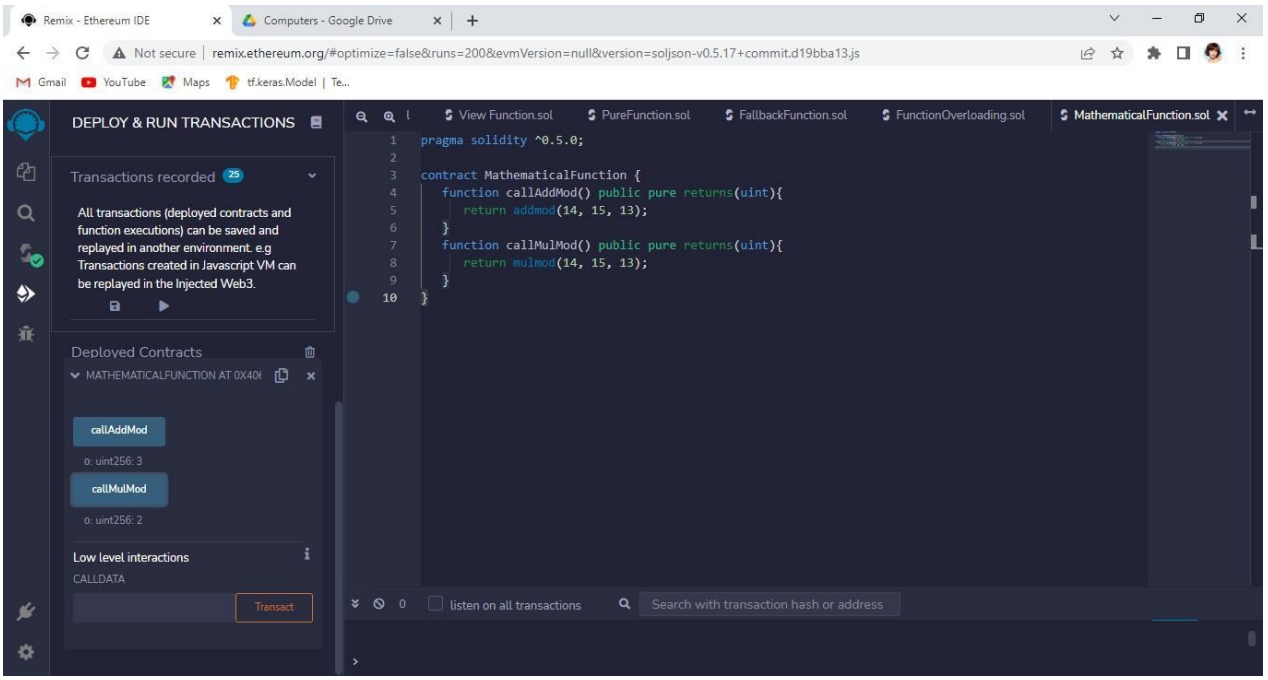




Mathematical Function

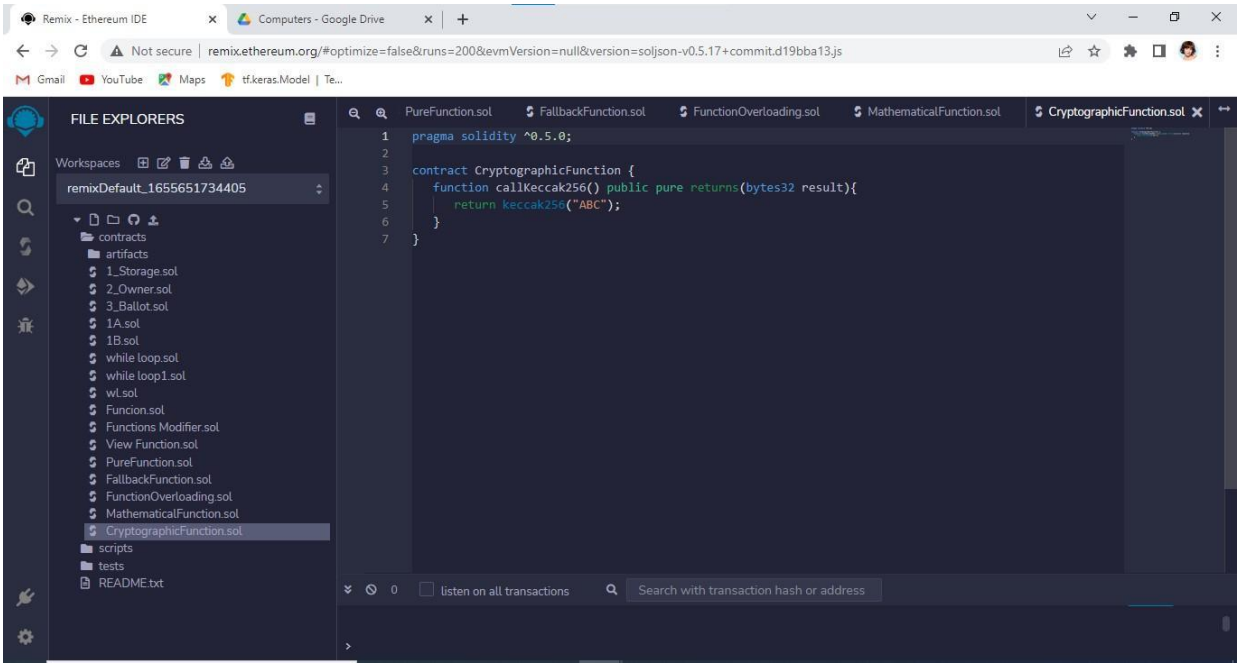
Code:

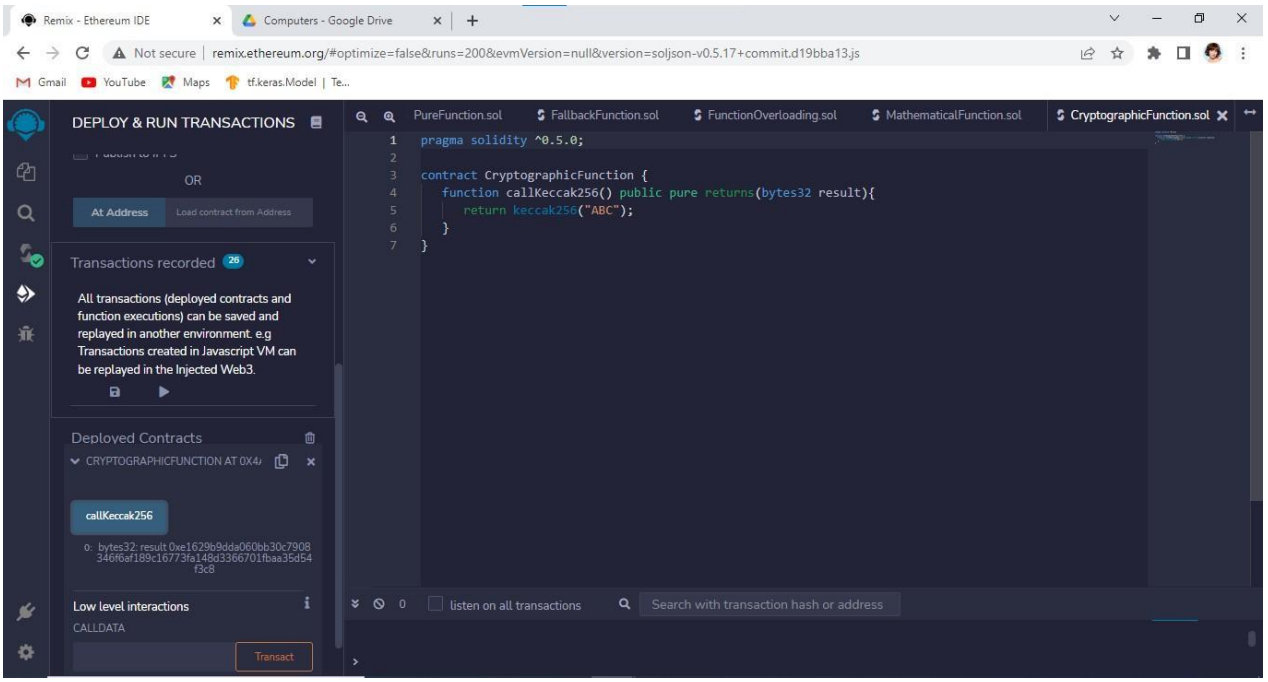




Cryptographic function

Code:

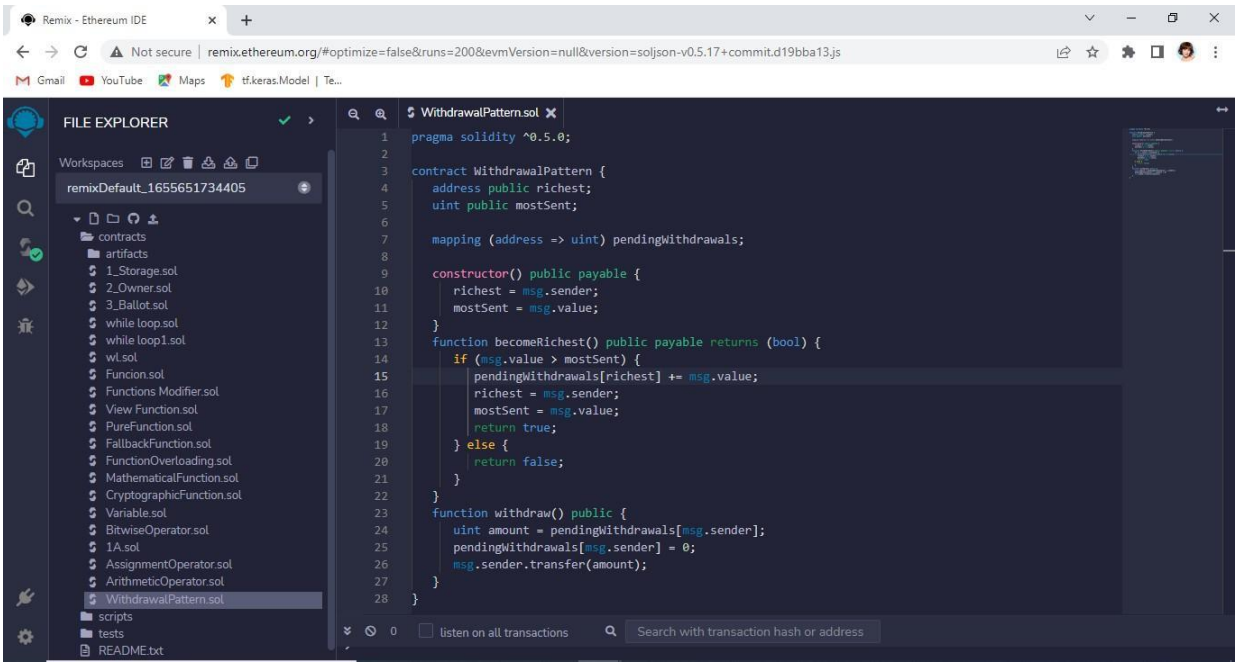


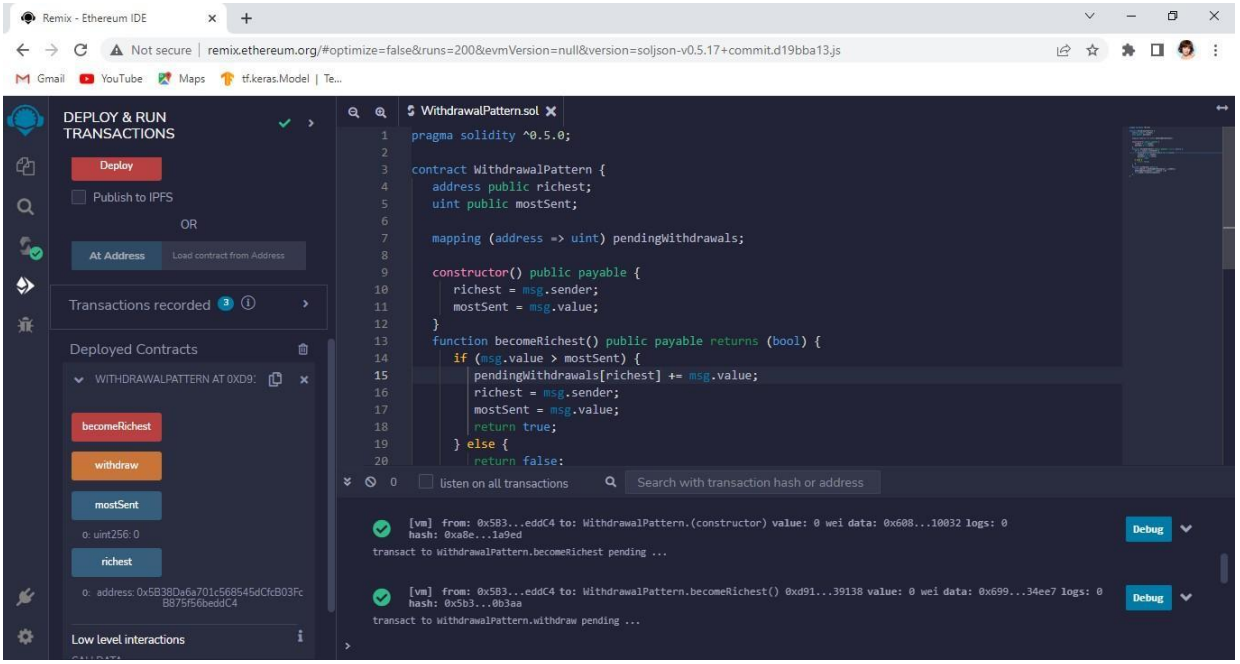


3a) Withdrawal Pattern, Restricted Access.

Withdrawal Pattern

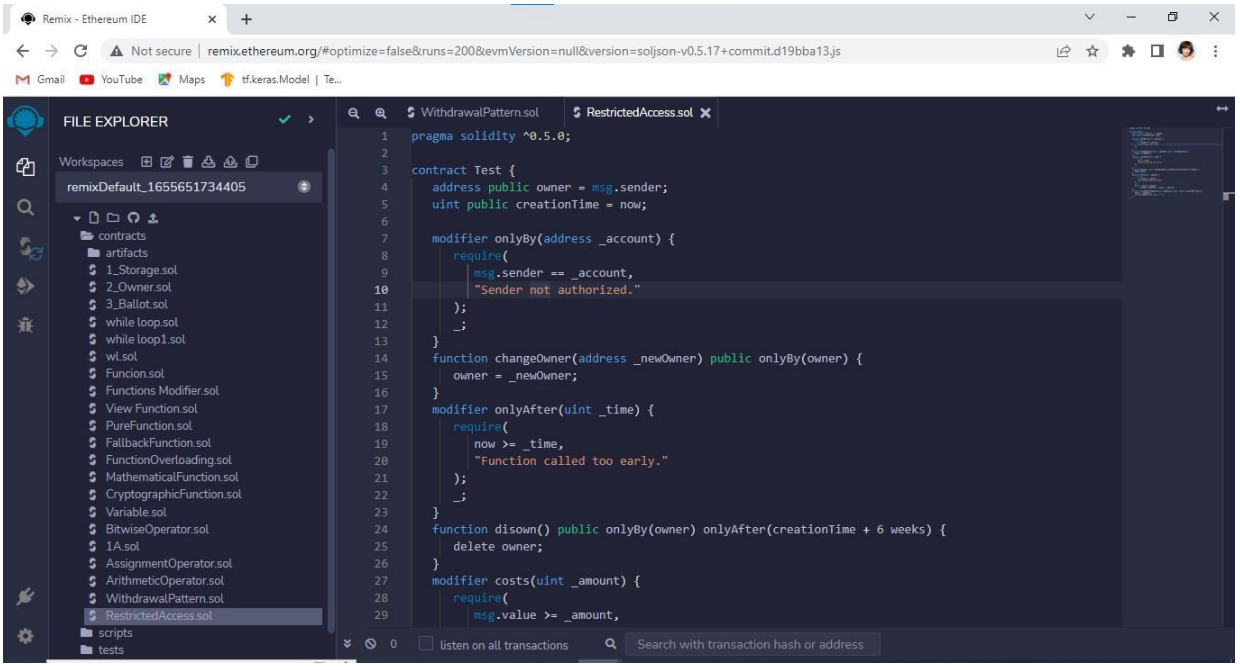
Code:

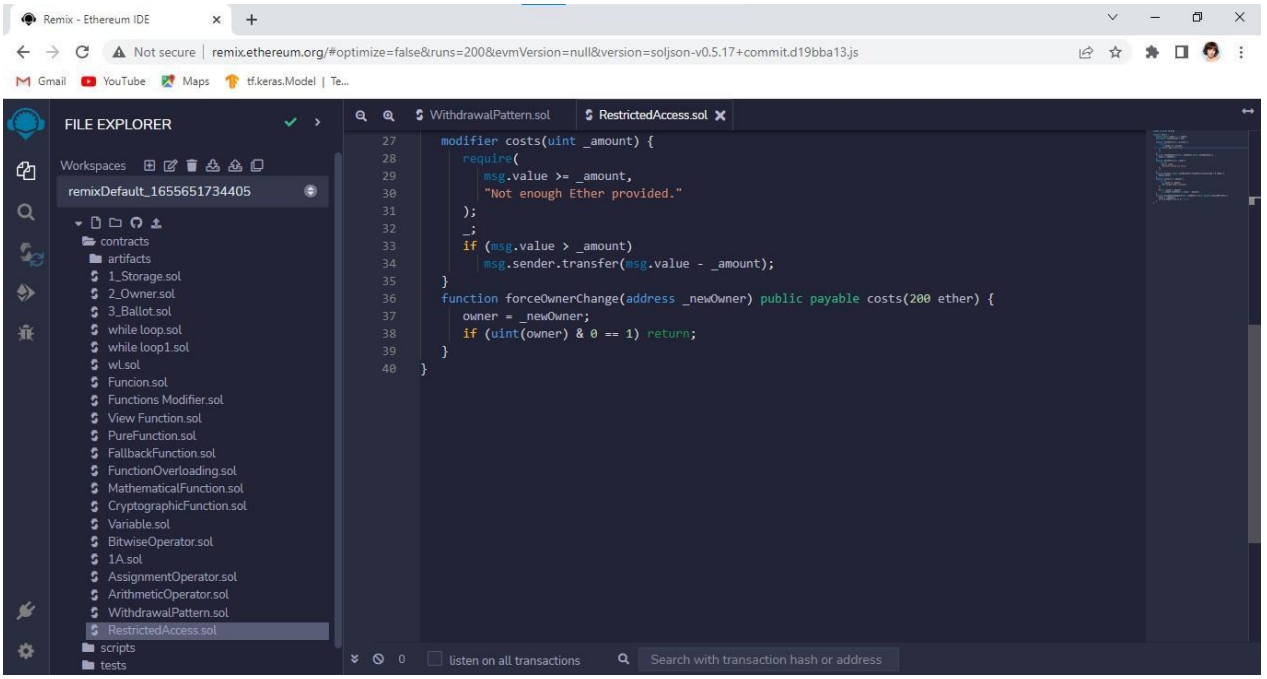




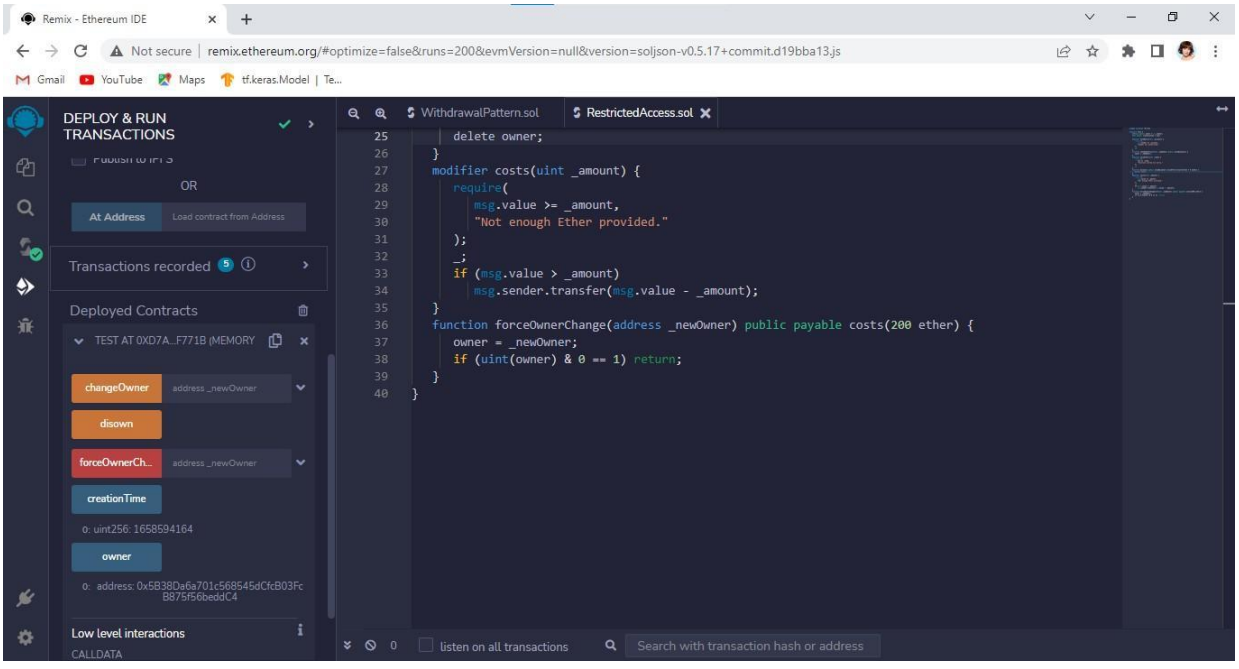
Restricted Access

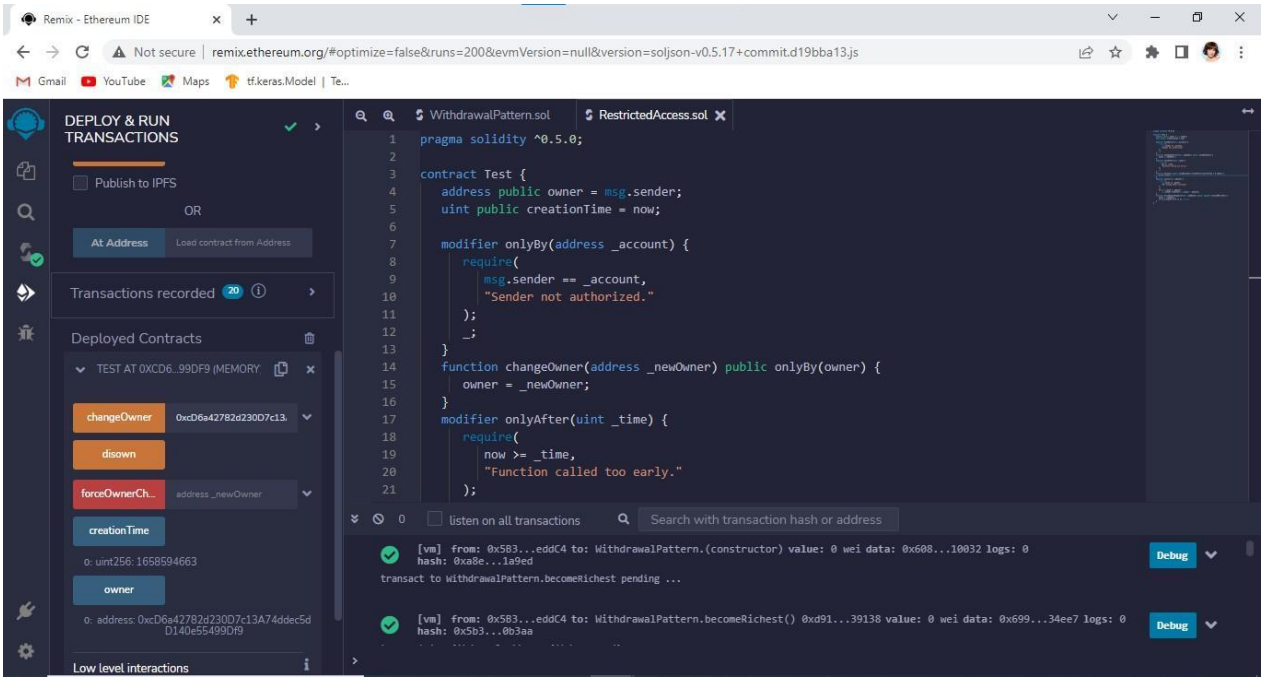
Code:




Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
scripts
tests
WithdrawalPattern.sol
RestrictedAccess.sol
27 modifier costs(uint _amount) {
28 require(
29 msg.value >= _amount,
30 "Not enough Ether provided."
31);
32 _;
33 if (msg.value > _amount)
34 msg.sender.transfer(msg.value - _amount);
35 }
36 function forceOwnerChange(address _newOwner) public payable costs(200 ether) {
37 owner = _newOwner;
38 if (uint(owner) & 0 == 1) return;
39 }
40 }
0
listen on all transactions
Search with transaction hash or address

Output:

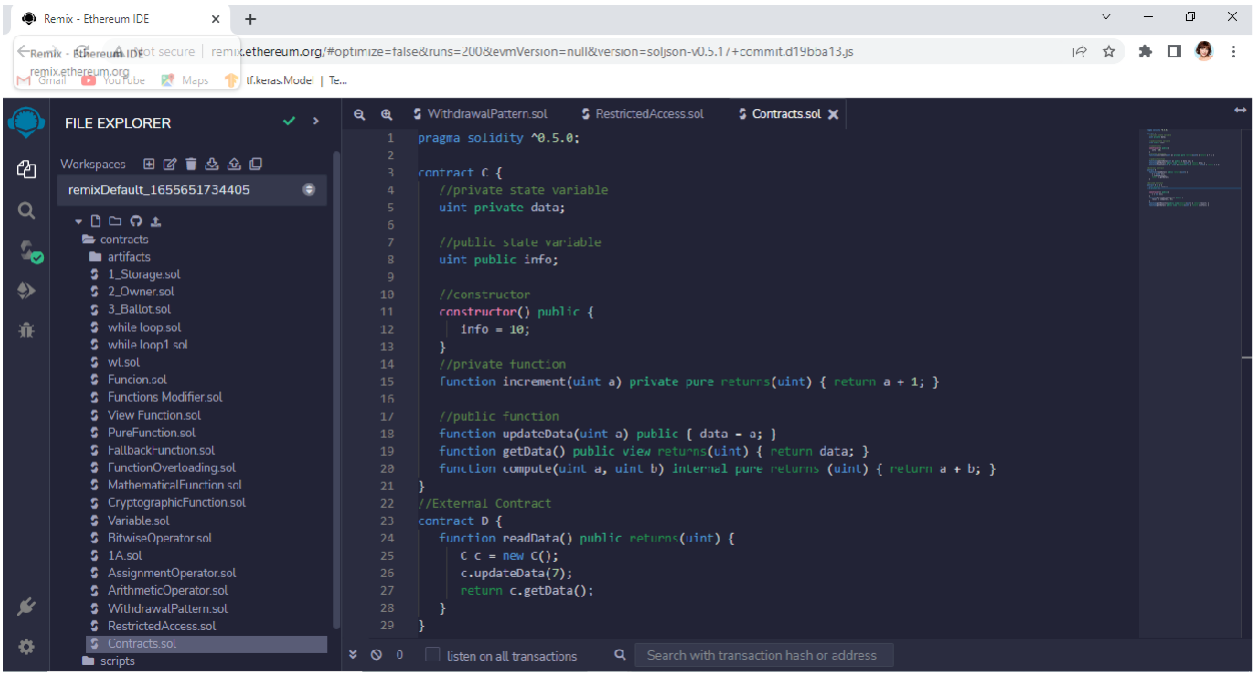

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
OR
At Address
Load contract from Address
Transactions recorded 5
Deployed Contracts
TEST AT 0XD7A...F771B (MEMORY
changeOwner
address _newOwner
disown
forceOwnerCh...
address _newOwner
creationTime
0: uint256: 1658594164
owner
0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
Low level interactions
CALLDATA
WithdrawalPattern.sol
RestrictedAccess.sol
25 delete owner;
26 }
27 modifier costs(uint _amount) {
28 require(
29 msg.value >= _amount,
30 "Not enough Ether provided."
31);
32 _;
33 if (msg.value > _amount)
34 msg.sender.transfer(msg.value - _amount);
35 }
36 function forceOwnerChange(address _newOwner) public payable costs(200 ether) {
37 owner = _newOwner;
38 if (uint(owner) & 0 == 1) return;
39 }
40 }
0
listen on all transactions
Search with transaction hash or address

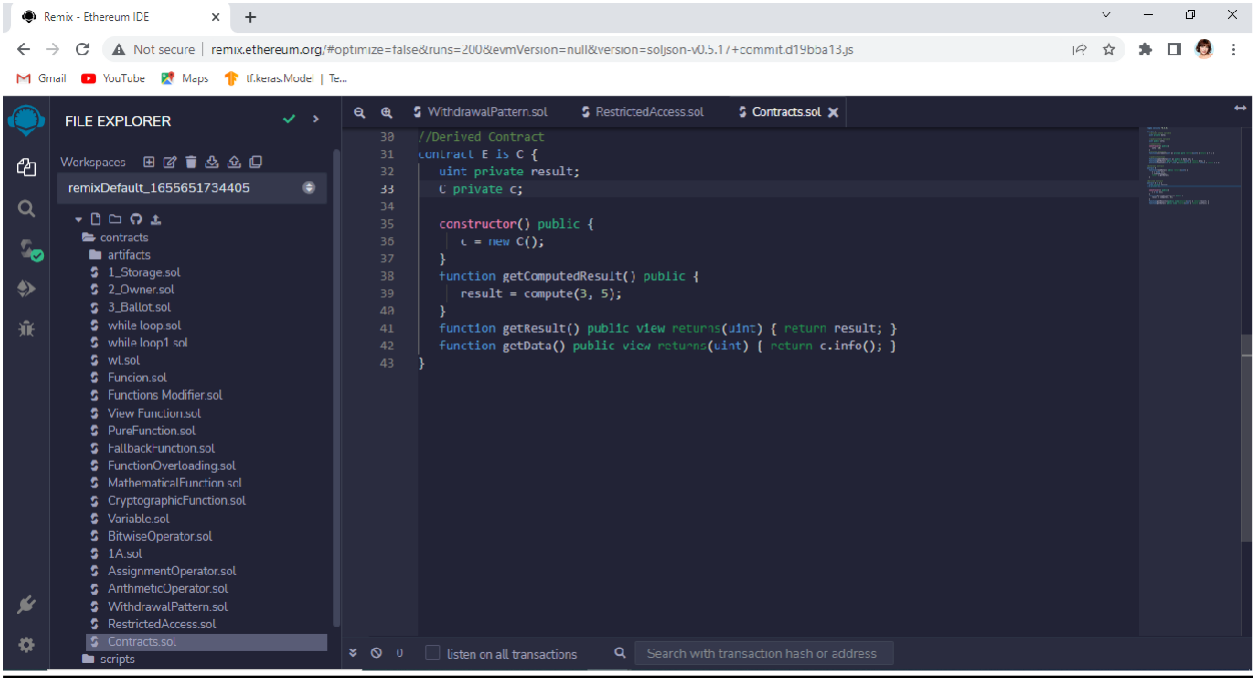


3b) Contracts, Inheritance, Constructors, Abstract Contracts, Interfaces.

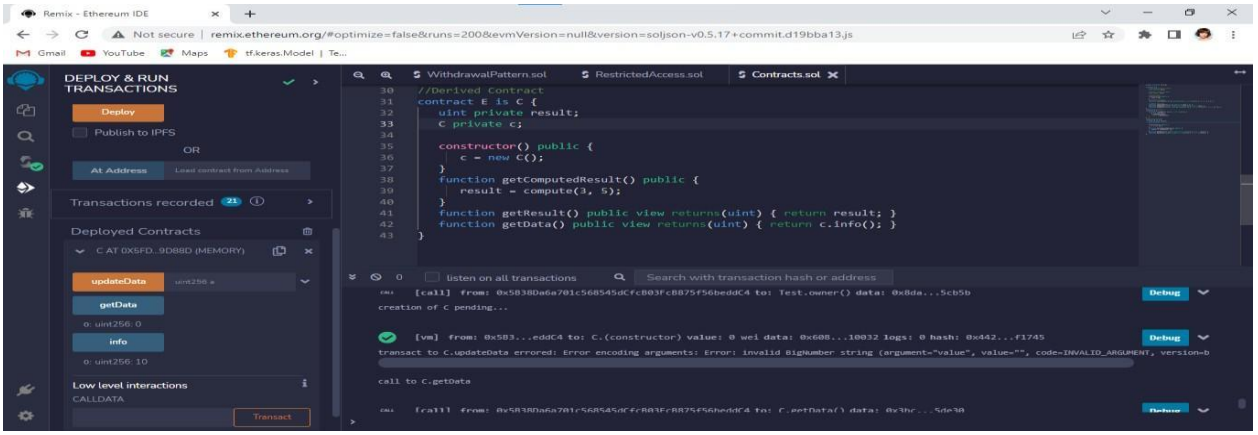
Contracts

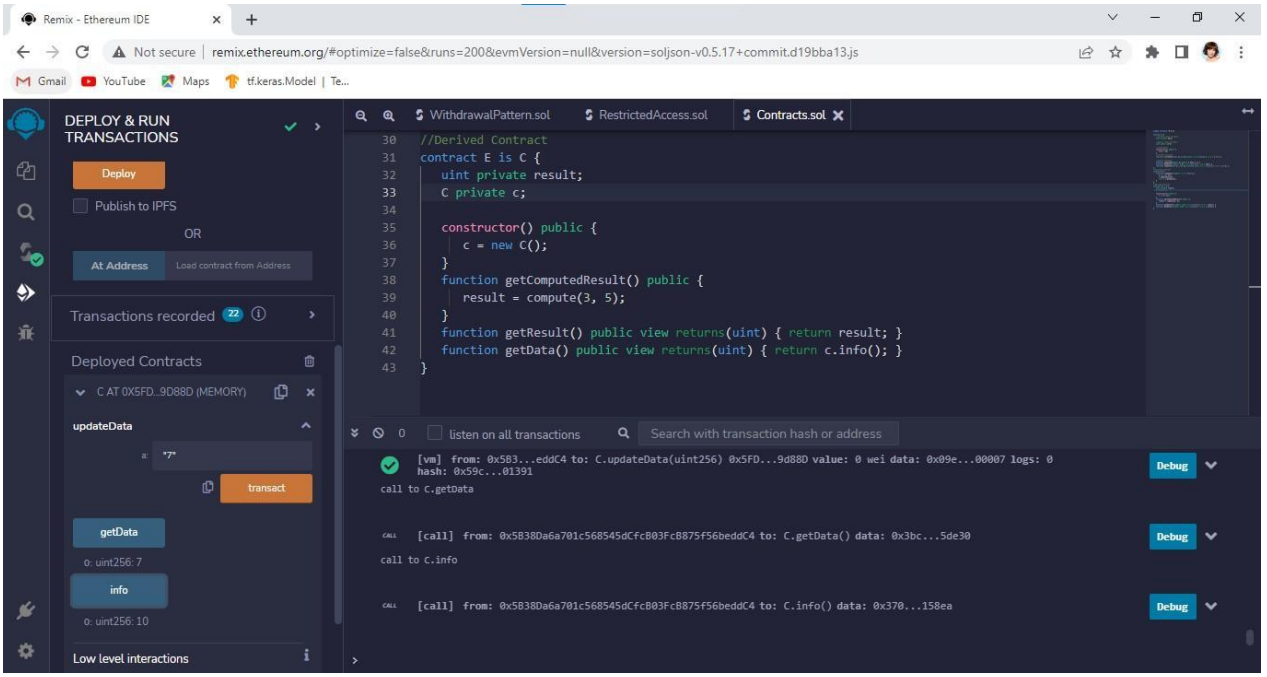
Code:




Remix - Ethereum IDE
FILE EXPLORER
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
//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(); }
}
Listen on all transactions
Search with transaction hash or address

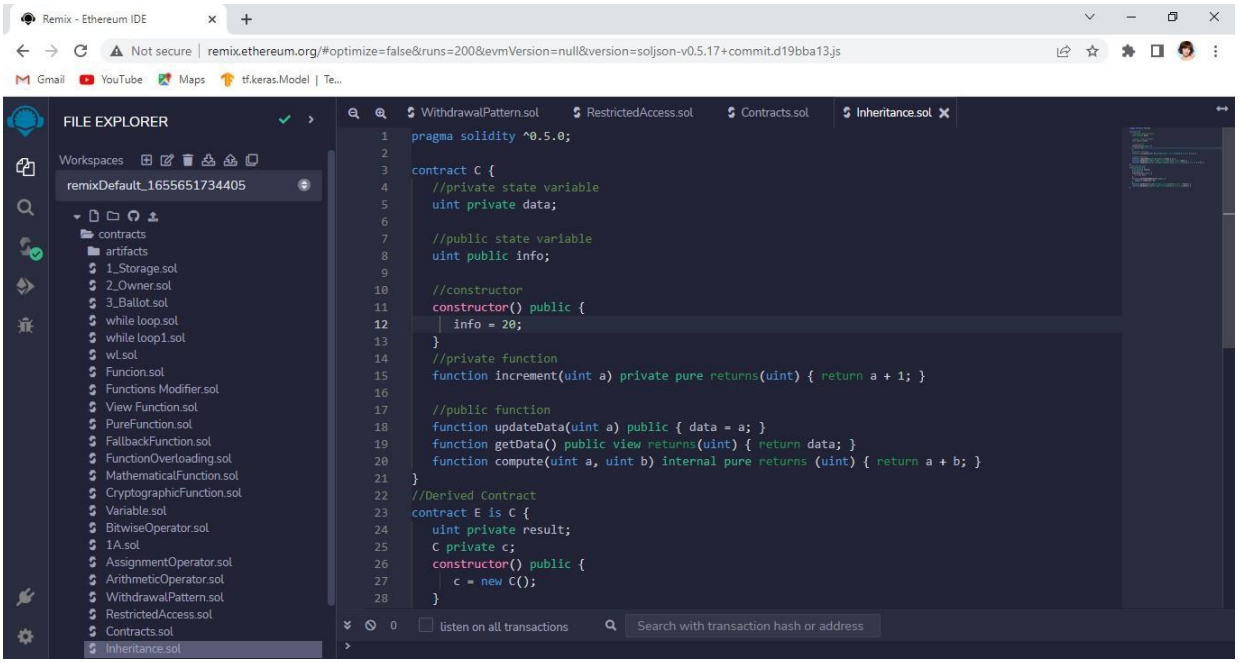
Output:

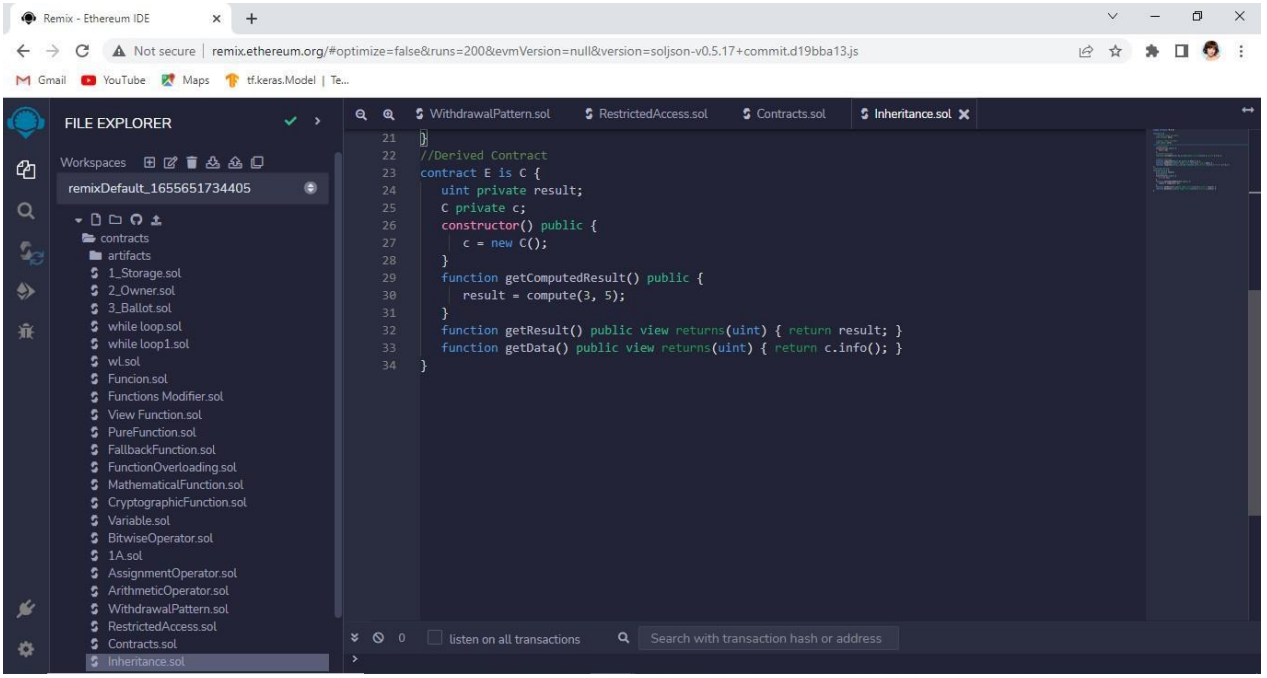

Remix - Ethereum IDE
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Transactions recorded
Deployed Contracts
updateData
getData
info
Low level interactions
CALLDATA
Transact
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Listen on all transactions
Search with transaction hash or address
creation of C pending...
call to C.getData



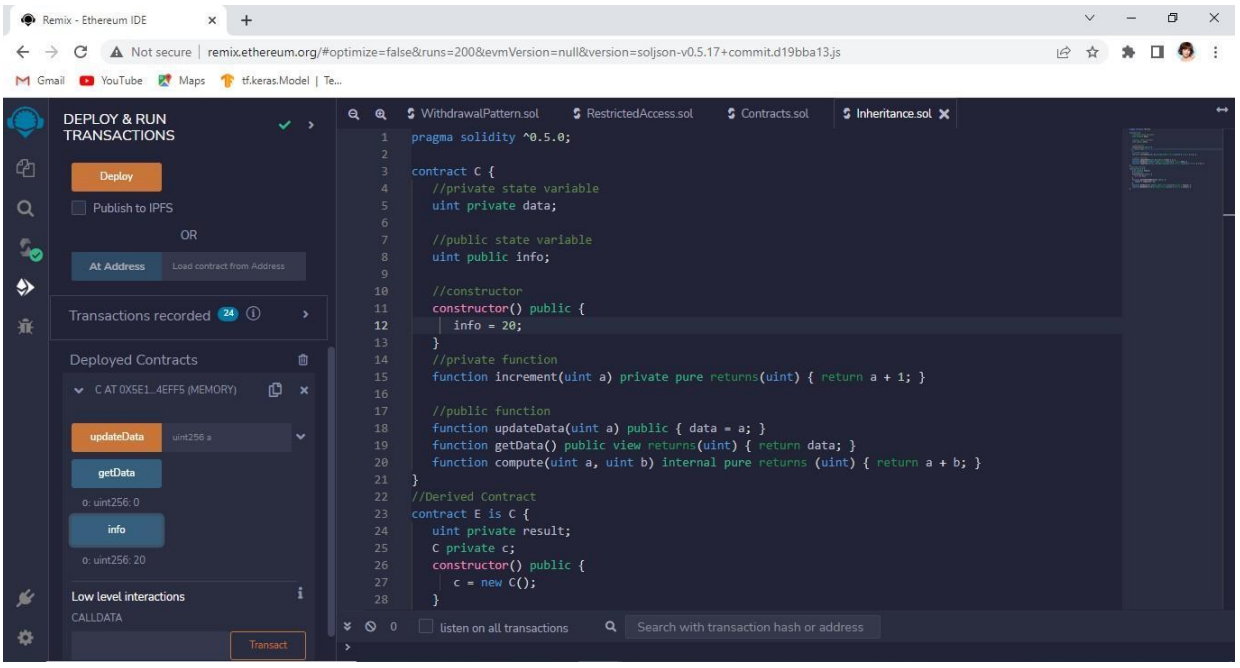
Inheritance

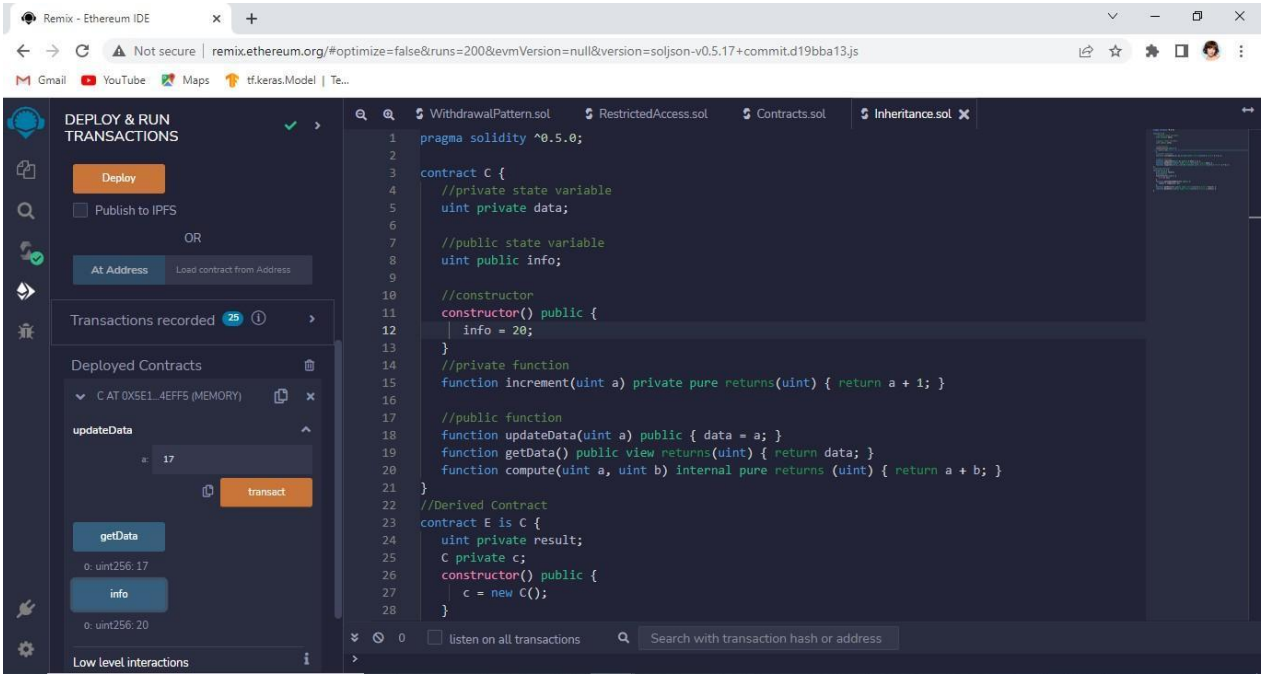
Code:




Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail YouTube Maps tf.keras.Model | Te...
FILE EXPLORER
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
}
//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(); }
}
0
listen on all transactions
Search with transaction hash or address

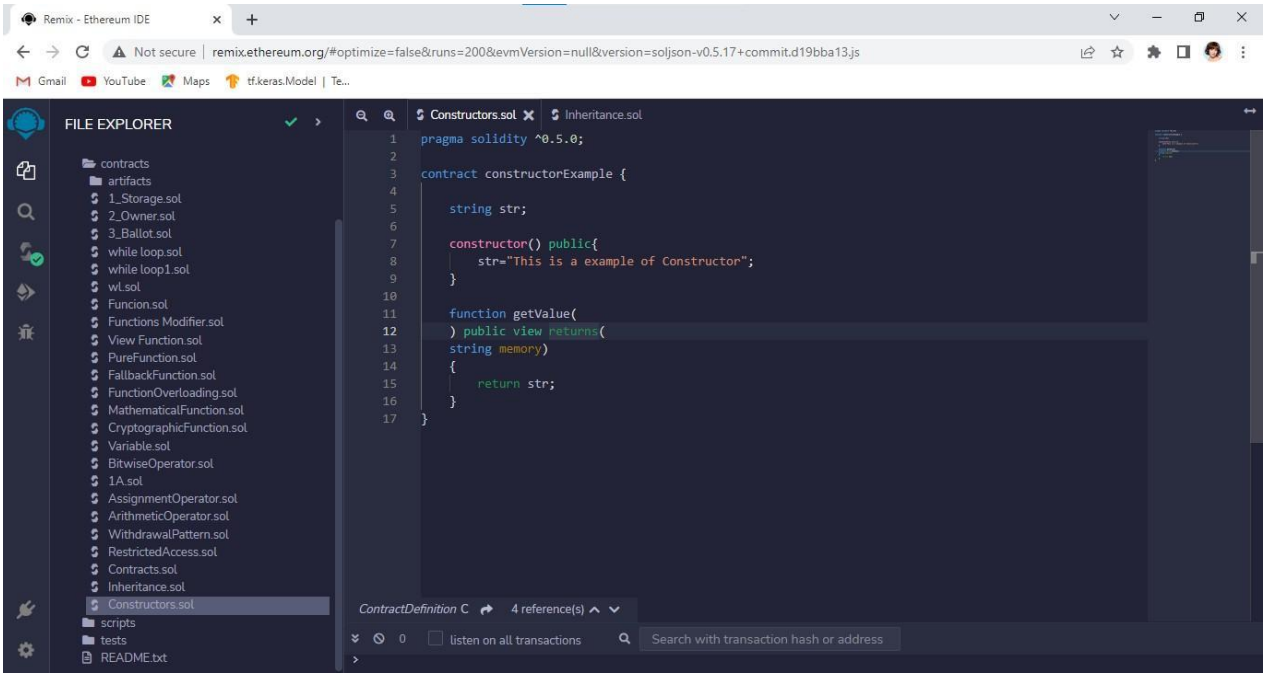
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail YouTube Maps tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 24
Deployed Contracts
C AT 0X5E1...4EFF5 (MEMORY)
updateData
uint256 a
getData
0: uint256: 0
info
0: uint256: 20
Low level interactions
CALLDATA
Transact
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
pragma solidity ^0.5.0;
contract C {
//private state variable
uint private data;
//public state variable
uint public info;
//constructor
constructor() public {
info = 20;
}
//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();
}
0
listen on all transactions
Search with transaction hash or address

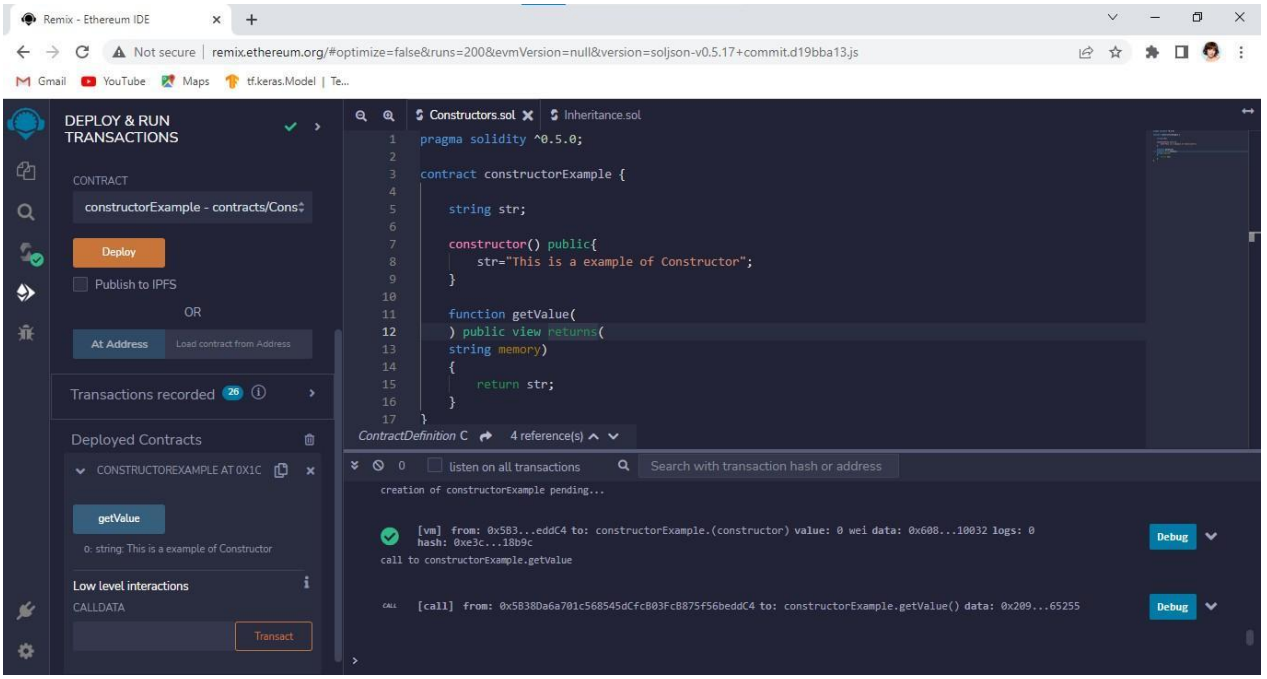


Constructors

Code:

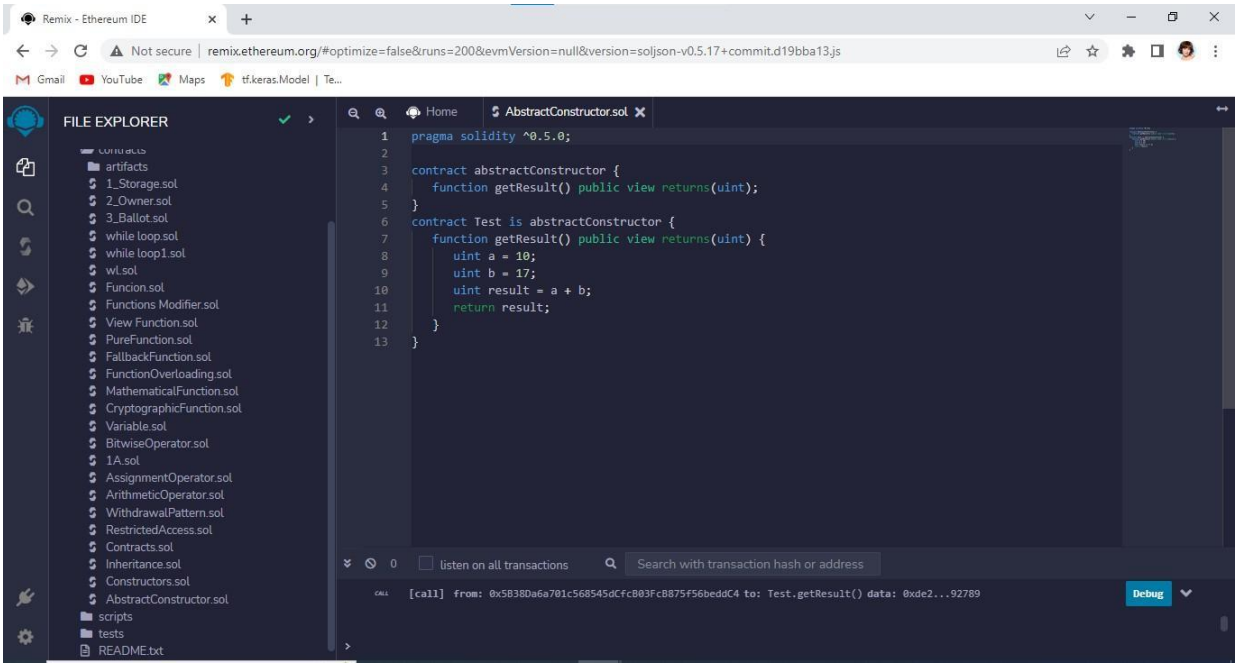


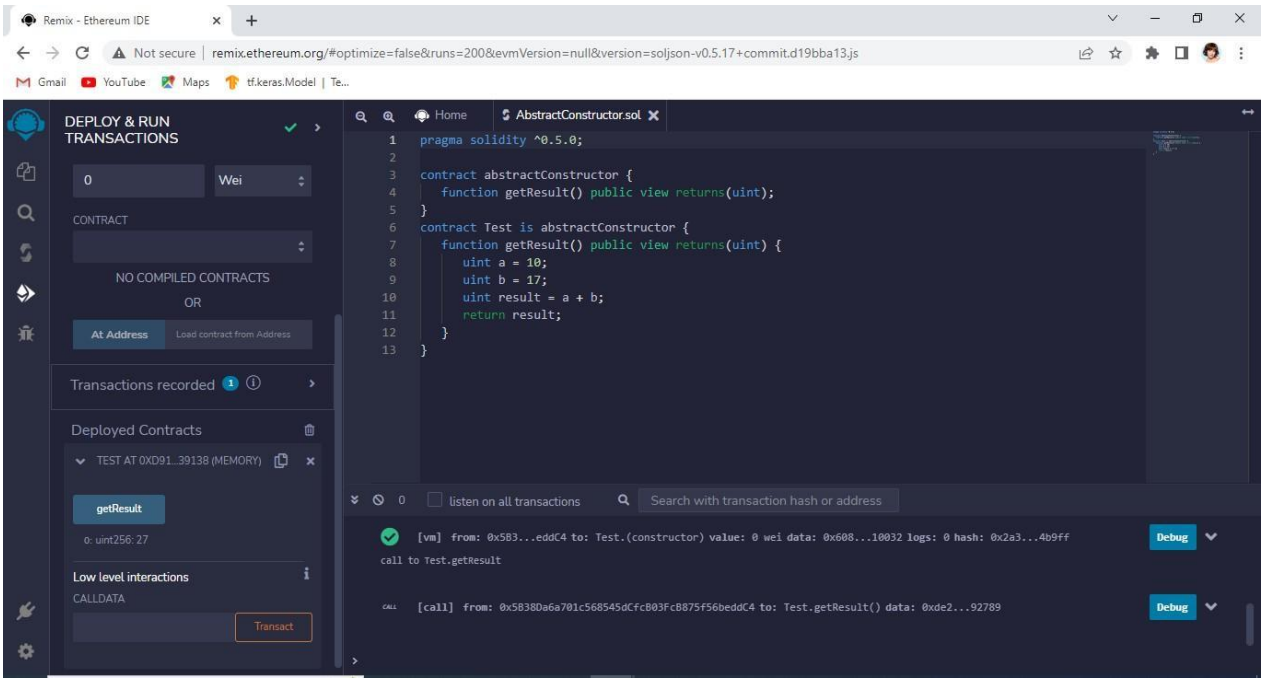
Output:



Abstract Contracts

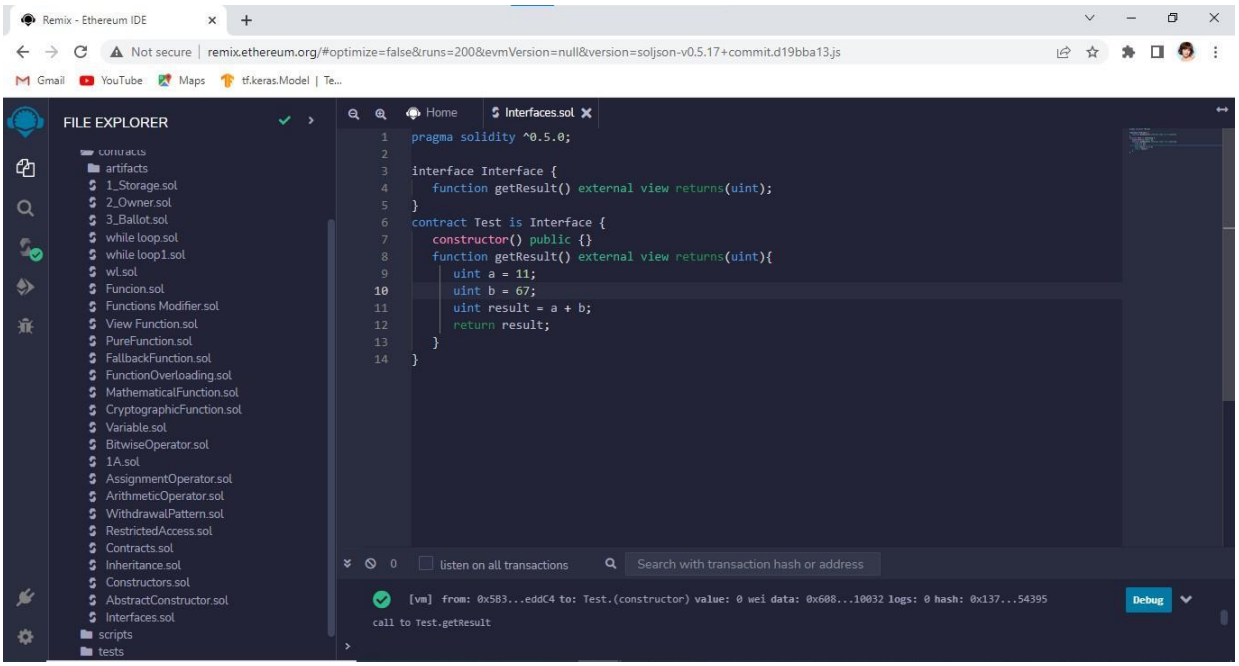
Code:

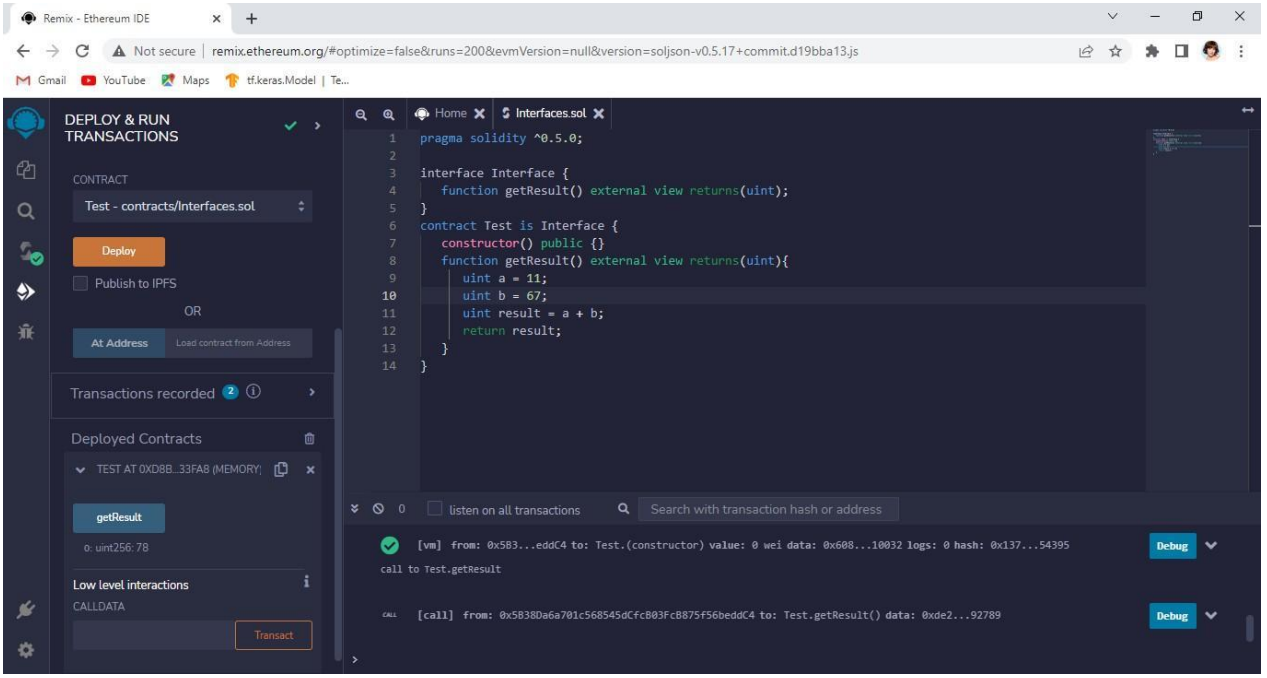




Interfaces

Code:

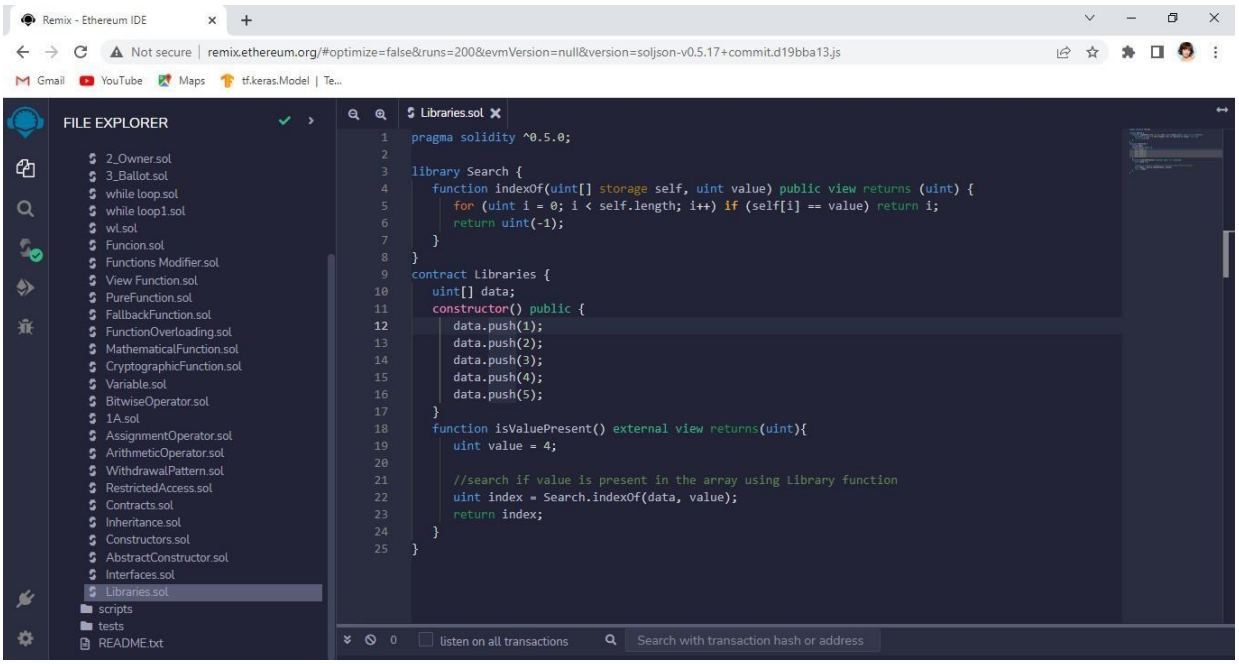


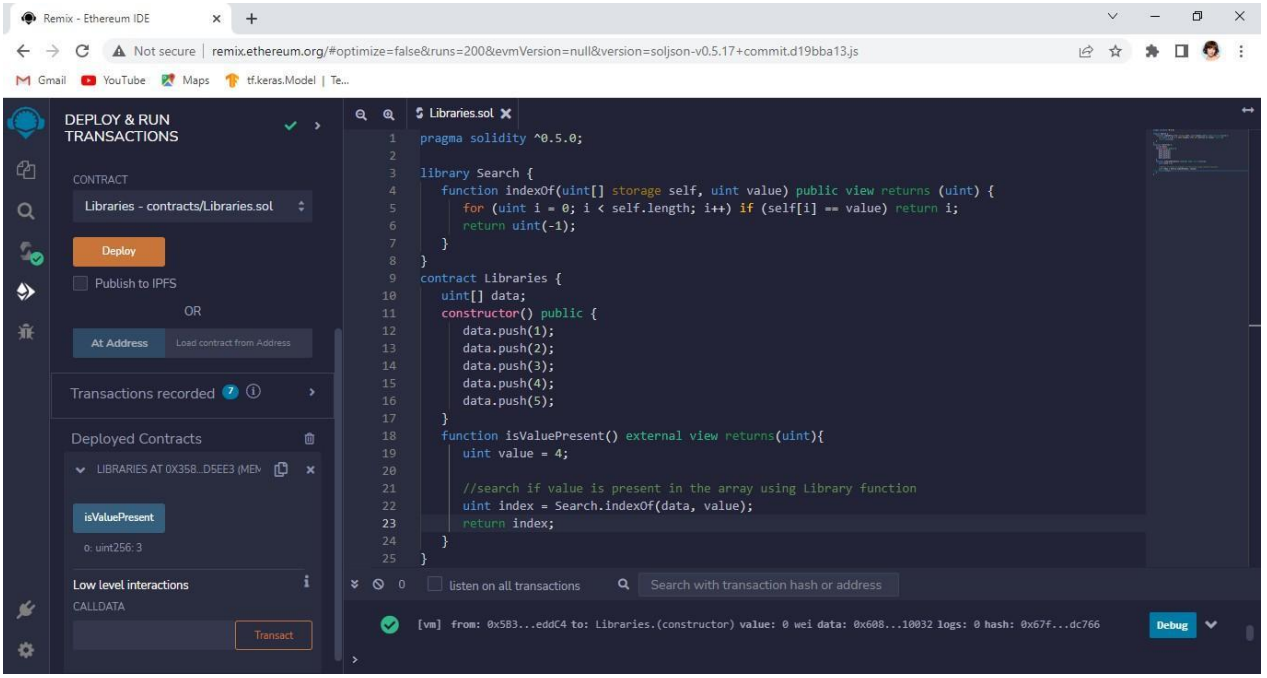


3c) Libraries, Assembly, Events, Error handling.

Libraries

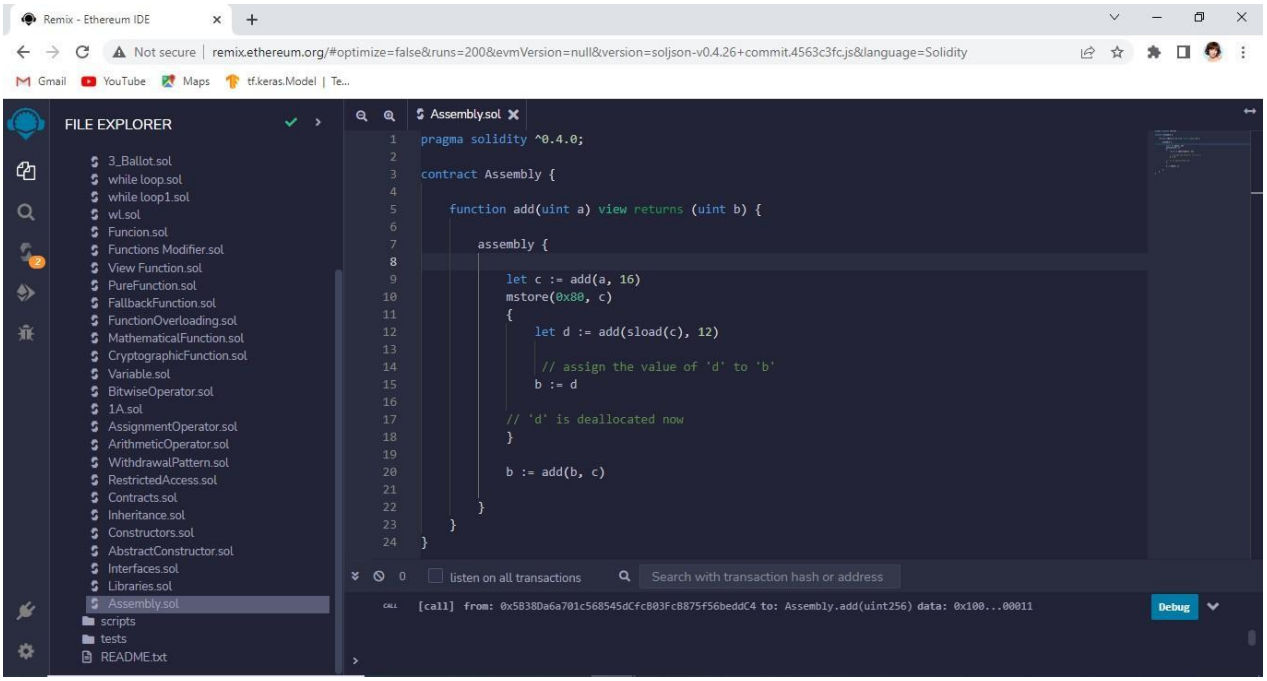
Code:



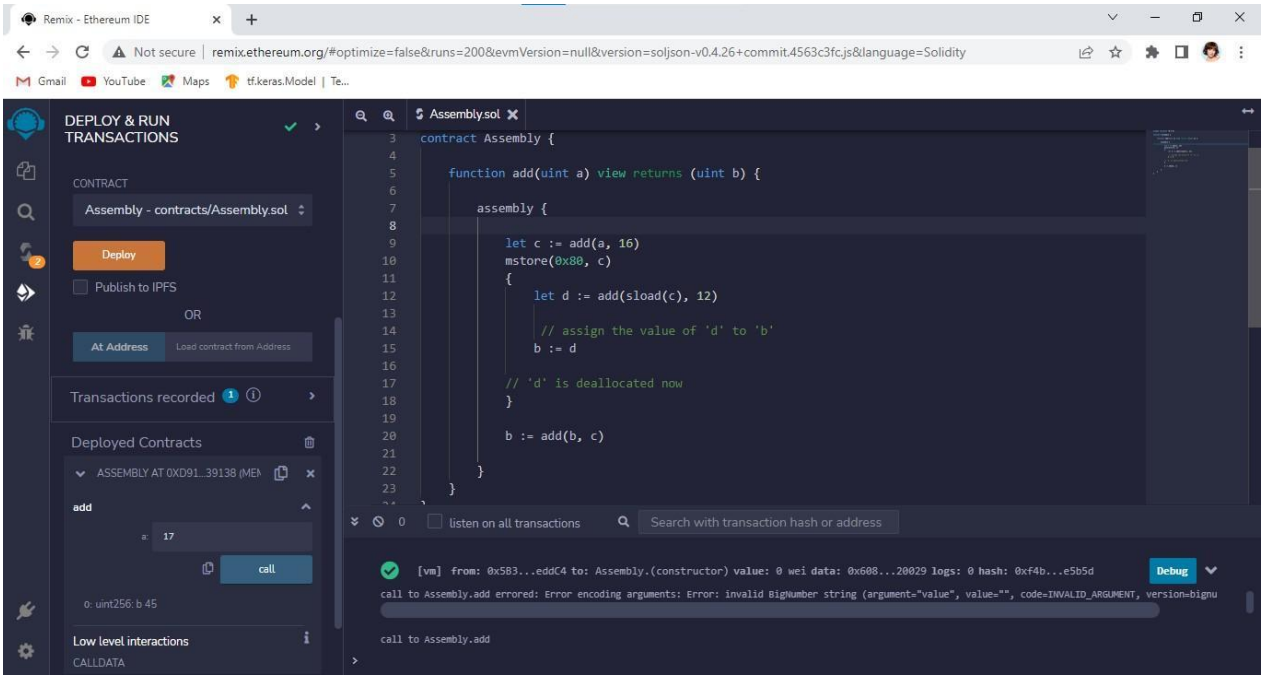


Assembly

Code:

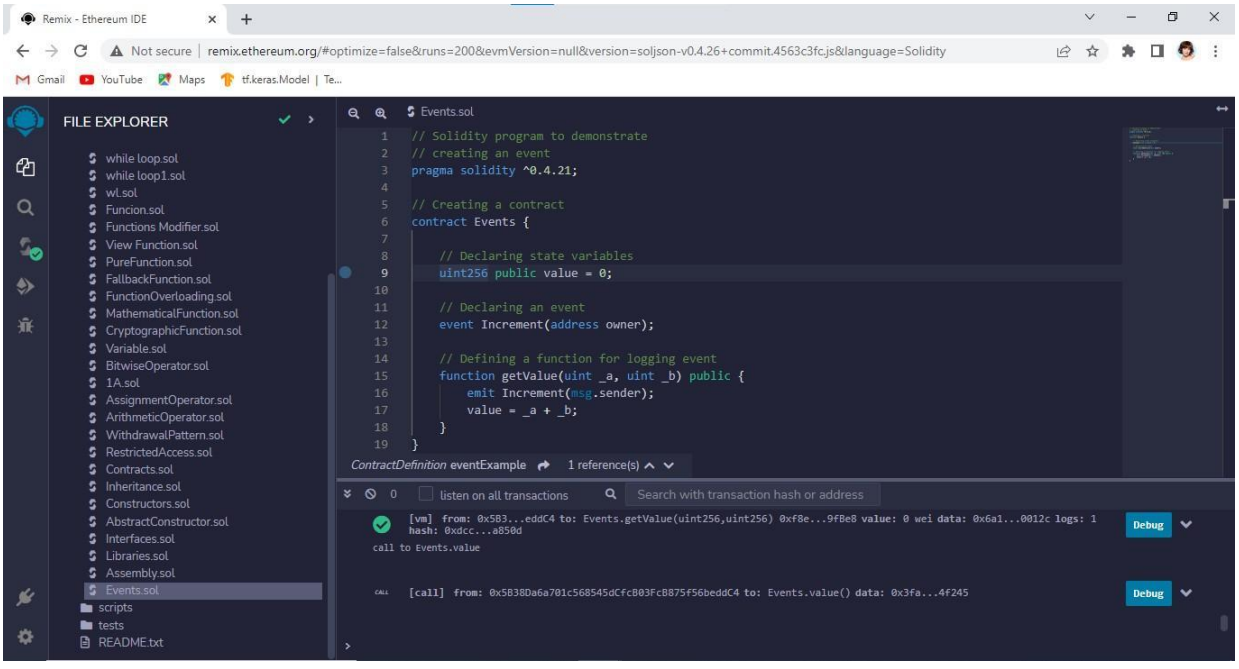


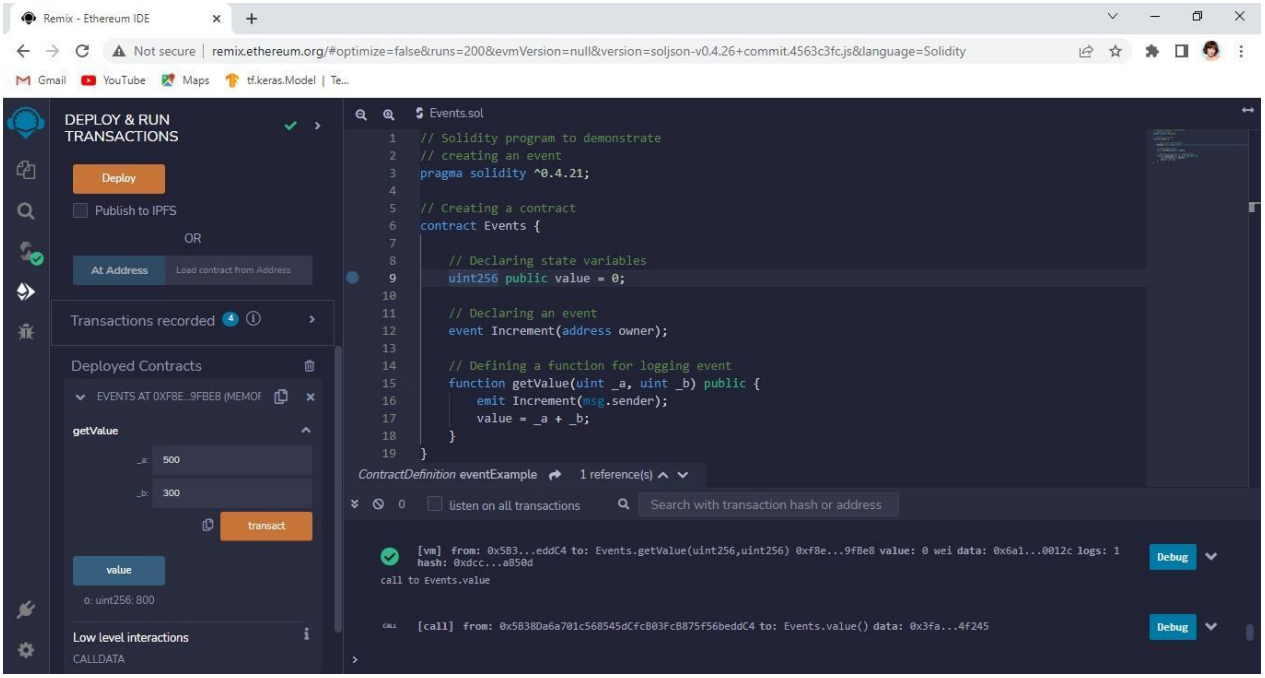
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
CONTRACT
Assembly - contracts/Assembly.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 1
Deployed Contracts
ASSEMBLY AT 0XD91...39138 (MEM
add
a: 17
call
0: uint256: b 45
Low level interactions
CALLDATA
Assembly.sol
contract Assembly {
function add(uint a) view returns (uint b) {
assembly {
let c := add(a, 16)
mstore(0x80, c)
{
let d := add(sload(c), 12)
// assign the value of 'd' to 'b'
b := d
// 'd' is deallocated now
}
b := add(b, c)
}
}
}
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: Assembly.(constructor) value: 0 wei data: 0x608...20029 logs: 0 hash: 0xf4b...e5b5d
Debug
call to Assembly.add errored: Error encoding arguments: Error: invalid BigNumber string (argument="value", value="", code=INVALID_ARGUMENT, version=bignu
call to Assembly.add

Events

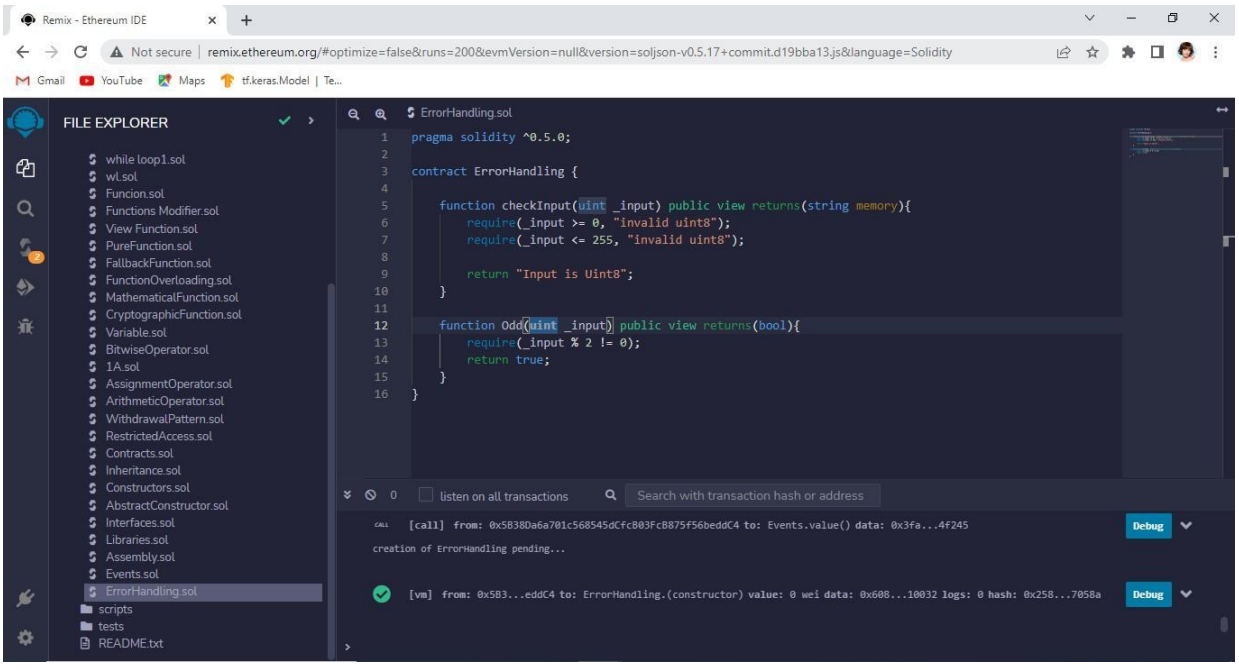
Code:

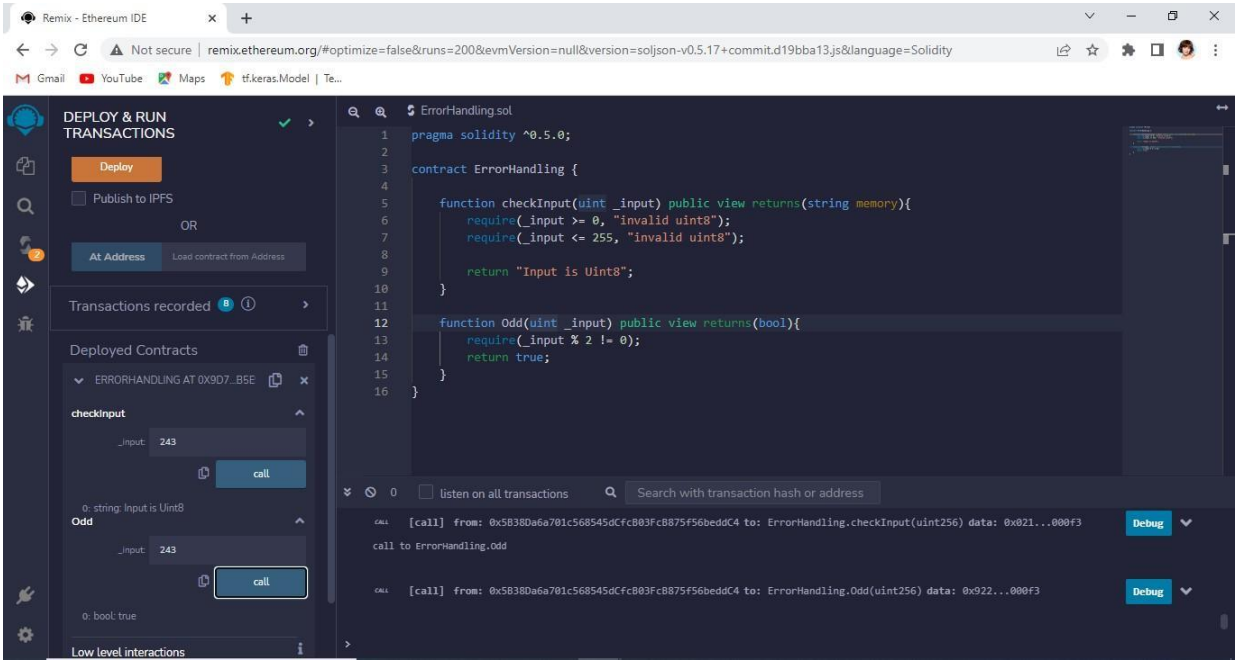

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
scripts
tests
README.txt
Events.sol
// Solidity program to demonstrate
// creating an event
pragma solidity ^0.4.21;
// Creating a contract
contract Events {
// Declaring state variables
uint256 public value = 0;
// Declaring an event
event Increment(address owner);
// Defining a function for logging event
function getValue(uint _a, uint _b) public {
emit Increment(msg.sender);
value = _a + _b;
}
}
ContractDefinition eventExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: Events.getValue(uint256,uint256) 0xf8e...9fBe8 value: 0 wei data: 0x6a1...0012c logs: 1 hash: 0xdcc...a850d
Debug
call to Events.value
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Events.value() data: 0x3fa...4f245
Debug



Error Handling

Code:




Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 8
Deployed Contracts
ERRORHANDLING AT 0X9D7...B5E
checkInput
_input: 243
call
0: string: Input is Uint8
Odd
_input: 243
call
0: bool: true
Low level interactions
ErrorHandling.sol
pragma solidity ^0.5.0;
contract ErrorHandling {
function checkInput(uint _input) public view returns(string memory){
require(_input >= 0, "invalid uint8");
require(_input <= 255, "invalid uint8");
return "Input is Uint8";
}
function Odd(uint _input) public view returns(bool){
require(_input % 2 != 0);
return true;
}
}
listen on all transactions
Search with transaction hash or address
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ErrorHandling.checkInput(uint256) data: 0x021...000f3
Debug
call to ErrorHandling.Odd
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ErrorHandling.Odd(uint256) data: 0x922...000f3
Debug

Practical 6

Aim: Demonstrate the use of Bitcoin Core API.

Code:

```
from hashlib import sha256
MAX_NONCE = 100000000000

def SHA256(text):
    return sha256(text.encode("ascii")).hexdigest()

def mine(block_number, transactions, previous_hash, prefix_zeros):
    prefix_str = '0'*prefix_zeros
    for nonce in range(MAX_NONCE):
        text = str(block_number) + transactions + previous_hash + str(n
once)                                       new_hash = SHA256(text)
                                            if new_hash.startswith(prefix_str):
                                                print(f"Yay! Successfully mined
                                                bitcoins with nonce value:{
nonce}")
            return new_hash

    raise BaseException(f"Couldn't find correct has after trying {MAX_N
ONCE} times")

if_name_=='_main__':
    transactions='''
    Dhaval->Bhavin->20,
    Mando->Cara->45
    '''
    difficulty=4 # try changing this to higher number and you will see
it will take more time for mining as difficulty increases
    import time
    start = time.time()
    print("start mining")
    new_hash = mine(5,transactions,'0000000xa036944e29568d0cff17edbe038
f81208fecf9a66be9a2b8321c6ec7', difficulty)
    total_time = str((time.time() - start))
    print(f"end mining. Mining took: {total_time} seconds")
    print(new_hash)
```

```
start mining
Yay! Successfully mined bitcoins with nonce value:2425
end mining. Mining took: 0.011358022689819336 seconds
0000de957fbfdfc77582e0d0b20c53d2d1d83d8bb8cfe3693521f672bf2a6021
```

Practical 7

Aim: Create your own blockchain and demonstrate its use.

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

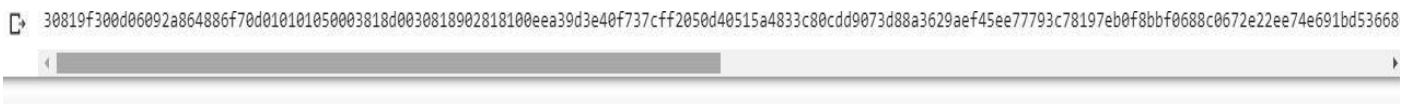
Code:

```
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.PublicKey import RSA
from Crypto.Signature import PKCS1_v1_5
import binascii

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(se
```

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

```
print (Dinesh.identity)
```

Output:



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

Code:

```
class Transaction:
                                    def __init__(self,
                                        sender,
                                        recipient,
                                        value):
                                        self.sender =
                                        sender
                                        s
                                        e
                                        l
                                        f
                                        .
                                        r
                                        e
                                        c
                                        i
                                        p
                                        i
                                        e
                                        n
                                        t
                                        =
                                        r
                                        e
                                        c
                                        i
                                        p
                                        i
                                        e
                                        n
                                        t
                                        s
                                        e
                                        l
                                        f
                                        .
                                        v
                                        a
```

```
l                                                   a
u                                                   l
e                                                   u
=                                                   e
v                                                   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')
Dinesh = Client()
Ramesh = Client()
t = Transaction(Dinesh,Ramesh.identity,5.0)

signature = t.sign_transaction()

print (signature)
```

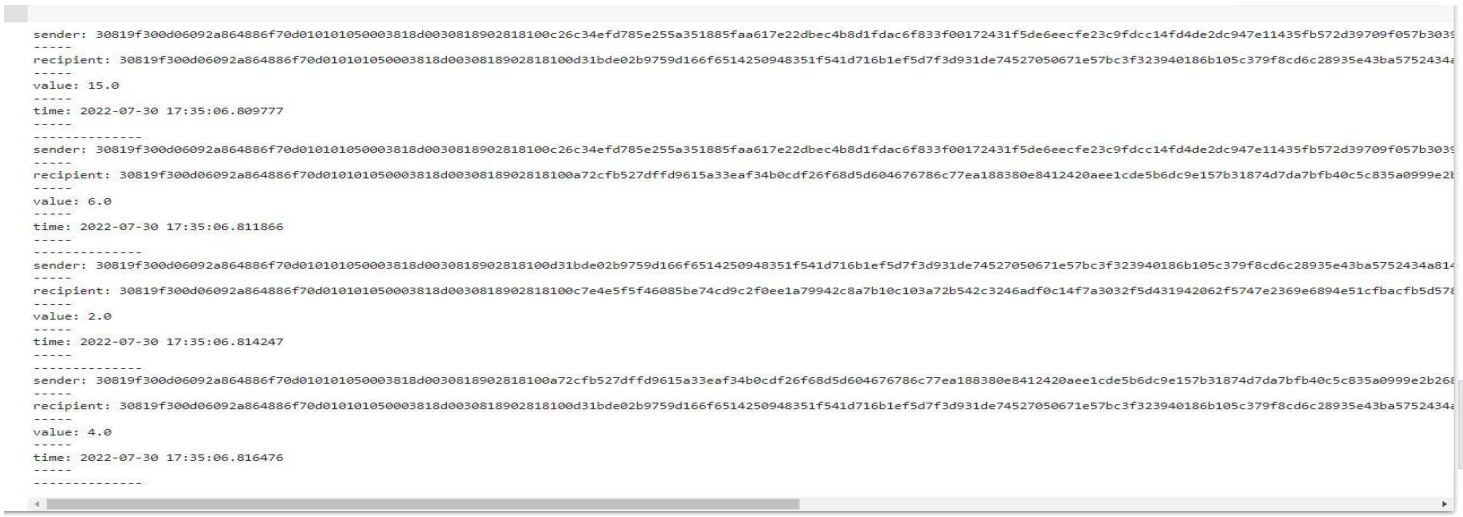
Output:

1c) Create Multiple Transaction and display them.

Code:

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

  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)

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

Output:



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

Code:

```
class Block:
                                    def __init(self):self.verified_transa
```

```
ctions=[]self.previous_block
                                    ock_hash=""self.Nonce=""
```

```
last_block_hash = ""

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 = []

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

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

Output:

```
Number of blocks in the chain: 1
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100a4a79ac0645d8483c7c97fb6b67e1bf90f17d87d1f
-----
value: 500.0
-----
time: 2022-07-30 17:50:16.288802
-----
--------------
====================================
```

1e) Create a mining function and test it.

Code:

```
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: "+ dig
est)
    return digest

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

Output:

```
after 238 iterations found nonce: 1124711db6185591392c6a06c24a3c2ebbaeca647fb8374824f939ada27c09e9
after 353 iterations found nonce: 11e0a4b57bb76496ecc6ab5a3c126165bc9dbbaf80f664e7e192a422360c3884
after 419 iterations found nonce: 11280dfd9ab05b3dbfa8699901537329414O8faa4a3b0832819b161f52321c08
after 511 iterations found nonce: 1150944bd7ea429acd052da390b148f27eb881686e0f8bfcdf77f833af878e2e
after 822 iterations found nonce: 110e61d41d94b48f6dfcb6f43f9ceafe2ab7168456dab088e05126d43d0366ef
after 924 iterations found nonce: 11147bf32bafeee4505b5cc40c3b227366d8a41ab3bd740e437c74400b7a8127
'6d80c4cf3cd4fb49aad25deced696c3a48e17974ecbfb1441e2128517f0152b2'
```

1f) Add blocks to the miner and dump the blockchain

Code:

```
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

    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)
```

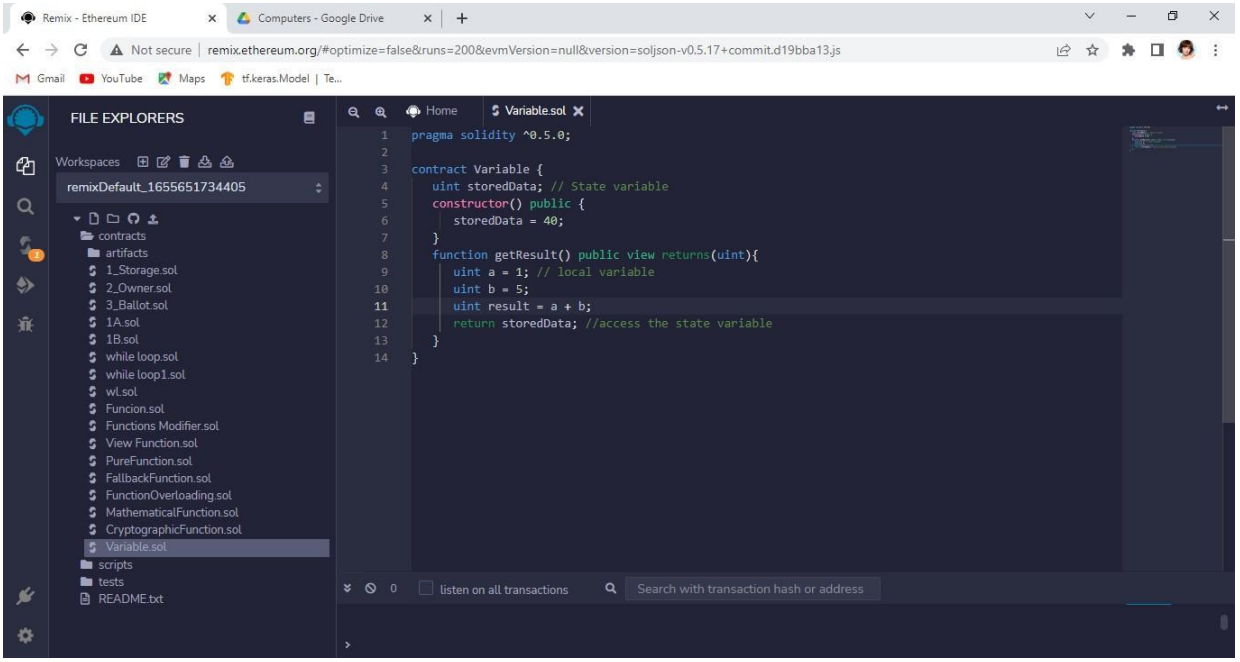
Output:

```
after 308 iterations found nonce: 111b1715b007a9920e76b241e9a4dde15a50b00faeed353bf9279311deb59804
after 398 iterations found nonce: 11a65418aca0201df85c2eeacabbcf0624092abb64bf6e159ccf5539cbb1cfe1
after 527 iterations found nonce: 11310267f14590e45be2375fe94e881c65f86020e3fcf7844a849241783680a7
after 718 iterations found nonce: 117ba8286d575afe73007e6f588b6f541580bec444d5165a3f71d9e4b623f3fa
after 308 iterations found nonce: 111b1715b007a9920e76b241e9a4dde15a50b00faeed353bf9279311deb59804
after 398 iterations found nonce: 11a65418aca0201df85c2eeacabbcf0624092abb64bf6e159ccf5539cbb1cfe1
after 527 iterations found nonce: 11310267f14590e45be2375fe94e881c65f86020e3fcf7844a849241783680a7
after 718 iterations found nonce: 117ba8286d575afe73007e6f588b6f541580bec444d5165a3f71d9e4b623f3fa
after 308 iterations found nonce: 111b1715b007a9920e76b241e9a4dde15a50b00faeed353bf9279311deb59804
after 398 iterations found nonce: 11a65418aca0201df85c2eeacabbcf0624092abb64bf6e159ccf5539cbb1cfe1
after 527 iterations found nonce: 11310267f14590e45be2375fe94e881c65f86020e3fcf7844a849241783680a7
after 718 iterations found nonce: 117ba8286d575afe73007e6f588b6f541580bec444d5165a3f71d9e4b623f3fa
```

2a) Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs, Mappings, Conversions, Ether Units, Special Variables. Variable

<u>Variable</u>

Code:

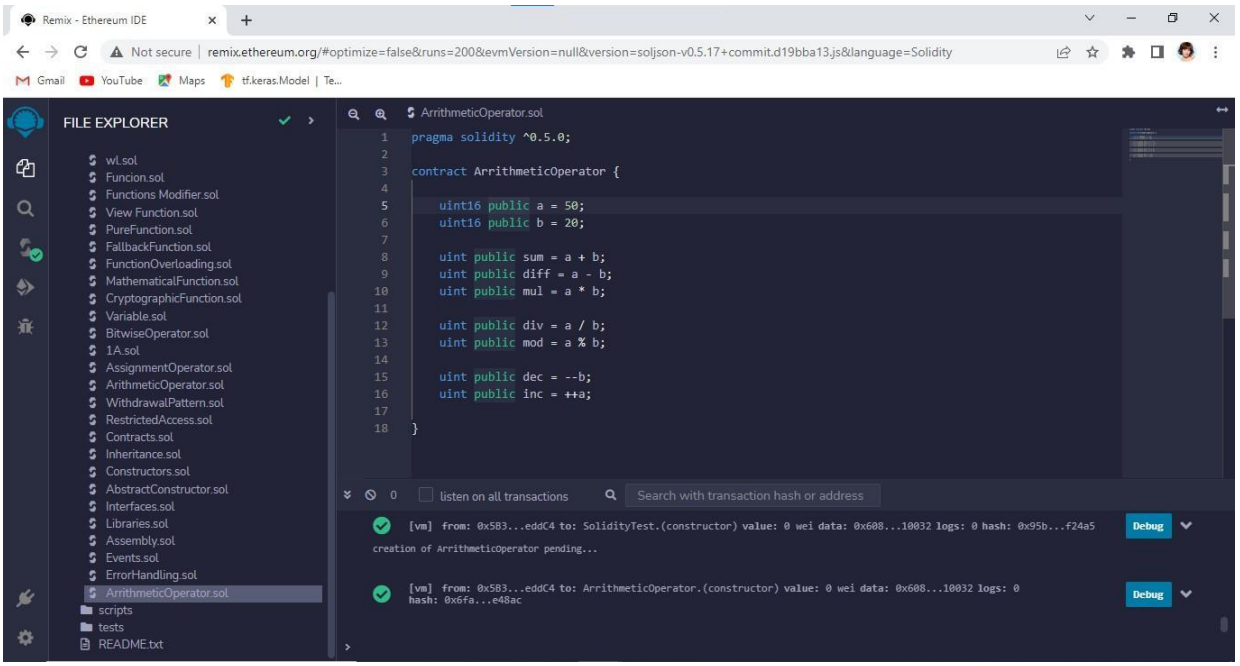


Output:

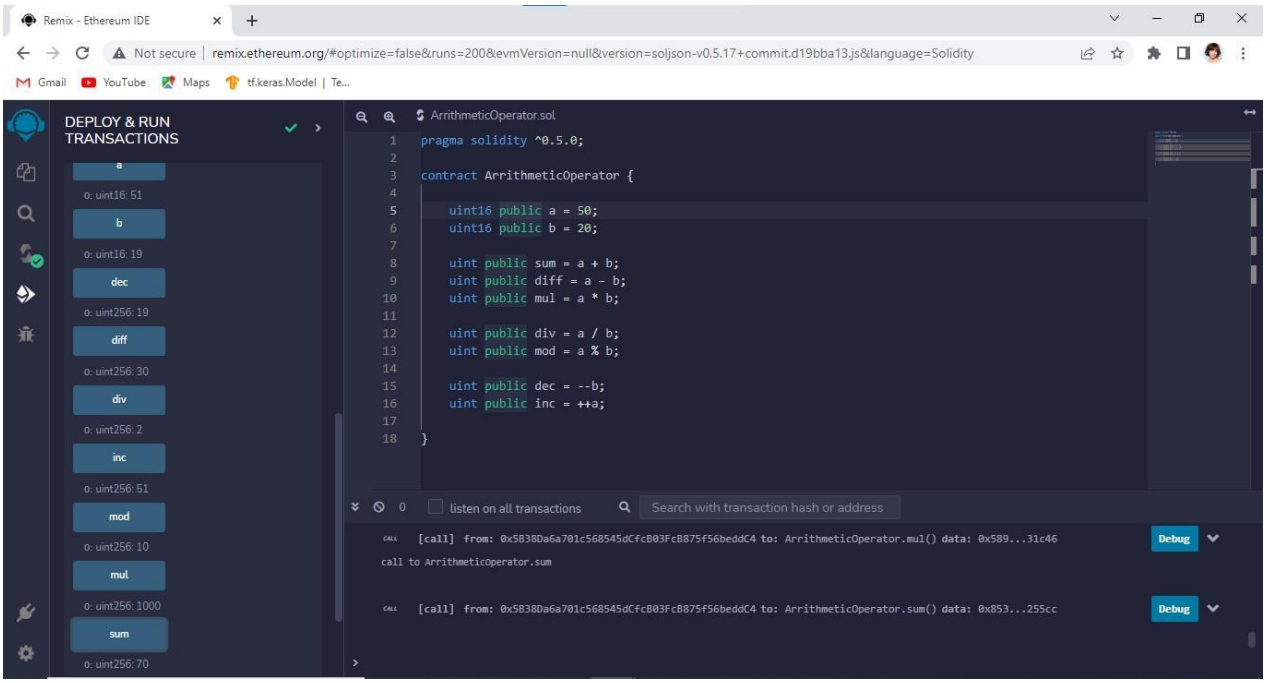
Operators:

Arithmetic Operator

Code:

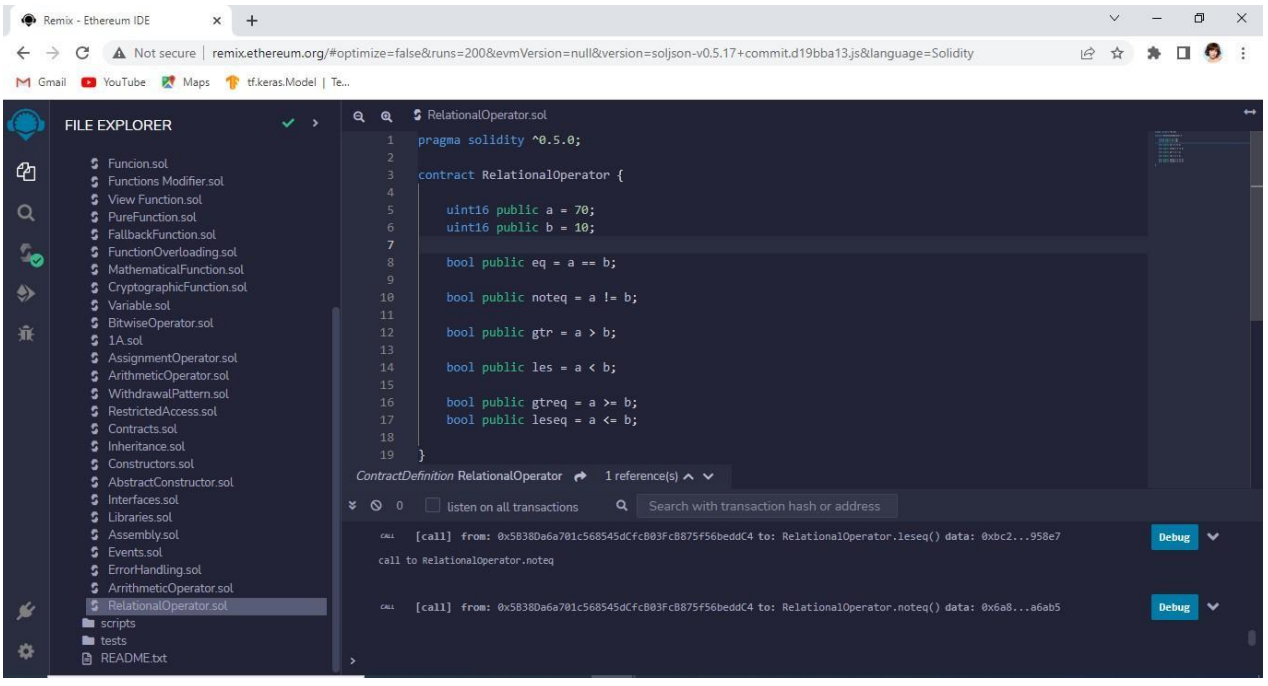


Output:

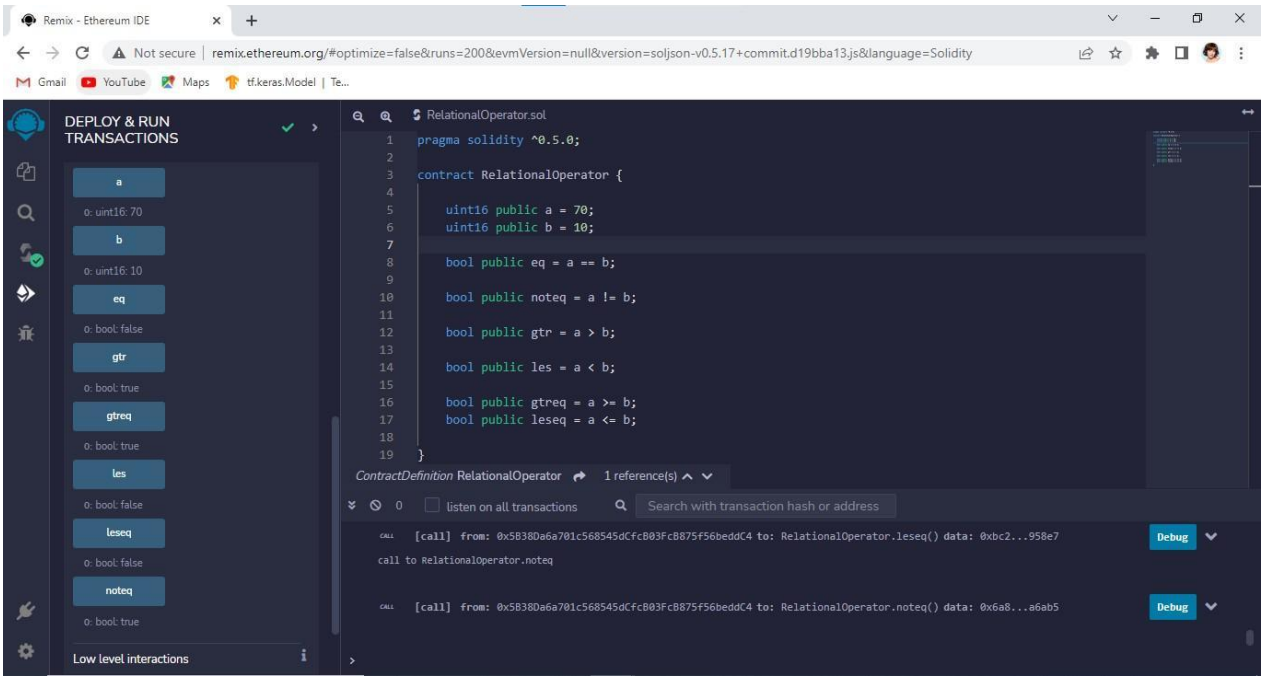


Relational

Operator Code:

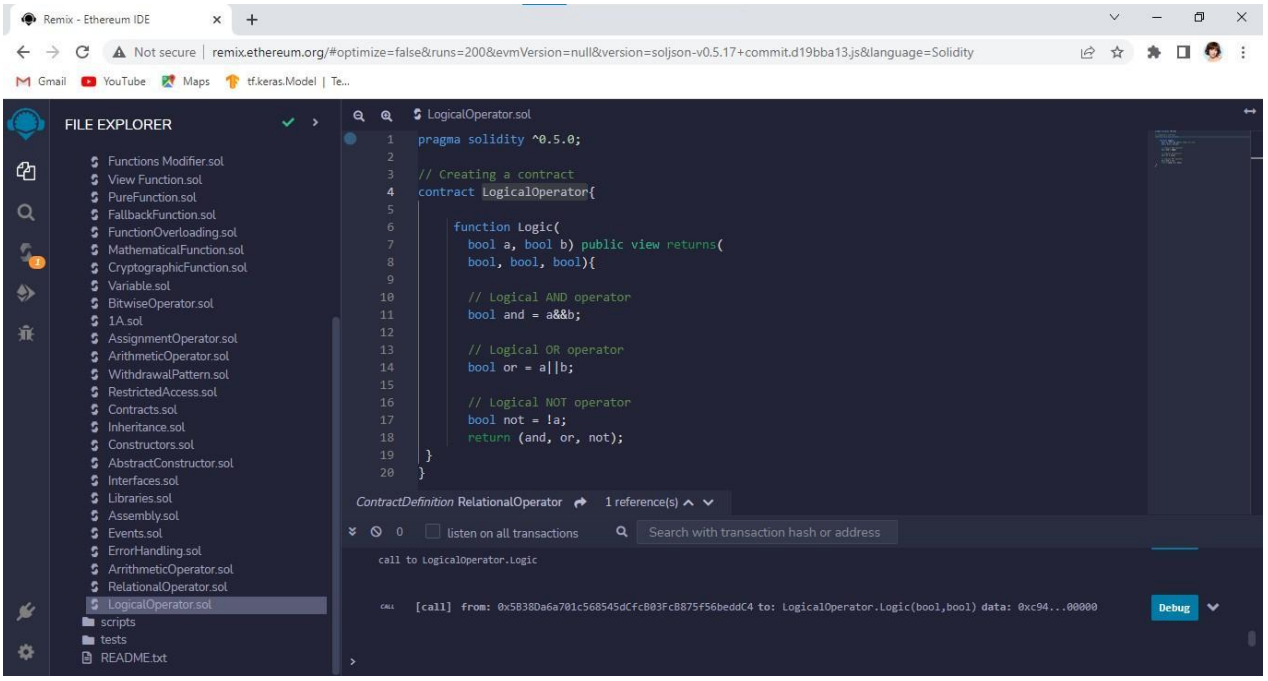


Output:

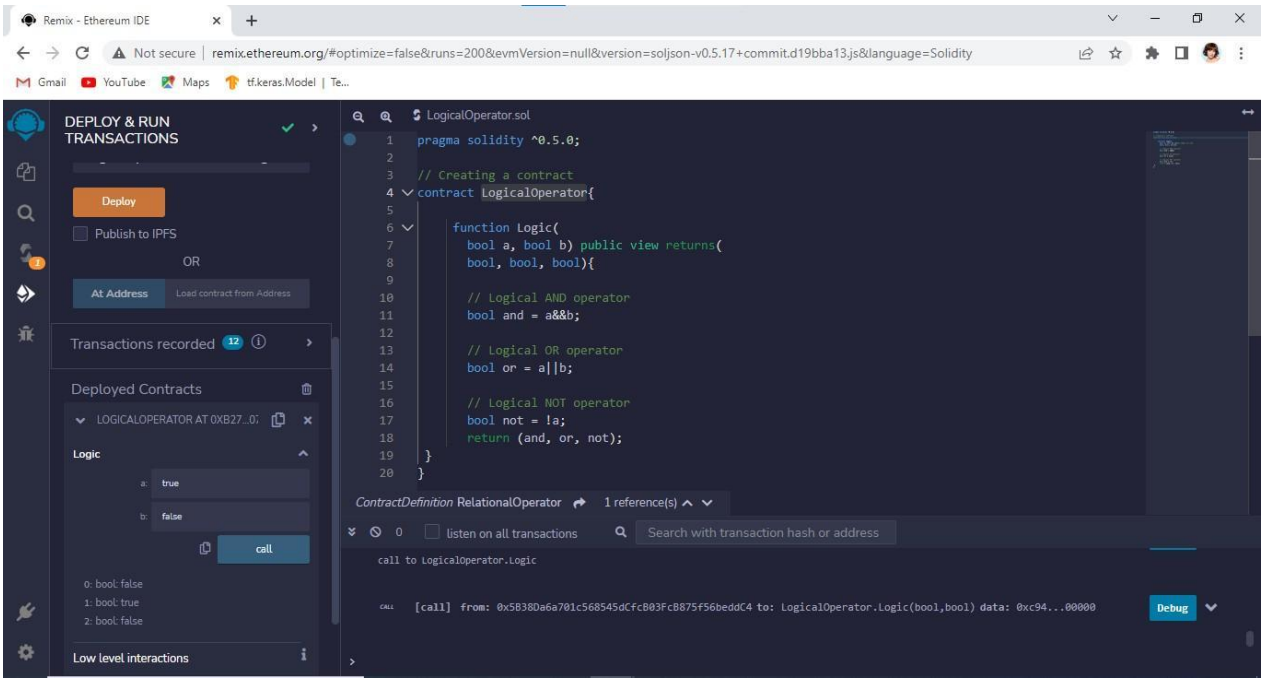


Logical Operator

Code:

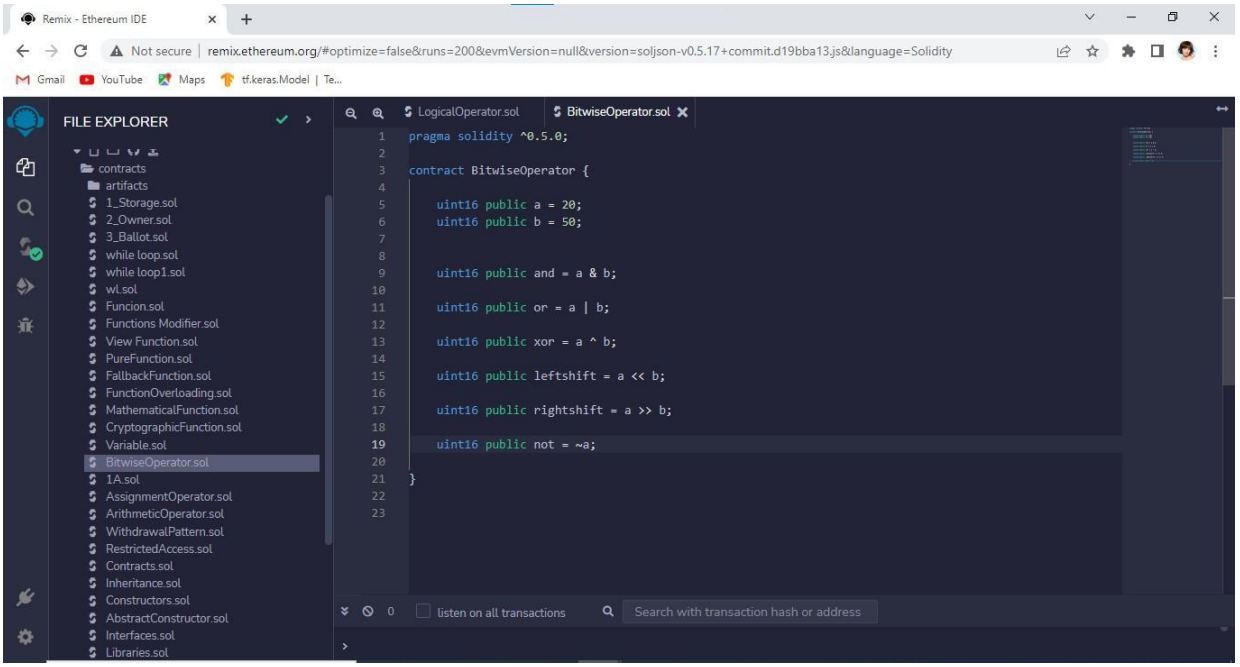


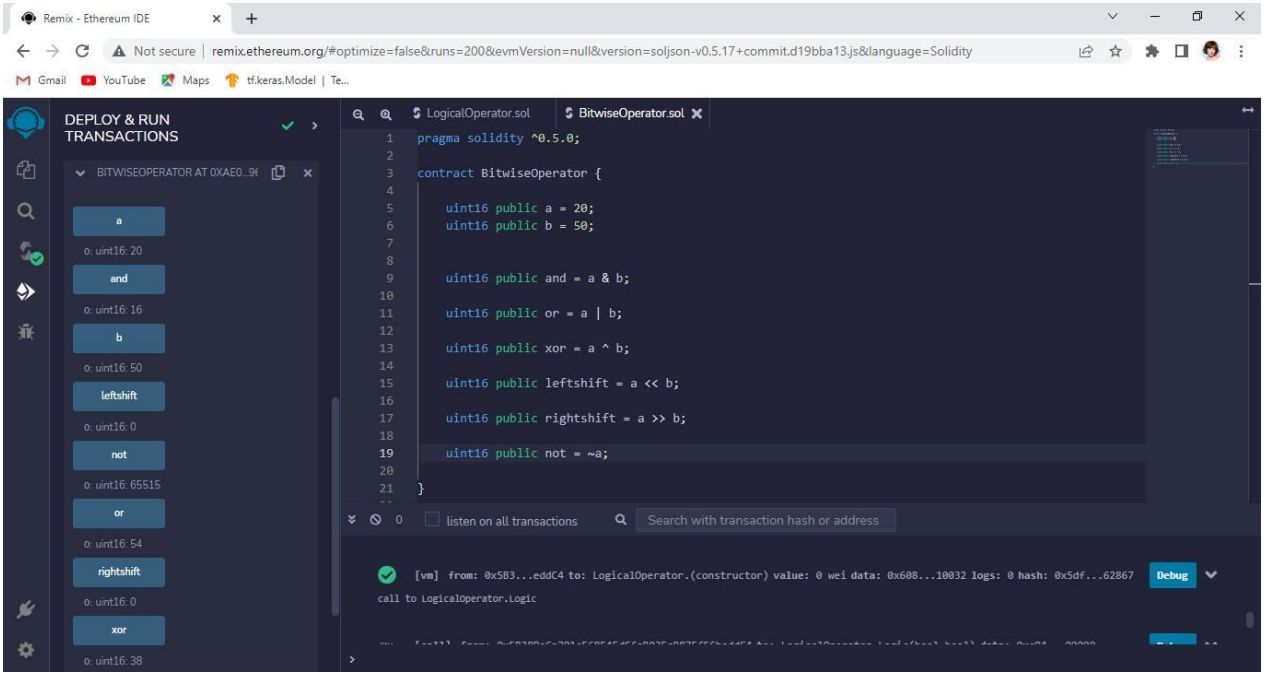
Output:



Bitwise Operator

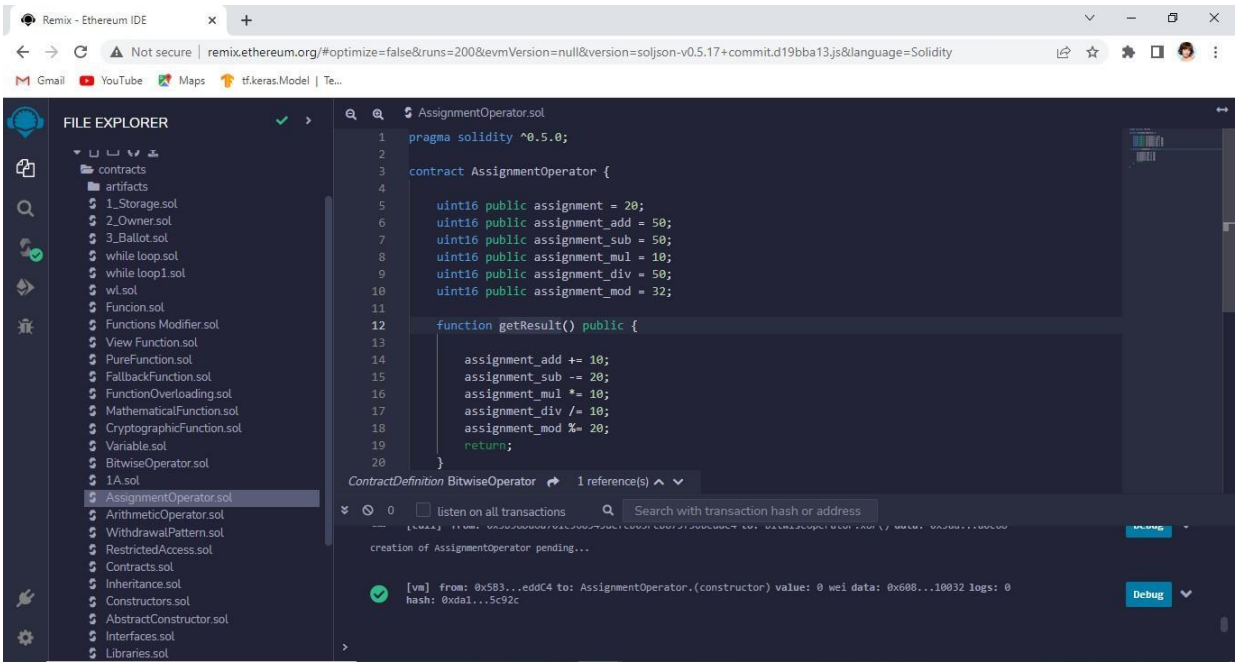
Code:

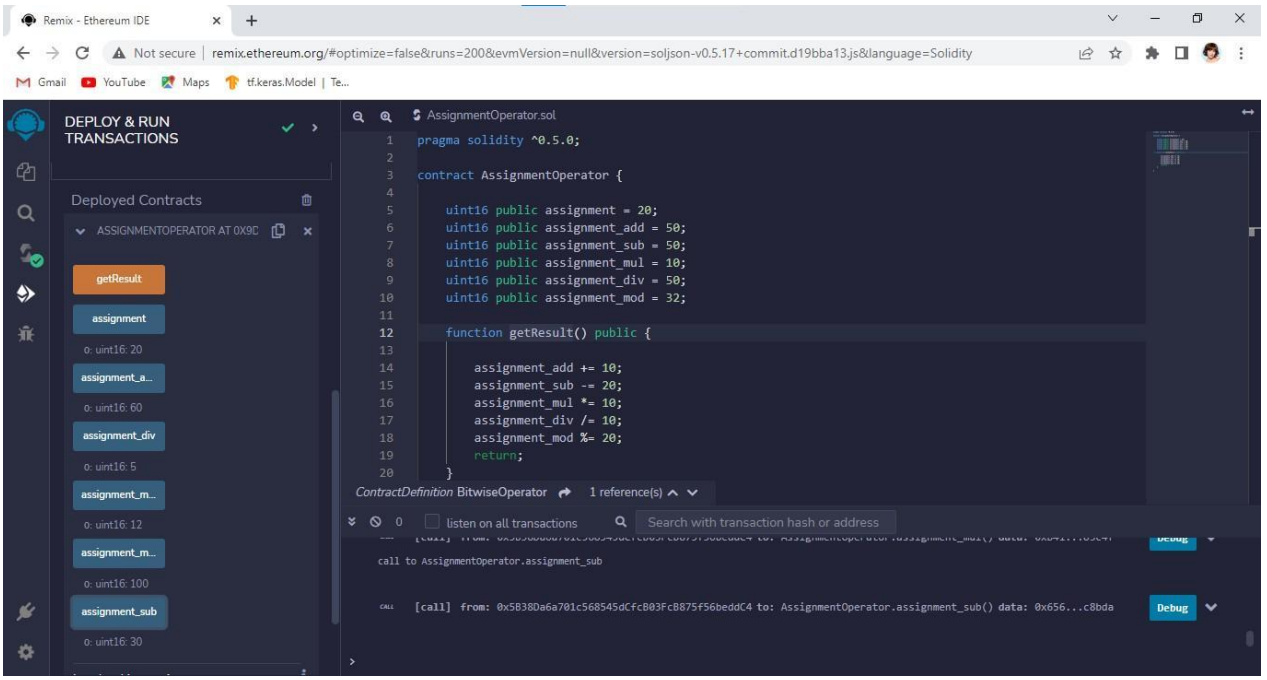




Assignment Operator

Code:

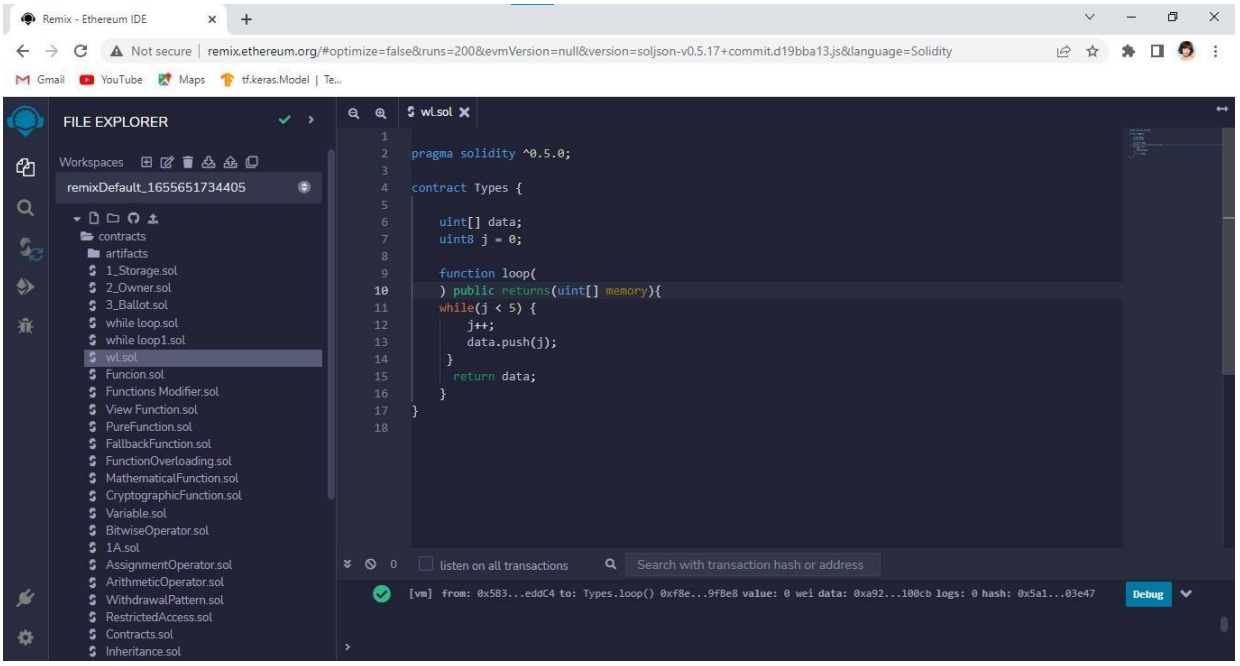


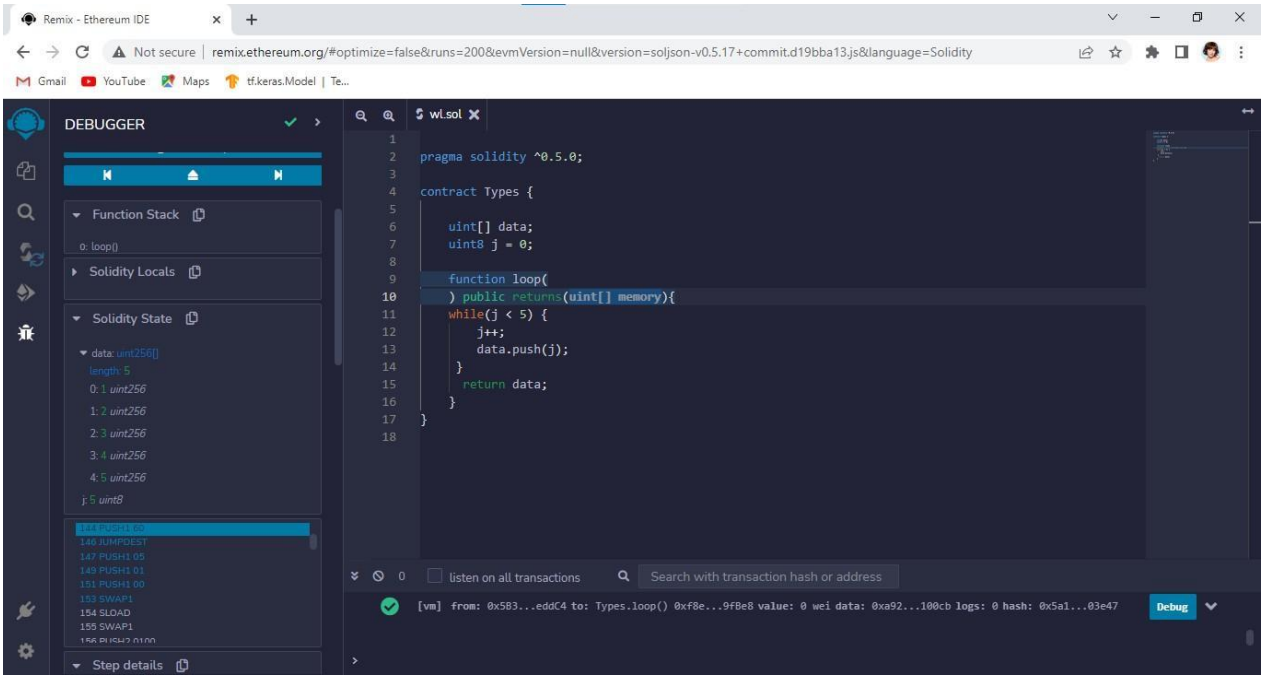


Loops

While Loop

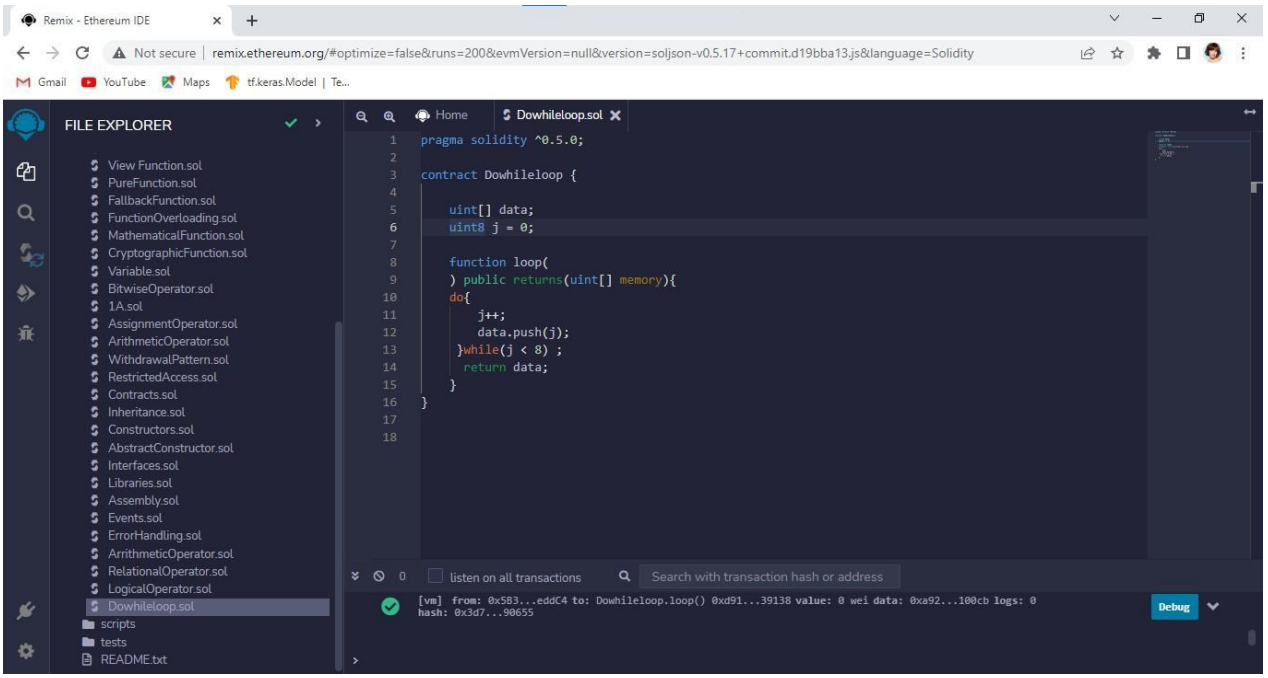
Code:



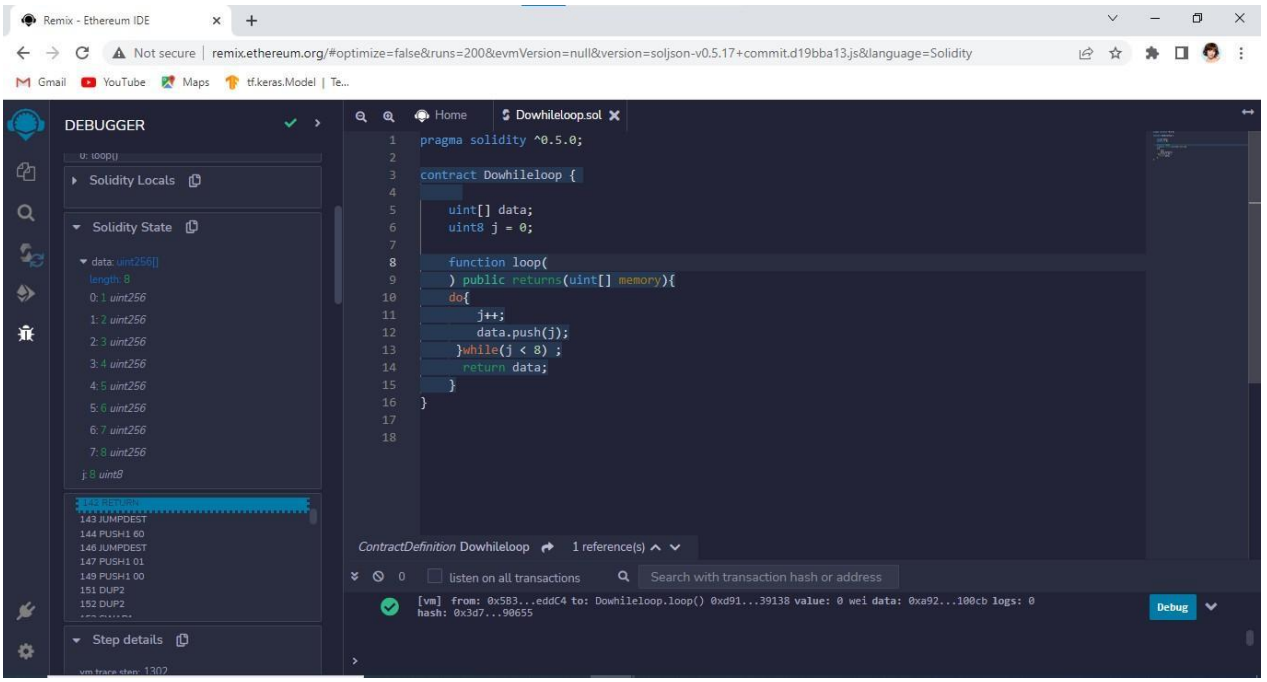


Dowhile loop

Code:

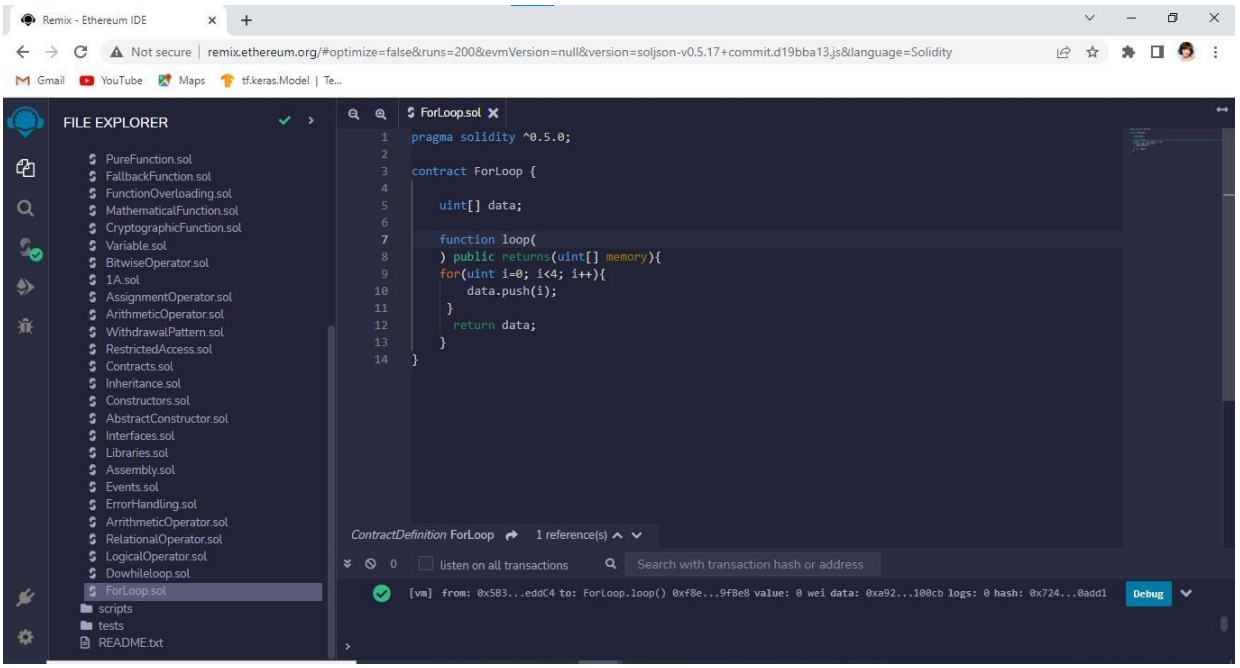


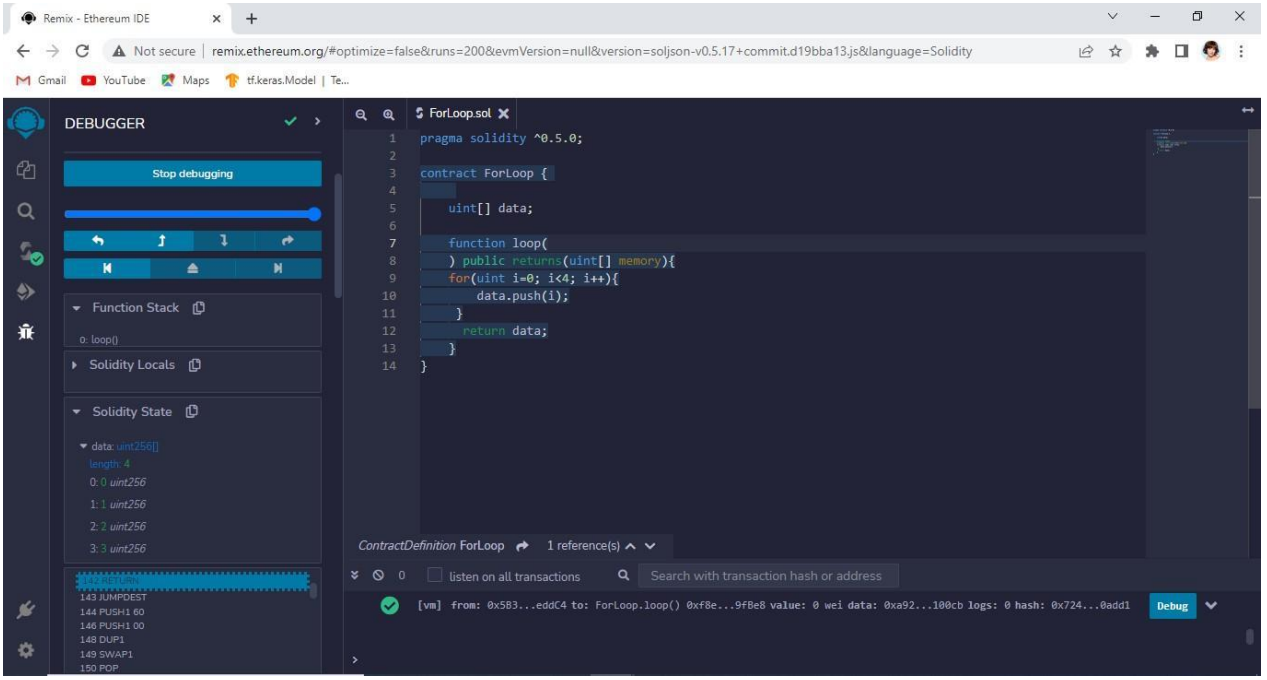
Output:



For loop

Code:

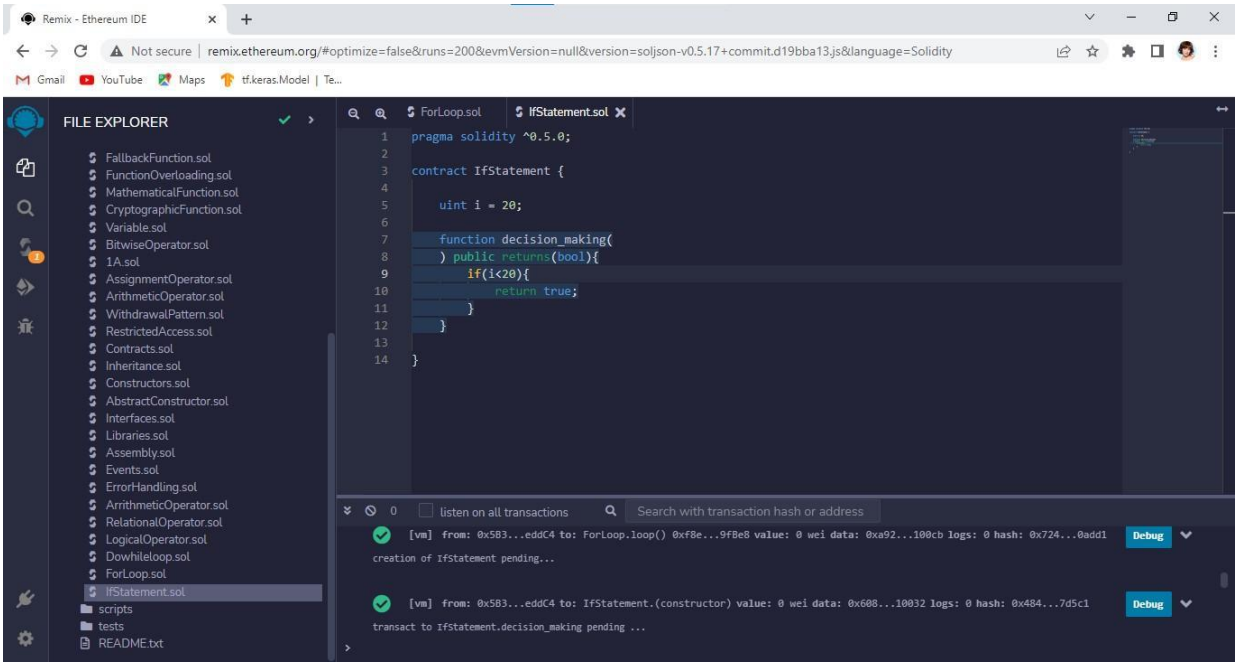


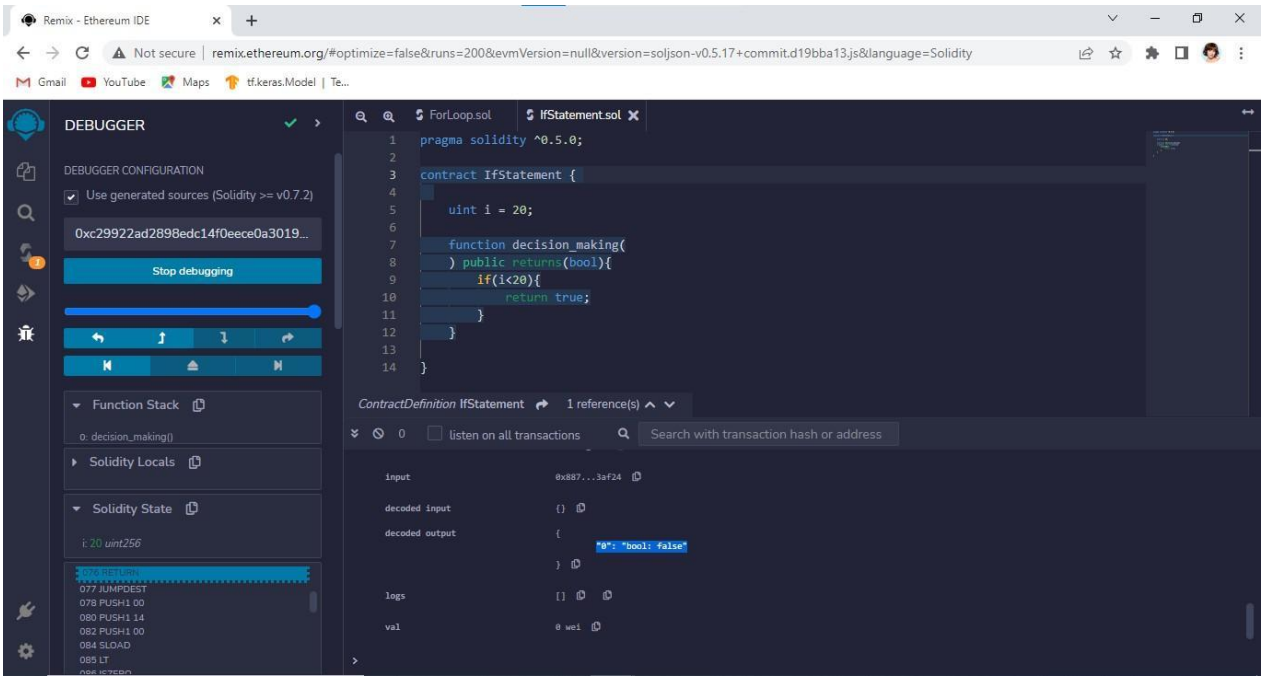


Decision making

If statement

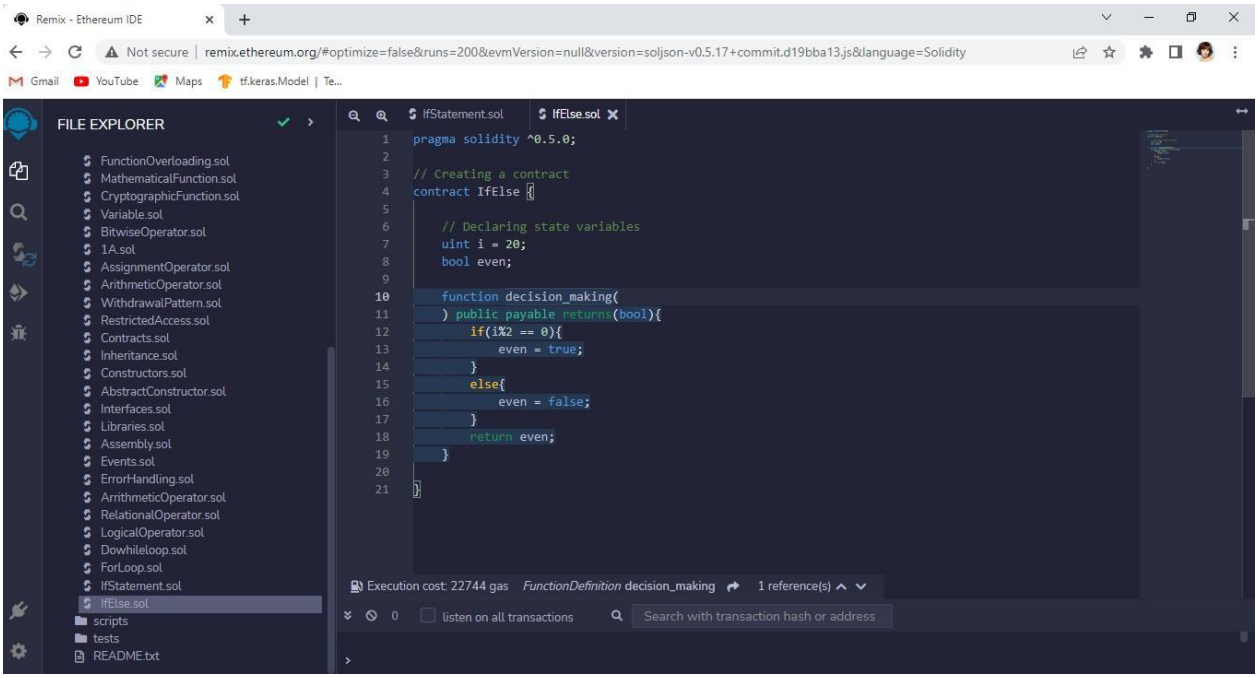
Code:



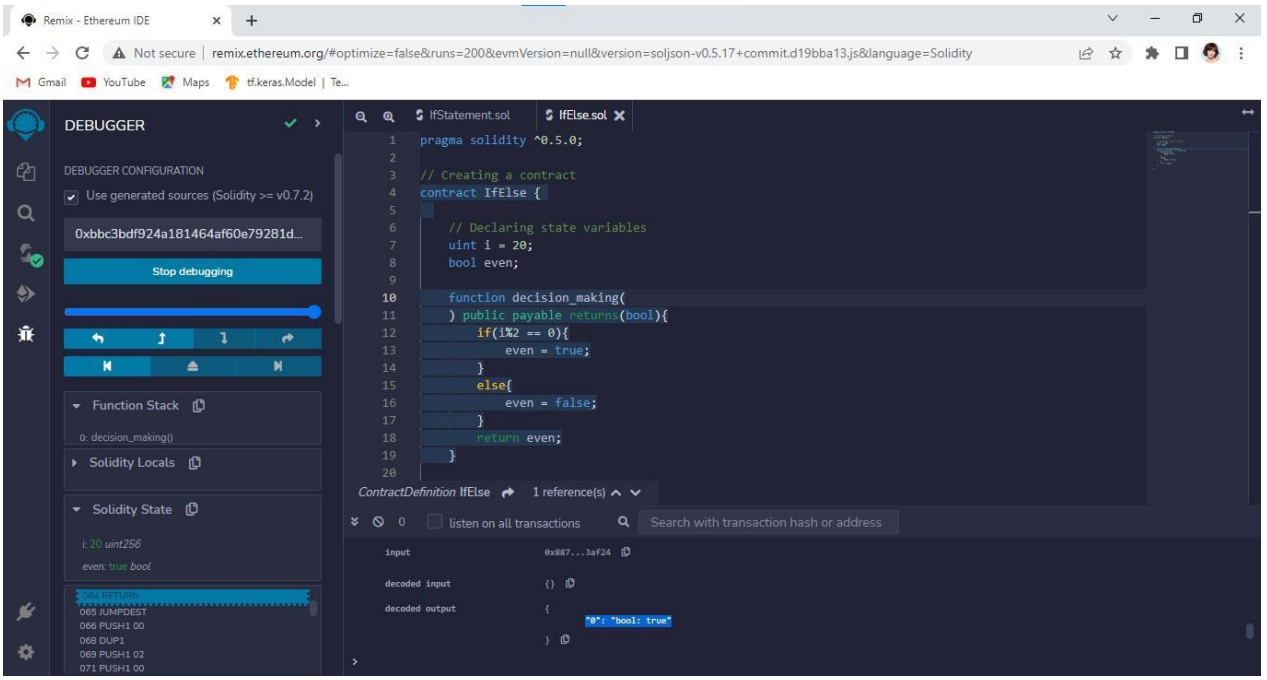


If…else

statement Code:

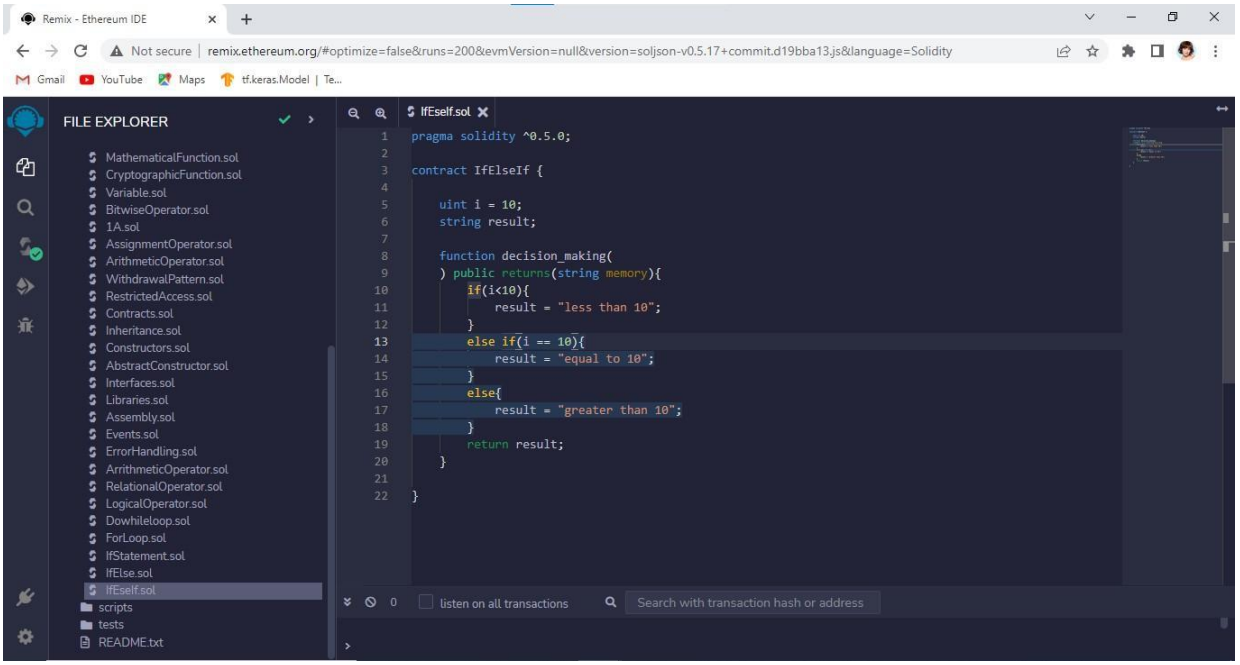


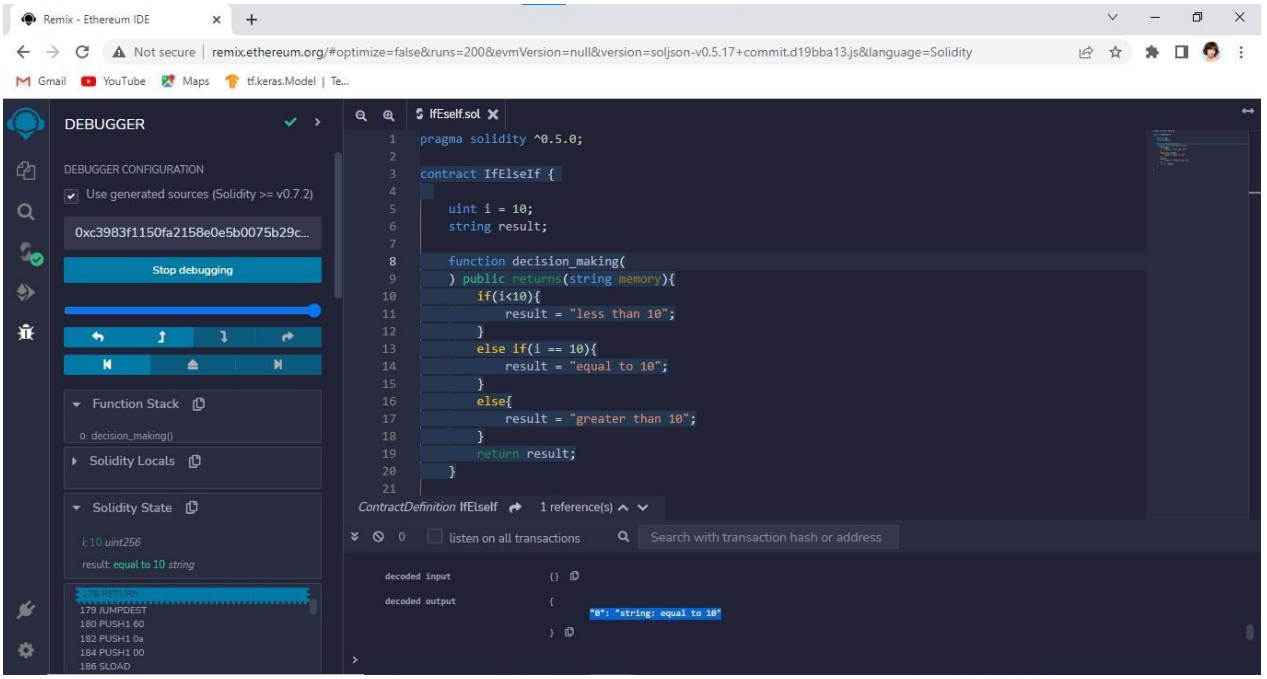
Output:



If…else if…else statement

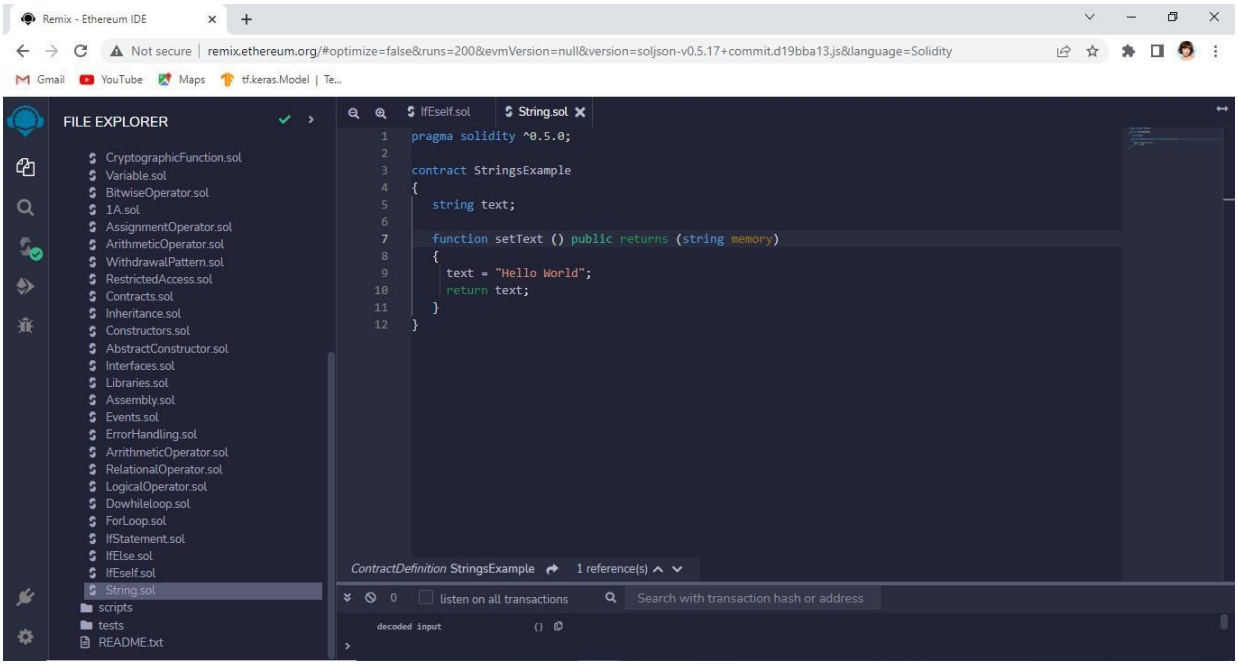
Code:

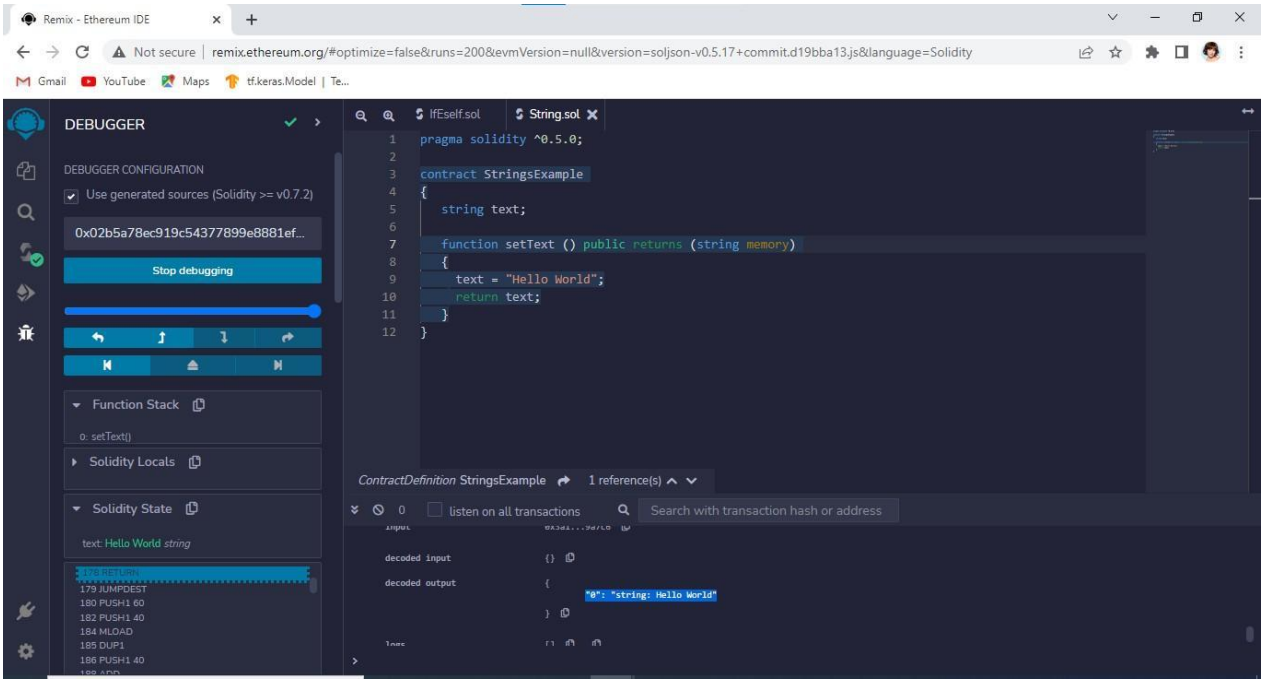



Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEBUGGER
DEBUGGER CONFIGURATION
Use generated sources (Solidity >= v0.7.2)
0xc3983f1150fa2158e0e5b0075b29c...
Stop debugging
Function Stack
0: decision_making()
Solidity Locals
Solidity State
i: 10 uint256
result: equal to 10 string
IfEself.sol
pragma solidity ^0.5.0;
contract IfElseIf {
uint i = 10;
string result;
function decision_making(
) public returns(string memory){
if(i<10){
result = "less than 10";
}
else if(i == 10){
result = "equal to 10";
}
else{
result = "greater than 10";
}
return result;
}
ContractDefinition IfElseIf
1 reference(s)
listen on all transactions
Search with transaction hash or address
decoded input
decoded output
"0": "string: equal to 10"

String

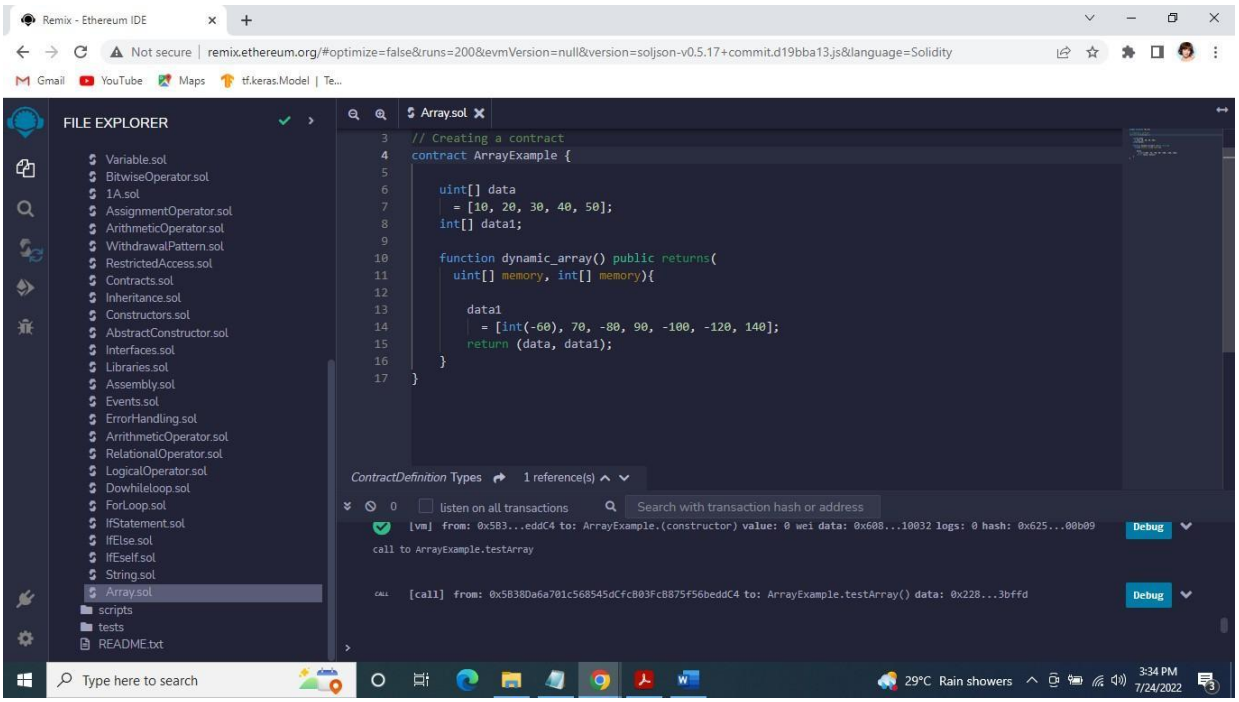
Code:

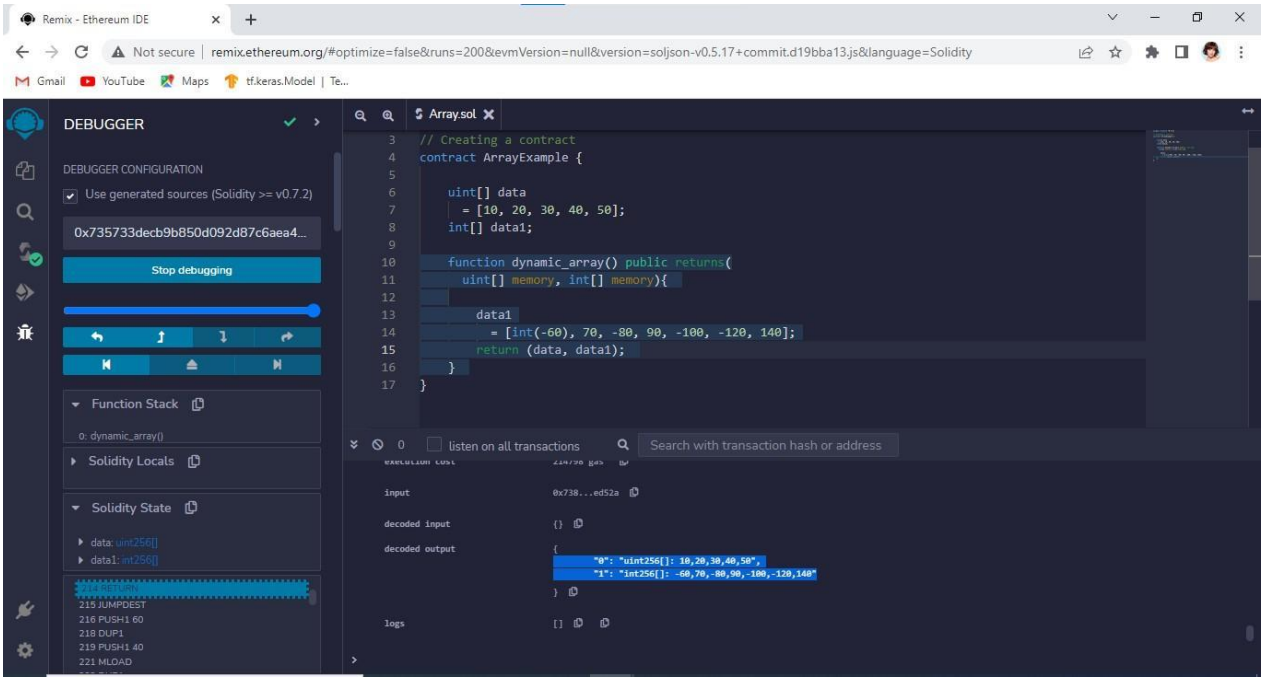

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
scripts
tests
README.txt
IfEself.sol
String.sol
pragma solidity ^0.5.0;
contract StringsExample
{
string text;
function setText () public returns (string memory)
{
text = "Hello World";
return text;
}
}
ContractDefinition StringsExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
decoded input


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEBUGGER
DEBUGGER CONFIGURATION
Use generated sources (Solidity >= v0.7.2)
0x02b5a78ec919c54377899e8881ef...
Stop debugging
Function Stack
0: setText()
Solidity Locals
Solidity State
text: Hello World string
178 RETURN
179 JUMPDEST
180 PUSH1 60
182 PUSH1 40
184 MLOAD
185 DUP1
186 PUSH1 40
IfEself.sol
String.sol
1 pragma solidity ^0.5.0;
2
3 contract StringsExample
4 {
5 string text;
6
7 function setText () public returns (string memory)
8 {
9 text = "Hello World";
10 return text;
11 }
12 }
ContractDefinition StringsExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
decoded input
decoded output
"0": "string: Hello World"

Array

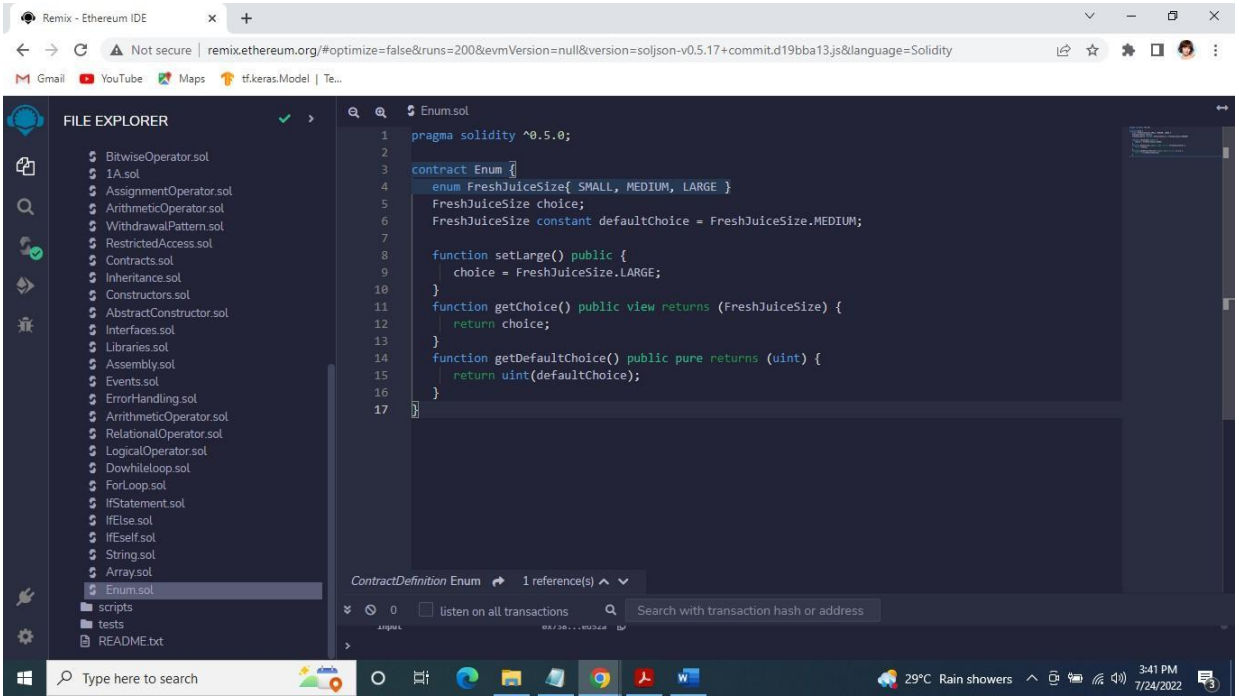
Code:

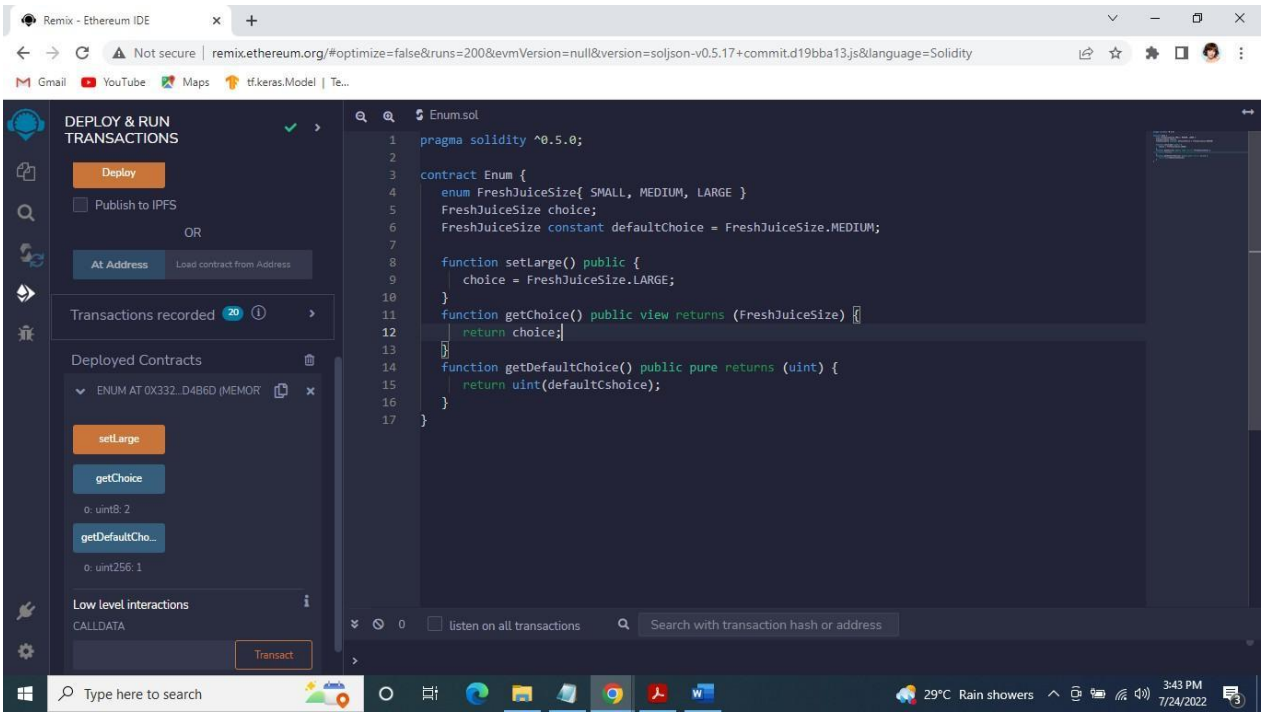

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
scripts
tests
README.txt
Array.sol
3 // Creating a contract
4 contract ArrayExample {
5
6 uint[] data
7 = [10, 20, 30, 40, 50];
8 int[] data1;
9
10 function dynamic_array() public returns(
11 uint[] memory, int[] memory){
12
13 data1
14 = [int(-60), 70, -80, 90, -100, -120, 140];
15 return (data, data1);
16 }
17 }
ContractDefinition Types
1 reference(s)
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: ArrayExample.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0x625...00b09
Debug
call to ArrayExample.testArray
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ArrayExample.testArray() data: 0x228...3bffd
Debug
Type here to search
29°C Rain showers
3:34 PM
7/24/2022


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEBUGGER
DEBUGGER CONFIGURATION
Use generated sources (Solidity >= v0.7.2)
0x735733decb9b850d092d87c6aea4...
Stop debugging
Function Stack
0: dynamic_array()
Solidity Locals
Solidity State
data: uint256[]
data1: int256[]
214 RETURN
215 JUMPDEST
216 PUSH1 60
218 DUP1
219 PUSH1 40
221 MLOAD
Array.sol
// Creating a contract
contract ArrayExample {
uint[] data
= [10, 20, 30, 40, 50];
int[] data1;
function dynamic_array() public returns(
uint[] memory, int[] memory){
data1
= [int(-60), 70, -80, 90, -100, -120, 140];
return (data, data1);
}
}
listen on all transactions
Search with transaction hash or address
input
0x738...ed52a
decoded input
{}
decoded output
{
"0": "uint256[]: 10,20,30,40,50",
"1": "int256[]: -60,70,-80,90,-100,-120,140"
}
logs
[]

Enums

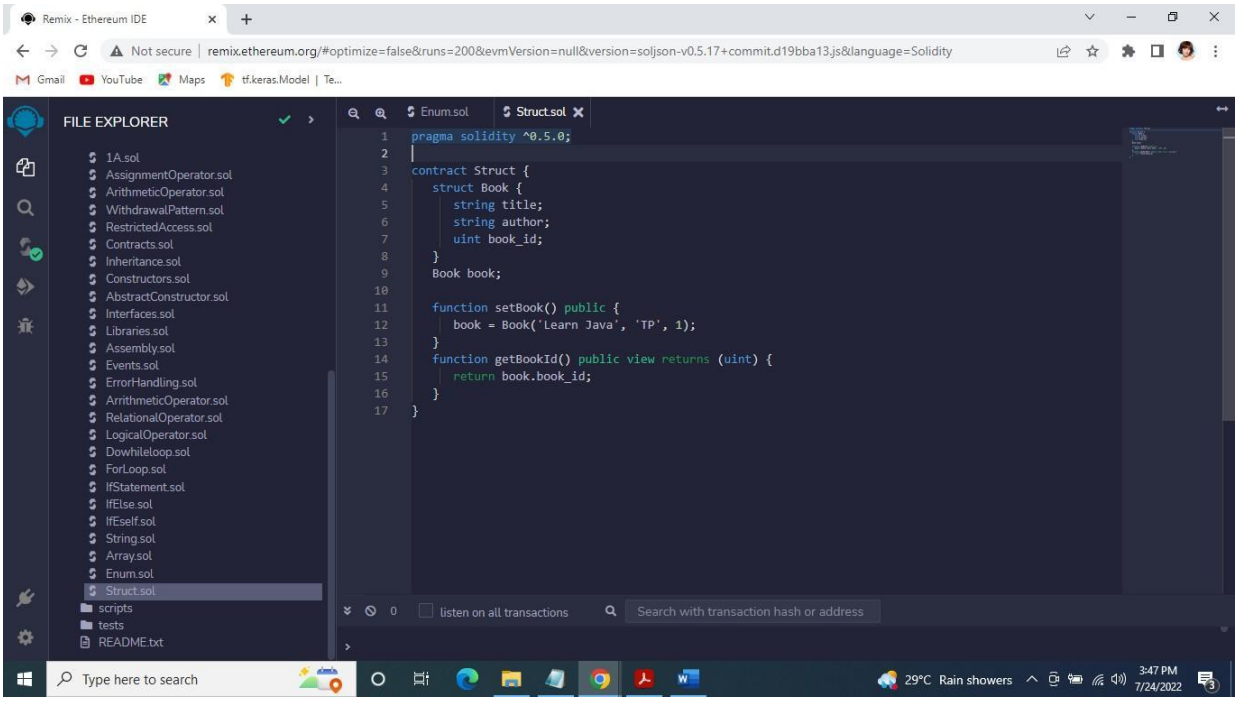
Code:

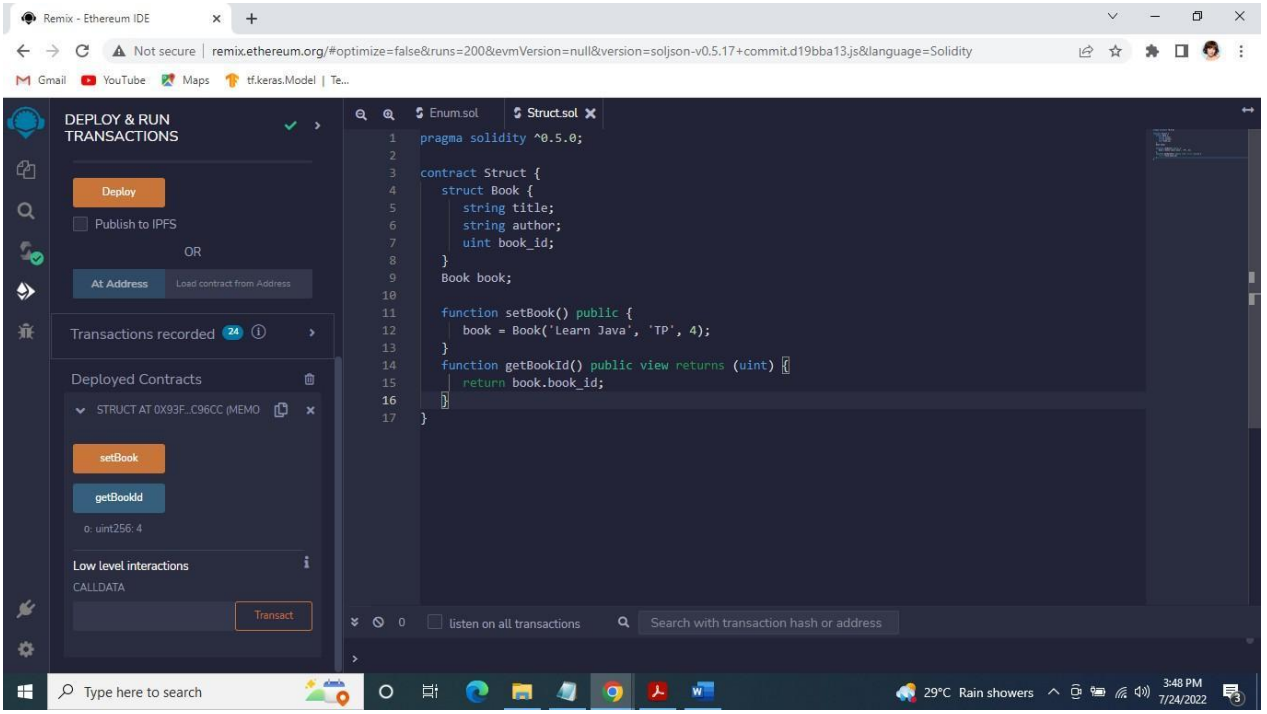

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
Enum.sol
scripts
tests
README.txt
Enum.sol
pragma solidity ^0.5.0;
contract Enum {
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);
}
}
ContractDefinition Enum
1 reference(s)
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:41 PM
7/24/2022


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 20
Deployed Contracts
ENUM AT 0X332...D4B6D (MEMOR
setLarge
getChoice
0: uint8: 2
getDefaultCho...
0: uint256: 1
Low level interactions
CALLDATA
Transact
Enum.sol
pragma solidity ^0.5.0;
contract Enum {
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(defaultCshoice);
}
}
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:43 PM
7/24/2022

Struct

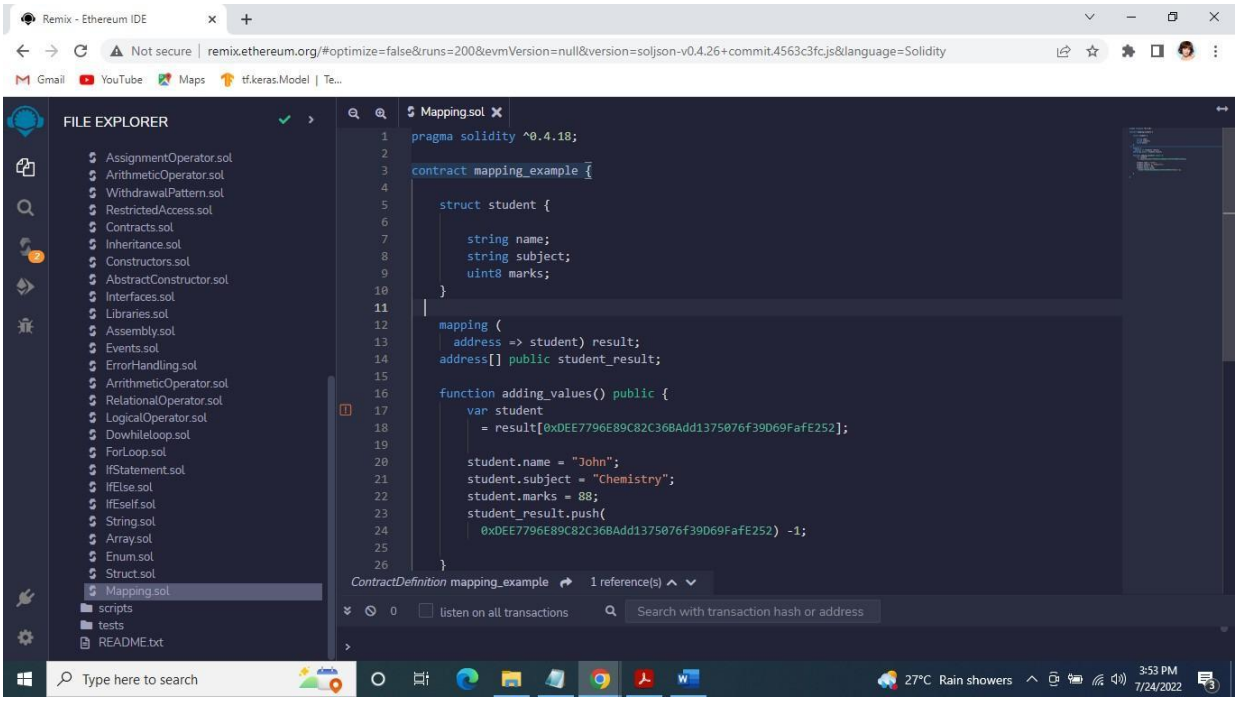
Code:

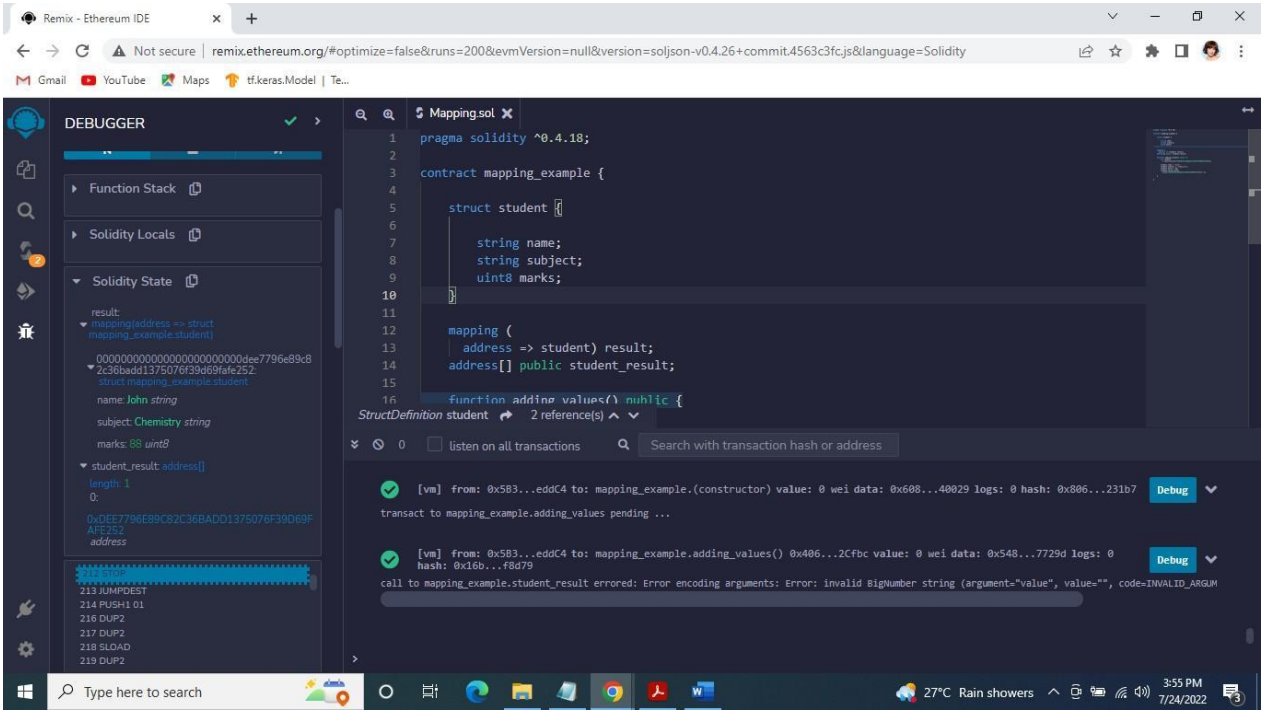

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
ErrorHandling.sol
ArrithmeticOperator.sol
RelationalOperator.sol
LogicalOperator.sol
Dowhileloop.sol
ForLoop.sol
IfStatement.sol
IfElse.sol
IfEself.sol
String.sol
Array.sol
Enum.sol
Struct.sol
scripts
tests
README.txt
Enum.sol
Struct.sol
pragma solidity ^0.5.0;
contract Struct {
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;
}
}
listen on all transactions
Search with transaction hash or address
Type here to search
29°C Rain showers
3:47 PM
7/24/2022



Mapping

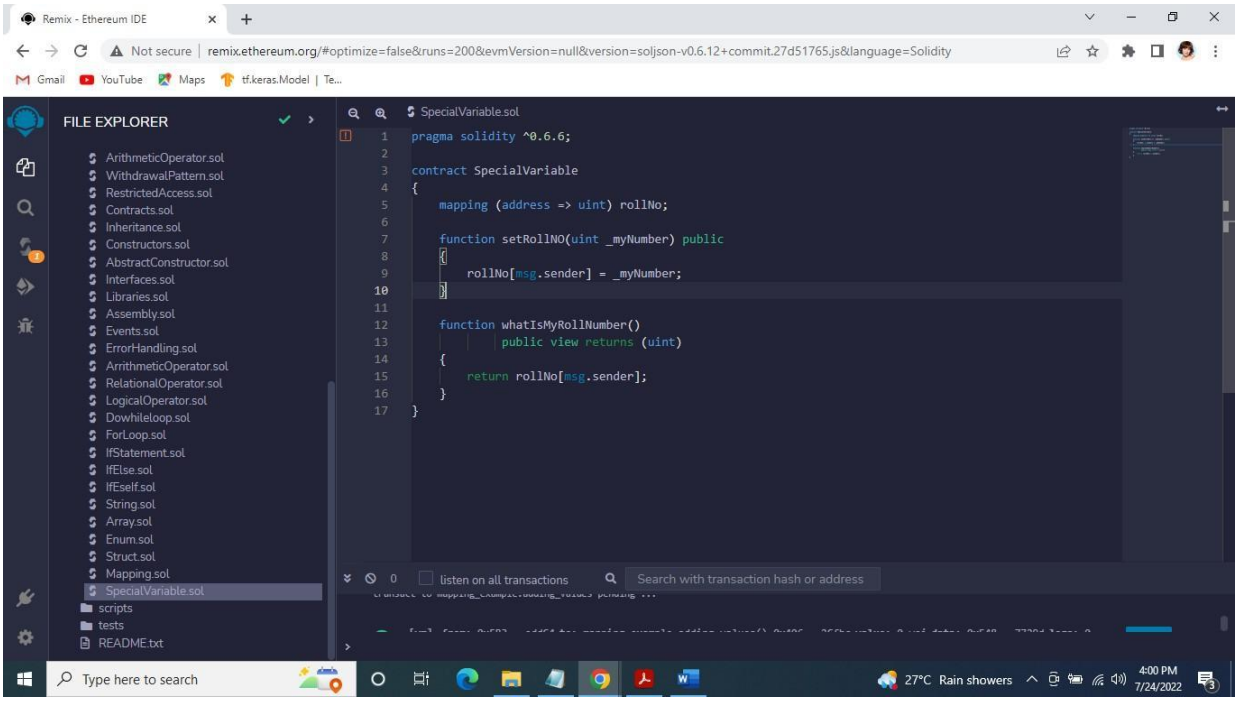
Code:

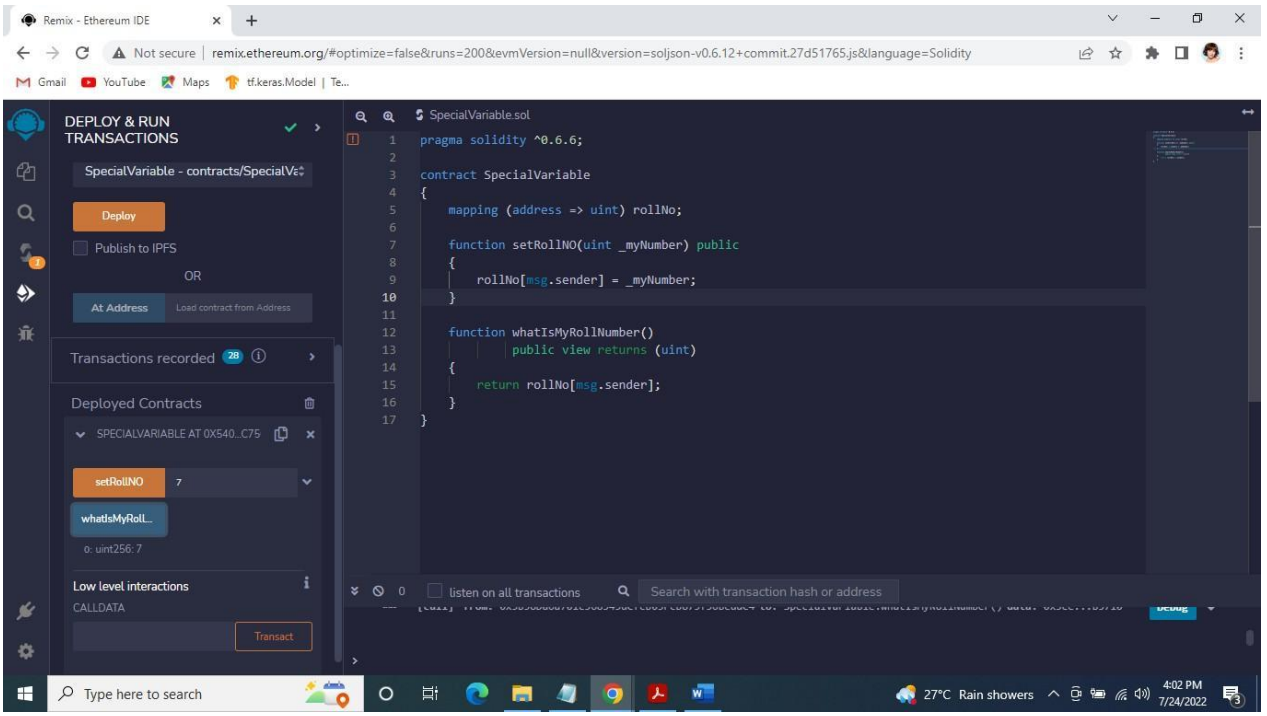




Special Variable

Code:

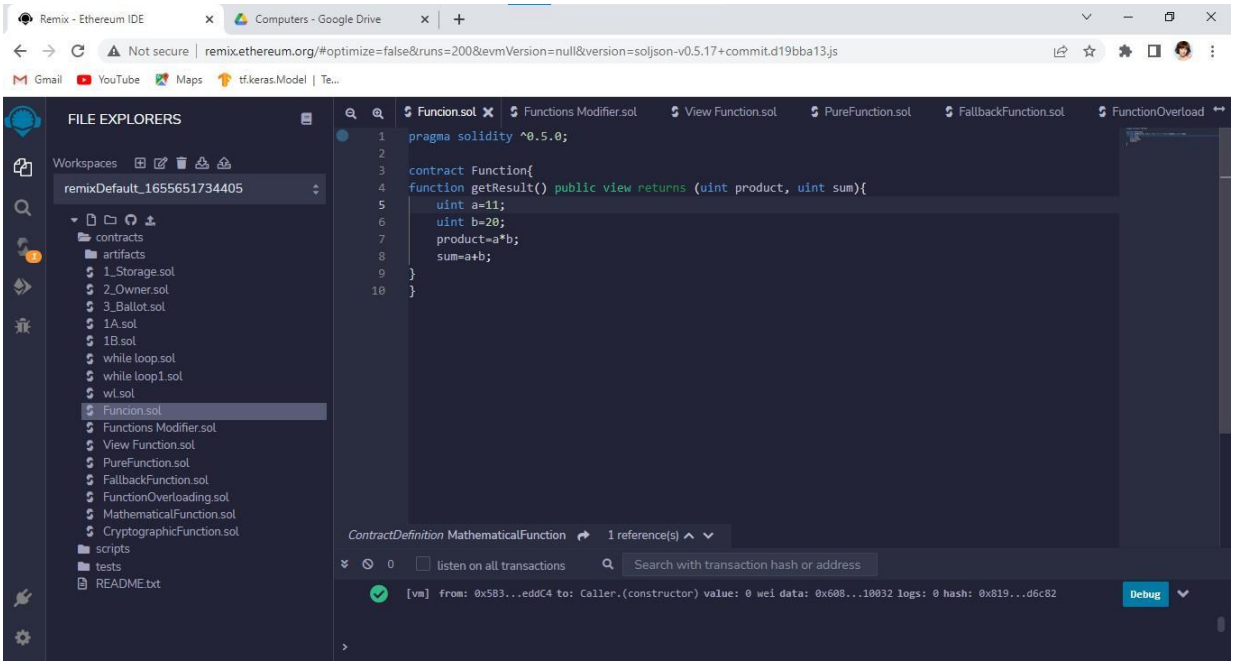


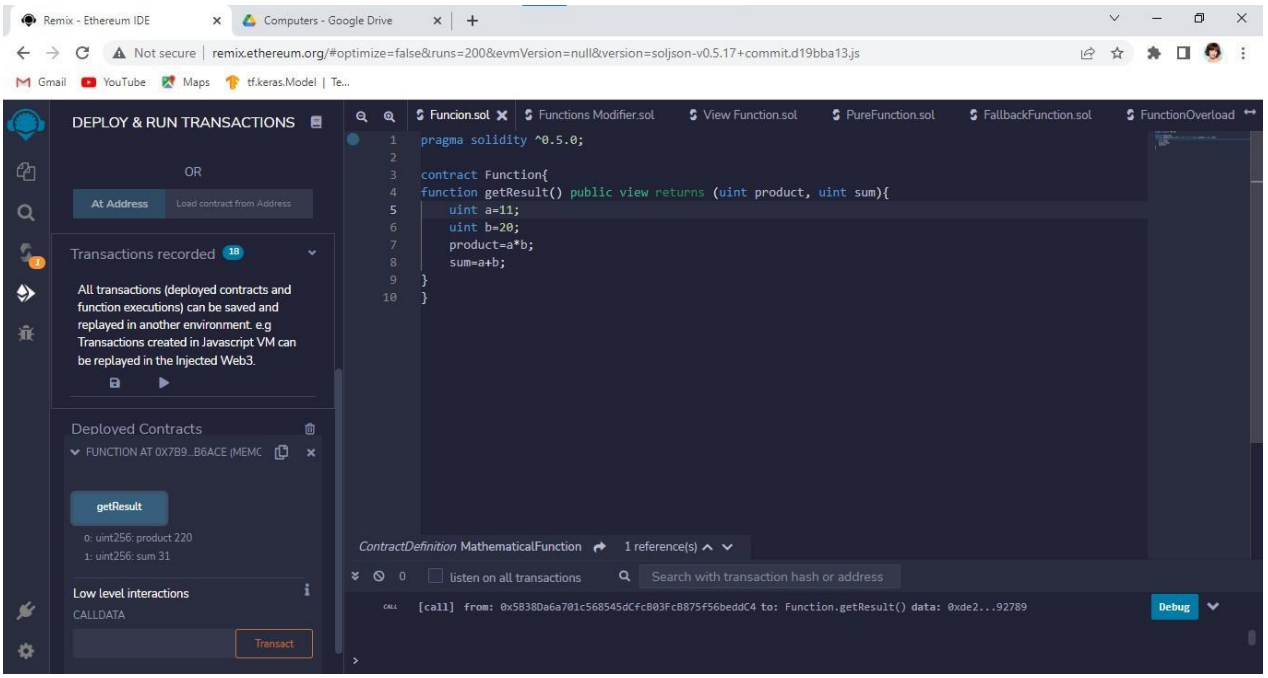


2b) Functions, Function Modifiers, View functions, Pure Functions, Fallback Function, Function Overloading, Mathematical functions, Cryptographic functions.

Function:

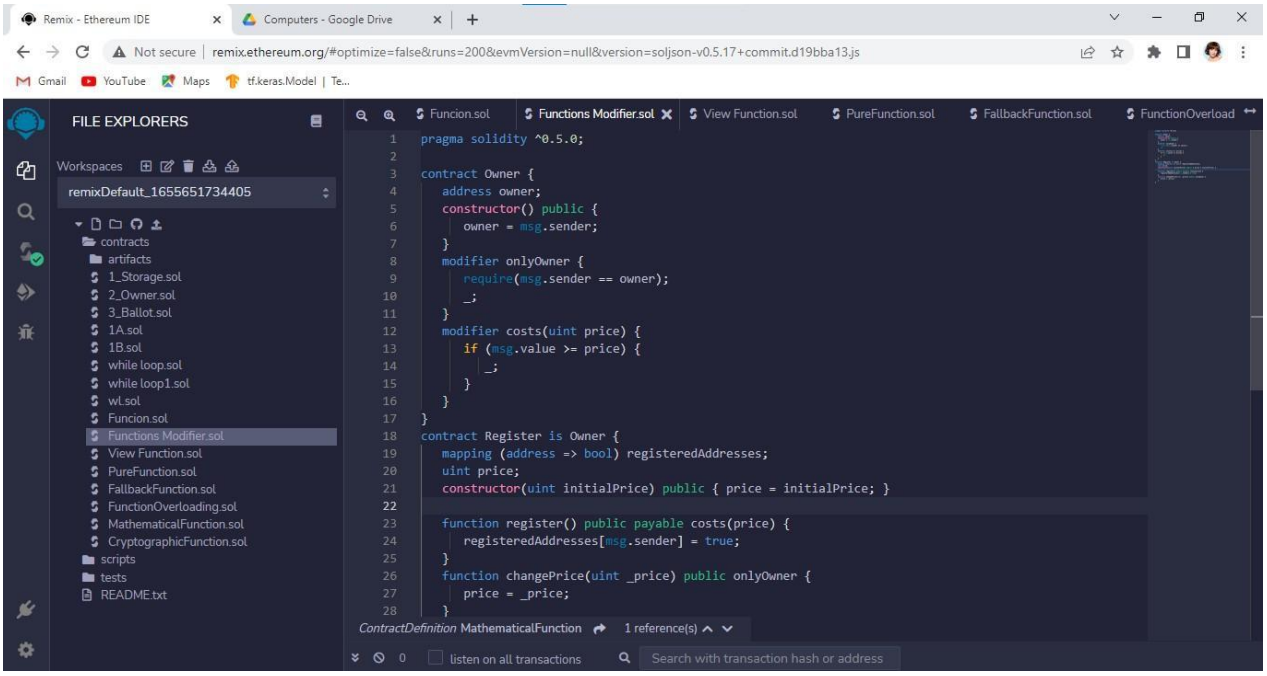
Code:



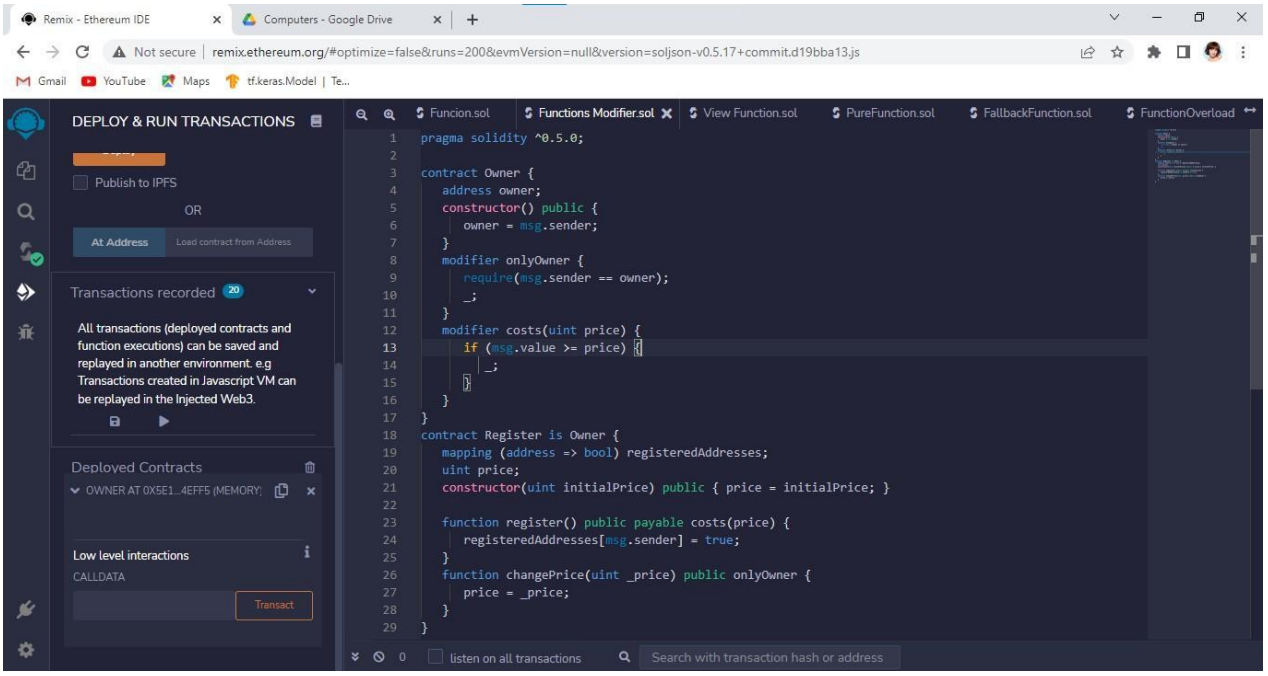


Functions Modifiers

Code:

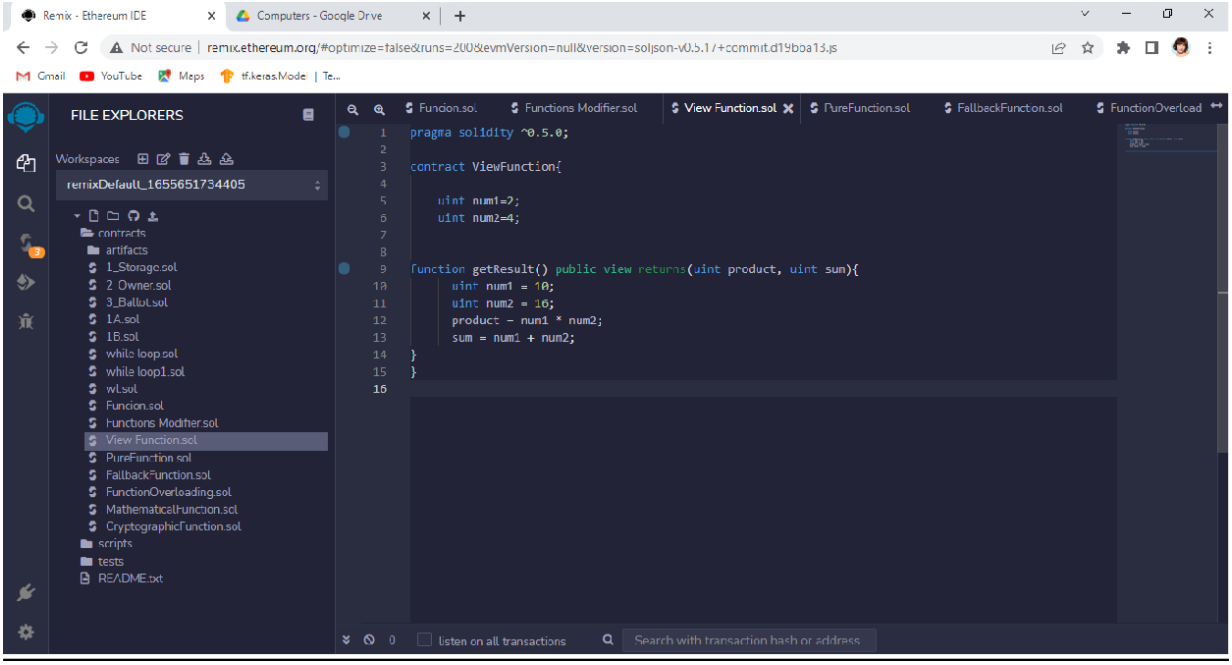


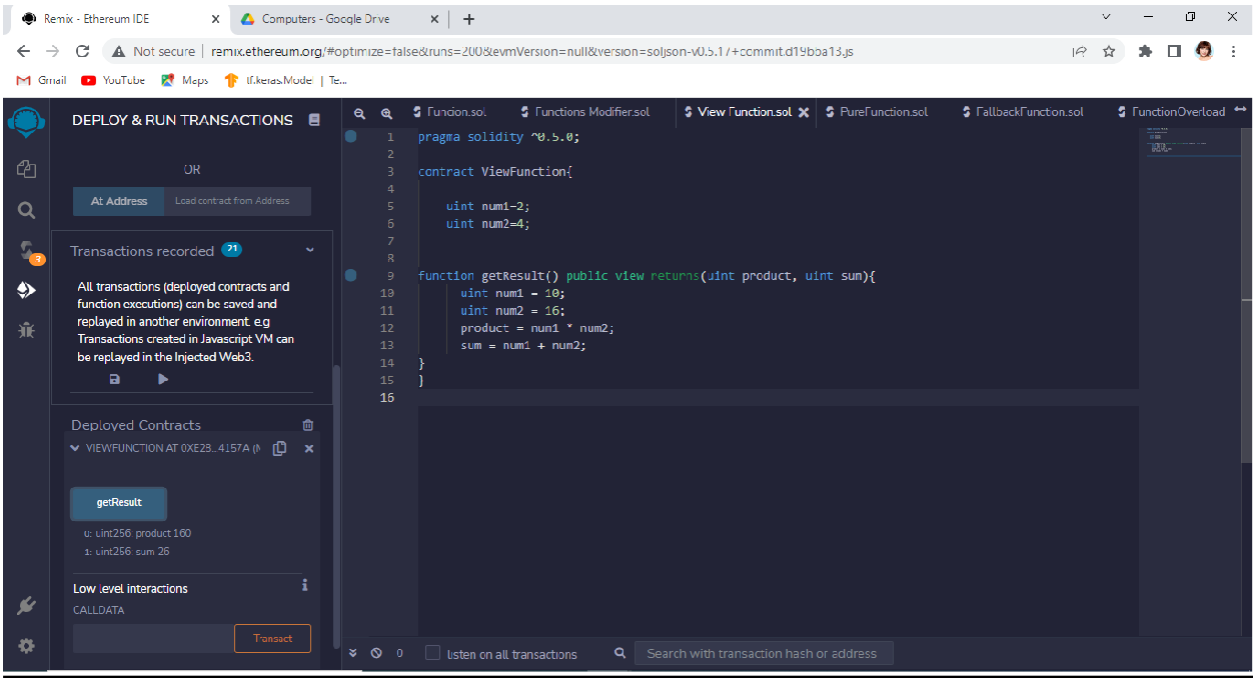
Output:



View function

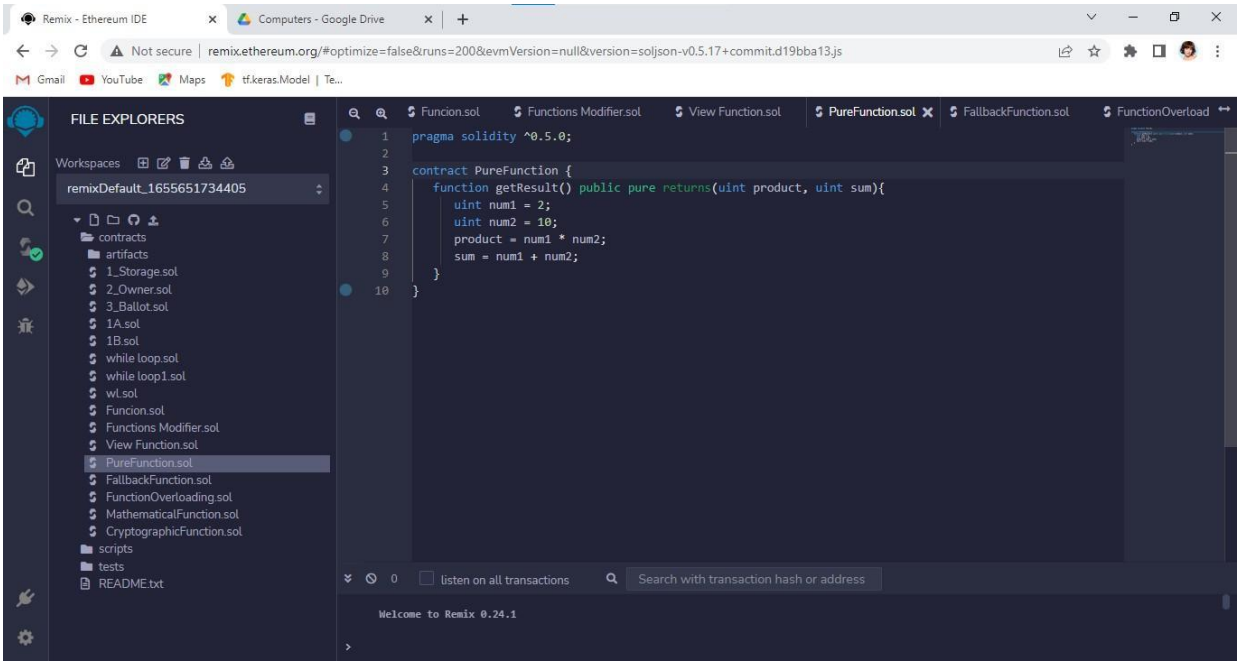
Code:

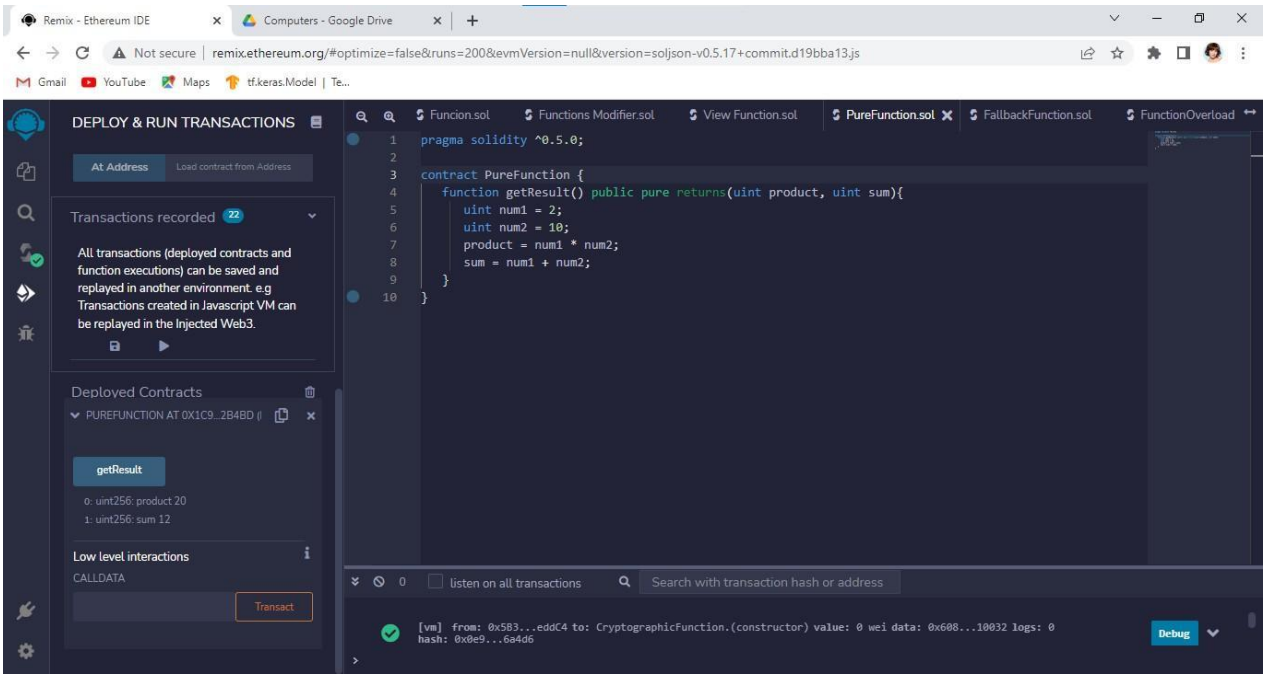




Pure function

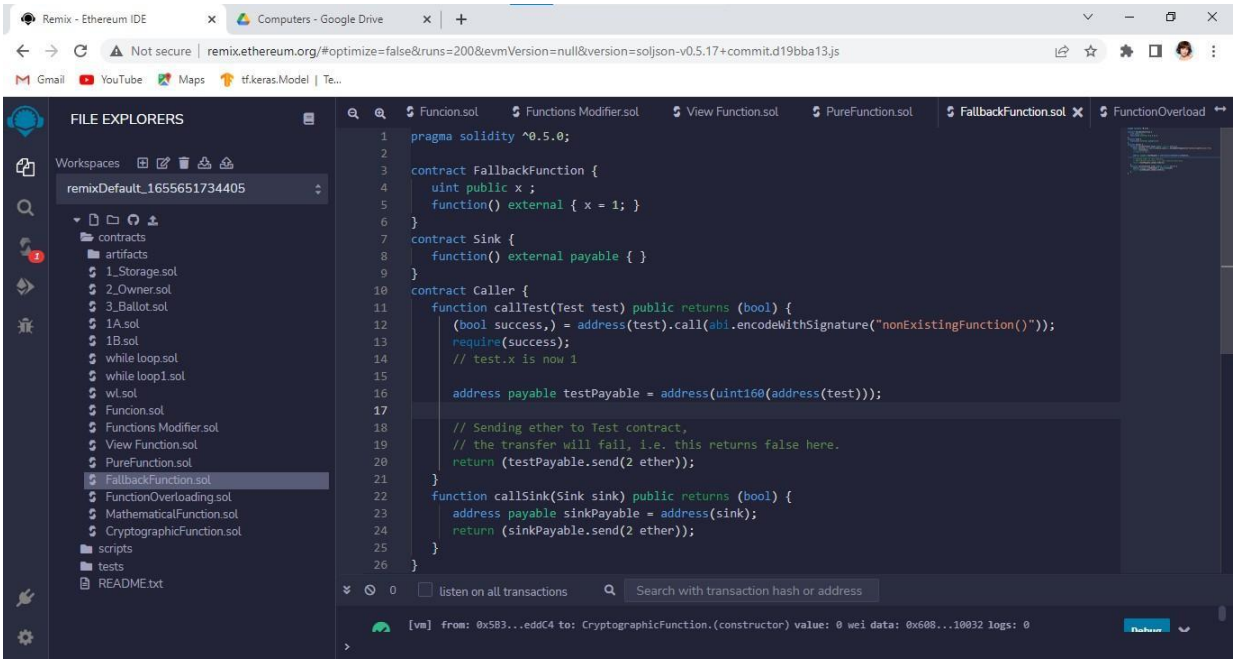
Code:

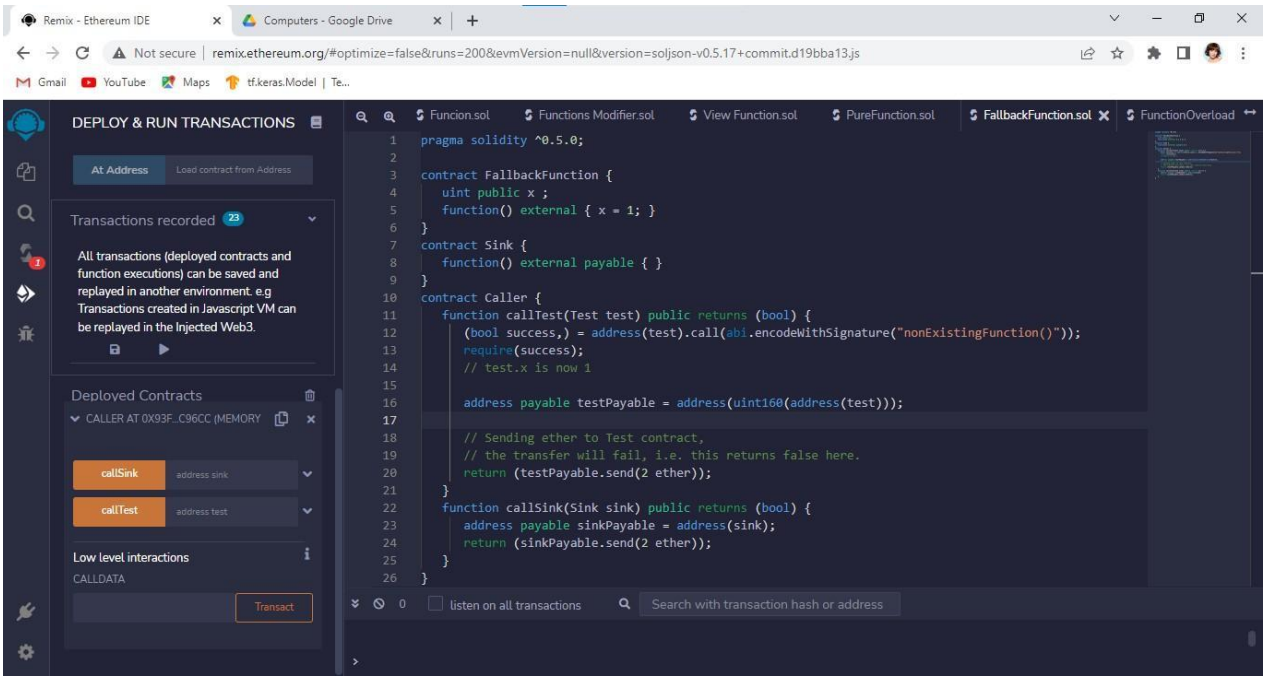




Fallback function

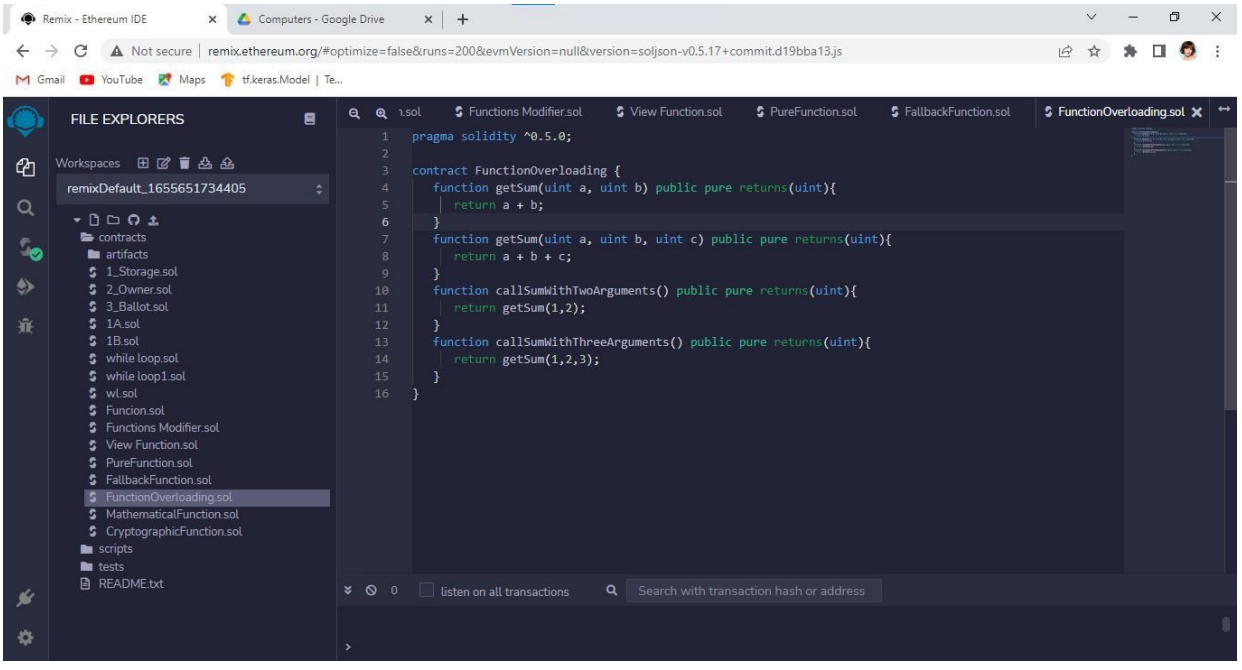
Code:

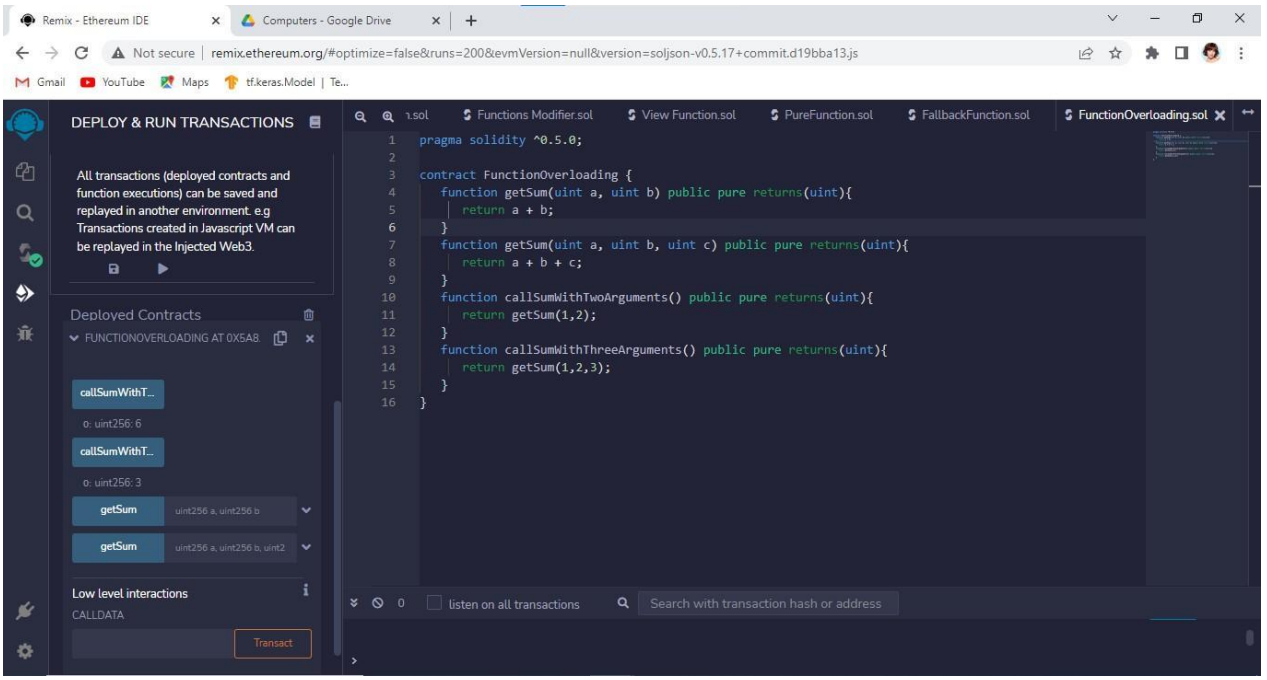




Function Overloading

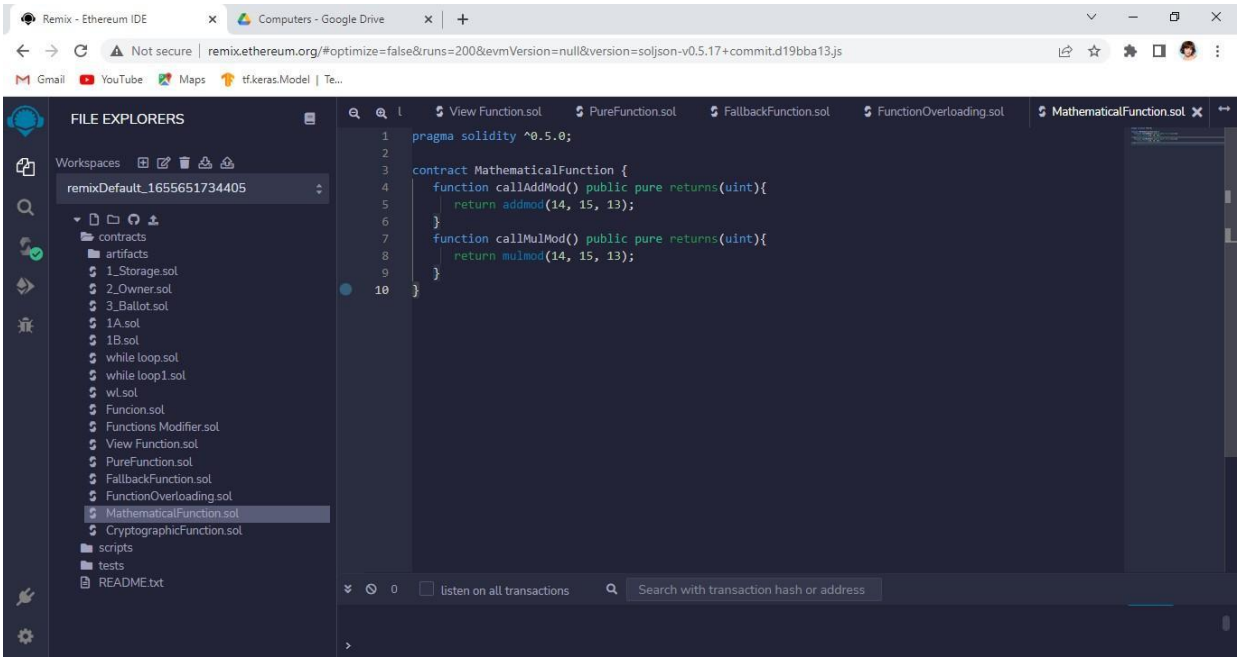
Code:

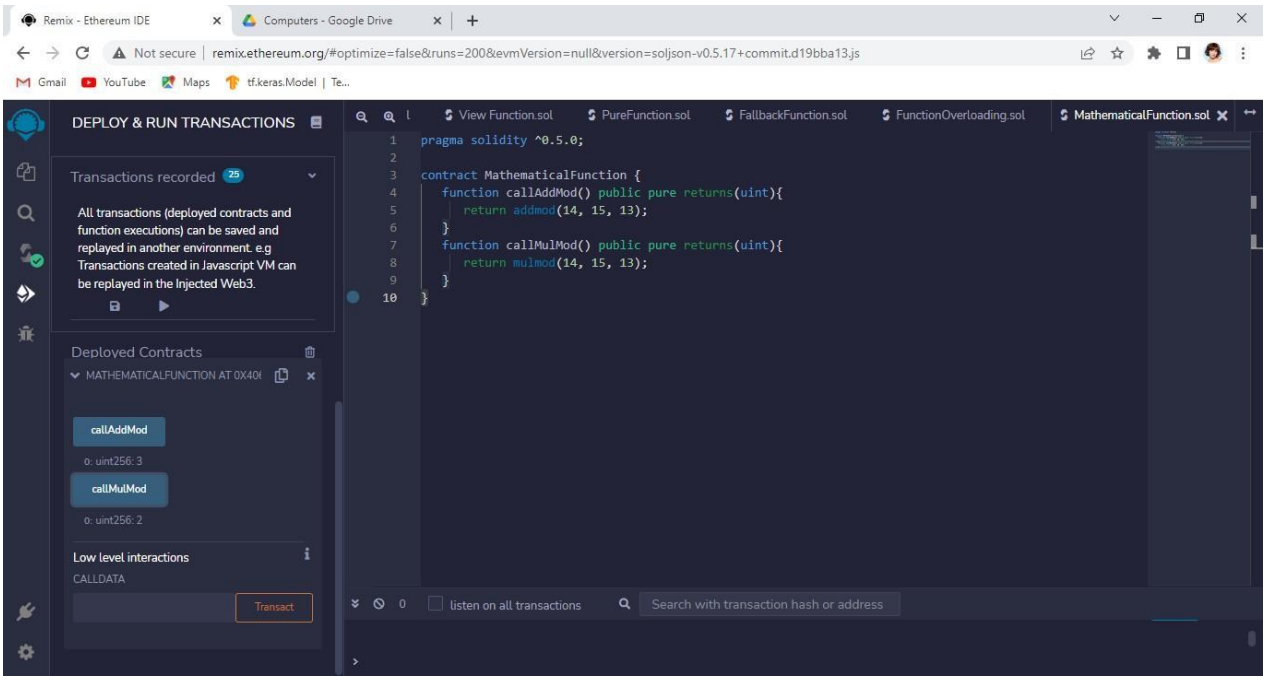




Mathematical Function

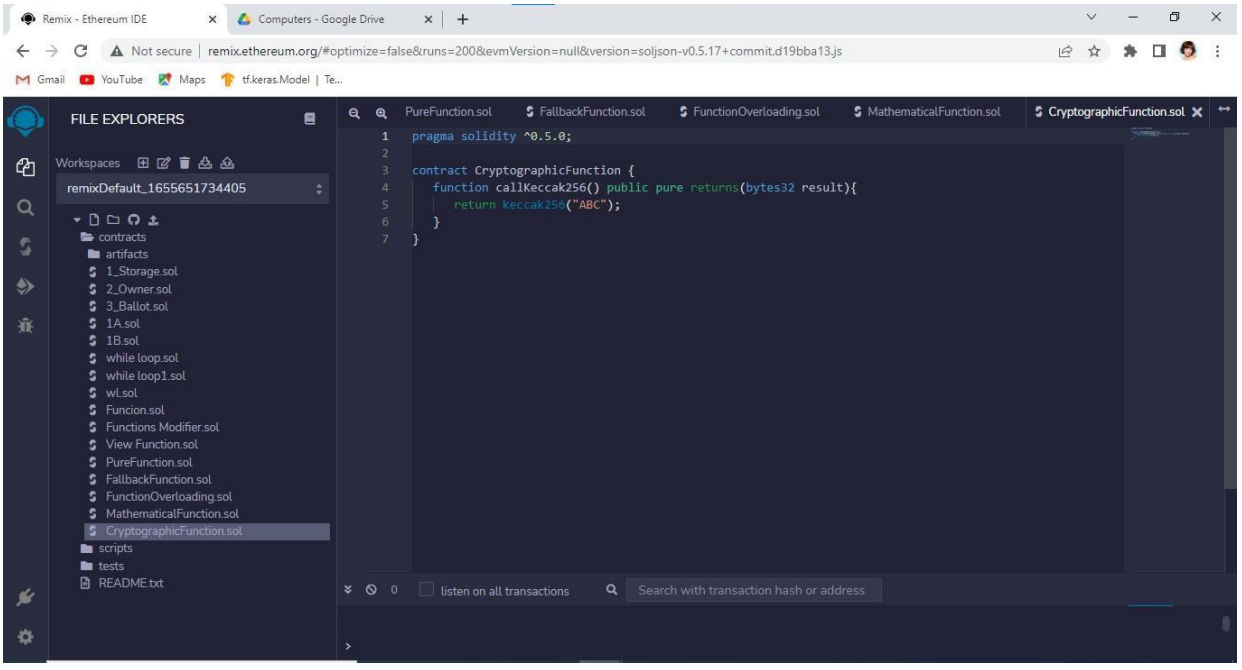
Code:

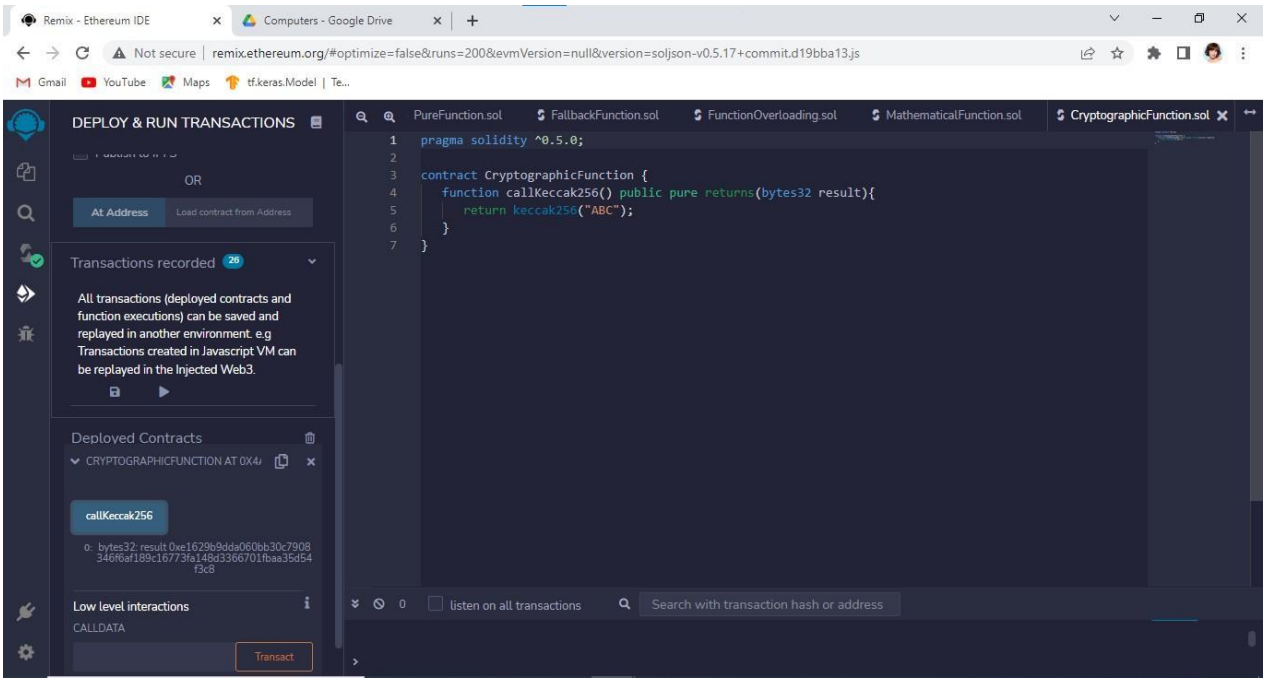




Cryptographic function

Code:

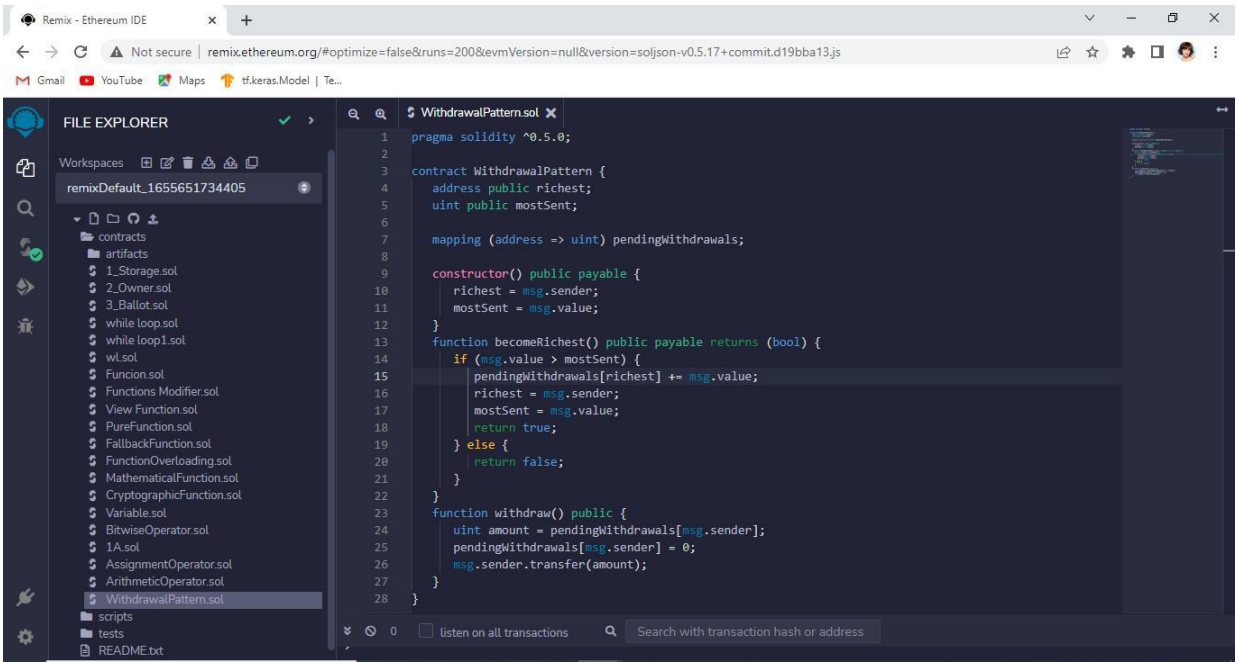


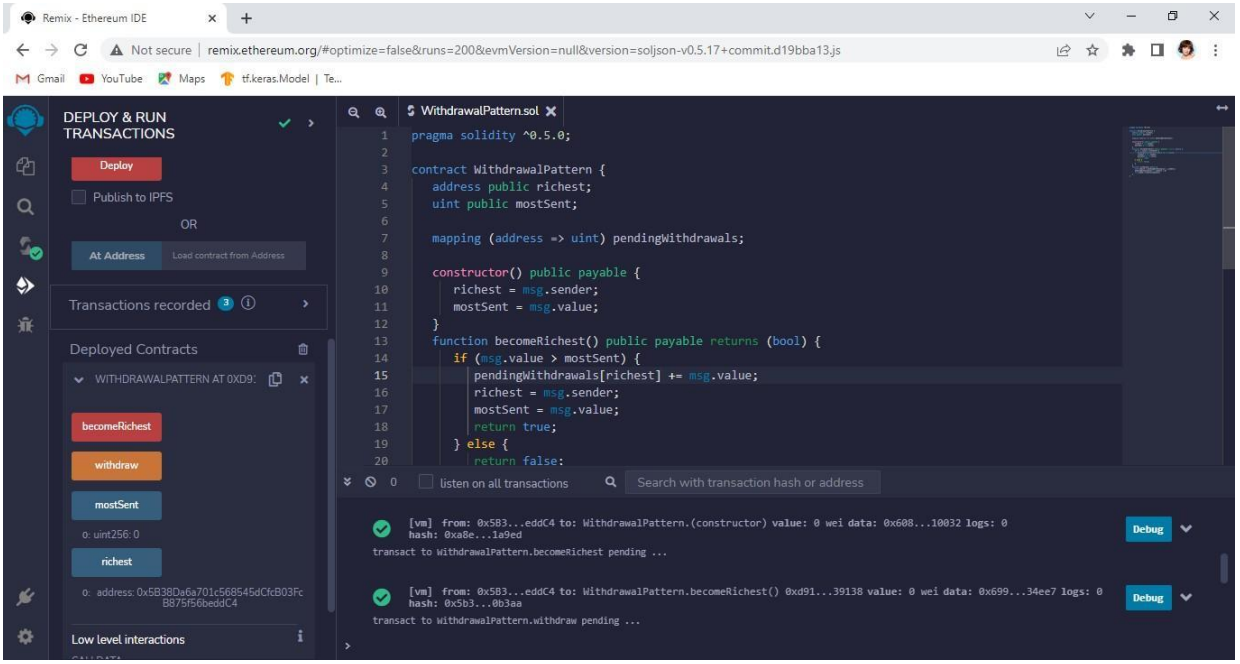


3a) Withdrawal Pattern, Restricted Access.

Withdrawal Pattern

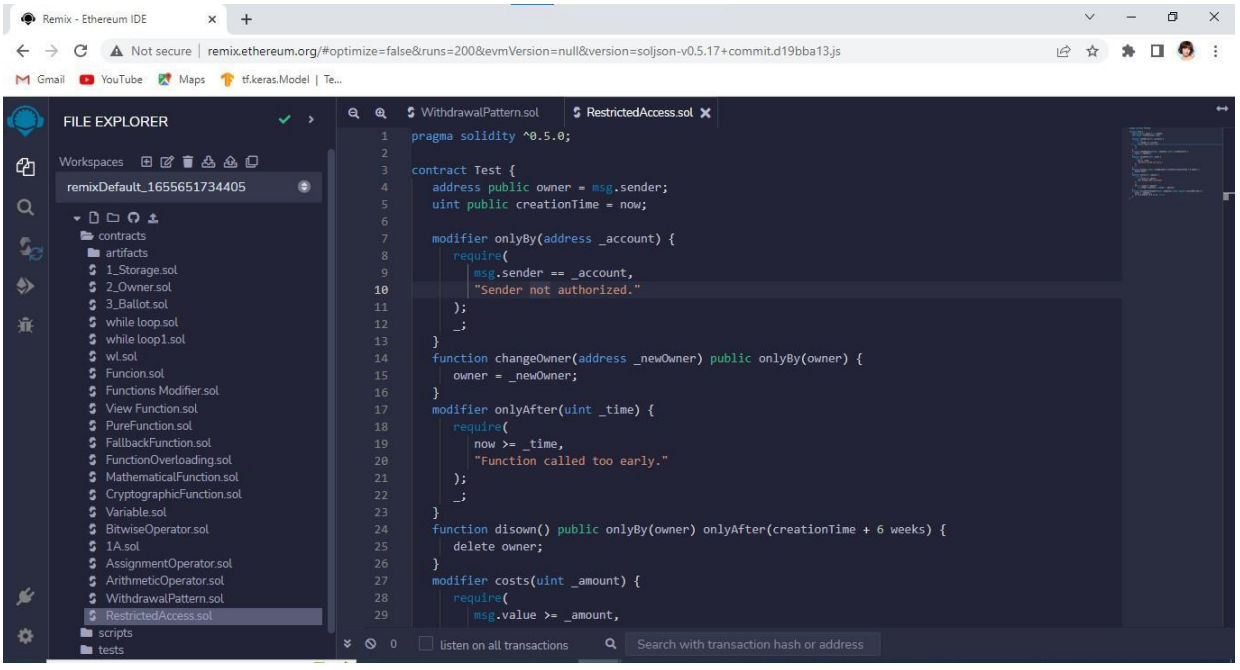
Code:

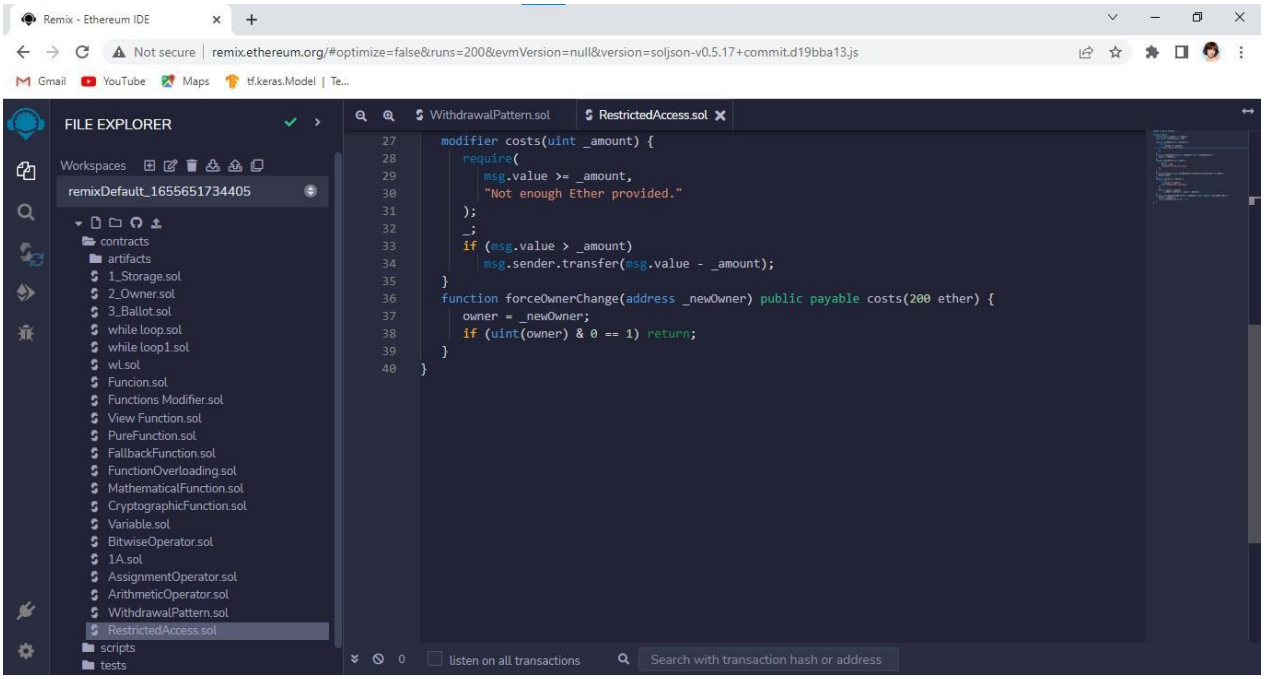




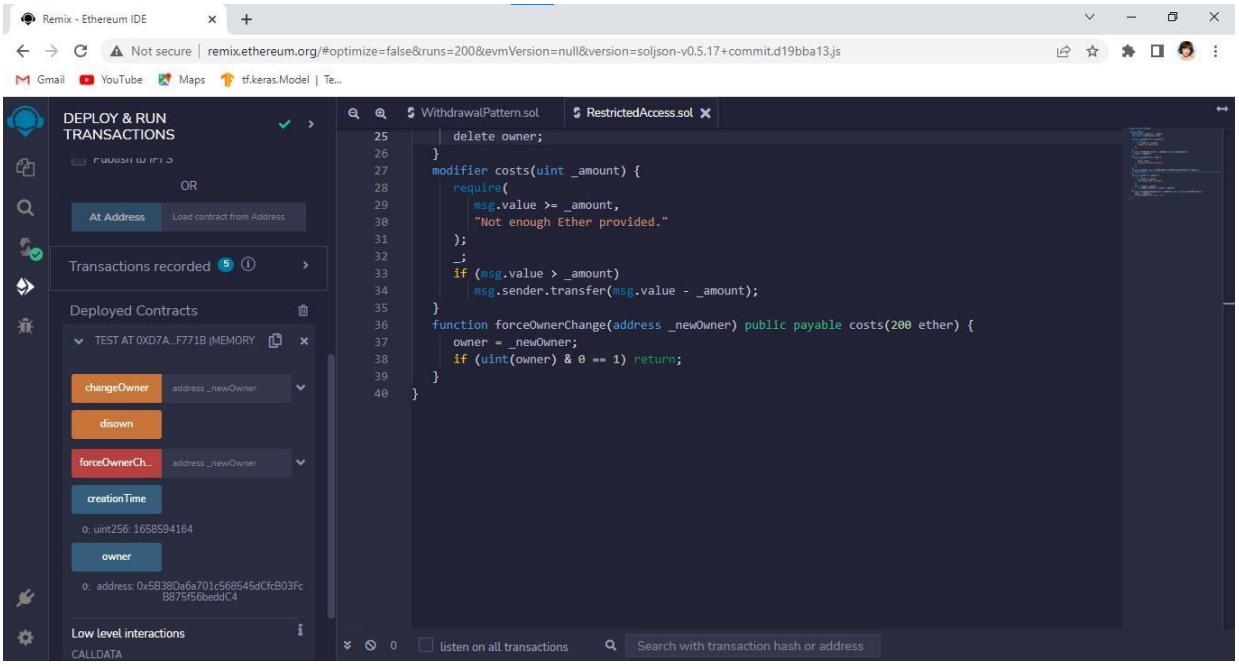
Restricted Access

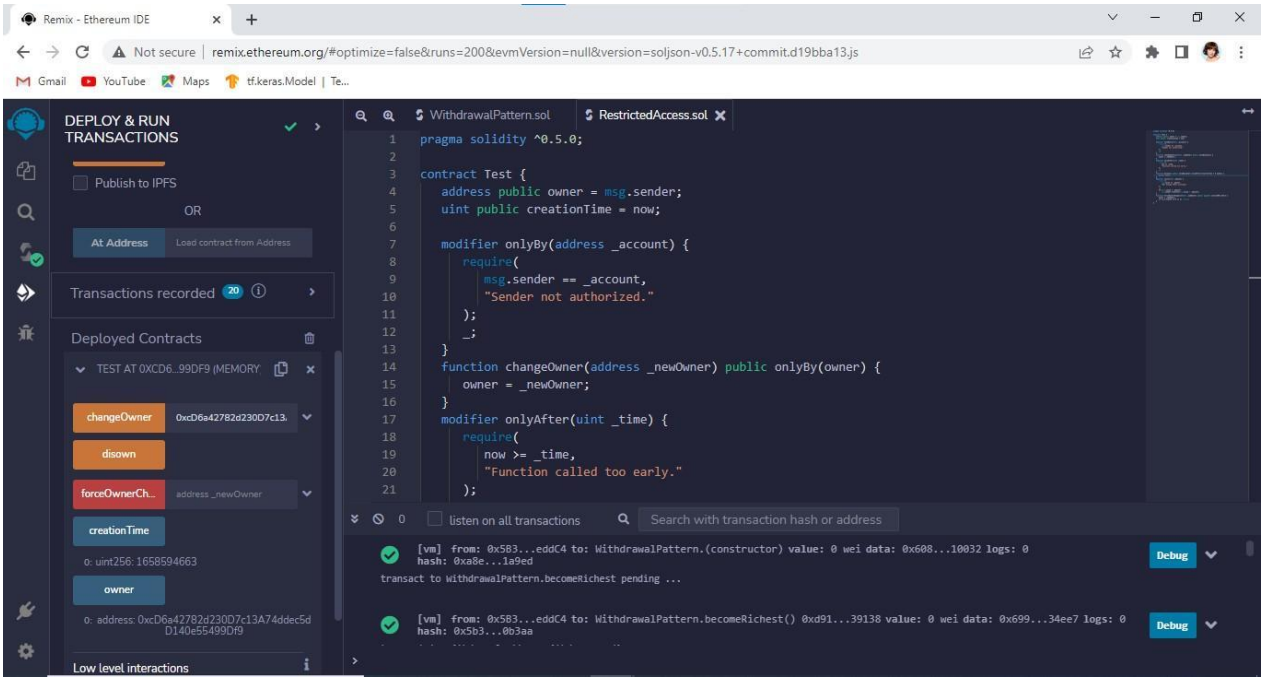
Code:




Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
scripts
tests
WithdrawalPattern.sol
RestrictedAccess.sol
27 modifier costs(uint _amount) {
28 require(
29 msg.value >= _amount,
30 "Not enough Ether provided."
31);
32 _;
33 if (msg.value > _amount)
34 msg.sender.transfer(msg.value - _amount);
35 }
36 function forceOwnerChange(address _newOwner) public payable costs(200 ether) {
37 owner = _newOwner;
38 if (uint(owner) & 0 == 1) return;
39 }
40 }
0
listen on all transactions
Search with transaction hash or address

Output:

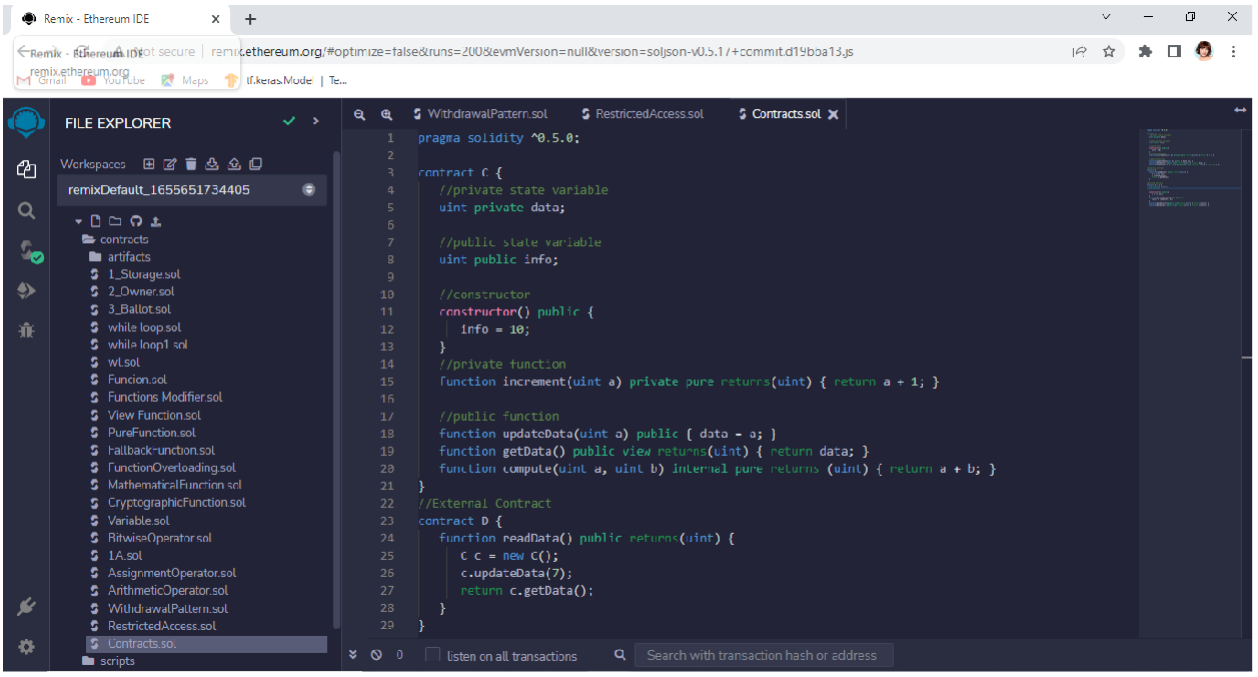

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
OR
At Address
Load contract from Address
Transactions recorded 5
Deployed Contracts
TEST AT 0XD7A...F771B (MEMORY
changeOwner
address _newOwner
disown
forceOwnerCh...
address _newOwner
creationTime
0: uint256: 1658594164
owner
0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
Low level interactions
CALLDATA
WithdrawalPattern.sol
RestrictedAccess.sol
25 delete owner;
26 }
27 modifier costs(uint _amount) {
28 require(
29 msg.value >= _amount,
30 "Not enough Ether provided."
31);
32 _;
33 if (msg.value > _amount)
34 msg.sender.transfer(msg.value - _amount);
35 }
36 function forceOwnerChange(address _newOwner) public payable costs(200 ether) {
37 owner = _newOwner;
38 if (uint(owner) & 0 == 1) return;
39 }
40 }
0
listen on all transactions
Search with transaction hash or address

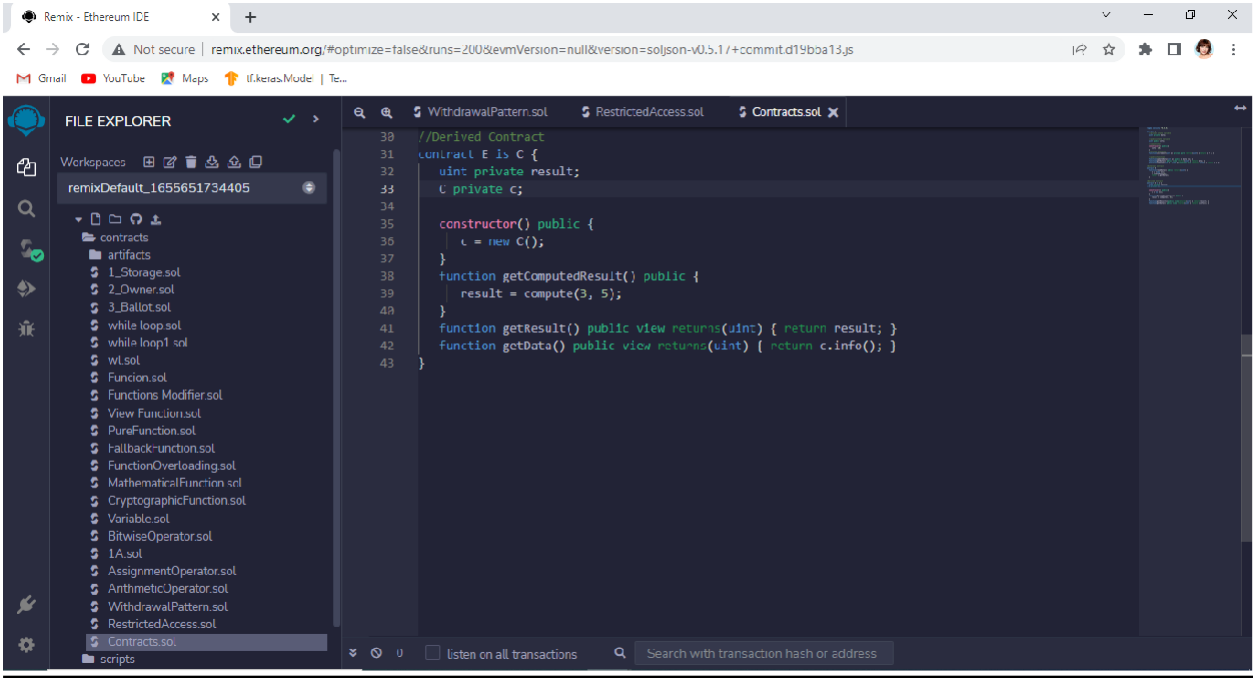


3b) Contracts, Inheritance, Constructors, Abstract Contracts, Interfaces.

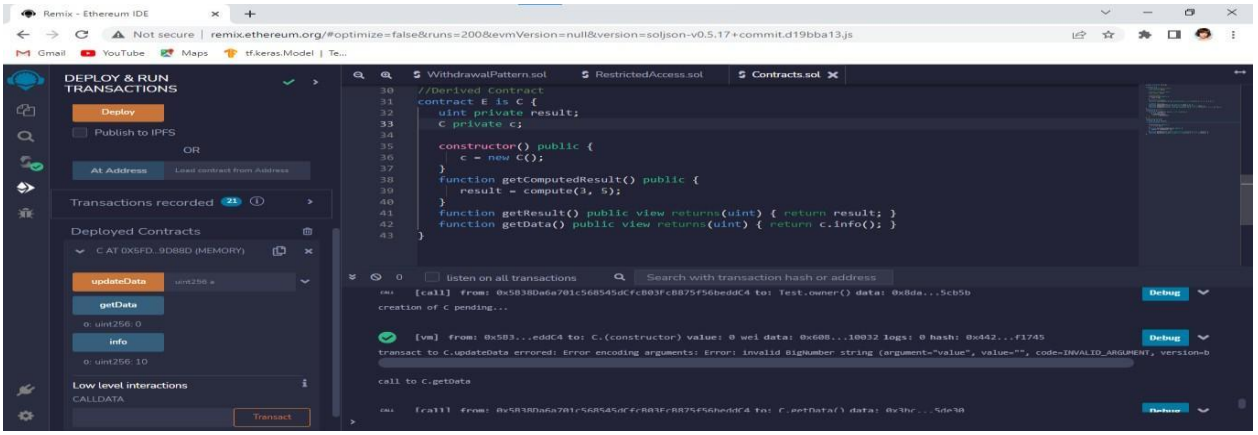
Contracts

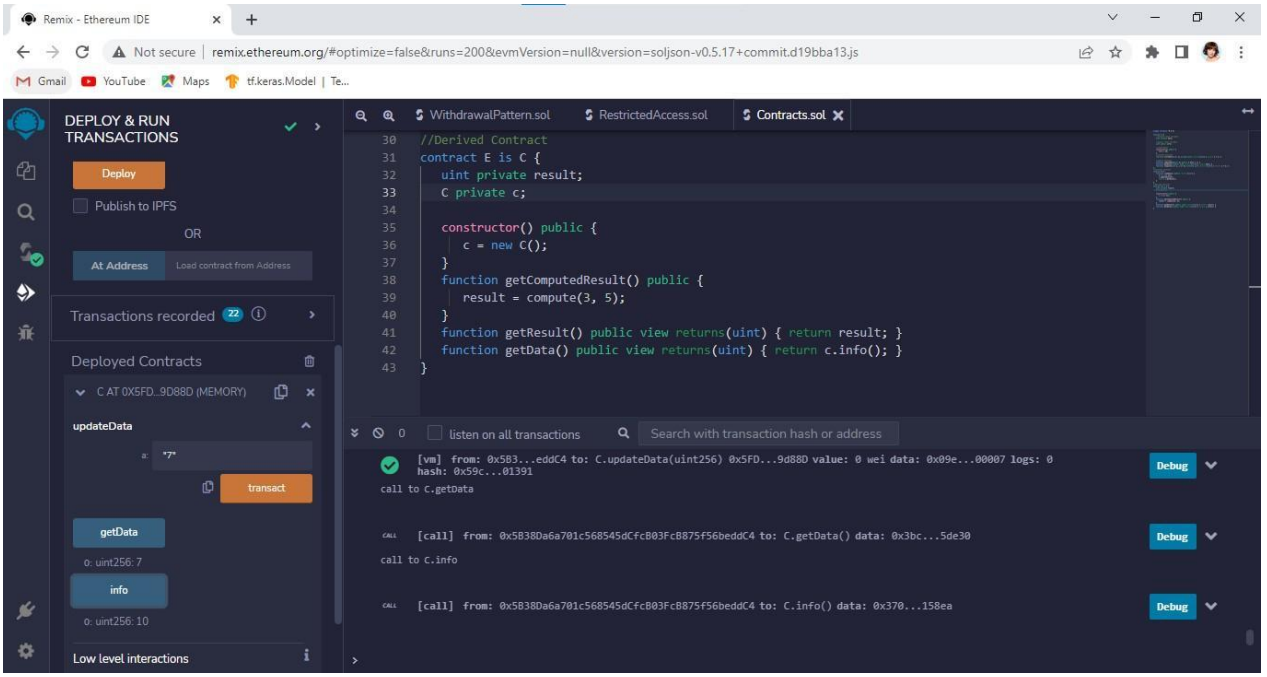
Code:




Remix - Ethereum IDE
FILE EXPLORER
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
//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(); }
}
Listen on all transactions
Search with transaction hash or address

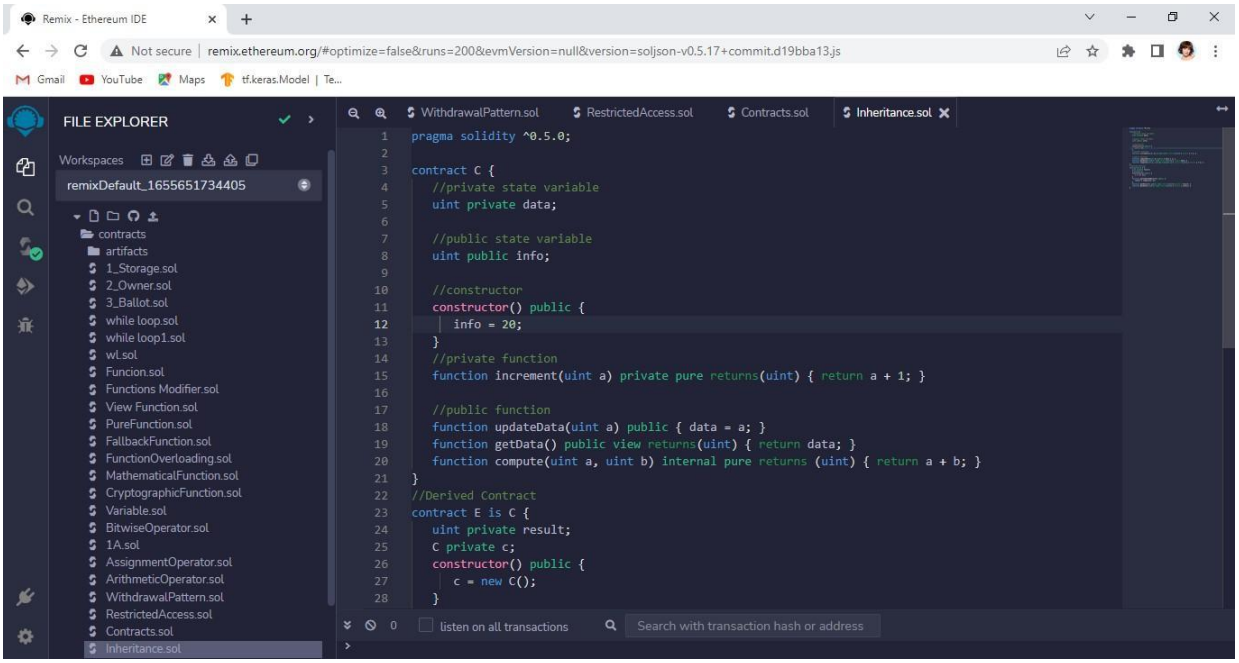
Output:

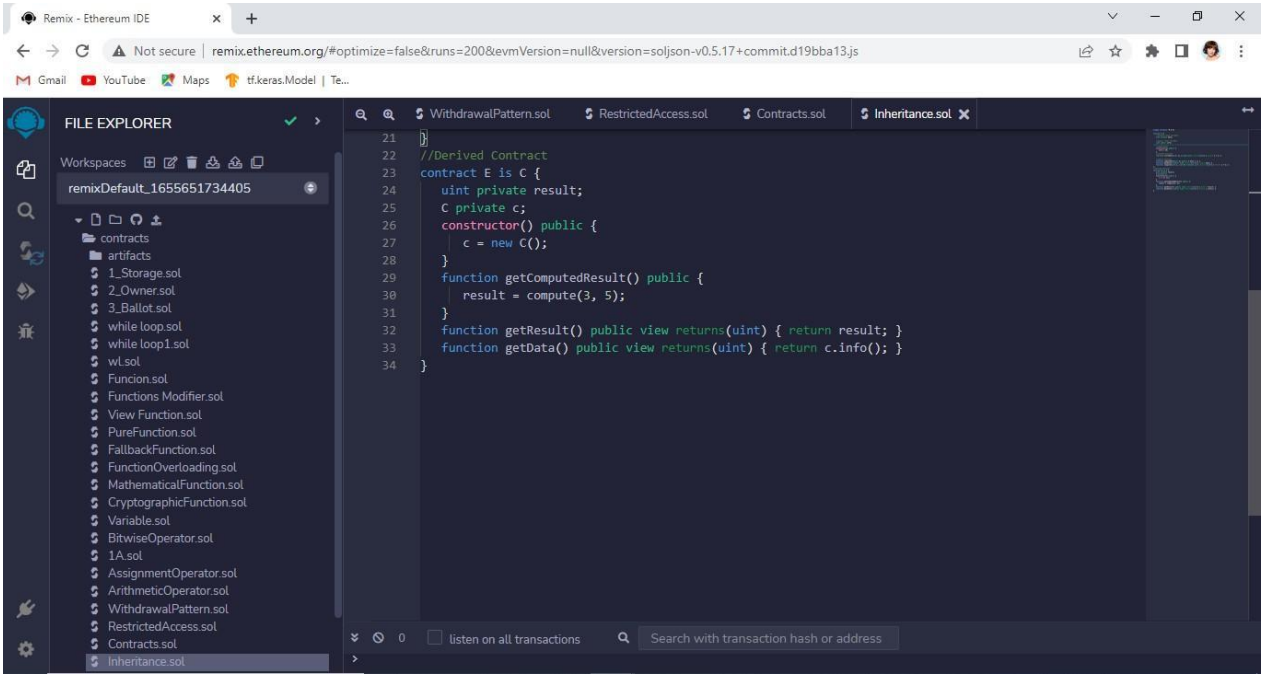

Remix - Ethereum IDE
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Transactions recorded
Deployed Contracts
updateData
getData
info
Low level interactions
CALLDATA
Transact
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Listen on all transactions
Search with transaction hash or address
creation of C pending...
call to C.getData



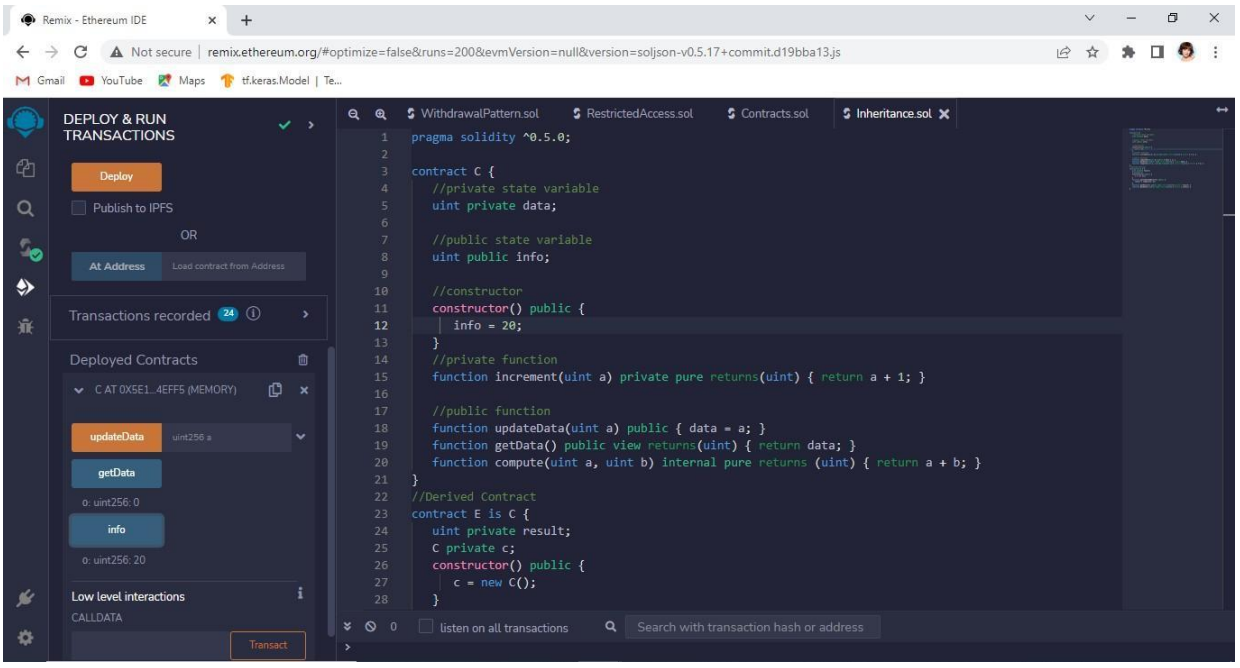
Inheritance

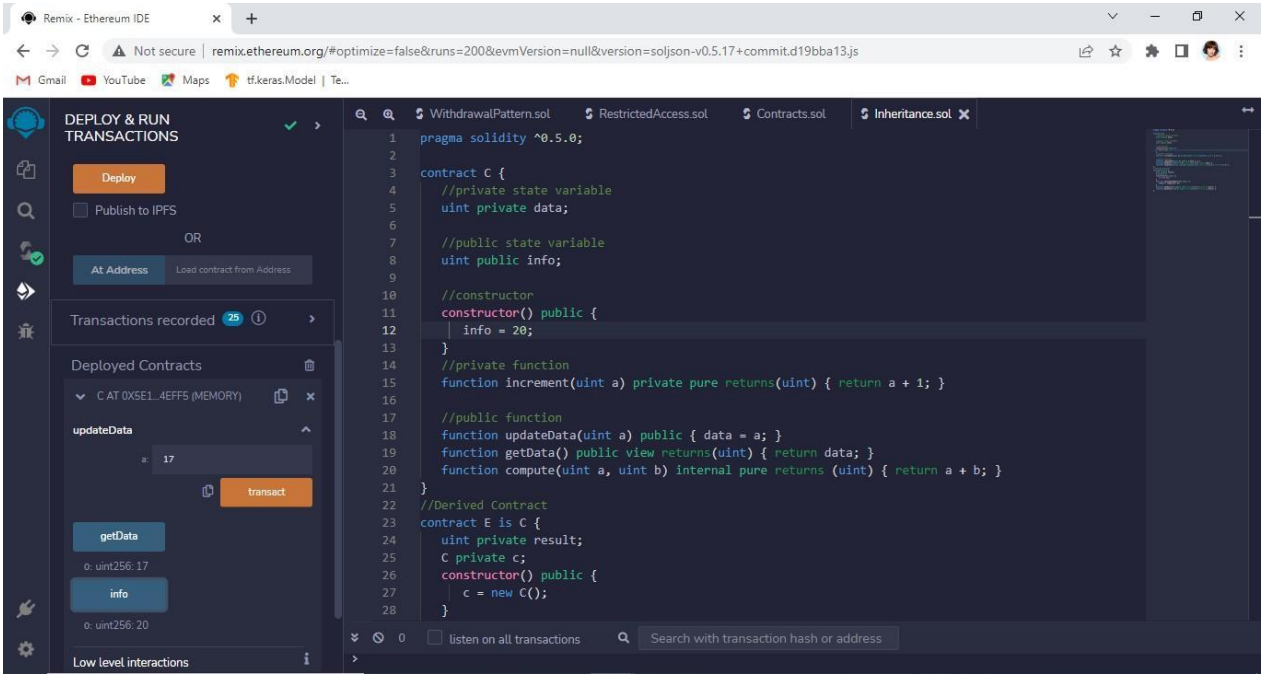
Code:




Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail YouTube Maps tf.keras.Model | Te...
FILE EXPLORER
Workspaces
remixDefault_1655651734405
contracts
artifacts
1_Storage.sol
2_Owner.sol
3_Ballot.sol
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
}
//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(); }
}
0
listen on all transactions
Search with transaction hash or address

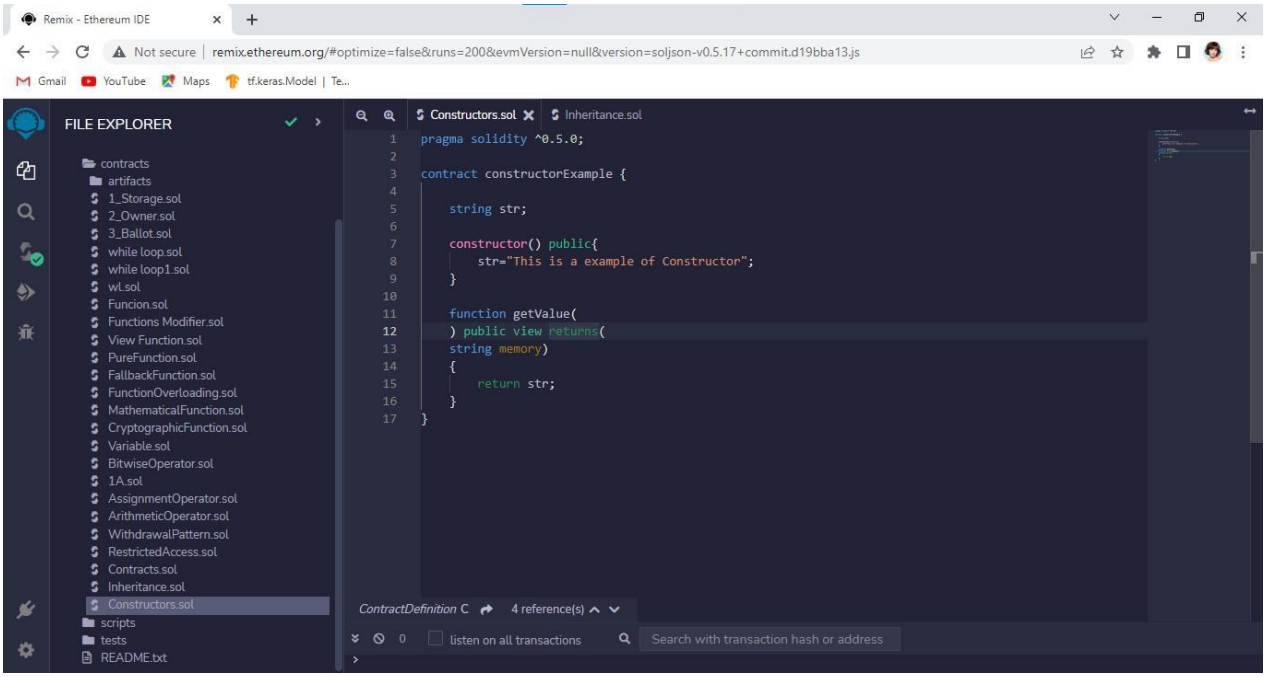
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js
Gmail YouTube Maps tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 24
Deployed Contracts
C AT 0X5E1...4EFF5 (MEMORY)
updateData
uint256 a
getData
0: uint256: 0
info
0: uint256: 20
Low level interactions
CALLDATA
Transact
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
pragma solidity ^0.5.0;
contract C {
//private state variable
uint private data;
//public state variable
uint public info;
//constructor
constructor() public {
info = 20;
}
//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();
}
0
listen on all transactions
Search with transaction hash or address

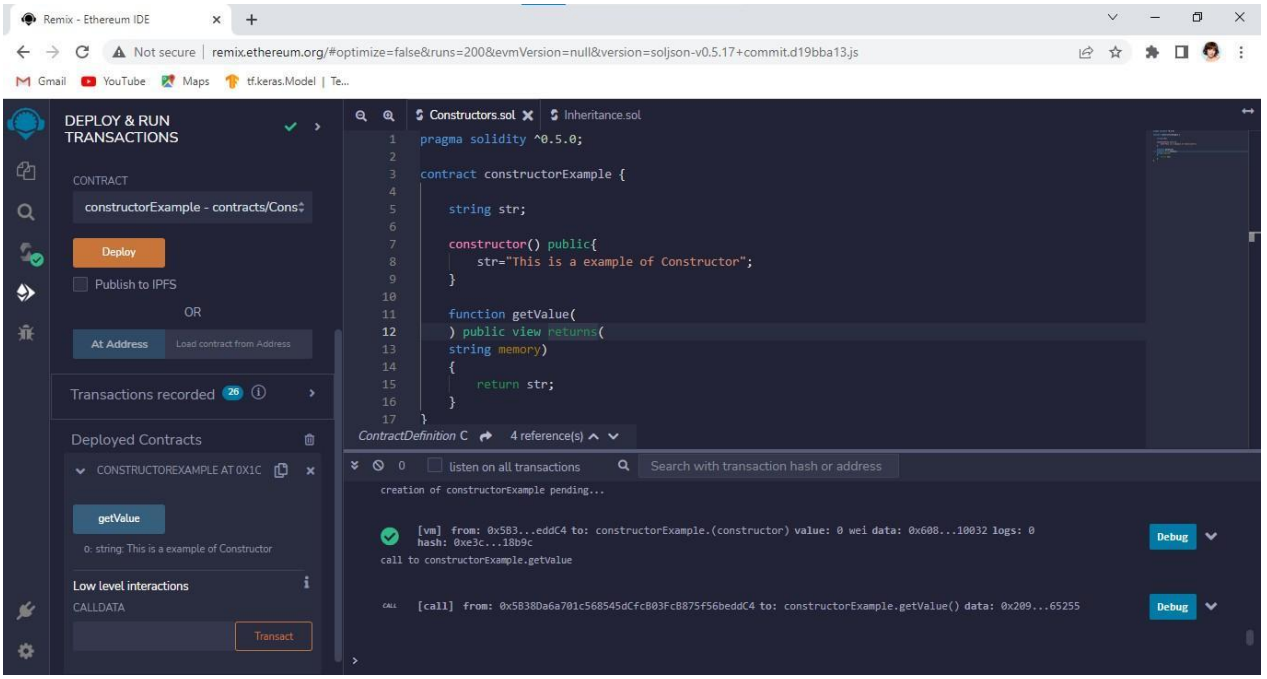


Constructors

Code:

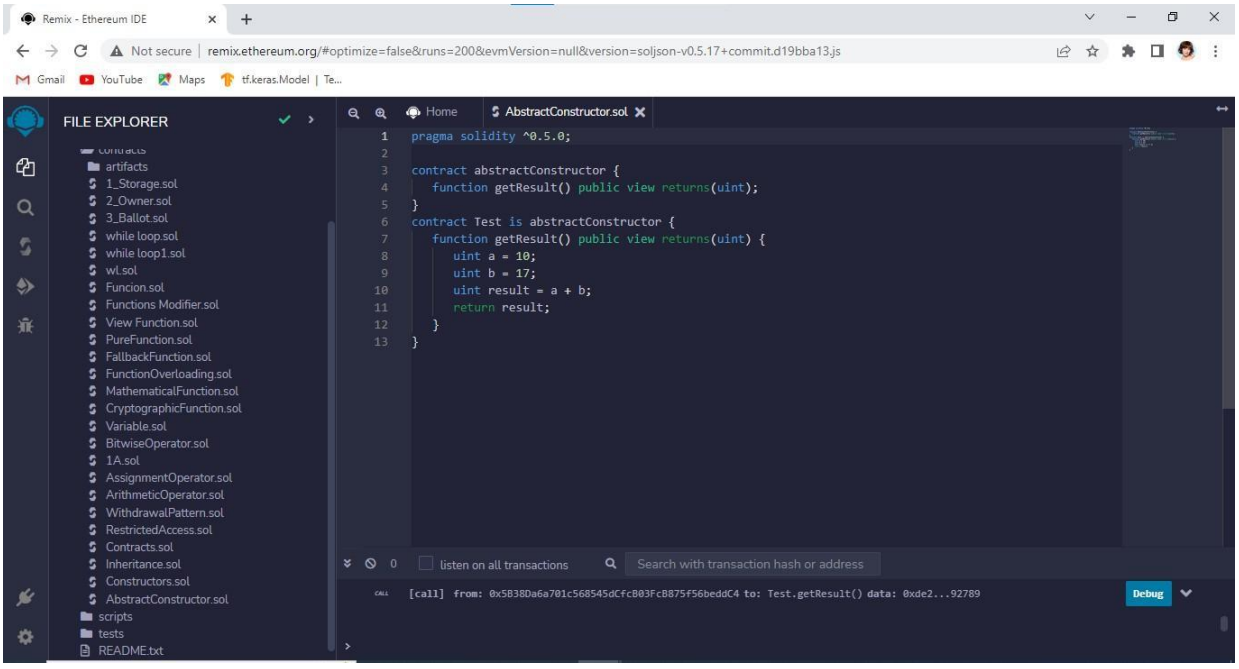


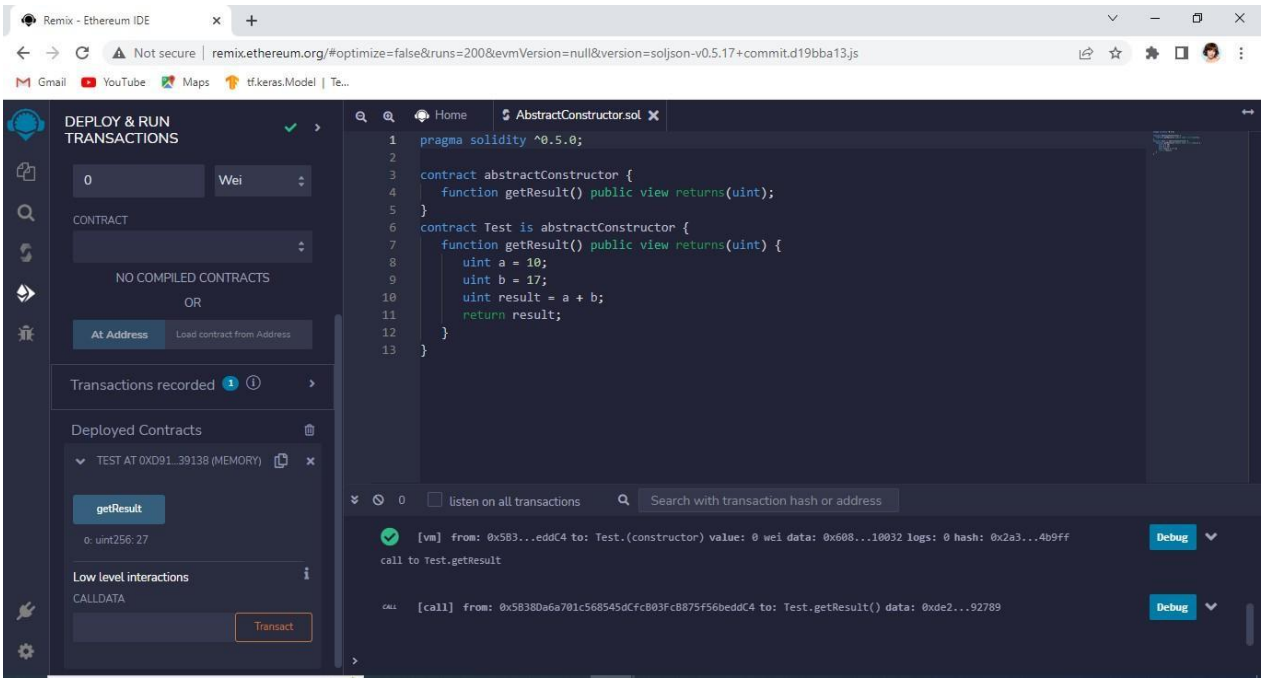
Output:



Abstract Contracts

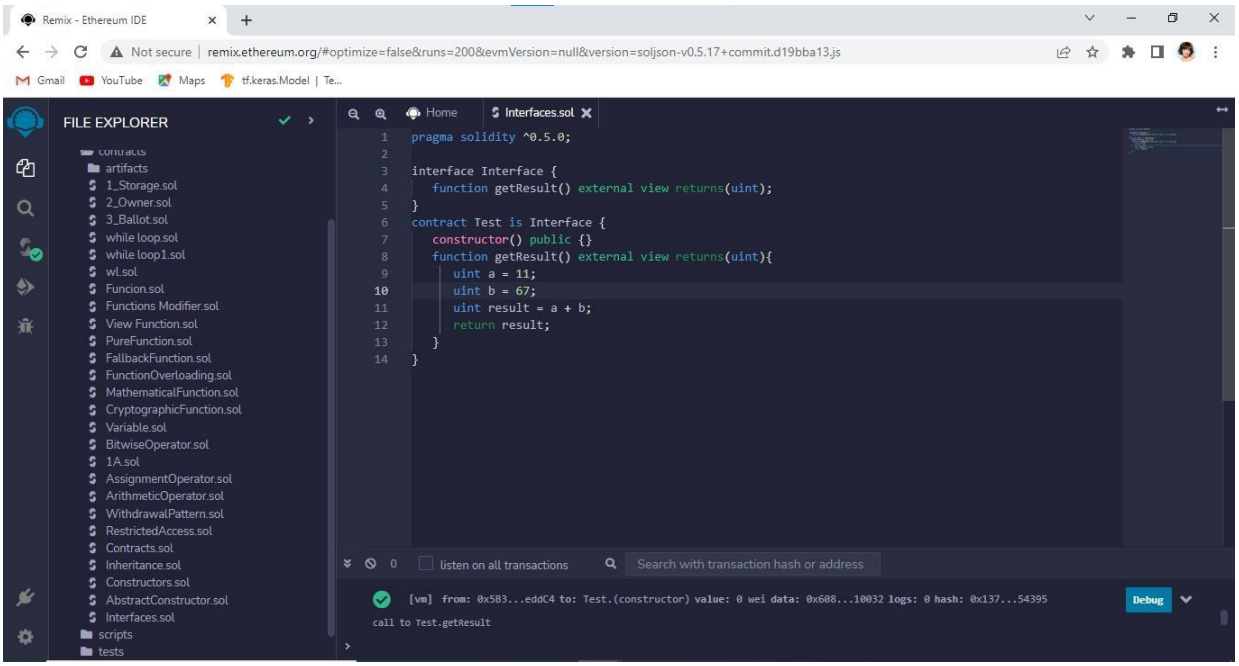
Code:

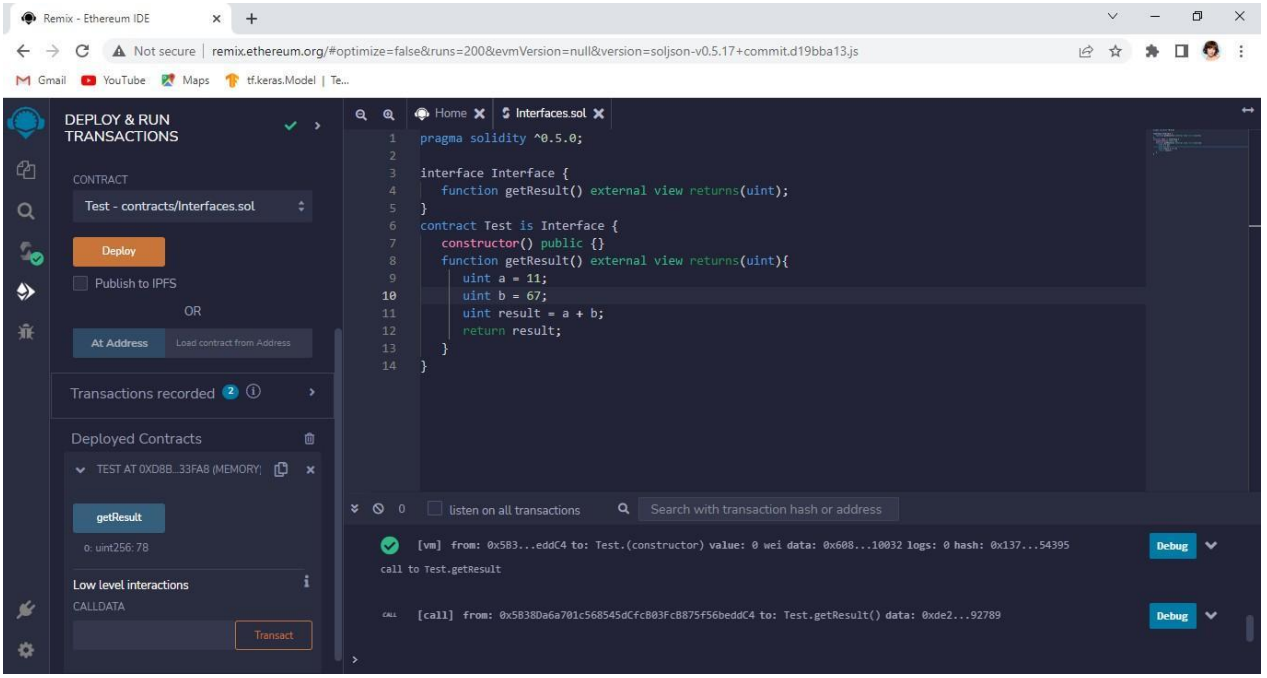




Interfaces

Code:

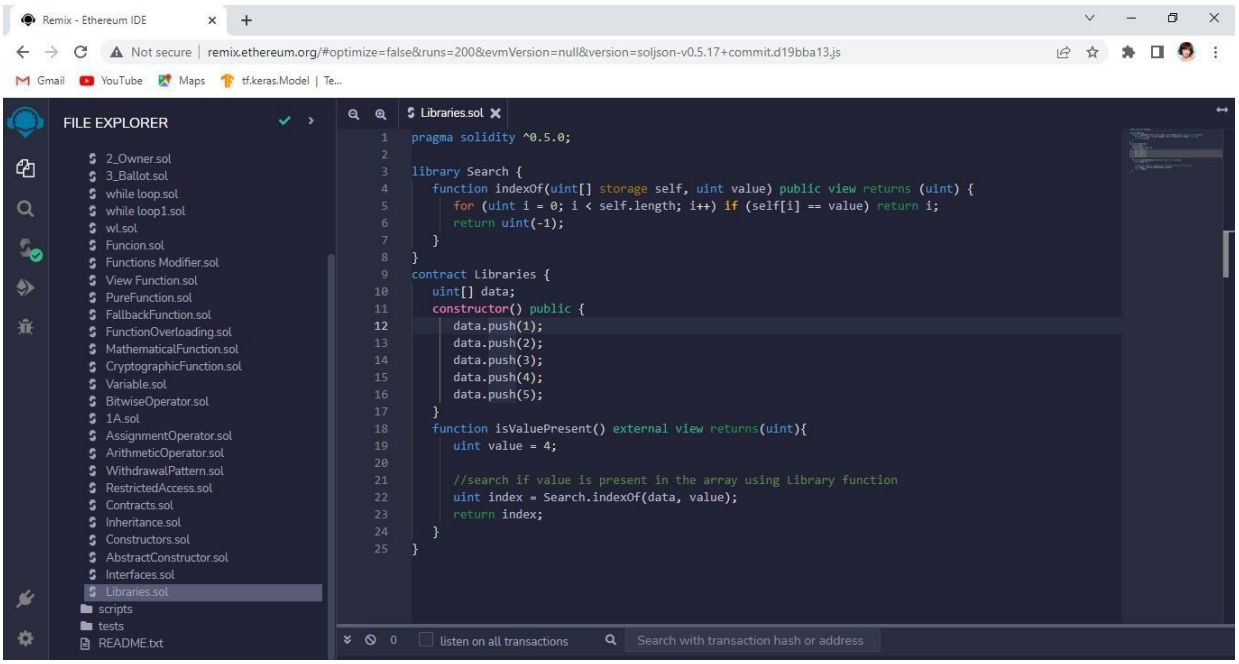


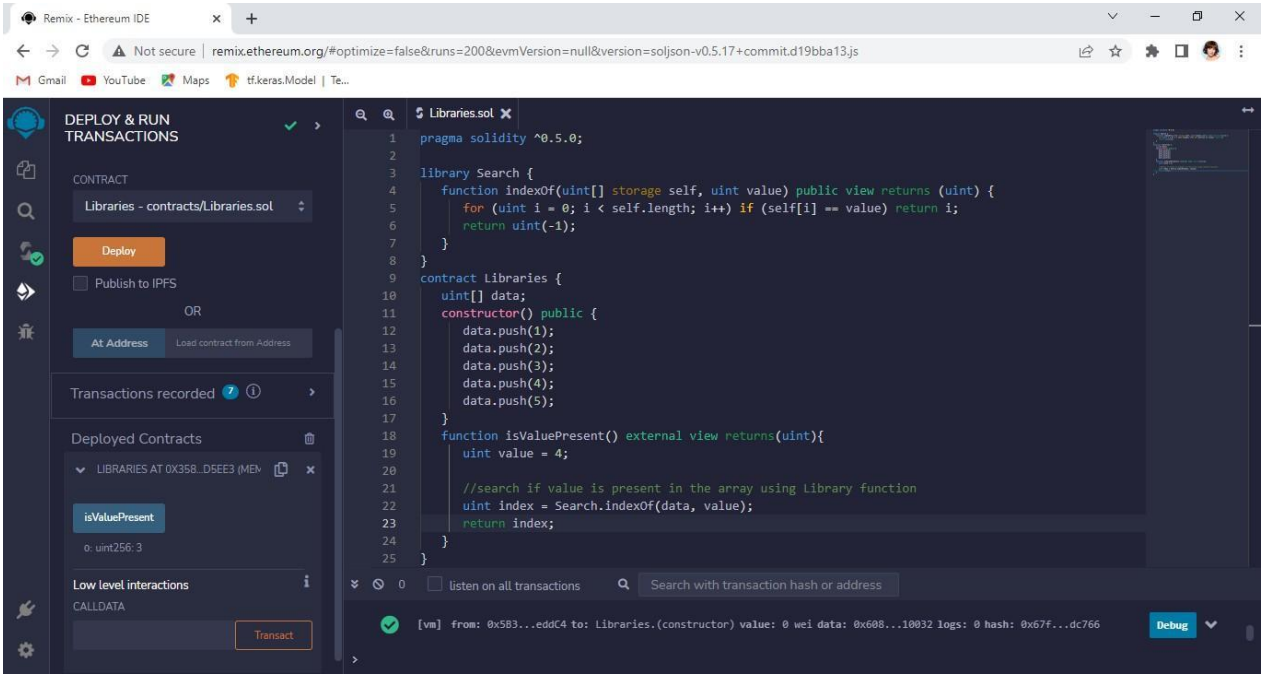


3c) Libraries, Assembly, Events, Error handling.

Libraries

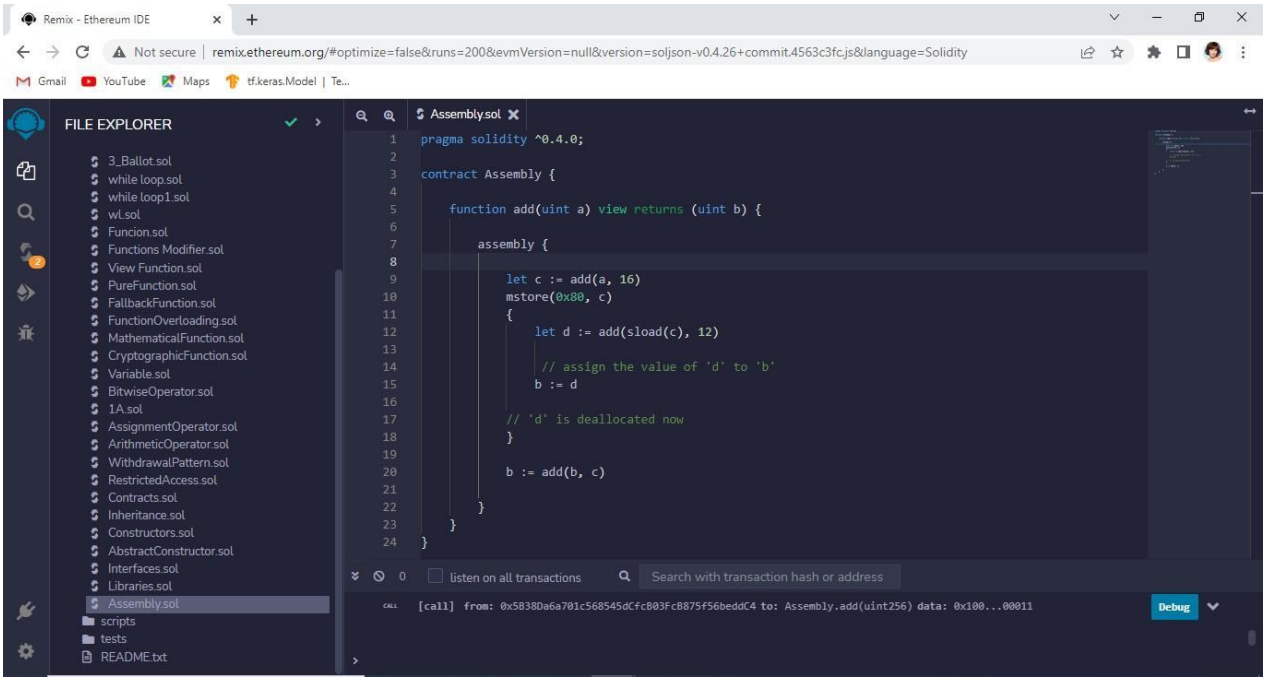
Code:



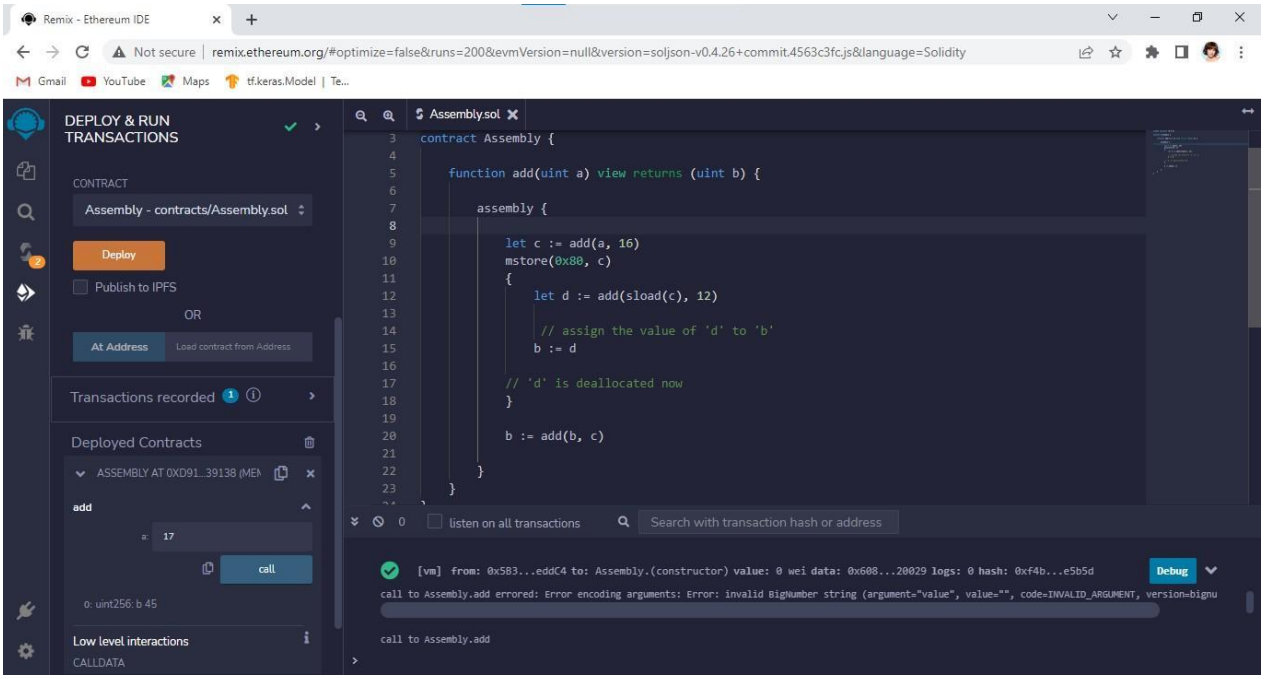


Assembly

Code:

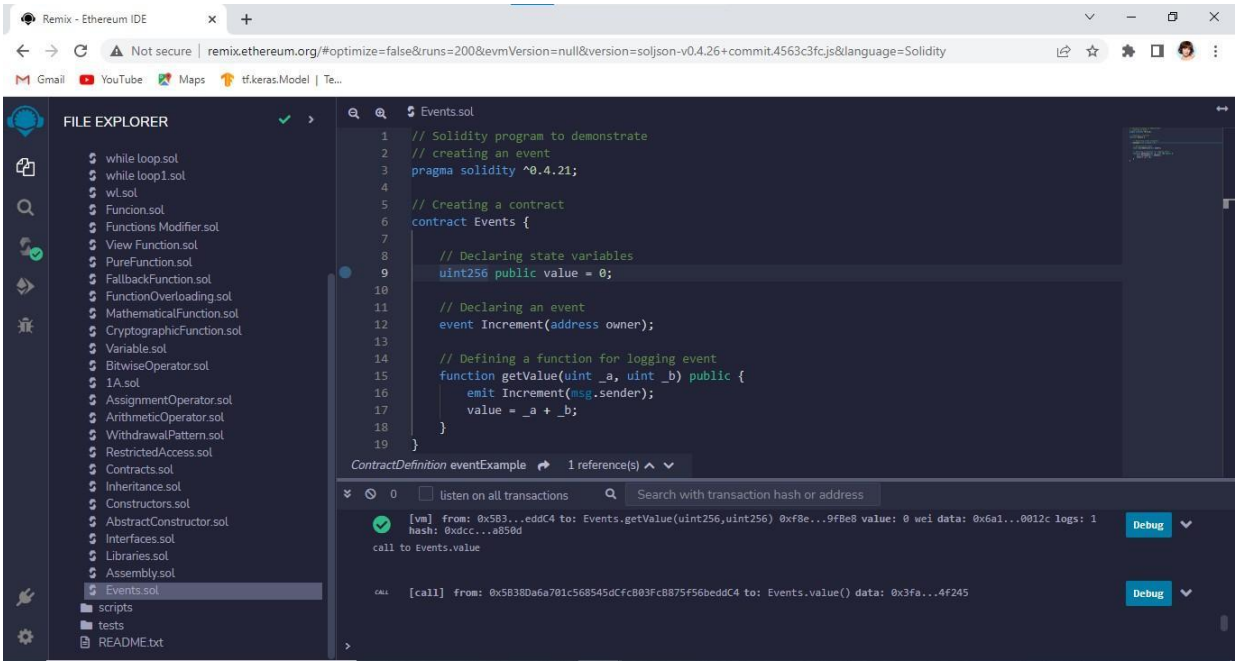


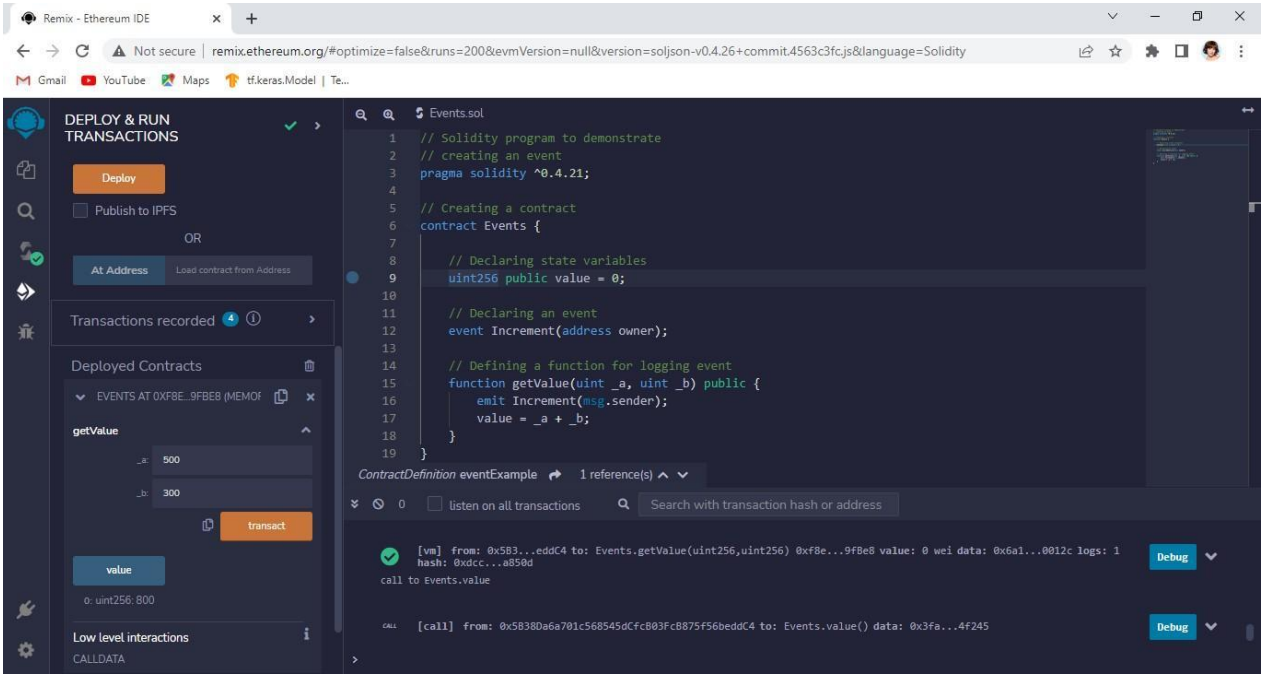
Output:


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
CONTRACT
Assembly - contracts/Assembly.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 1
Deployed Contracts
ASSEMBLY AT 0XD91...39138 (MEM
add
a: 17
call
0: uint256: b 45
Low level interactions
CALLDATA
Assembly.sol
contract Assembly {
function add(uint a) view returns (uint b) {
assembly {
let c := add(a, 16)
mstore(0x80, c)
{
let d := add(sload(c), 12)
// assign the value of 'd' to 'b'
b := d
// 'd' is deallocated now
}
b := add(b, c)
}
}
}
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: Assembly.(constructor) value: 0 wei data: 0x608...20029 logs: 0 hash: 0xf4b...e5b5d
Debug
call to Assembly.add errored: Error encoding arguments: Error: invalid BigNumber string (argument="value", value="", code=INVALID_ARGUMENT, version=bignu
call to Assembly.add

Events

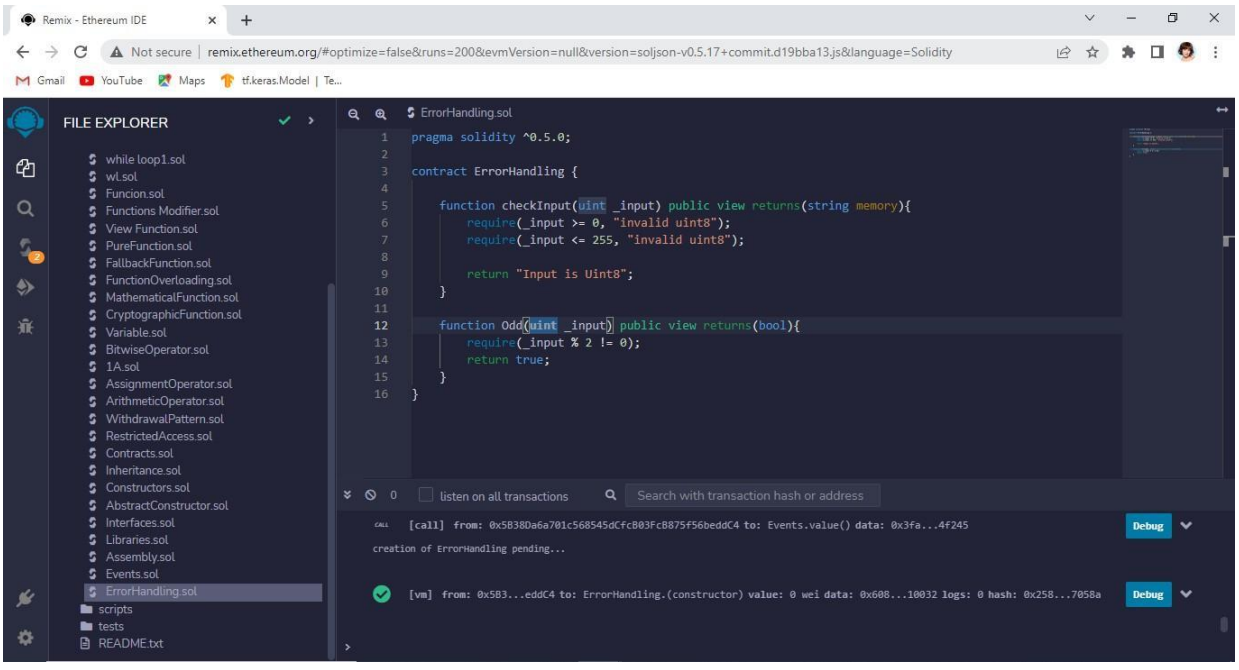
Code:

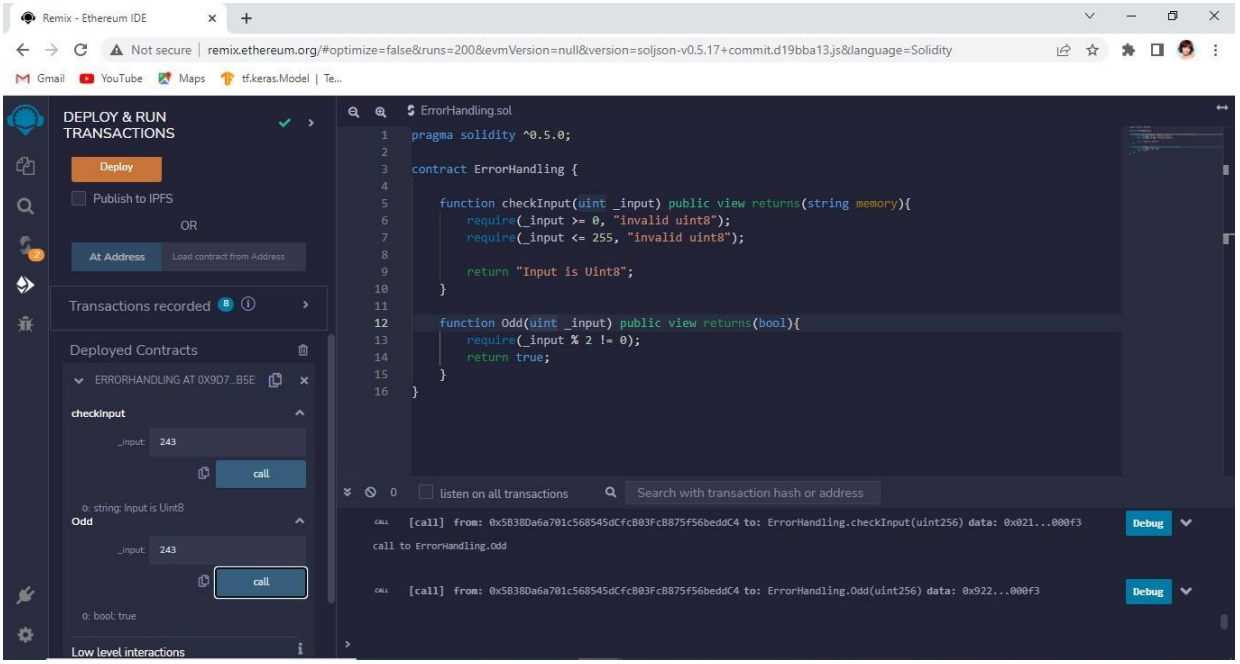

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
FILE EXPLORER
while loop.sol
while loop1.sol
wl.sol
Funcion.sol
Functions Modifier.sol
View Function.sol
PureFunction.sol
FallbackFunction.sol
FunctionOverloading.sol
MathematicalFunction.sol
CryptographicFunction.sol
Variable.sol
BitwiseOperator.sol
1A.sol
AssignmentOperator.sol
ArithmeticOperator.sol
WithdrawalPattern.sol
RestrictedAccess.sol
Contracts.sol
Inheritance.sol
Constructors.sol
AbstractConstructor.sol
Interfaces.sol
Libraries.sol
Assembly.sol
Events.sol
scripts
tests
README.txt
Events.sol
// Solidity program to demonstrate
// creating an event
pragma solidity ^0.4.21;
// Creating a contract
contract Events {
// Declaring state variables
uint256 public value = 0;
// Declaring an event
event Increment(address owner);
// Defining a function for logging event
function getValue(uint _a, uint _b) public {
emit Increment(msg.sender);
value = _a + _b;
}
}
ContractDefinition eventExample
1 reference(s)
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: Events.getValue(uint256,uint256) 0xf8e...9fBe8 value: 0 wei data: 0x6a1...0012c logs: 1 hash: 0xdcc...a850d
Debug
call to Events.value
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Events.value() data: 0x3fa...4f245
Debug



Error Handling

Code:




Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.5.17+commit.d19bba13.js&language=Solidity
Gmail
YouTube
Maps
tf.keras.Model | Te...
DEPLOY & RUN TRANSACTIONS
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 8
Deployed Contracts
ERRORHANDLING AT 0X9D7...B5E
checkInput
_input: 243
call
0: string: Input is Uint8
Odd
_input: 243
call
0: bool: true
Low level interactions
ErrorHandling.sol
pragma solidity ^0.5.0;
contract ErrorHandling {
function checkInput(uint _input) public view returns(string memory){
require(_input >= 0, "invalid uint8");
require(_input <= 255, "invalid uint8");
return "Input is Uint8";
}
function Odd(uint _input) public view returns(bool){
require(_input % 2 != 0);
return true;
}
}
listen on all transactions
Search with transaction hash or address
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ErrorHandling.checkInput(uint256) data: 0x021...000f3
Debug
call to ErrorHandling.Odd
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ErrorHandling.Odd(uint256) data: 0x922...000f3
Debug

NATURAL

LANGUAGE

PROCESSING

INDEX

Sr. No	Practicals	Sign
1	a) Install NLTK. b) Convert the given text to speech. c) Convert audio file Speech to Text.	
2	a) Study of various Corpus – Brown, Inaugural, Reuters, udhr with various methods like fields, raw, words, sents, categories. b) Create and use your own corpora (plaintext, categorical) c) Study Conditional frequency distributions d) Study of tagged corpora with methods like tagged_sents, tagged_words. e) Write a program to find the most frequent noun tags. f) Map Words to Properties Using Python Dictionaries g) Study Default Tagger, Regular expression tagger, UnigramTagger h) Find different words from a given plain text without any space by comparing this text with a given corpus of words. Also find the score of words.	
3	a) Study of Wordnet Dictionary with methods as synsets, definitions, examples, antonyms b) Study lemmas, hyponyms, hypernyms. c) Write a program using python to find synonym and antonym of word "active" using Wordnet. d) Compare two nouns e) Handling stopword:	
4	a) Tokenization using Python's split() function code b) Tokenization using Regular Expressions (RegEx) c) Tokenization using NLTK d) Tokenization using the spaCy library e) Tokenization using Keras f) Tokenization using Gensim	
5	a) word tokenization in Hindi Source code b) Generate similar sentences from a given Hindi text input Source code c) Identify the Indian language of a text Source code	
6	a) Part of speech Tagging and chunking of user defined tex b) Named Entity recognition of user defined text.	

	c) Named Entity recognition with diagram using NLTK corpus – treebank	
7	a) Define grammar using nltk. Analyze a sentence using the same b) Accept the input string with Regular expression of Finite Automaton c) Accept the input string with Regular expression of FA d) Implementation of Deductive Chart Parsing using context free grammar and a given sentence	
8	Study PorterStemmer, LancasterStemmer, RegexpStemmer, SnowballStemmer Study WordNetLemmatize	
9	Implement Naive Bayes classifier.	
10	a) Speech Tagging b) Statistical parsing c) Malt parsing	
11	a) Multiword Expressions in NLP b) Normalized Web Distance and Word Similarity Source c) Word Sense Disambiguation Source	

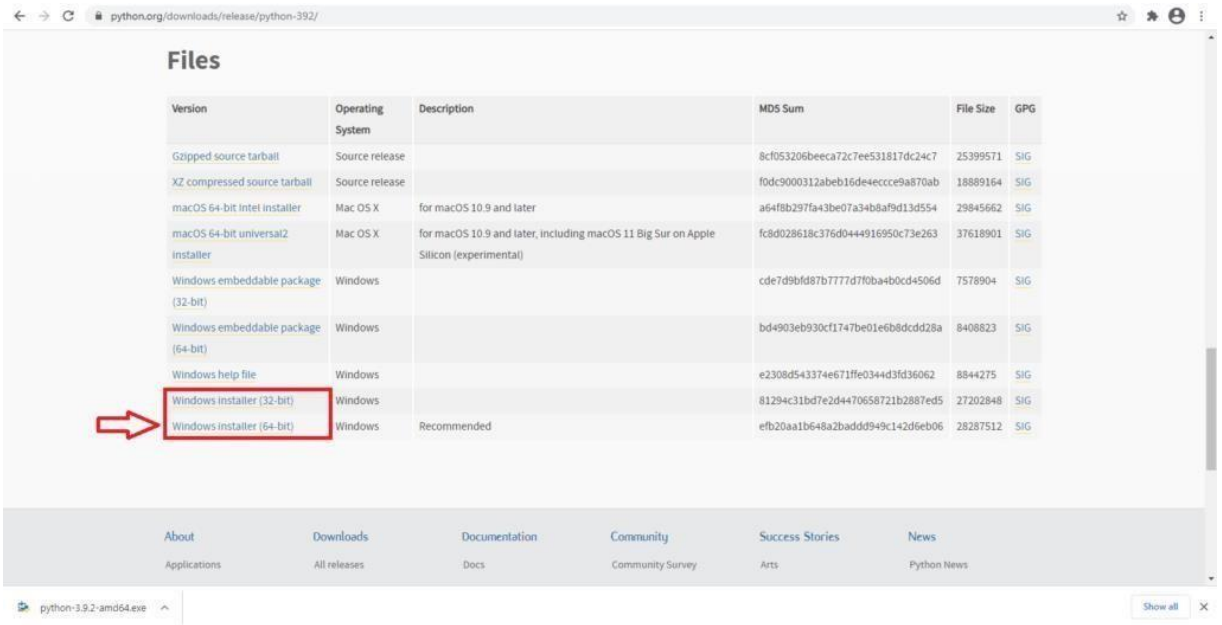
Practical 1

a) Install NLTK

Python 3.9.2 Installation on Windows

Step 1) **Go to link** https://www.python.org/downloads/, **and select the latest version for windows.**

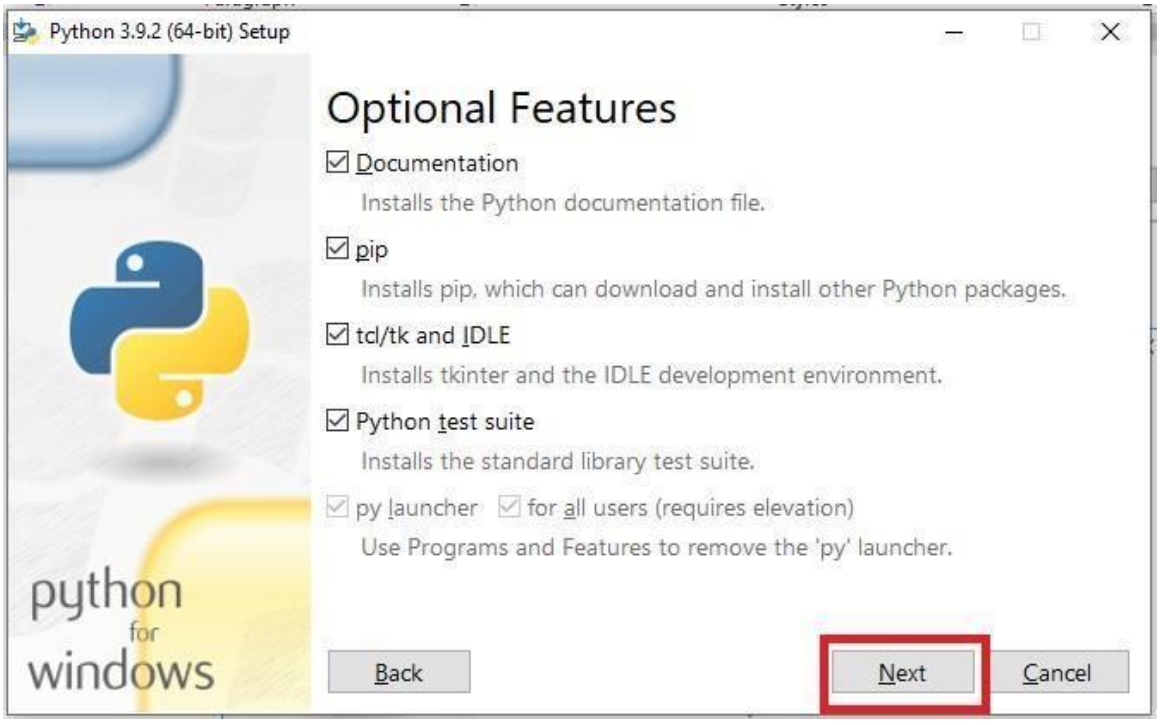
Note: If you don't want to download the latest version, you can visit the download tab and see all releases.



Step 2) Click on the Windows installer (64 bit)

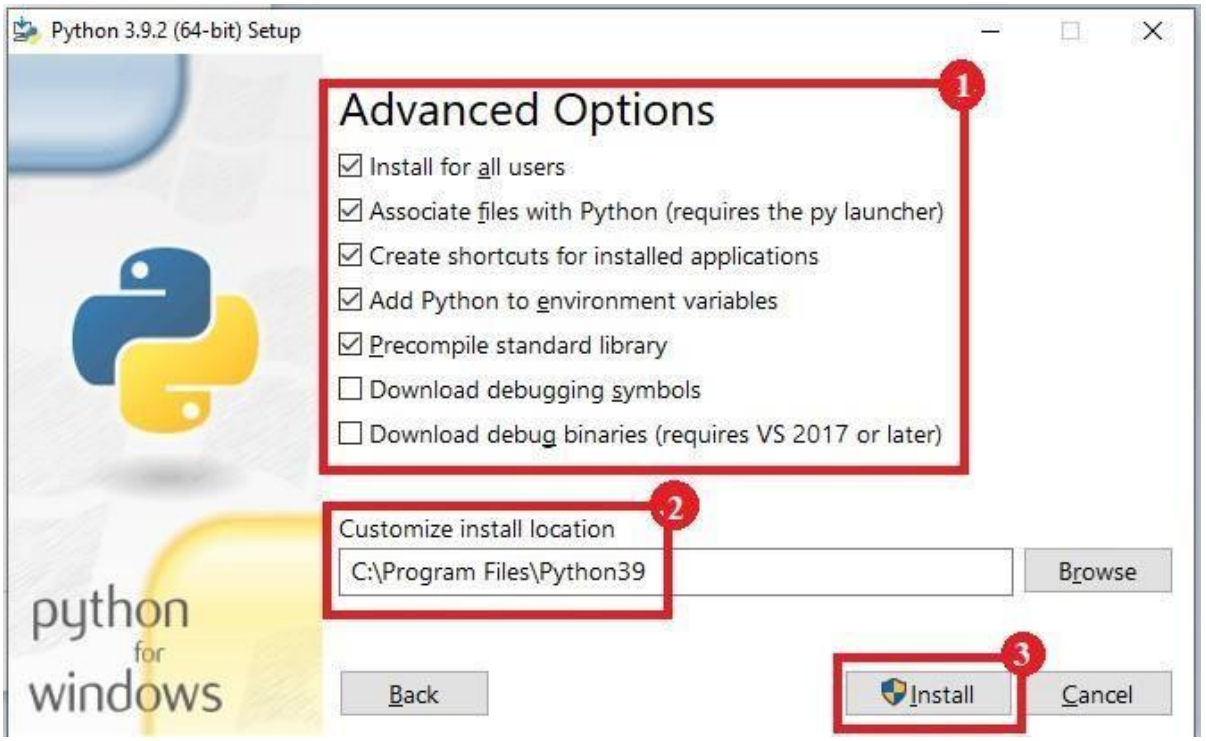
Step 3) Select Customize Installation

Step 4) Click NEXT



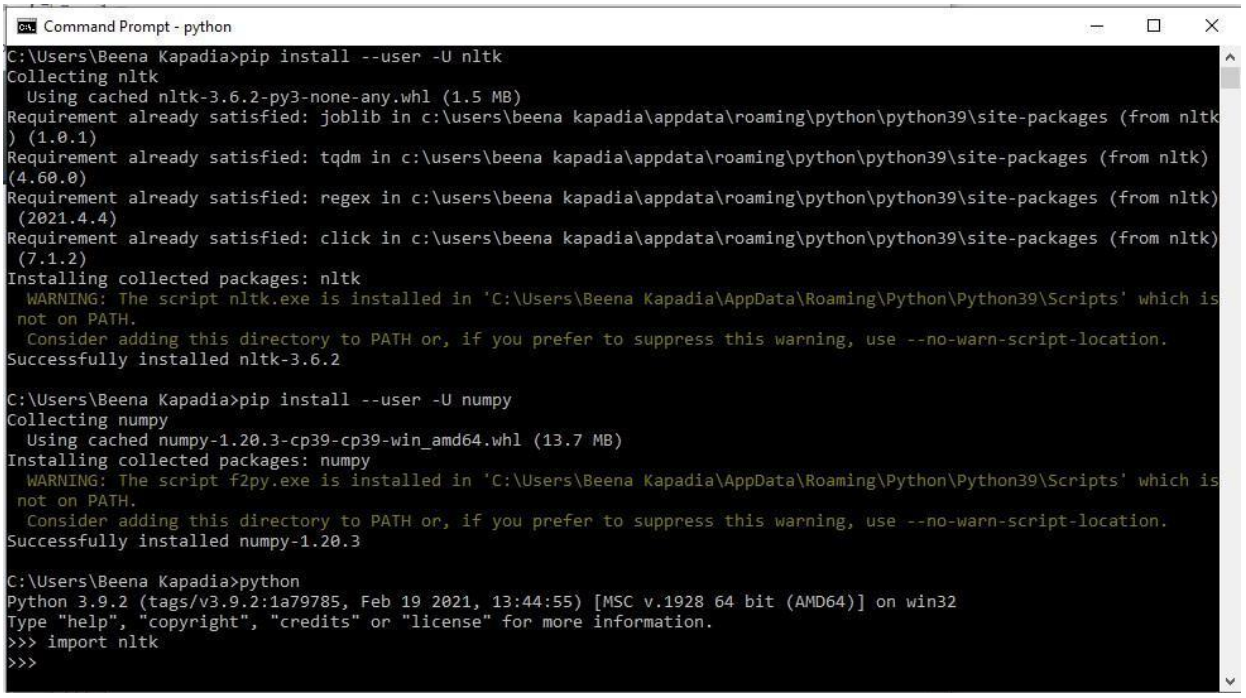
Step 5) In next screen

1. Select the advanced options
2. Give a Custom install location. Keep the default folder as c:\Program files\Python39
3. Click Install



Step 6) Click Close button once install is done.

Step 7) open command prompt window and run the following commands:
C:\Users\Beena Kapadia>pip install --upgrade pip
C:\Users\Beena Kapadia> pip install --user -U nltk
C:\Users\Beena Kapadia> >pip install --user -U numpy
C:\Users\Beena Kapadia>python
>>> import nltk
>>>



b) Convert the given text to speech.

Source code:

```
# text to speech
# pip install gtts
# pip install playsound

from playsound import playsound
# import required for text to speech conversion

from gtts import gTTS

mytext = "Welcome to Natural Language programming" language = "en"
myobj = gTTS(text=mytext, lang=language, slow=False) myobj.save("myfile.mp3")
playsound("myfile.mp3")
```

Output:

welcomeNLP.mp3 audio file is getting created and it plays the file with playsound() method, while running the program.

c) Convert audio file Speech to Text. Source code:

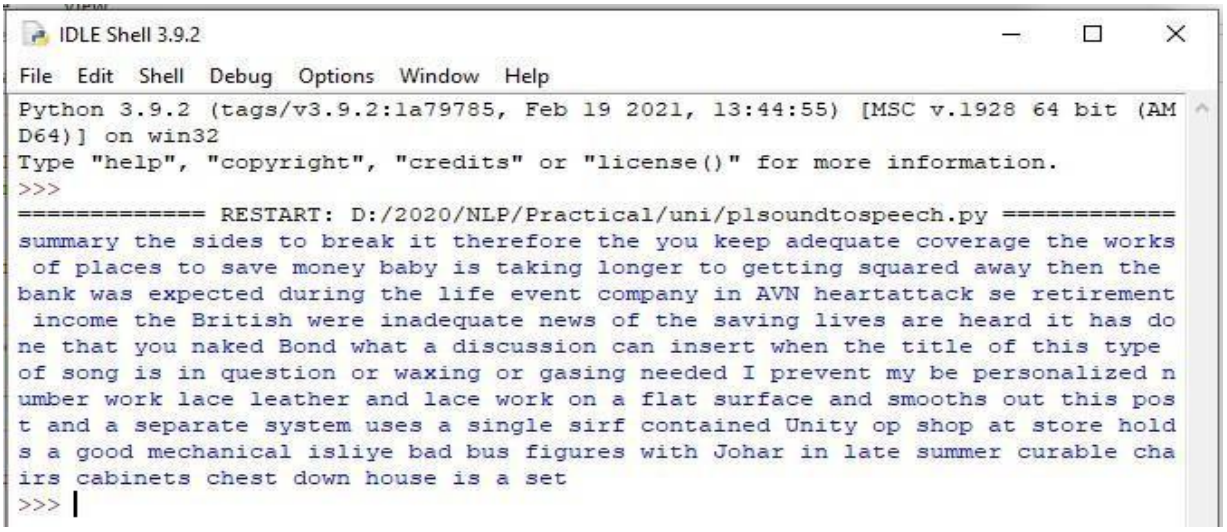
Note: required to store the input file "male.wav" in the current folder before running the program.

```
#pip3 install SpeechRecognition pydub

import speech_recognition as sr
filename = "male.wav"

# initialize the recognizer
r = sr.Recognizer()

# open the file with sr.AudioFile(filename) as source:     # listen for the data (load audio to
memory)     audio_data = r.record(source)
   # recognize (convert from speech to text)     text = r.recognize_google(audio_data)
print(text)
```

Input:

male.wav (any wav file)

Output:


IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/2020/NLP/Practical/uni/plsoundtospeech.py ==========
summary the sides to break it therefore the you keep adequate coverage the works
of places to save money baby is taking longer to getting squared away then the
bank was expected during the life event company in AVN heartattack se retirement
income the British were inadequate news of the saving lives are heard it has do
ne that you naked Bond what a discussion can insert when the title of this type
of song is in question or waxing or gasing needed I prevent my be personalized n
umber work lace leather and lace work on a flat surface and smooths out this pos
t and a separate system uses a single sirf contained Unity op shop at store hold
s a good mechanical isliye bad bus figures with Johar in late summer curable cha
irs cabinets chest down house is a set
>>>

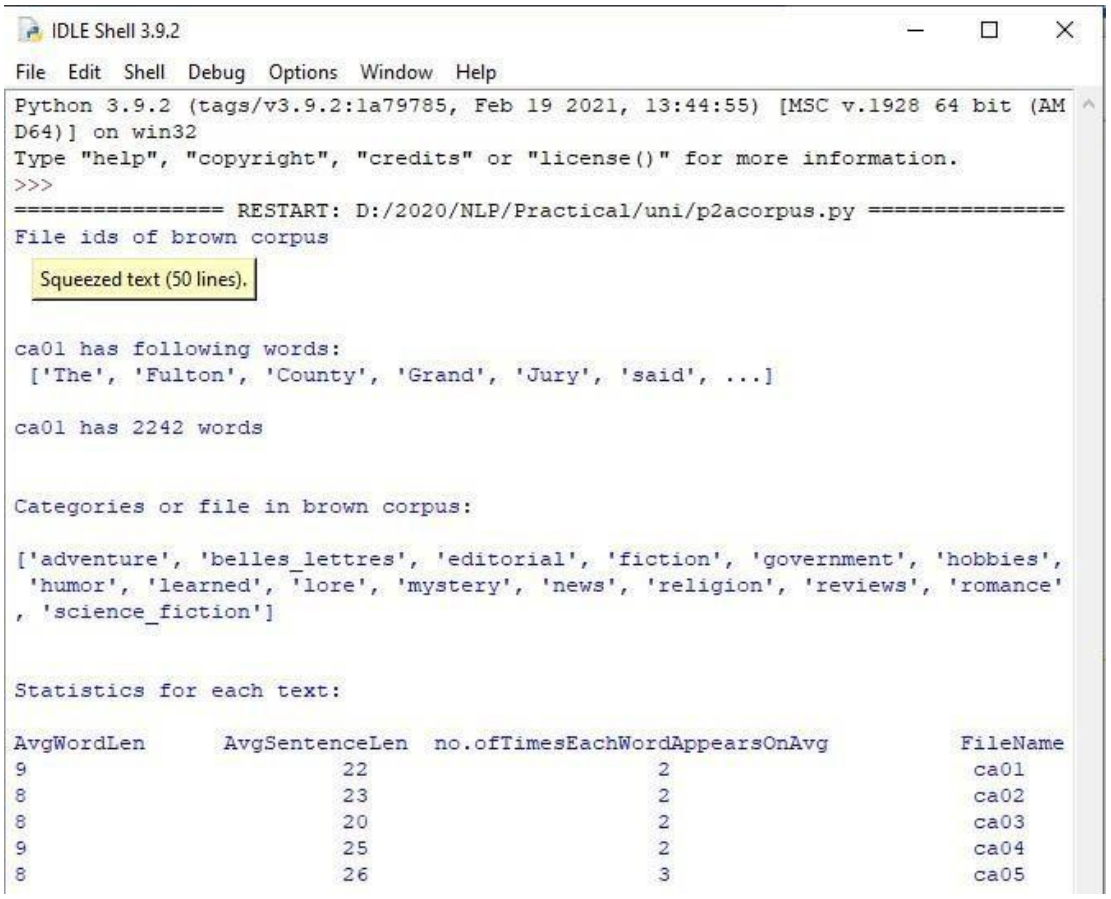
Practical 2

a. Study of various Corpus – Brown, Inaugural, Reuters, udhr with various methods like fields, raw, words, sents, categories,

source code:

```
'''NLTK includes a small selection of texts from the Project brown electronic text archive, which
  contains some 25,000 free electronic books, hosted at http://www.brown.org/. We begin by
  getting the Python interpreter to load the NLTK package, then ask to see
  nltk.corpus.brown.fileids(), the file identifiers in this corpus:'''

  import nltk
  from nltk.corpus import brown
  print ('File ids of brown corpus\n',brown.fileids())

  '''Let's pick out the first of these texts — Emma by Jane Austen — and give it a short name,
  emma, then find out how many words it contains:''' ca01 = brown.words('ca01')

  # display first few words
  print('\nca01 has following words:\n',ca01)

  # total number of words in ca01
  print('\nca01 has',len(ca01),'words')


  #categories or files print ('\n\nCategories or file in brown corpus:\n') print
  (brown.categories())

'''display other information about each text, by looping over all the values of fileid
  corresponding to the brown file identifiers listed earlier and then computing statistics for
  each text.'''
  print ('\n\nStatistics for each text:\n') print
('AvgWordLen\tAvgSentenceLen\tno.ofTimesEachWordAppearsOnAvg\t\tFileName') for fileid in
  brown.fileids():
     num_chars = len(brown.raw(fileid))    num_words = len(brown.words(fileid))    num_sents
  = len(brown.sents(fileid))
     num_vocab = len(set([w.lower() for w in brown.words(fileid)]))
     print (int(num_chars/num_words),'\t\t\t', int(num_words/num_sents),'\t\t\t',
  int(num_words/num_vocab),'\t\t\t', fileid)
```

output:



b. Create and use your own corpora (plaintext, categorical) source code:

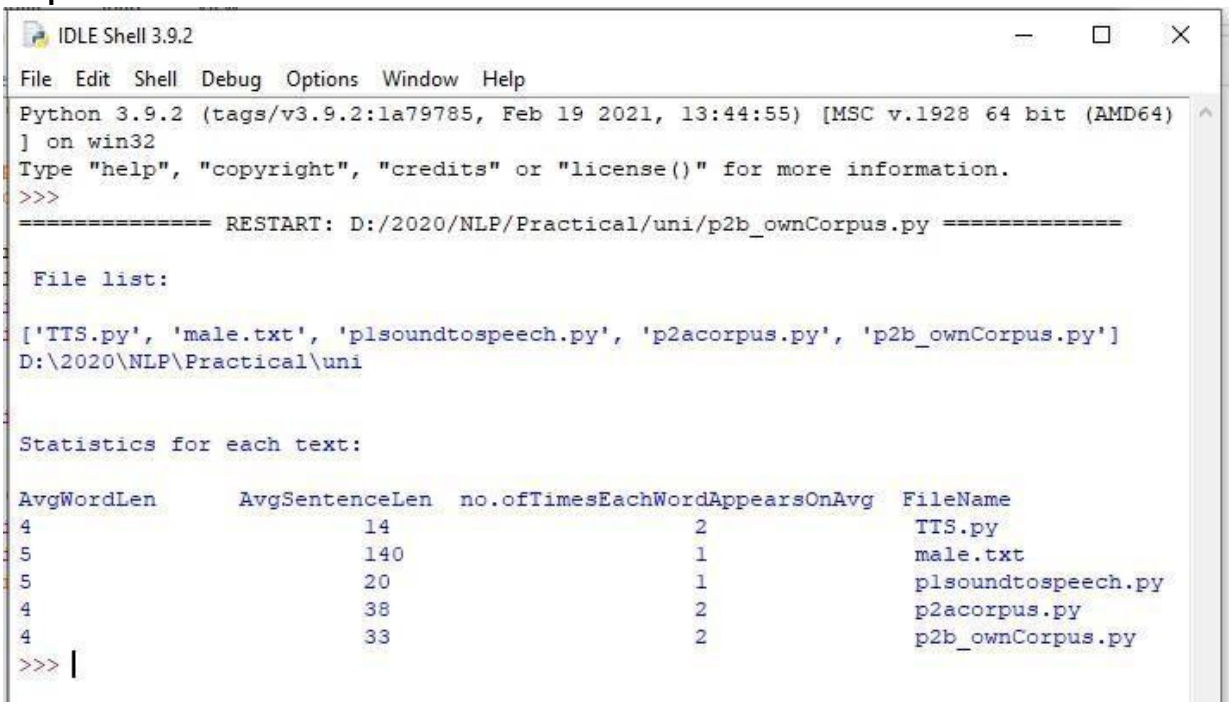
'"NLTK includes a small selection of texts from the Project filelist electronic text archive, which contains some 25,000 free electronic books, hosted at http://www.filelist.org/. We begin by getting the Python interpreter to load the NLTK package, then ask to see nltk.corpus.filelist.fileids(), the file identifiers in this corpus:'"

import nltk from nltk.corpus import PlaintextCorpusReader

corpus_root = 'D:/2020/NLP/Practical/uni' filelist = PlaintextCorpusReader(corpus_root, '.*') print ('\n File list: \n')

```
print (filelist.fileids())


print (filelist.root)


'''display other information about each text, by looping over all the values of fileid
corresponding to the filelist file identifiers listed earlier and then computing statistics for each
text.'''
print ('\n\nStatistics for each text:\n') print
('AvgWordLen\tAvgSentenceLen\tno.ofTimesEachWordAppearsOnAvg\tFileName')
for fileid in filelist.fileids():
    num_chars = len(filelist.raw(fileid))     num_words = len(filelist.words(fileid))     num_sents
= len(filelist.sents(fileid))
    num_vocab = len(set([w.lower() for w in filelist.words(fileid)]))     print
(int(num_chars/num_words),'\t\t\t', int(num_words/num_sents),'\t\t\t',
int(num_words/num_vocab),'\t\t', fileid)
```

output:

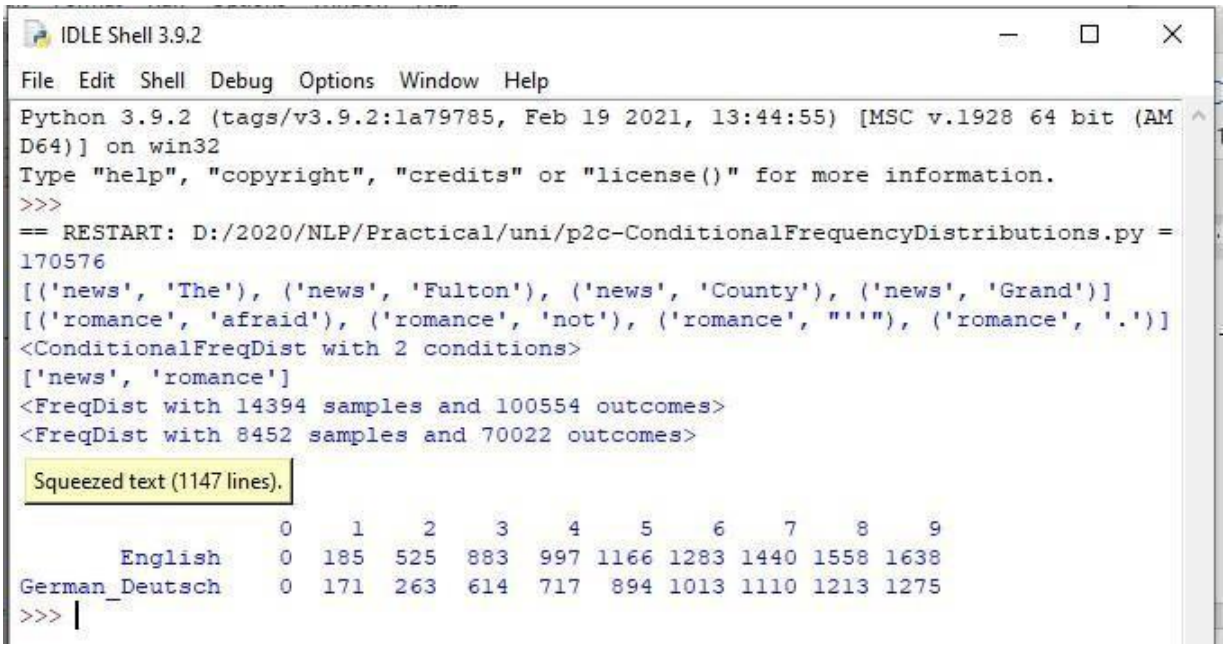


c. Study Conditional frequency distributions source code:

```
#process a sequence of pairs
text = ['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...] pairs = [('news', 'The'), ('news',
'Fulton'), ('news', 'County'), ...] import nltk
from nltk.corpus import brown fd = nltk.ConditionalFreqDist(
        (genre, word)
        for genre in brown.categories()
        for word in brown.words(categories=genre))
```

```
genre_word = [(genre, word)
        for genre in ['news', 'romance']
        for word in brown.words(categories=genre)]

print(len(genre_word))

print(genre_word[:4])

print(genre_word[-4:])

cfd = nltk.ConditionalFreqDist(genre_word)

print(cfd)

print(cfd.conditions())

print(cfd['news']) print(cfd['romance'])
print(list(cfd['romance']))

from nltk.corpus import inaugural
cfd = nltk.ConditionalFreqDist(
      (target, fileid[:4])      for fileid in inaugural.fileids()      for w in
inaugural.words(fileid)      for target in ['america', 'citizen']
      if w.lower().startswith(target))

from nltk.corpus import udhr languages = ['Chickasaw', 'English', 'German_Deutsch',
'Greenlandic_Inuktikut', 'Hungarian_Magyar', 'Ibibio_Efik'] cfd =
nltk.ConditionalFreqDist( (lang, len(word))      for lang in languages
      for word in udhr.words(lang + '-Latin1'))

cfd.tabulate(conditions=['English', 'German_Deutsch'],      samples=range(10),
cumulative=True)
```

output:



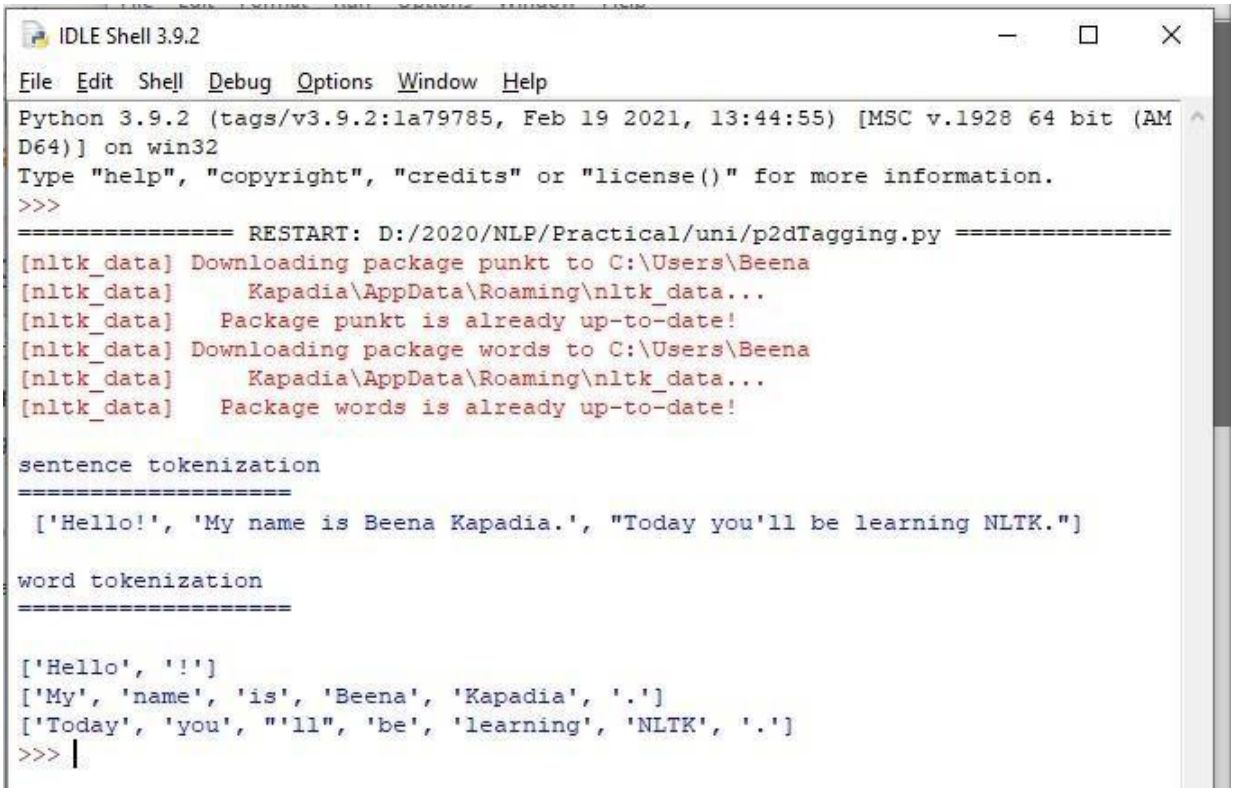
d. Study of tagged corpora with methods like tagged_sents, tagged_words.

Source code: import nltk from nltk import tokenize nltk.download('punkt') nltk.download('words')

para = "Hello! My name is Beena Kapadia. Today you'll be learning NLTK." sents = tokenize.sent_tokenize(para)
print("\nsentence tokenization\n==================\n",sents)

word tokenization print("\nword tokenization\n==================\n") for index in range(len(sents)): words = tokenize.word_tokenize(sents[index]) print(words)

output:



e. Write a program to find the most frequent noun tags. Code:

import nltk
from collections import defaultdict
text = nltk.word_tokenize("Nick likes to play football. Nick does not like to play cricket.")
tagged = nltk.pos_tag(text) print(tagged)

# checking if it is a noun or not addNounWords = [] count=0 for words in tagged:
    val = tagged[count][1]   if(val == 'NN' or val == 'NNS' or val == 'NNPS' or val == 'NNP'):
addNounWords.append(tagged[count][0])   count+=1

print (addNounWords)

temp = defaultdict(int)
# memoizing count for sub in addNounWords: for wrd in sub.split(): temp[wrd] +=

1 # getting max frequency

res = max(temp, key=temp.get)

# printing result
print("Word with maximum frequency : " + str(res))
output:

```
=============== RESTART: D:/2020/NLP/Practical/uni/p2emostFreq.py ==============
[('Nick', 'NNP'), ('likes', 'VBZ'), ('to', 'TO'), ('play', 'VB'), ('football', '
NN'), ('.', '.'), ('Nick', 'NNP'), ('does', 'VBZ'), ('not', 'RB'), ('like', 'VB'
), ('to', 'TO'), ('play', 'VB'), ('cricket', 'NN'), ('.', '.')]
['Nick', 'football', 'Nick', 'cricket']
Word with maximum frequency : Nick
>>>
```

f. Map Words to Properties Using Python Dictionaries code:
#creating and printing a dictionay by mapping word with its properties thisdict = { "brand":
"Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict) print(thisdict["brand"]) print(len(thisdict))
print(type(thisdict))

output:

```
================= RESTART: D:/2020/NLP/Practical/uni/p2fMap.py =================
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
Ford
3
<class 'dict'>
```

g. Study i) DefaultTagger, ii) Regular expression tagger, iii) UnigramTagger

i) DefaultTagger code:
import nltk from nltk.tag import DefaultTagger exptagger = DefaultTagger('NN') from
nltk.corpus import treebank testsentences = treebank.tagged_sents() [1000:]
print(exptagger.evaluate (testsentences))

#Tagging a list of sentences import nltk from nltk.tag import DefaultTagger
exptagger = DefaultTagger('NN')
print(exptagger.tag_sents([['Hi', ','], ['How', 'are', 'you', '?']]))

output

```
============ RESTART: D:/2020/NLP/Practical/uni/p2g1DefaultTagger.py ===========
0.13198749536374715
[[('Hi', 'NN'), (',', 'NN')], [('How', 'NN'), ('are', 'NN'), ('you', 'NN'), ('?'
, 'NN')]]
>>>
```

ii) Regular expression tagger, code: from nltk.corpus import brown from nltk.tag
import RegexpTagger test_sent = brown.sents(categories='news')[0]

```
regexp_tagger = RegexpTagger(
    [(r'^-?[0-9]+(.[0-9]+)?$', 'CD'), # cardinal numbers
     (r'(The|the|A|a|An|an)$', 'AT'), # articles
     (r'.*able$', 'JJ'),              # adjectives
     (r'.*ness$', 'NN'),              # nouns formed from adjectives
     (r'.*ly$', 'RB'),                # adverbs
     (r'.*s$', 'NNS'),                # plural nouns
     (r'.*ing$', 'VBG'),              # gerunds
     (r'.*ed$', 'VBD'),               # past tense verbs
     (r'.*', 'NN')                    # nouns (default)
])
print(regexp_tagger) print(regexp_tagger.tag(test_sent)) output:
```

```
============ RESTART: D:/2020/NLP/Practical/uni/p2g2RegularExp.py ============
<Regexp Tagger: size=9>
[('The', 'AT'), ('Fulton', 'NN'), ('County', 'NN'), ('Grand', 'NN'), ('Jury', 'N
N'), ('said', 'NN'), ('Friday', 'NN'), ('an', 'AT'), ('investigation', 'NN'), ('
of', 'NN'), ("Atlanta's", 'NNS'), ('recent', 'NN'), ('primary', 'NN'), ('electio
n', 'NN'), ('produced', 'VBD'), ('``', 'NN'), ('no', 'NN'), ('evidence', 'NN'),
("''", 'NN'), ('that', 'NN'), ('any', 'NN'), ('irregularities', 'NNS'), ('took',
 'NN'), ('place', 'NN'), ('.', 'NN')]
```

iii) UnigramTagger code:

```
# Loading Libraries from nltk.tag import UnigramTagger
from nltk.corpus import treebank

# Training using first 10 tagged sentences of the treebank corpus as data.
# Using data
train_sents = treebank.tagged_sents()[:10]

# Initializing
tagger = UnigramTagger(train_sents)

# Lets see the first sentence # (of the treebank corpus) as list
print(treebank.sents()[0]) print('\n',tagger.tag(treebank.sents()[0]))

#Finding the tagged results after training.
tagger.tag(treebank.sents()[0])

#Overriding the context model
tagger = UnigramTagger(model ={'Pierre': 'NN'})
print('\n',tagger.tag(treebank.sents()[0])) output:
```

```
=============== RESTART: D:/2020/NLP/Practical/uni/p2g3Unigram.py ==============
['Pierre', 'Vinken', ',', '61', 'years', 'old', ',', 'will', 'join', 'the', 'boa
rd', 'as', 'a', 'nonexecutive', 'director', 'Nov.', '29', '.']

 [('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ('61', 'CD'), ('years', 'NNS
'), ('old', 'JJ'), (',', ','), ('will', 'MD'), ('join', 'VB'), ('the', 'DT'), ('
board', 'NN'), ('as', 'IN'), ('a', 'DT'), ('nonexecutive', 'JJ'), ('director', '
NN'), ('Nov.', 'NNP'), ('29', 'CD'), ('.', '.')]

 [('Pierre', 'NN'), ('Vinken', None), (',', None), ('61', None), ('years', None)
, ('old', None), (',', None), ('will', None), ('join', None), ('the', None), ('b
oard', None), ('as', None), ('a', None), ('nonexecutive', None), ('director', No
ne), ('Nov.', None), ('29', None), ('.', None)]
>>>
```

h. Find different words from a given plain text without any space by comparing this text with a given corpus of words. Also find the score of words.

Question:

Initialize the hash tag test data or URL test data and convert to plain text without any space.. Read a text file of different words and compare the plain text data with the words exist in that text file and find out different words available in that plain text. Also find out how many words could be found. (for example, text = "#whatismyname" or text = www.whatismyname.com. Convert that to plain text without space as:
whatismyname and read text file as words.txt. Now compare plain text with words given in a file and find the words form the plain text and the count of words which could be found)

Source code:

```
from__future___import with_statement #with statement for reading file import re  # Regular expression

words = [] # corpus file words testword = [] # test words        ans = []    # words matches with corpus

print("MENU")
print("             ") print(" 1 . Hash tag segmentation ") print(" 2 . URL segmentation ")
print("enter the input choice for performing word segmentation") choice = int(input())

if choice == 1:     text = "#whatismyname"         # hash tag test data to segment
print("input with HashTag",text)                    pattern=re.compile("[^\w']")           a =
pattern.sub('', text) elif choice == 2:
   text = "www.whatismyname.com"       # url test data to segment     print("input with
URL",text)     a=re.split('\s|(?<!\d)[,.](?!\d)', text)
   splitwords = ["www","com","in"]      # remove the words which is containg in the list     a
="".join([each for each in a if each not in splitwords]) else:     print("wrong choice   try again")
print(a)

for each in a:     testword.append(each) #test word
test_lenth = len(testword)       # lenth of the test data

# Reading the corpus with open('words.txt', 'r') as f:     lines = f.readlines()
   words =[(e.strip()) for e in lines]

def Seg(a,lenth):
   ans =[]     for k in range(0,lenth+1): # this loop checks char by char in the corpus

      if a[0:k] in words:
         print(a[0:k],"-appears in the corpus")            ans.append(a[0:k])
         break     if ans != []:
      g = max(ans,key=len)
      return g

test_tot_itr = 0    #each iteration value
answer = []     # Store the each word contains the corpus
Score = 0 # initial value for score
N = 37    # total no of corpus
M = 0 C = 0 while test_tot_itr < test_lenth:        ans_words = Seg(a,test_lenth)     if
ans_words != 0:
      test_itr = len(ans_words)        answer.append(ans_words)        a = a[test_itr:test_lenth]
      test_tot_itr += test_itr
```

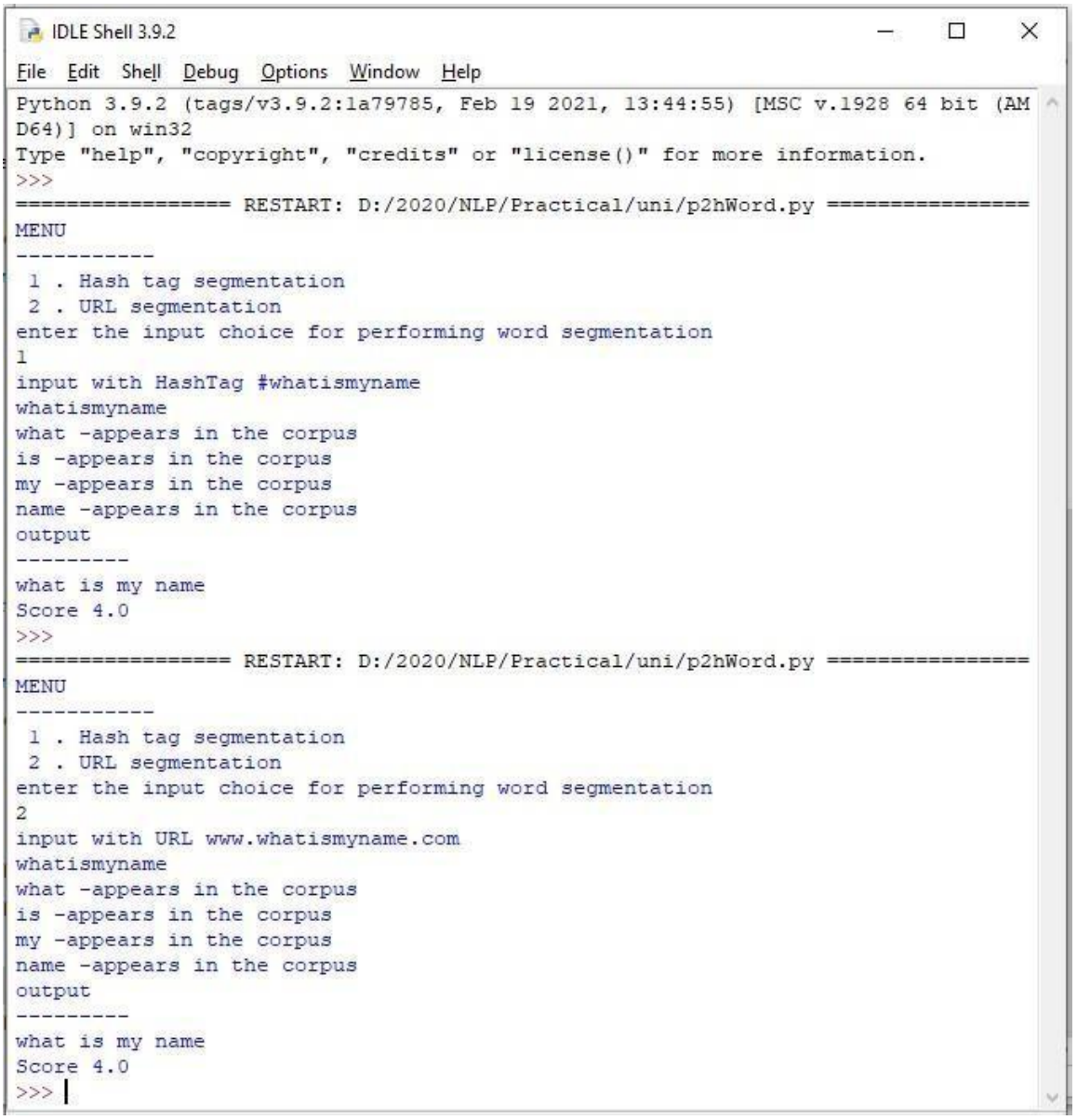
```
Aft_Seg = " ".join([each for each in answer]) # print segmented words in the list
print("output") print("           ")
print(Aft_Seg) # print After segmentation the input

# Calculating Score C = len(answer)
score = C * N / N       # Calculate the
score print("Score",score)
```

Input:

Words.txt

```
_________-
check domain big rocks name cheap being human back
current rates social
ought to media
go 30
down apple domains honesty hour follow seconds
earth
this is
insane it
time
what is
my
name
let us
go
```

Output:


IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/2020/NLP/Practical/uni/p2hWord.py ==========
MENU
----
1 . Hash tag segmentation
2 . URL segmentation
enter the input choice for performing word segmentation
1
input with HashTag #whatismyname
whatismyname
what -appears in the corpus
is -appears in the corpus
my -appears in the corpus
name -appears in the corpus
output
----
what is my name
Score 4.0
>>>
========== RESTART: D:/2020/NLP/Practical/uni/p2hWord.py ==========
MENU
----
1 . Hash tag segmentation
2 . URL segmentation
enter the input choice for performing word segmentation
2
input with URL www.whatismyname.com
whatismyname
what -appears in the corpus
is -appears in the corpus
my -appears in the corpus
name -appears in the corpus
output
----
what is my name
Score 4.0
>>>

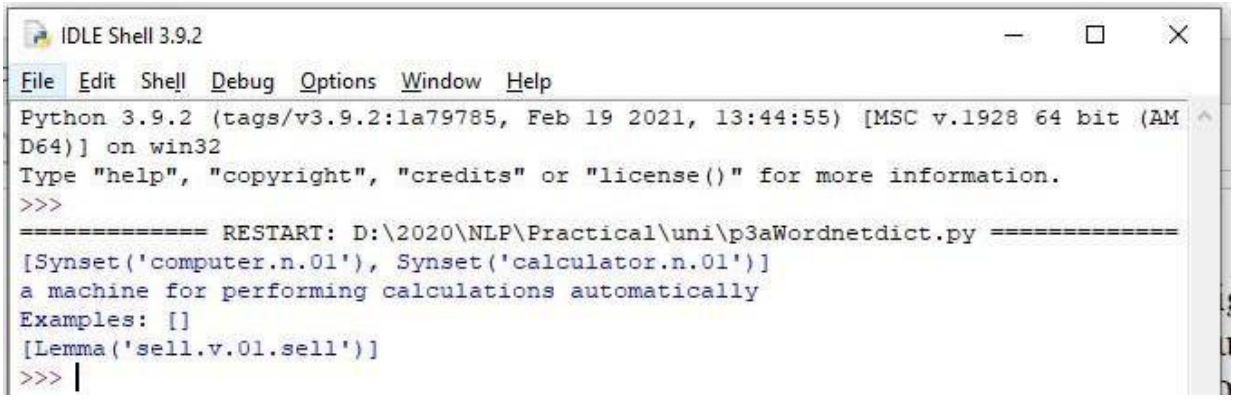
Practical 3

a. **Study of Wordnet Dictionary with methods as synsets, definitions, examples, antonyms**

Source code:

```
'''WordNet provides synsets which is the collection of synonym words also called
      “lemmas”''' import nltk from nltk.corpus import wordnet
print(wordnet.synsets("computer"))

# definition and example of the word ‘computer’
print(wordnet.synset("computer.n.01").definition())

#examples
print("Examples:", wordnet.synset("computer.n.01").examples())

#get Antonyms
print(wordnet.lemma('buy.v.01.buy').antonyms())
```

output:



b. Study lemmas, hyponyms, hypernyms.

Source code:

```
import nltk from nltk.corpus import wordnet print(wordnet.synsets("computer"))
print(wordnet.synset("computer.n.01").lemma_names()) #all lemmas for each synset.
for e in wordnet.synsets("computer"):     print(f'{e} --> {e.lemma_names()}')

#print all lemmas for a given synset
print(wordnet.synset('computer.n.01').lemmas())

#get the synset corresponding to lemma
print(wordnet.lemma('computer.n.01.computing_device').synset())

#Get the name of the lemma
print(wordnet.lemma('computer.n.01.computing_device').name())

#Hyponyms give abstract concepts of the word that are much more specific #the list of
hyponyms words of the computer
```

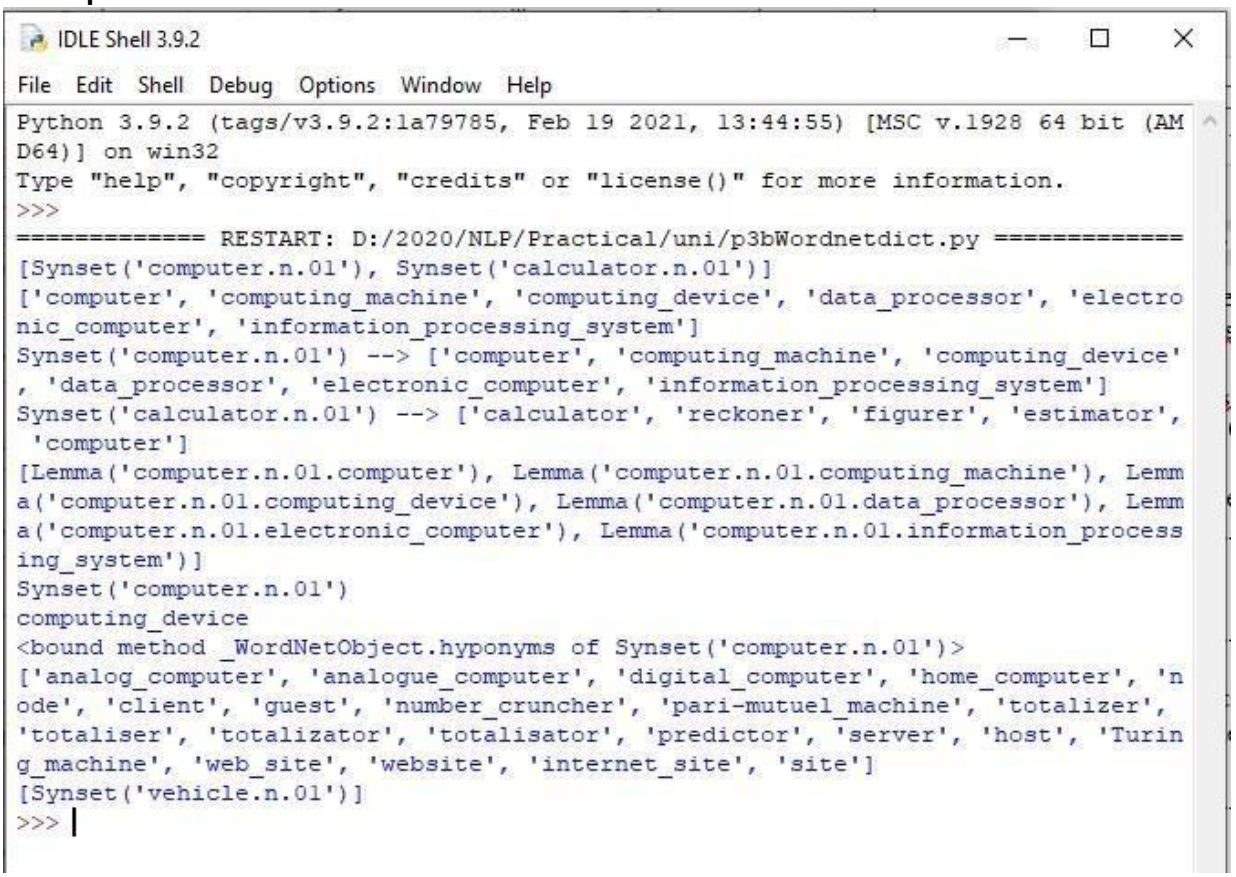
syn = wordnet.synset('computer.n.01') print(syn.hyponyms)

print([lemma.name() for synset in syn.hyponyms() for lemma in synset.lemmas()])

#the semantic similarity in WordNet vehicle = wordnet.synset('vehicle.n.01') car = wordnet.synset('car.n.01')

print(car.lowest_common_hypernyms(vehicle))

Output:

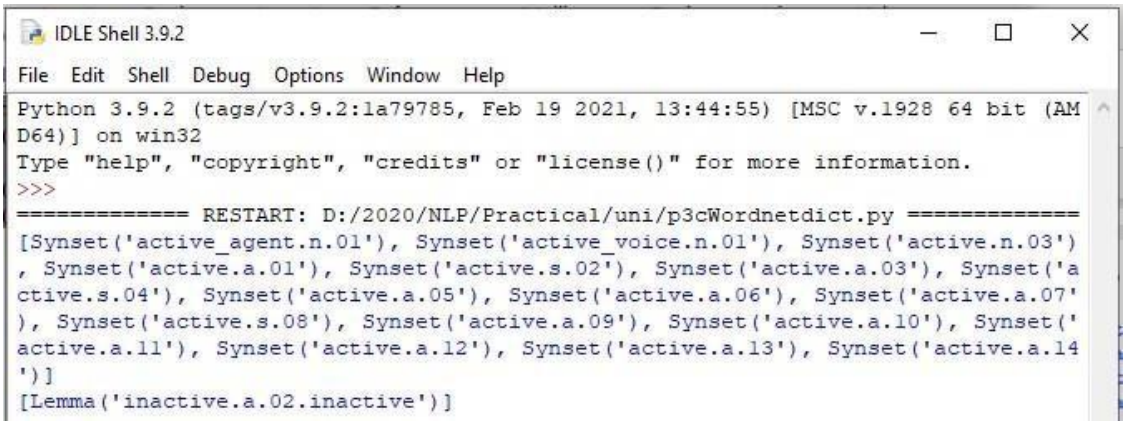


c. **Write a program using python to find synonym and antonym of word "active" using Wordnet.**

Source code: from nltk.corpus import wordnet print(wordnet.synsets("active"))

print(wordnet.lemma('active.a.01.active').antonyms())

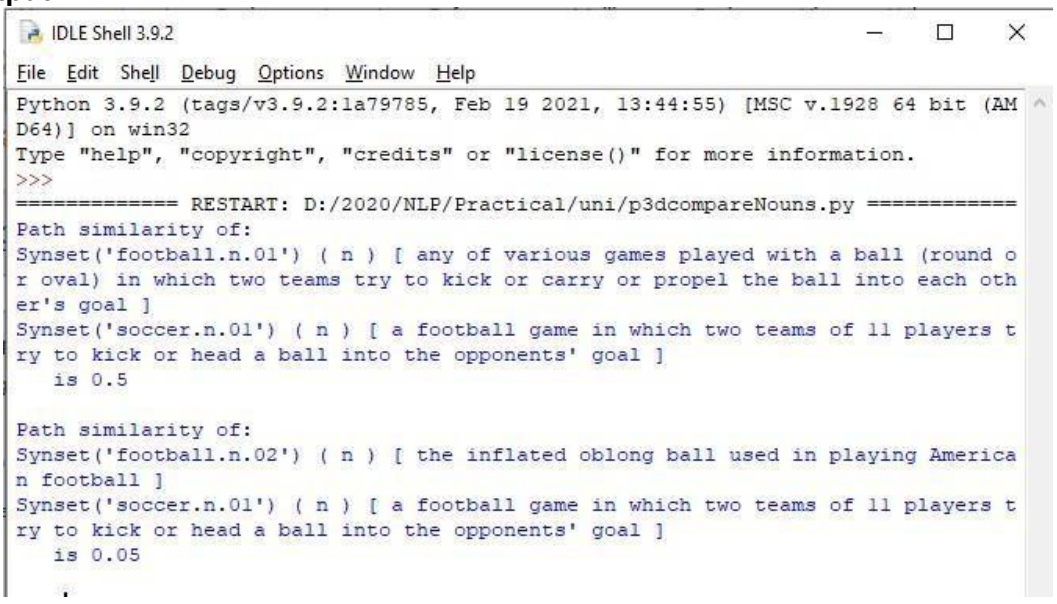
Output:



d. Compare two nouns source code:

```
import nltk
from nltk.corpus import wordnet

syn1 = wordnet.synsets('football')
syn2 = wordnet.synsets('soccer')

# A word may have multiple synsets, so need to compare each synset of word1 with synset of
word2
for s1 in syn1:    for s2 in syn2:        print("Path similarity of: ")        print(s1, '(', s1.pos(), ')',
'[', s1.definition(), ']')        print(s2, '(', s2.pos(), ')', '[', s2.definition(), ']')
      print(" is", s1.path_similarity(s2))          print()
```

output:



e. Handling stopword:

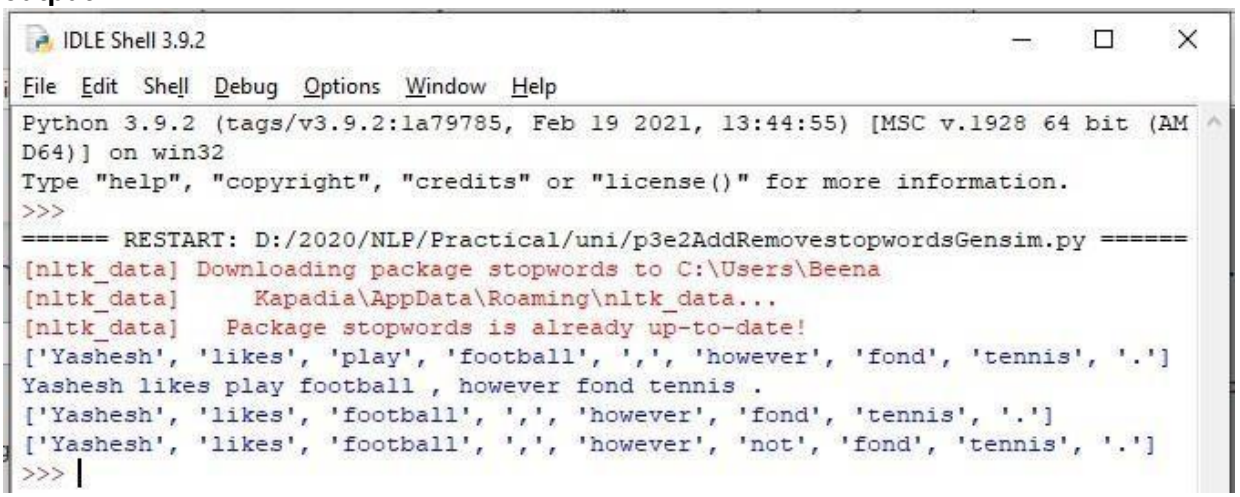
i) Using nltk Adding or Removing Stop Words in NLTK's Default Stop Word List

code:

```
import nltk from nltk.corpus import stopwords nltk.download('stopwords')
from nltk.tokenize import word_tokenize
```

```
text = "Yashesh likes to play football, however he is not too fond of tennis." text_tokens =
word_tokenize(text)

tokens_without_sw = [word for word in text_tokens if not word in stopwords.words()]

print(tokens_without_sw)

#add the word play to the NLTK stop word collection all_stopwords =
stopwords.words('english') all_stopwords.append('play')

text_tokens = word_tokenize(text)
tokens_without_sw = [word for word in text_tokens if not word in all_stopwords]

print(tokens_without_sw)

#remove 'not' from stop word collection all_stopwords.remove('not')

text_tokens = word_tokenize(text)
tokens_without_sw = [word for word in text_tokens if not word in all_stopwords]

print(tokens_without_sw)
```

output



ii) Using Gensim Adding and Removing Stop Words in Default Gensim Stop Words List code:

```
#pip install gensim import gensim
from gensim.parsing.preprocessing import remove_stopwords

text = "Yashesh likes to play football, however he is not too fond of tennis." filtered_sentence
= remove_stopwords(text)

print(filtered_sentence)

all_stopwords = gensim.parsing.preprocessing.STOPWORDS print(all_stopwords)
```

'''The following script adds likes and play to the list of stop words in Gensim:'''

```
from gensim.parsing.preprocessing import STOPWORDS

all_stopwords_gensim = STOPWORDS.union(set(['likes', 'play']))

text = "Yashesh likes to play football, however he is not too fond of tennis." text_tokens =
word_tokenize(text)
tokens_without_sw = [word for word in text_tokens if not word in all_stopwords_gensim]

print(tokens_without_sw)

'''Output:

['Yashesh', 'football', ',', 'fond', 'tennis', '.']
The following script removes the word "not" from the set of stop words in Gensim:'''
from gensim.parsing.preprocessing import STOPWORDS
all_stopwords_gensim = STOPWORDS sw_list = {"not"}
all_stopwords_gensim = STOPWORDS.difference(sw_list)

text = "Yashesh likes to play football, however he is not too fond of tennis." text_tokens =
word_tokenize(text)
tokens_without_sw = [word for word in text_tokens if not word in all_stopwords_gensim]

print(tokens_without_sw)
```

output

Microsoft Visual C++ 14.0 is required. Get it with "Build Tools for Visual Studio": https://visualstudio.microsoft.com/downloads/

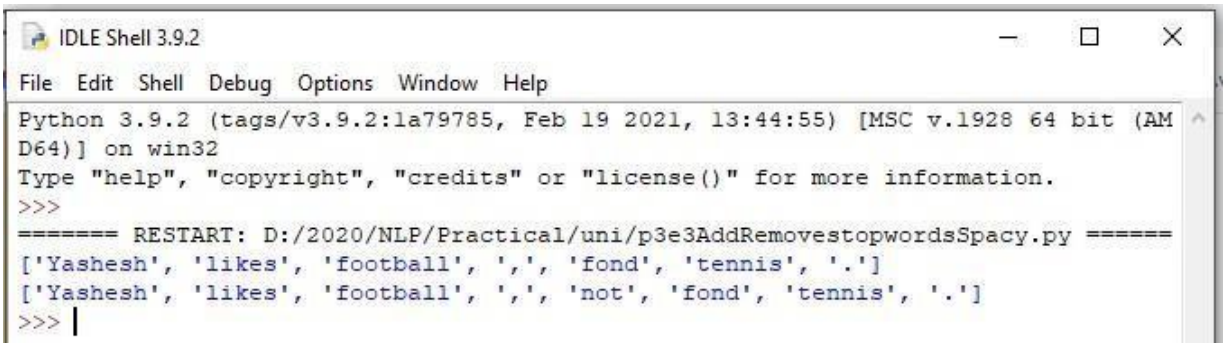
iii) Using Spacy Adding and Removing Stop Words in Default Spacy Stop Words List

code:

```
#pip install spacy
#python -m spacy download en_core_web_sm
#python -m spacy download en

import spacy import nltk
from nltk.tokenize import word_tokenize

sp = spacy.load('en_core_web_sm')

#add the word play to the NLTK stop word collection all_stopwords =
sp.Defaults.stop_words all_stopwords.add("play")

text = "Yashesh likes to play football, however he is not too fond of tennis." text_tokens =
word_tokenize(text)
```

```
tokens_without_sw = [word for word in text_tokens if not word in all_stopwords]

print(tokens_without_sw)

#remove 'not' from stop word collection all_stopwords.remove('not')

tokens_without_sw = [word for word in text_tokens if not word in all_stopwords]

print(tokens_without_sw)
```

output:



Practical 4

a. **Tokenization using Python's split() function code:** text = """ This tool is an a beta stage. Alexa developers can use Get Metrics API to seamlessly analyse metric. It also supports custom skill model, prebuilt Flash Briefing model, and the Smart Home Skill API. You can use this tool for creation of monitors, alarms, and dashboards that spotlight changes. The release of these three tools will enable developers to create visual rich skills for Alexa devices with screens. Amazon describes these tools as the collection of tech and tools for creating visually rich and interactive voice experiences. """
data = text.split('.') for i in data: print (i)

output:

```
>>>
========== RESTART: D:/2020/NLP/Practical/uni/p4a.py ==========
This tool is an a beta stage
 Alexa developers can use Get Metrics API to seamlessly analyse metric
 It also supports custom skill model, prebuilt Flash Briefing model, and the Sma
rt Home Skill API
 You can use this tool for creation of monitors, alarms, and dashboards that spo
tlight changes
 The release of these three tools will enable developers to create visual rich s
kills for Alexa devices with screens
 Amazon describes these tools as the collection of tech and tools for creating v
isually rich and interactive voice experiences
```

b. **Tokenization using Regular Expressions (RegEx)**

code:

```
import nltk
# import RegexpTokenizer() method from nltk
from nltk.tokenize import RegexpTokenizer

# Create a reference variable for Class RegexpTokenizer tk = RegexpTokenizer('\s+', gaps =
True)

# Create a string input
str = "I love to study Natural Language Processing in Python"

# Use tokenize method
tokens = tk.tokenize(str)

print(tokens)
```

output:

```
>>>
========== RESTART: D:/2020/NLP/Practical/uni/p4b.py ==========
['I', 'love', 'to', 'study', 'Natural', 'Language', 'Processing', 'in', 'Python'
]
>>>
```

c. Tokenization using NLTK

code:

```
import nltk
from nltk.tokenize import word_tokenize

# Create a string input
str = "I love to study Natural Language Processing in Python"

# Use tokenize method
print(word_tokenize(str))
```

output:

```
=================== RESTART: D:/2020/NLP/Practical/uni/p4c.py ==================
['I', 'love', 'to', 'study', 'Natural', 'Language', 'Processing', 'in', 'Python'
]
>>>
```

d. Tokenization using the spaCy library

code:

```
import spacy
nlp = spacy.blank("en")

# Create a string input
str = "I love to study Natural Language Processing in Python"

# Create an instance of document;
# doc object is a container for a sequence of Token objects. doc = nlp(str)

# Read the words; Print the words
# words = [word.text for word in doc] print(words)
```

output:

```
=================== RESTART: D:/2020/NLP/Practical/uni/p4d.py ==================
['I', 'love', 'to', 'study', 'Natural', 'Language', 'Processing', 'in', 'Python'
]
>>>
```

e. Tokenization using Keras code:

```
#pip install keras #pip install tensorflow import keras
from keras.preprocessing.text import text_to_word_sequence

# Create a string input
str = "I love to study Natural Language Processing in Python"

# tokenizing the text
tokens = text_to_word_sequence(str)
print(tokens)
```

output:

```
>>>
========== RESTART: D:\2020\NLP\Practical\uni\p4e.py ==========
['i', 'love', 'to', 'study', 'natural', 'language', 'processing', 'in', 'python'
]
```

f. Tokenization using Gensim

code:

```
#pip install gensim
from gensim.utils import tokenize

# Create a string input

str = "I love to study Natural Language Processing in Python"

# tokenizing the text
list(tokenize(str))
```

output:

Microsoft Visual C++ 14.0 is required. Get it with "Build Tools for Visual Studio":
https://visualstudio.microsoft.com/downloads/

Practical 5

Import NLP Libraries for Indian Languages and perform: Note: Execute this practical in https://colab.research.google.com/

a) word tokenization in Hindi Source code:

```
!pip install torch==1.3.1+cpu -f https://download.pytorch.org/whl/torch_stable.html

!pip install inltk

!pip install tornado==4.5.3

from inltk.inltk import setup
setup('hi')

from inltk.inltk import tokenize

hindi_text = """प्राकृ तिक भाषा सीखना बहुत तिलचस्प है।"""

# tokenize(input text, language code) tokenize(hindi_text, "hi")
```

output

['_प्राकृ तिक', '_भाषा', '_सीखना', '_बहुत ', '_तिलचस्प', '_है', '।']

b) Generate similar sentences from a given Hindi text input Source code:

```
!pip install torch==1.3.1+cpu -f https://download.pytorch.org/whl/torch_stable.html

!pip install inltk

!pip install tornado==4.5.3

from inltk.inltk import setup
setup('hi')

from inltk.inltk import get_similar_sentences

# get similar sentences to the one given in hindi
output = get_similar_sentences('मैं आज बहुत खुश हूं', 5, 'hi')

print(output)
```

Output:

['मैं आजकल बहुत खुश हूं', 'मैं आज अत्यतिक खुश हूं', 'मैं अभी बहुत खुश हूं', 'मैं तिमान बहुत खुश हूं', 'मैं तिमान बहुत खुश हूं']

c) Identify the Indian language of a text Source code:

```
!pip install torch==1.3.1+cpu -f https://download.pytorch.org/whl/torch_stable.html

!pip install inltk
```

```
!pip install tornado==4.5.3

from inltk.inltk import setup setup('gu')

from inltk.inltk import identify_language #Identify the Lnaguage of given text
identify_language('બીના કાપડિયા')
```

Output: gujarati

Practical 6

a. **Part of speech Tagging and chunking of user defined text.**
b. **Named Entity recognition of user defined text.**
c. **Named Entity recognition with diagram using NLTK corpus – treebank**

POS Tagging, ch unking and NER:
a) sentence tokenization, word tokenization, Part of speech Tagging and chunking of user defined text. Source code: import nltk from nltk import tokenize nltk.download('punkt') from nltk import tag from nltk import chunk nltk.download('averaged_perceptron_tagger') nltk.download('maxent_ne_chunker') nltk.download('words')

para = "Hello! My name is Beena Kapadia. Today you'll be learning NLTK." sents = tokenize.sent_tokenize(para)
print("\nsentence tokenization\n==========\n",sents)

word tokenization print("\nword tokenization\n==========\n") for index in range(len(sents)): words = tokenize.word_tokenize(sents[index]) print(words)

POS Tagging

tagged_words = [] for index in range(len(sents)):
tagged_words.append(tag.pos_tag(words))
print("\nPOS Tagging\n==========\n",tagged_words)

chunking

tree = [] for index in range(len(sents)):
tree.append(chunk.ne_chunk(tagged_words[index])) print("\nchunking\n========\n")
print(tree)

Output:
sentence tokenization
==========
['Hello!', 'My name is Beena Kapadia.', "Today you'll be learning NLTK."]

word tokenization
==========

['Hello', '!']
['My', 'name', 'is', 'Beena', 'Kapadia', '.']
['Today', 'you', "'ll", 'be', 'learning', 'NLTK', '.']

POS Tagging

==========
[[('Today', 'NN'), ('you', 'PRP'), ("'ll", 'MD'), ('be', 'VB'), ('learning', 'VBG'), ('NLTK', 'NNP'), ('.', '.')], [('Today', 'NN'), ('you', 'PRP'), ("'ll", 'MD'), ('be', 'VB'), ('learning', 'VBG'), ('NLTK', 'NNP'), ('.', '.')], [('Today', 'NN'), ('you', 'PRP'), ("'ll", 'MD'), ('be', 'VB'), ('learning', 'VBG'), ('NLTK', 'NNP'), ('.', '.')]]

chunking
========

[Tree('S', [('Today', 'NN'), ('you', 'PRP'), ("'ll", 'MD'), ('be', 'VB'), ('learning', 'VBG'), Tree('ORGANIZATION', [('NLTK', 'NNP')]), ('.', '.')]), Tree('S', [('Today', 'NN'), ('you', 'PRP'), ("'ll", 'MD'), ('be', 'VB'), ('learning', 'VBG'), Tree('ORGANIZATION', [('NLTK', 'NNP')]), ('.', '.')]), Tree('S', [('Today', 'NN'), ('you', 'PRP'), ("'ll", 'MD'), ('be', 'VB'), ('learning', 'VBG'), Tree('ORGANIZATION', [('NLTK', 'NNP')]), ('.', '.')])]

b) Named Entity recognition using user defined text. Source code:

```
!pip install -U spacy
!python -m spacy download en_core_web_sm
import spacy

# Load English tokenizer, tagger, parser and NER nlp = spacy.load("en_core_web_sm")

# Process whole documents text = ("When Sebastian Thrun started working on self-driving cars at "  "Google in 2007, few people outside of the company took him "
"seriously. “I can tell you very senior CEOs of major American "
    "car companies would shake my hand and turn away because I wasn’t "
    "worth talking to,” said Thrun, in an interview with Recode earlier "
    "this week.")
doc = nlp(text)

# Analyse syntax print("Noun phrases:", [chunk.text for chunk in doc.noun_chunks])
print("Verbs:", [token.lemma_ for token in doc if token.pos_ == "VERB"])
```

Output:

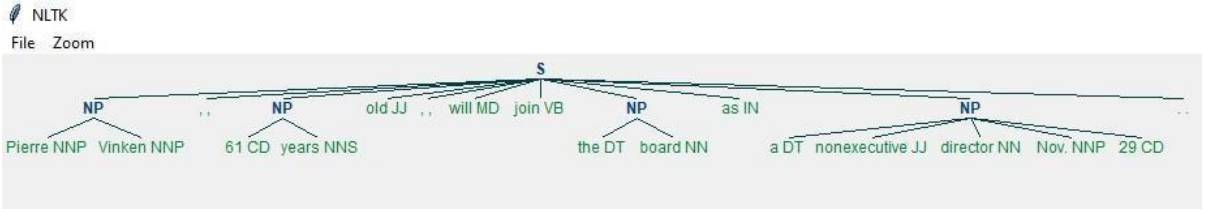
Noun phrases: ['Sebastian Thrun', 'self-driving cars', 'Google', 'few people', 'the company', 'him', 'I', 'you', 'very senior CEOs', 'major American car companies', 'my hand', 'I', 'Thrun', 'an interview', 'Recode']
Verbs: ['start', 'work', 'drive', 'take', 'tell', 'shake', 'turn', 'be', 'talk', 'say']

c) Named Entity recognition with diagram using NLTK corpus – treebank. Source code:

Note: It runs on Python IDLE

```
import nltk

nltk.download('treebank') from nltk.corpus import treebank_chunk
treebank_chunk.tagged_sents()[0]

treebank_chunk.chunked_sents()[0]
treebank_chunk.chunked_sents()[0].draw()
```

Output:

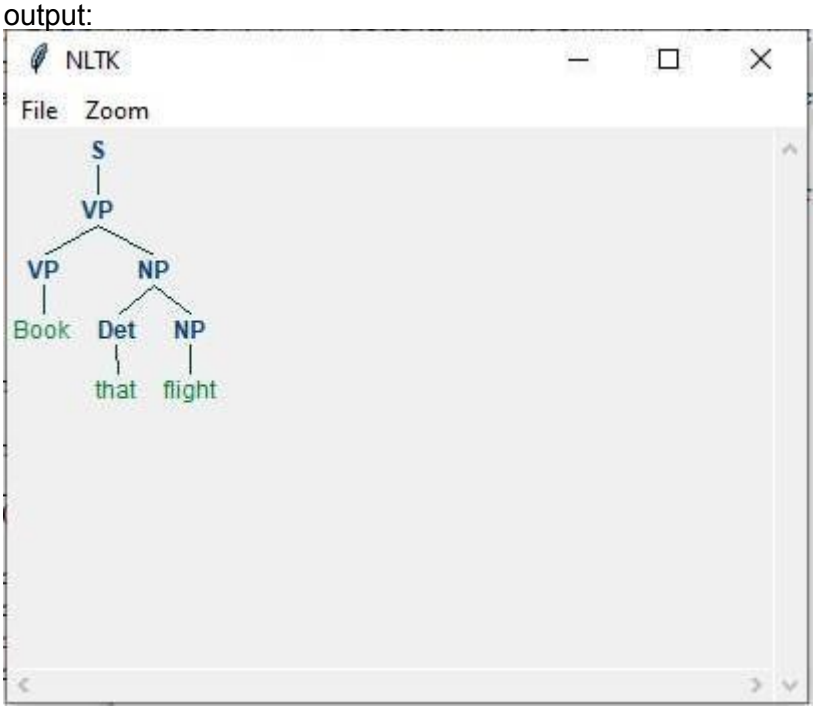

NLTK
File Zoom
S
NP
Pierre NNP Vinken NNP
, ,
NP
61 CD years NNS
old JJ
, ,
will MD
join VB
NP
the DT board NN
as IN
NP
a DT nonexecutive JJ director NN Nov. NNP 29 CD
. .

Practical 7

a) Define grammar using nltk. Analyze a sentence using the same. Code:

```
import nltk from nltk import tokenize grammar1 = nltk.CFG.fromstring("""
                    S -> VP
        VP -> VP NP
        NP -> Det NP
        Det -> 'that'
        NP -> singular Noun
        NP -> 'flight'
        VP -> 'Book'
                    """)
sentence = "Book that flight"

for index in range(len(sentence)):
  all_tokens = tokenize.word_tokenize(sentence)
print(all_tokens)

parser = nltk.ChartParser(grammar1) for tree in parser.parse(all_tokens):
    print(tree)
    tree.draw()
```

output:



b) Accept the input string with Regular expression of Finite Automaton: 101+.

Source code:

```
def FA(s):
#if the length is less than 3 then it can't be accepted, Therefore end the process.     if
len(s)<3:
        return "Rejected"
```

```
#first three characters are fixed. Therefore, checking them using index     if s[0]=='1':         if
s[1]=='0':             if s[2]=='1':
              # After index 2 only "1" can appear. Therefore break the process if any other
character is detected                 for i in range(3,len(s)):                     if s[i]!='1':
                      return "Rejected"
              return "Accepted" # if all 4 nested if true              return "Rejected" # else of 3rd
if return "Rejected" # else of 2nd if  return "Rejected" # else of 1st if
inputs=['1','10101','101','10111','01010','100','','10111101','1011111'] for i in inputs:
print(FA(i))
```

Output:

```
Rejected
Rejected
Accepted
Accepted
Rejected
Rejected
Rejected
Rejected
Accepted
```

c) Accept the input string with Regular expression of FA: (a+b)*bba.

Code:

```
def FA(s):     size=0
#scan complete string and make sure that it contains only 'a' & 'b'     for i in s:
       if i=='a' or i=='b':
          size+=1         else:
          return "Rejected"
#After checking that it contains only 'a' & 'b' #check it's length it should be 3 atleast     if
size>=3:
#check the last 3 elements         if s[size-3]=='b':
          if s[size-2]=='b':                    if s[size-1]=='a':                     return "Accepted" # if all
4 if true              return "Rejected" # else of 4th if             return "Rejected" # else of 3rd if
return "Rejected" # else of 2nd if
   return "Rejected" # else of 1st if

inputs=['bba', 'ababbba', 'abba','abb', 'baba','bbb',''] for i in inputs:     print(FA(i))
```

output:

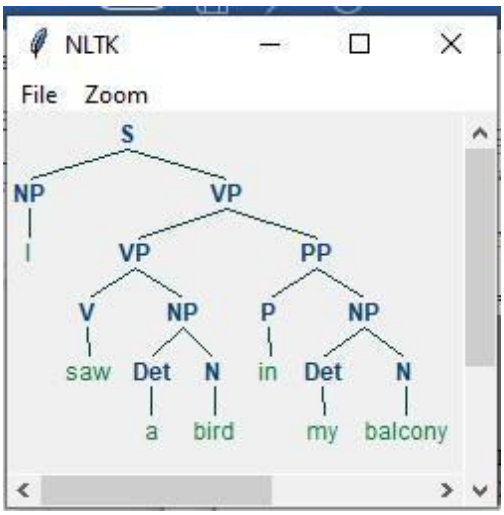
```
Rejected
Rejected
Accepted
Accepted
Rejected
Rejected
Rejected
Rejected
Accepted
```

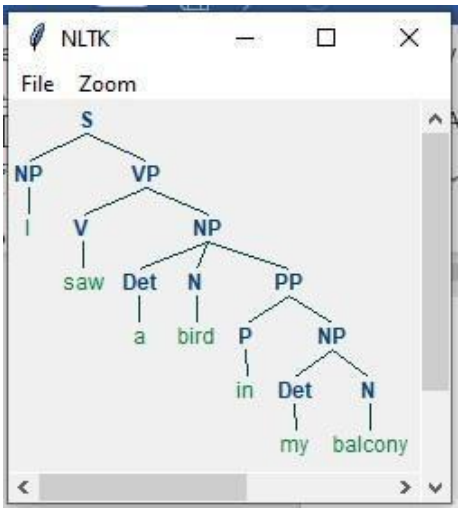
d) Implementation of Deductive Chart Parsing using context free grammar and a given sentence. Source code:

```
import nltk from nltk import tokenize grammar1 =
nltk.CFG.fromstring("""
              S -> NP VP
```

```
            PP -> P NP
            NP -> Det N | Det N PP | 'I'
            VP -> V NP | VP PP
            Det -> 'a' | 'my'
            N -> 'bird' | 'balcony'
            V -> 'saw'
        P -> 'in'           """)
sentence = "I saw a bird in my balcony"

for index in range(len(sentence)):
  all_tokens = tokenize.word_tokenize(sentence)
print(all_tokens)

# all_tokens = ['I', 'saw', 'a', 'bird', 'in', 'my', 'balcony']
parser = nltk.ChartParser(grammar1) for tree in parser.parse(all_tokens):
    print(tree)
    tree.draw()
```

output:





Practical 8

Study PorterStemmer, LancasterStemmer, RegexpStemmer, SnowballStemmer Study WordNetLemmatizer

Code:

# **PorterStemmer** import nltk
from nltk.stem import PorterStemmer word_stemmer = PorterStemmer()
print(word_stemmer.stem('writing')) **Output:**

```
============ RESTART: D:/2020/NLP/Practical/uni/p8aPorterStemmer.py ============
write
>>>
```

#LancasterStemmer

import nltk from nltk.stem import LancasterStemmer Lanc_stemmer = LancasterStemmer()
print(Lanc_stemmer.stem('writing')) **Output:**

```
========== RESTART: D:/2020/NLP/Practical/uni/p8bLancasterStemmer.py ==========
writ
>>>
```

#RegexpStemmer

```
import nltk
from nltk.stem import RegexpStemmer
Reg_stemmer = RegexpStemmer('ing$|s$|e$|able$', min=4)
print(Reg_stemmer.stem('writing'))
```

output

```
============ RESTART: D:/2020/NLP/Practical/uni/p8cRegexprStemmer.py ===========
writ
>>>
```

#SnowballStemmer import nltk
from nltk.stem import SnowballStemmer english_stemmer = SnowballStemmer('english')
print(english_stemmer.stem ('writing'))

output

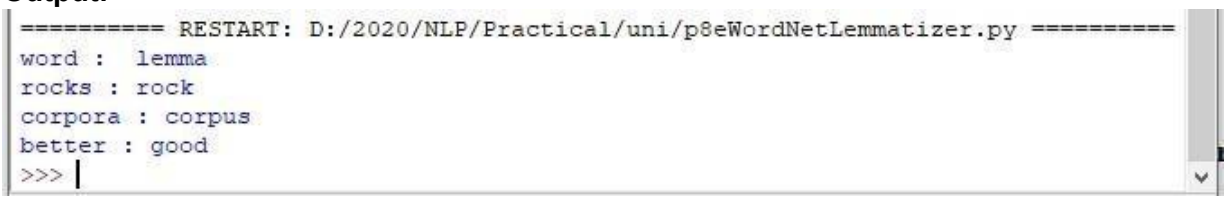
```
========== RESTART: D:/2020/NLP/Practical/uni/p8dSnowballStemmer.py ==========
write
>>>
```

#WordNetLemmatizer

```
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

print("word :\tlemma") print("rocks :", lemmatizer.lemmatize("rocks"))
print("corpora :", lemmatizer.lemmatize("corpora"))

# a denotes adjective in "pos"
```

```
print("better :", lemmatizer.lemmatize("better", pos ="a"))
```

Output:



Practical 9

Implement Naive Bayes classifier

Code:

```
        #pip install pandas
#pip install sklearn

import pandas as pd import numpy as np

sms_data = pd.read_csv("spam.csv", encoding='latin-1')

import re import nltk from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

stemming = PorterStemmer() corpus = [] for i in range
   (0,len(sms_data)): s1 = re.sub('[^a-zA-Z]',repl = ' ',string =
   sms_data['v2'][i])                                          s1.lower()
   s1 = s1.split()
   s1 = [stemming.stem(word) for word in s1 if word not in set(stopwords.words('english'))]
s1 = ' '.join(s1)
   corpus.append(s1)

from sklearn.feature_extraction.text import CountVectorizer countvectorizer
=CountVectorizer()

x  = countvectorizer.fit_transform(corpus).toarray() print(x)

y  = sms_data['v1'].values
print(y)

from sklearn.model_selection import train_test_split x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size = 0.3,
stratify=y,random_state=2)

#Multinomial Naïve Bayes.
from sklearn.naive_bayes import MultinomialNB multinomialnb = MultinomialNB()
multinomialnb.fit(x_train,y_train)

# Predicting on test data:
y_pred = multinomialnb.predict(x_test) print(y_pred)
#Results of our Models
from sklearn.metrics import classification_report, confusion_matrix from sklearn.metrics
import accuracy_score

print(classification_report(y_test,y_pred))
print("accuracy_score: ",accuracy_score(y_test,y_pred))
```

input: spam.csv file from github

output:

```
========= RESTART: D:\2020\NLP\Practical\uni\p9NaiveBayesClassifier.py =========
[[0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0
  1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 2 0 2 1 1 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0
  0 0 1 1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
  0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
  0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0
  0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1
  0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1]
 [1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 0 0 0 0 1 2 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0
  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 2 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0
  0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0]]
['ham' 'ham' 'spam' 'ham' 'ham' 'spam' 'ham' 'ham' 'spam']
['ham' 'ham' 'ham']
              precision    recall  f1-score   support

         ham       0.67      1.00      0.80         2
        spam       0.00      0.00      0.00         1

    accuracy                           0.67         3
   macro avg       0.33      0.50      0.40         3
weighted avg       0.44      0.67      0.53         3

accuracy_score:  0.6666666666666666
>>>
```

Practical 10

i. Speech tagging using spacy

code import spacy
sp = spacy.load('en_core_web_sm')
sen = sp(u"I like to play football. I hated it in my childhood though") print(sen.text)
print(sen[7].pos_) print(sen[7].tag_) print(spacy.explain(sen[7].tag_)) for word in sen:
print(f'{word.text:{12}} {word.pos_:{10}} {word.tag_:{8}}
{spacy.explain(word.tag_)}')

sen = sp(u'Can you go ogle it?') word = sen[2]
print(f'{word.text:{12}} {word.pos_:{10}}
{word.tag_:{8}}
{spacy.explain(word.tag_)}') sen = sp(u'Can you search it on google?') word = sen[5]

print(f'{word.text:{12}} {word.pos_:{10}} {word.tag_:{8}}
{spacy.explain(word.tag_)}')

#Finding the Number of POS Tags
sen = sp(u"I like to play football. I hated it in my childhood though")

num_pos = sen.count_by(spacy.attrs.POS)
num_pos

for k,v in sorted(num_pos.items()):
    print(f'{k}. {sen.vocab[k].text:{8}}: {v}')

#Visualizing Parts of Speech Tags
from spacy import displacy

sen = sp(u"I like to play football. I hated it in my childhood though") displacy.serve(sen,
style='dep', options={'distance': 120})

output:
To view the dependency tree, type the following address in your browser:
http://127.0.0.1:5000/. You will see the following dependency tree:

ii. Speech tagging
using nktl code:
import nltk
from nltk.corpus import state_union
from nltk.tokenize import PunktSentenceTokenizer

#create our training and testing data:

```
================== RESTART: D:\2020\NLP\Practical\uni\p10a1.py ================
I like to play football. I hated it in my childhood though
VERB
VBD
verb, past tense
I             PRON        PRP       pronoun, personal
like          VERB        VBP       verb, non-3rd person singular present
to            PART        TO        infinitival "to"
play          VERB        VB        verb, base form
football      NOUN        NN        noun, singular or mass
.             PUNCT       .         punctuation mark, sentence closer
I             PRON        PRP       pronoun, personal
hated         VERB        VBD       verb, past tense
it            PRON        PRP       pronoun, personal
in            ADP         IN        conjunction, subordinating or preposition
my            PRON        PRP$      pronoun, possessive
childhood     NOUN        NN        noun, singular or mass
though        ADV         RB        adverb
google        VERB        VB        verb, base form
google        PROPN       NNP       noun, proper singular
85. ADP      : 1
86. ADV      : 1
92. NOUN     : 2
94. PART     : 1
95. PRON     : 4
97. PUNCT    : 1
100. VERB     : 3

Using the 'dep' visualizer
Serving on http://0.0.0.0:5000 ...
```

```
train_text = state_union.raw("2005-GWBush.txt")
sample_text = state_union.raw("2006-GWBush.txt")

#train the Punkt tokenizer like:
custom_sent_tokenizer = PunktSentenceTokenizer(train_text)

# tokenize:
tokenized = custom_sent_tokenizer.tokenize(sample_text)

def process_content():     try:         for i in tokenized[:2]:             words = nltk.word_tokenize(i)
tagged = nltk.pos_tag(words)             print(tagged)

    except Exception as e:
        print(str(e))


process_content()
```

output:

b. Statistical parsing:
i. Usage of Give and Gave in the Penn Treebank sample Source code:

```
#probabilitistic parser
#Usage of Give and Gave in the Penn Treebank sample

import nltk import nltk.parse.viterbi
import nltk.parse.pchart
```


```
def give(t):
   return t.label() == 'VP' and len(t) > 2 and t[1].label() == 'NP'\        and (t[2].label() == 'PP-
DTV' or t[2].label() == 'NP')\        and ('give' in t[0].leaves() or 'gave' in t[0].leaves())


def sent(t):     return ' '.join(token for token in t.leaves() if token[0] not in '*-0') def
print_node(t, width):     output = "%s %s: %s / %s: %s" %\
       (sent(t[0]), t[1].label(), sent(t[1]), t[2].label(), sent(t[2]))     if len(output) > width:
output = output[:width] + "..."     print (output)

for tree in nltk.corpus.treebank.parsed_sents():     for t in tree.subtrees(give):
       print_node(t, 72)
```

Output:

ii. probabilistic parser Source code: import nltk from nltk import PCFG

```
grammar = PCFG.fromstring('''
NP -> NNS [0.5] | JJ NNS [0.3] | NP CC NP [0.2]
NNS -> "men" [0.1] | "women" [0.2] | "children" [0.3] | NNS CC NNS [0.4]
JJ -> "old" [0.4] | "young" [0.6]
CC -> "and" [0.9] | "or" [0.1]
''')

print(grammar)

viterbi_parser = nltk.ViterbiParser(grammar)
```

```
================== RESTART: D:/2020/NLP/Practical/uni/pl0bl.py ==================
gave NP: the chefs / NP: a standing ovation
give NP: advertisers / NP: discounts for maintaining or increasing ad sp...
give NP: it / PP-DTV: to the politicians
gave NP: them / NP: similar help
give NP: them / NP:
give NP: only French history questions / PP-DTV: to students in a Europe...
give NP: federal judges / NP: a raise
give NP: consumers / NP: the straight scoop on the U.S. waste crisis
gave NP: Mitsui / NP: access to a high-tech medical product
give NP: Mitsubishi / NP: a window on the U.S. glass industry
give NP: much thought / PP-DTV: to the rates she was receiving , nor to ...
give NP: your Foster Savings Institution / NP: the gift of hope and free...
give NP: market operators / NP: the authority to suspend trading in futu...
gave NP: quick approval / PP-DTV: to $ 3.18 billion in supplemental appr...
give NP: the Transportation Department / NP: up to 50 days to review any...
give NP: the president / NP: such power
give NP: me / NP: the heebie-jeebies
give NP: holders / NP: the right , but not the obligation , to buy a cal...
gave NP: Mr. Thomas / NP: only a `` qualified '' rating , rather than ``...
give NP: the president / NP: line-item veto power
>>>
```

```
token = "old men and women".split()

obj = viterbi_parser.parse(token)

print("Output: ") for x in obj:     print(x)
```

Output:

c. Malt parsing:

Parse a sentence and draw a tree using malt parsing.

Note: 1) Java should be installed.

2) maltparser-1.7.2 zip file should be copied in C:\Users\Beena Kapadia\AppData\Local\Programs\Python\Python39 folder and should be extracted in the same folder.

3) engmalt.linear-1.7.mco file should be copied to C:\Users\Beena Kapadia\AppData\Local\Programs\Python\Python39 folder **Source code:**

copy maltparser-1.7.2(unzipped version) and engmalt.linear-1.7.mco files to

```
================== RESTART: D:/2020/NLP/Practical/uni/pl0b2.py ==================
Grammar with 11 productions (start state = NP)
    NP -> NNS [0.5]
    NP -> JJ NNS [0.3]
    NP -> NP CC NP [0.2]
    NNS -> 'men' [0.1]
    NNS -> 'women' [0.2]
    NNS -> 'children' [0.3]
    NNS -> NNS CC NNS [0.4]
    JJ -> 'old' [0.4]
    JJ -> 'young' [0.6]
    CC -> 'and' [0.9]
    CC -> 'or' [0.1]
Output:
(NP (JJ old) (NNS (NNS men) (CC and) (NNS women))) (p=0.000864)
>>>
```

C:\Users\Beena Kapadia\AppData\Local\Programs\Python\Python39 folder

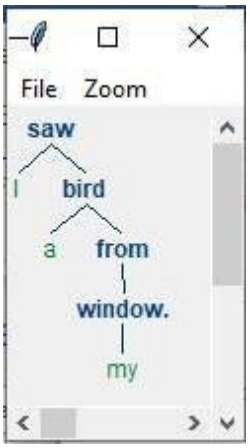
java should be installed

environment variables should be set - MALT_PARSER - C:\Users\Beena Kapadia\AppData\Local\Programs\Python\Python39\maltparser-1.7.2 and MALT_MODEL - C:\Users\Beena Kapadia\AppData\Local\Programs\Python\Python39\engmalt.linear-1.7.mco

```
from nltk.parse import malt
mp = malt.MaltParser('maltparser-1.7.2', 'engmalt.linear-1.7.mco')#file        t =
mp.parse_one('I saw a bird from my window.'.split()).tree() print(t)
t.draw()
```

Output:

(saw I (bird a (from (window. my))))


File Zoom
saw
I
bird
a
from
window.
my

Practical 11

a) Multiword Expressions in NLP Source code:

```
# Multiword Expressions in NLP

from nltk.tokenize import MWETokenizer from nltk import sent_tokenize, word_tokenize
s = "'Good cake cost Rs.1500\kg in Mumbai. Please buy me one of them.\n\nThanks."' mwe
= MWETokenizer([('New', 'York'), ('Hong', 'Kong')], separator='_') for sent in
sent_tokenize(s):     print(mwe.tokenize(word_tokenize(sent))) Output:
```

```
================== RESTART: D:/2020/NLP/Practical/uni/plla.py ==================
['Good', 'cake', 'cost', 'Rs.1500\\kg', 'in', 'Mumbai', '.']
['Please', 'buy', 'me', 'one', 'of', 'them', '.']
['Thanks', '.']
>>>
```

b) Normalized Web Distance and Word Similarity Source code:

```
# Normalized Web Distance and Word Similarity

#convert

#Reliance
supermarket
#Reliance
hypermarket
#Reliance
#Reliance
#Reliance downtown
#Relianc market
#Mumbai
#Mumbai Hyper
#Mumbai dxb
#mumbai airport
#k.m trading
#KM Trading
#KM trade
#K.M. Trading
#KM.Trading

#into

#Reliance
#Reliance
#Reliance
#Reliance
#Reliance
#Reliance
#Mumbai
#Mumbai
#Mumbai
#Mumbai
#KM Trading
#KM Trading
```

```
#KM Trading
#KM Trading
#KM Trading

import numpy as np
import re
import textdistance # pip install textdistance
# we will need scikit-learn>=0.21 import sklearn #pip install sklearn
from sklearn.cluster import AgglomerativeClustering

texts = [
  'Reliance supermarket', 'Reliance hypermarket', 'Reliance', 'Reliance', 'Reliance downtown',
'Relianc market',
   'Mumbai', 'Mumbai Hyper', 'Mumbai dxb', 'mumbai airport',
  'k.m trading', 'KM Trading', 'KM trade', 'K.M. Trading', 'KM.Trading'
]

def normalize(text):
  """ Keep only lower-cased text and numbers""" return re.sub('[^a-z0-9]+', ' ', text.lower())

def group_texts(texts, threshold=0.4):
  """ Replace each text with the representative of its cluster""" normalized_texts =
np.array([normalize(text) for text in texts]) distances = 1 - np.array([
      [textdistance.jaro_winkler(one, another) for one in normalized_texts]       for another in
normalized_texts
  ])
  clustering = AgglomerativeClustering(
    distance_threshold=threshold, # this parameter needs to be tuned carefully
affinity="precomputed", linkage="complete", n_clusters=None
  ).fit(distances) centers = dict() for cluster_id in set(clustering.labels_):        index =
clustering.labels_ == cluster_id     centrality = distances[:, index][index].sum(axis=1)
    centers[cluster_id] = normalized_texts[index][centrality.argmin()] return [centers[i] for i in
clustering.labels_]

print(group_texts(texts))
```

Output:

```
========== RESTART: D:/2020/NLP/Practical/uni/p11b.py ==========
['reliance', 'reliance', 'reliance', 'reliance', 'reliance', 'reliance', 'mumbai
', 'mumbai', 'mumbai', 'mumbai', 'km trading', 'km trading', 'km trading', 'km t
rading', 'km trading']
>>>
```

c) Word Sense Disambiguation Source code:

```
#Word Sense Disambiguation from nltk.corpus import wordnet as wn

def get_first_sense(word, pos=None):     if pos:         synsets = wn.synsets(word,pos)     else:
        synsets = wn.synsets(word)     return synsets[0]

best_synset = get_first_sense('bank')
print ('%s: %s' % (best_synset.name, best_synset.definition)) best_synset =
get_first_sense('set','n')
```

print ('%s: %s' % (best_synset.name, best_synset.definition)) best_synset = get_first_sense('set','v') print ('%s: %s' % (best_synset.name, best_synset.definition))

Output:

DEEP

LEARNING

INDEX

Sr. No	Practicals	Sign
1	Get the result of XOR gate as the output by using perceptron model	
2	Find the local minima by using gradient descent	
3	Do image classification by using Convolutional neural network	
4	Do Stock market prediction of Reliance company by using Recurrent neural network	
5	Do Stock market prediction of Reliance company by using Recurrent neural network	
6	Do feature extraction using Linear factor model	
7	Do auto encoding of the images	
8	Convert speech (audio format) into text format by neural network	
9	Predict the perfect emoji for the given text	
10	To remove the noise from the image using autoencoders	
11	To detect the object from image accurately.	
12	To classify the disease type by using CNN mode	

Practical 1

Aim : To get the result of XOR gate as the output by using perceptron model

Theory : First we build the function for classification of 0 and 1 that is. a unitStep function, and after that we've made a simple perceptron model which is doing dot product of the inputs, then we have added NOT,AND,OR,XOR logic functions and we have assigned weights and biases and passed the values to perceptron model for each function respectively finally we pass truth table values and we get correct result.

Input:

In [1]:
```
import numpy as np
```

In [2]:
```
#Defin unit step function
```

In [3]:
```
def unitStep(v):
    if v>=0:
        return 1
    else:
        return 0
```

In [4]:
```
#design perceptron model
```

In [5]:
```
def perceptronModel(x,w,b):
    v=np.dot(x,w)+b
    y=unitStep(v)
    return y
```

In [6]:
```
#Not logic
def NOT_logicFunction(x):
    wNOT = -1
    bNOT = 0.5
    return perceptronModel(x, wNOT, bNOT)
```

In [7]:
```
#And logic
def AND_logicFunction(x):
    w = np.array([1, 1])
    bAND = -1.5
    return perceptronModel(x, w, bAND)
```

In [8]:
```
#OR logic
def OR_logicFunction(x):
    w = np.array([1, 1])
    bOR = -0.5
    return perceptronModel(x, w, bOR)
```

In [9]:
```
#XOR Logic Function
def XOR_logicFunction(x):
    y1 = AND_logicFunction(x)
    y2 = OR_logicFunction(x)
    y3 = NOT_logicFunction(y1)
    final_x = np.array([y2, y3])
    finalOutput = AND_logicFunction(final_x)
    return finalOutput
```

In [10]:
```
test1 = np.array([0, 0])
test2 = np.array([0, 1])
test3 = np.array([1, 0])
test4 = np.array([1, 1])
```

Output:

In [11]:
```
print("XOR({}, {}) = {}".format(0, 1, XOR_logicFunction(test1)))
print("XOR({}, {}) = {}".format(0, 1, XOR_logicFunction(test2)))
print("XOR({}, {}) = {}".format(1, 0, XOR_logicFunction(test3)))
print("XOR({}, {}) = {}".format(1, 1, XOR_logicFunction(test4)))
```

```
XOR(0, 1) = 0
XOR(0, 1) = 1
XOR(1, 0) = 1
XOR(1, 1) = 0
```

Conclusion: We can easily build the XOR gate by help of simple basic perceptron model

Practical 2

Aim : To find the local minima by using gradient descent

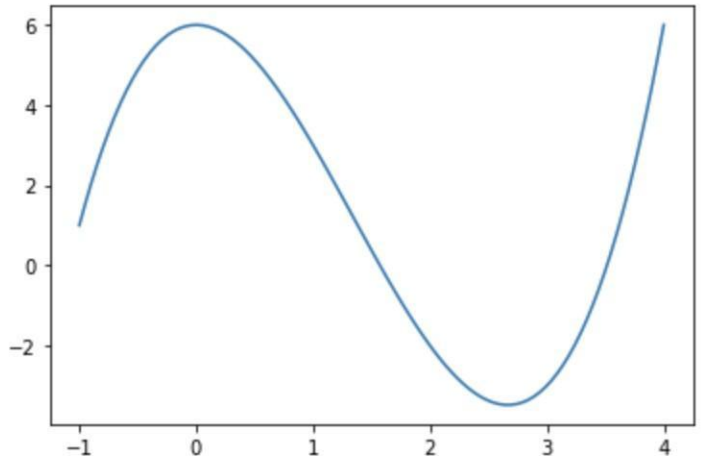
Theory : We pass the current value as 3, step size, maximum iterations and precision at the start and we loop through each iteration and we obtain the derivation function after looping on values then we are ready to plot our derivative function after finding derivative function plot we will plot our final graph and we zoom in to the graph to see more details finally we get the local minima as 2.66 after running through multiple epochs

Input:

In [1]:
```
#experiment 2 gradient descent (finding local minima)
cur_x = 3 # The algorithm starts at x=3
rate = 0.01 # Learning rate
precision = 0.000001 #This tells us when to stop the algorithm
previous_step_size = 1 #
max_iters = 10000 # maximum number of iterations
iters = 0 #iteration counter
df = lambda x: 2*(x+5) #Gradient of our function

while previous_step_size > precision and iters < max_iters:
    prev_x = cur_x #Store current x value in prev_x
    cur_x = cur_x - rate * df(prev_x) #Grad descent
    previous_step_size = abs(cur_x - prev_x) #Change in x
    iters = iters+1 #iteration count
    print("Iteration",iters,"\nX value is",cur_x) #Print iterations

print("The local minimum occurs at", cur_x)
```

```
Iteration 1
X value is 2.84
Iteration 2
X value is 2.6832
Iteration 3
X value is 2.529536
Iteration 4
X value is 2.37894528
Iteration 5
X value is 2.2313663744
Iteration 6
X value is 2.0867390469119997
Iteration 7
X value is 1.9450042659737599
Iteration 8
X value is 1.8061041806542846
Iteration 9
X value is 1.669982097041199
Iteration 10
```

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt

f_x = lambda x: (x**3)-4*(x**2)+6
x = np.linspace(-1,4,100)

#Plot the curve
plt.plot(x, f_x(x))
plt.show()
#plotting of first order derivative

f_x_derivative = lambda x: 3*(x**2)-8*x
```

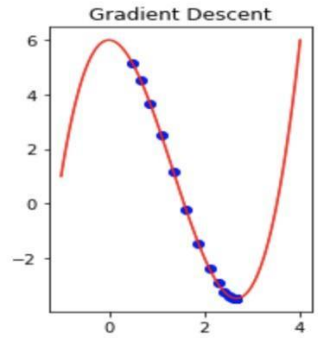


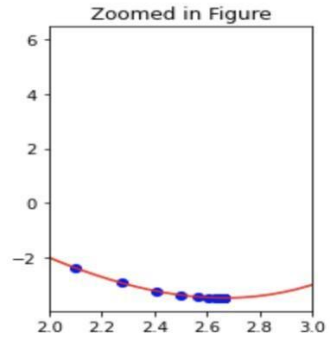
In [3]:
```
def plot_gradient(x, y, x_vis, y_vis):
    plt.subplot(1,2,2)
    plt.scatter(x_vis, y_vis, c = "b")
    plt.plot(x, f_x(x), c = "r")
    plt.title("Gradient Descent")
    plt.show()

    plt.subplot(1,2,1)
    plt.scatter(x_vis, y_vis, c = "b")
    plt.plot(x,f_x(x), c = "r")
    plt.xlim([2.0,3.0])
    plt.title("Zoomed in Figure")
    plt.show()
```

```
def gradient_iteration(x_start, iterations, learning_rate):


    x_grad = [x_start]
    y_grad = [f_x(x_start)]

    for i in range(iterations):


        x_start_derivative = - f_x_derivative(x_start)

        x_start += (learning_rate * x_start_derivative)

        x_grad.append(x_start)
        y_grad.append(f_x(x_start))

    print ("Local minimum occurs at: {:.2f}".format(x_start))
    print ("Number of steps: ",len(x_grad)-1)
    plot_gradient(x, f_x(x) ,x_grad, y_grad)

gradient_iteration(0.5, 1000, 0.05)
```

Output:

```
Local minimum occurs at: 2.67
Number of steps:  1000
```





Conclusion : We are able to create the gradient descent function perfectly and we are getting accurate local minima results

Practical 3

Aim : To do image classification by using Convolutional neural network

Theory : We are going to use inbuilt cifar dataset for image classification task in this experiment, we have provided classes and tags for each image after that we have trained our CNN model with training dataset and finally we are getting perfect output our model is doing fantastic job with 79% accuracy on training data.

Input:

In [1]:
```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
import numpy as np
from keras.datasets import cifar10
```

In [2]:
```
(X_train, y_train), (X_test,y_test) = datasets.cifar10.load_data()
X_train.shape
```

Out[2]:
```
(50000, 32, 32, 3)
```

In [3]:
```
X_test.shape
```

Out[3]:
```
(10000, 32, 32, 3)
```

In [4]:
```
y_train.shape
```

Out[4]:
```
(50000, 1)
```

In [5]:
```
y_train[:5]
```

Out[5]:
```
array([[6],
       [9],
       [9],
       [4],
       [1]], dtype=uint8)
```

In [6]:
```
y_train = y_train.reshape(-1,)
y_train[:5]
```

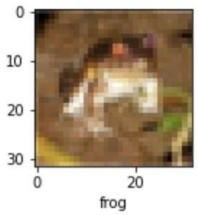
Out[6]:
```
array([6, 9, 9, 4, 1], dtype=uint8)
```

In [7]:
```
y_test = y_test.reshape(-1,)
```

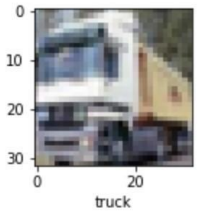
In [8]:
```
classes = ["airplane","automobile","bird","cat","deer","dog","frog","horse","ship","truck"]
```

In [9]:
```
def plot_sample(X, y, index):
    plt.figure(figsize = (15,2))
    plt.imshow(X[index])
    plt.xlabel(classes[y[index]])
```

In [10]:
```
plot_sample(X_train, y_train, 0)
```



In [11]:
```
plot_sample(X_train, y_train, 1)
```



In [12]:
```
X_train = X_train / 255.0
X_test = X_test / 255.0
```

In [13]:
```
cnn = models.Sequential([
    layers.Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(32, 32, 3)),
    layers.MaxPooling2D((2, 2)),

    layers.Conv2D(filters=64, kernel_size=(3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),

    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10, activation='softmax')
])
```

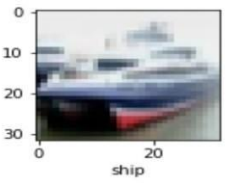
In [14]:
```
cnn.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

In [15]:
```
cnn.fit(X_train, y_train, epochs=10)
```

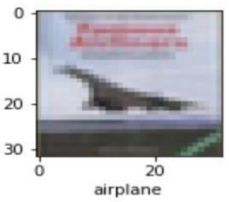
```
In [15]: cnn.fit(X_train, y_train, epochs=10)

Epoch 1/10
1563/1563 [==============================] - 21s 13ms/step - loss: 1.7069 - accuracy: 0.3754
Epoch 2/10
1563/1563 [==============================] - 21s 14ms/step - loss: 1.1595 - accuracy: 0.5890
Epoch 3/10
1563/1563 [==============================] - 20s 13ms/step - loss: 1.0082 - accuracy: 0.6475
Epoch 4/10
1563/1563 [==============================] - 19s 12ms/step - loss: 0.9102 - accuracy: 0.6857
Epoch 5/10
1563/1563 [==============================] - 18s 12ms/step - loss: 0.8445 - accuracy: 0.7043
Epoch 6/10
1563/1563 [==============================] - 18s 11ms/step - loss: 0.7804 - accuracy: 0.7293
Epoch 7/10
1563/1563 [==============================] - 18s 11ms/step - loss: 0.7259 - accuracy: 0.7471
Epoch 8/10
1563/1563 [==============================] - 204s 131ms/step - loss: 0.6719 - accuracy: 0.7662
Epoch 9/10
1563/1563 [==============================] - 20s 13ms/step - loss: 0.6319 - accuracy: 0.7798
Epoch 10/10
1563/1563 [==============================] - 20s 13ms/step - loss: 0.5854 - accuracy: 0.7989
Out[15]: <tensorflow.python.keras.callbacks.History at 0x7fda4084b400>

In [16]: cnn.evaluate(X_test,y_test)
313/313 [==============================] - 1s 2ms/step - loss: 0.9267 - accuracy: 0.6983
Out[16]: [0.9267432689666748, 0.6983000040054321]
```

Output:

```
In [17]: print(y_test[:4])
         plot_sample(X_test,y_test,1)
[3 8 8 0]
```




```
In [18]: y_pred = cnn.predict(X_test)
         y_pred[:4]

In [19]: y_classes = [np.argmax(element) for element in y_pred]
         y_classes[:4]
Out[19]: [3, 8, 1, 0]

In [20]: y_test[:4]
Out[20]: array([3, 8, 8, 0], dtype=uint8)

In [21]: plot_sample(X_test, y_test,3)
```




```
In [22]: classes[y_classes[3]]
Out[22]: 'airplane'
```

Conclusion : We got 79% accuracy and model is classifying airplane image perfectly in output.

Practical 4

Aim : To do Stock market prediction of Reliance company by using Recurrent neural network

Theory : First we have downloaded data of stock history of reliance from 1st jan 1996 to 17th april 2021 and we have tried to do prediction by using RNN model and we have used tensorflow library version 1.12, we have sorted the dataset by cleaning the data then we have prepared the model using tensorflow library which is giving 90% accuracy.

Input:

In [1]:
```
import numpy as np
import pandas as pd
import math
import sklearn
import sklearn.preprocessing
import datetime
import os
import matplotlib.pyplot as plt
import tensorflow as tf
```

In [2]:
```
dataset = pd.read_csv('RELIANCE.NS.csv',index_col = 0)
df_stock = dataset.copy()
df_stock = df_stock.dropna()
df_stock = df_stock[['Open','High','Low','Close']]
```

In [3]:
```
def normalize_data(df):
    min_max_scaler = sklearn.preprocessing.MinMaxScaler()
    df['Open'] = min_max_scaler.fit_transform(df.Open.values.reshape(-1,1))
    df['High'] = min_max_scaler.fit_transform(df.High.values.reshape(-1,1))
    df['Low'] = min_max_scaler.fit_transform(df.Low.values.reshape(-1,1))
    df['Close'] = min_max_scaler.fit_transform(df['Close'].values.reshape(-1,1))
    return df
df_stock_norm = df_stock.copy()
df_stock_norm = normalize_data(df_stock_norm)
```

In [4]:
```
valid_set_size_percentage = 10
test_set_size_percentage = 10
seq_len = 20

def load_data(stock,seq_len):
    data_raw = stock.to_numpy()
    data = []
    for index in range(len(data_raw) - seq_len):
        data.append(data_raw[index: index + seq_len])
    data = np.array(data);
    valid_set_size = int(np.round(valid_set_size_percentage/100*data.shape[0]));
    test_set_size = int(np.round(test_set_size_percentage/100*data.shape[0]));
    train_set_size = data.shape[0] - (valid_set_size + test_set_size);
    x_train = data[:train_set_size,:-1,:]
    y_train = data[:train_set_size,-1,:]
    x_valid = data[train_set_size:train_set_size+valid_set_size,:-1,:]
    y_valid = data[train_set_size:train_set_size+valid_set_size,-1,:]
    x_test = data[train_set_size+valid_set_size:,:-1,:]
    y_test = data[train_set_size+valid_set_size:,-1,:]
    return [x_train,y_train,x_valid,y_valid,x_test,y_test]

x_train,y_train,x_valid,y_valid,x_test,y_test = load_data(df_stock_norm,seq_len)
print('x_train.shape=',x_train.shape)
print('y_train.shape=',y_train.shape)
print('x_valid.shape=',x_valid.shape)
print('y_valid.shape=',y_valid.shape)
print('x_test.shape=',x_test.shape)
print('y_test.shape=',y_test.shape)
```

```
x_train.shape= (5070, 19, 4)
y_train.shape= (5070, 4)
x_valid.shape= (634, 19, 4)
y_valid.shape= (634, 4)
x_test.shape= (634, 19, 4)
y_test.shape= (634, 4)
```

In [5]:
```
from tensorflow.python.framework import ops

n_steps  = seq_len - 1
n_inputs = 4
n_neurons = 200
n_outputs = 4
n_layers = 2
learning_rate = 0.001
batch_size = 50
n_epochs = 100
train_set_size = x_train.shape[0]
test_set_size = x_test.shape[0]
ops.reset_default_graph()
X = tf.placeholder(tf.float32,[None, n_steps, n_inputs])
Y = tf.placeholder(tf.float32,[None, n_outputs])
```

In [6]:
```
index_in_epoch = 0;
perm_array = np.arange(x_train.shape[0])
np.random.shuffle(perm_array)

def get_next_batch(batch_size):
    global index_in_epoch, x_train, perm_array
    start = index_in_epoch
    index_in_epoch += batch_size
    if index_in_epoch > x_train.shape[0]:
        np.random.shuffle(perm_array)
        start = 0
        index_in_epoch = batch_size
    end = index_in_epoch
    return x_train[perm_array[start:end]], y_train[perm_array[start:end]]
```

In [7]:
```
layers = [tf.contrib.rnn.BasicRNNCell(num_units = n_neurons, activation = tf.nn.elu) for layer in range(n_layers)]
multi_layer_cell = tf.contrib.rnn.MultiRNNCell(layers)
rnn_outputs, states = tf.nn.dynamic_rnn(multi_layer_cell,X, dtype = tf.float32)
stacked_rnn_outputs = tf.reshape(rnn_outputs,[-1, n_neurons])
stacked_outputs = tf.layers.dense(stacked_rnn_outputs,n_outputs)
outputs = tf.reshape(stacked_outputs,[-1,n_steps,n_outputs])
outputs = outputs[:,n_steps-1,:]
```

In [9]:
```
loss = tf.reduce_mean(tf.square(outputs - Y))
```

In [10]:
```
optimizer = tf.train.AdamOptimizer(learning_rate=learning_rate)
training_op = optimizer.minimize(loss)
```

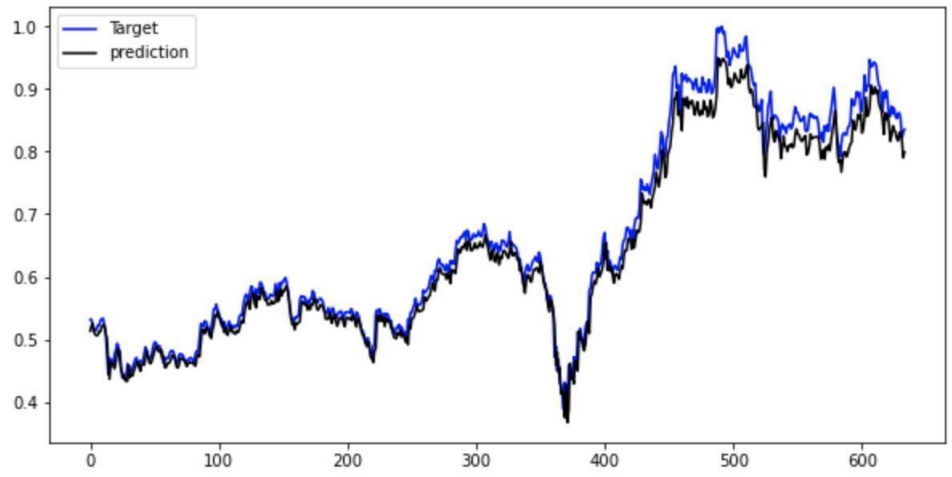
```
In [12]: with tf.Session() as sess:
             sess.run(tf.global_variables_initializer())
             for iteration in range(int(n_epochs*train_set_size/batch_size)):
                 x_batch, y_batch = get_next_batch(batch_size)
                 sess.run(training_op,feed_dict={X:x_batch,Y:y_batch})
                 if iteration % int(5*train_set_size/batch_size) == 0:
                     mse_train = loss.eval(feed_dict={X:x_train,Y:y_train})
                     mse_valid = loss.eval(feed_dict={X:x_valid,Y:y_valid})
                     print('%.2f epochs : MSE train/valid = %.6f/%.6f'%(iteration*batch_size/train_set_size,mse_train,mse_valid)
             y_test_pred = sess.run(outputs,feed_dict={X:x_test})
```

```
0.00 epochs : MSE train/valid = 0.024536/0.076188
5.00 epochs : MSE train/valid = 0.000024/0.000085
10.00 epochs : MSE train/valid = 0.000029/0.000254
15.00 epochs : MSE train/valid = 0.000013/0.000045
20.00 epochs : MSE train/valid = 0.000015/0.000034
25.00 epochs : MSE train/valid = 0.000009/0.000042
30.00 epochs : MSE train/valid = 0.000009/0.000044
35.00 epochs : MSE train/valid = 0.000009/0.000047
40.00 epochs : MSE train/valid = 0.000008/0.000029
45.00 epochs : MSE train/valid = 0.000013/0.000072
50.00 epochs : MSE train/valid = 0.000010/0.000045
55.00 epochs : MSE train/valid = 0.000009/0.000022
60.00 epochs : MSE train/valid = 0.000013/0.000076
65.00 epochs : MSE train/valid = 0.000010/0.000030
70.00 epochs : MSE train/valid = 0.000007/0.000032
75.00 epochs : MSE train/valid = 0.000008/0.000022
80.00 epochs : MSE train/valid = 0.000008/0.000021
85.00 epochs : MSE train/valid = 0.000009/0.000027
90.00 epochs : MSE train/valid = 0.000008/0.000041
95.00 epochs : MSE train/valid = 0.000007/0.000026
```

```
In [13]: y_test_pred.shape
Out[13]: (634, 4)
```

Output:

```
In [14]: comp = pd.DataFrame({'Column1':y_test[:,3],'Column2':y_test_pred[:,3]})
         plt.figure(figsize=(10,5))
         plt.plot(comp['Column1'],color='blue',label='Target')
         plt.plot(comp['Column2'],color='black',label='prediction')
         plt.legend()
         plt.show()
```



Conclusion : As we can see in the output the black line is predicted results all prediction was done by our model it is very close to the actual data we can conclude that RNN works very well for stock market prediction with very negligible loss rate.

Practical 5

Aim : To do Stock market prediction of TATA company by using Long short term memory

Theory : First we have downloaded data of stock history of TATA for training and test dataset and we have tried to do prediction by using LSTM model and we have used tensorflow keras library We trained our LSTM model by adding a sequential model and obtaining good results by help of LSTM .

Input:

In [10]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
```

In [11]:
```
dataset_train = pd.read_csv('NSE-TATAGLOBAL.csv')
training_set = dataset_train.iloc[:, 1:2].values
```

In [12]:
```
dataset_train.head()
```

Out[12]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509	7999.55

In [13]:
```
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
```

In [14]:
```
X_train = []
y_train = []
for i in range(60, 2035):
    X_train.append(training_set_scaled[i-60:i, 0])
    y_train.append(training_set_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)

X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
```

In [15]:
```
regressor = Sequential()

regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))

regressor.add(Dense(units = 1))

regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')

regressor.fit(X_train, y_train, epochs = 100, batch_size = 32)
```

```
Epoch 92/100
62/62 [==============================] - 3s 51ms/step - loss: 6.9065e-04
Epoch 93/100
62/62 [==============================] - 3s 51ms/step - loss: 6.7258e-04
Epoch 94/100
62/62 [==============================] - 3s 51ms/step - loss: 6.1093e-04
Epoch 95/100
62/62 [==============================] - 3s 51ms/step - loss: 6.4324e-04
Epoch 96/100
62/62 [==============================] - 3s 51ms/step - loss: 7.5046e-04
Epoch 97/100
62/62 [==============================] - 3s 51ms/step - loss: 5.5045e-04
Epoch 98/100
62/62 [==============================] - 3s 50ms/step - loss: 6.2113e-04
Epoch 99/100
62/62 [==============================] - 3s 50ms/step - loss: 8.3116e-04
Epoch 100/100
62/62 [==============================] - 3s 50ms/step - loss: 5.8380e-04
```

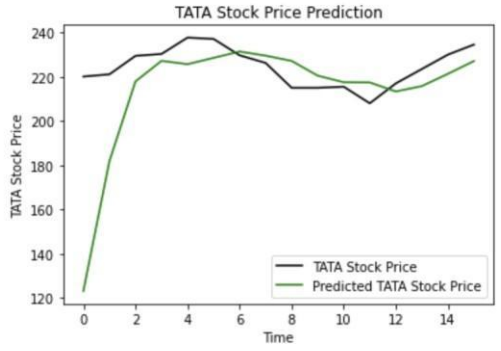
Out[15]:
```
<tensorflow.python.keras.callbacks.History at 0x7f951d3308b0>
```

In [16]:
```
dataset_test = pd.read_csv('tatatest.csv')
real_stock_price = dataset_test.iloc[:, 1:2].values
```

In [17]:
```
dataset_total = pd.concat((dataset_train['Open'], dataset_test['Open']), axis = 0)
inputs = dataset_total[len(dataset_total) - len(dataset_test) - 60:].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs)
X_test = []
for i in range(60, 76):
    X_test.append(inputs[i-60:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
predicted_stock_price = regressor.predict(X_test)
predicted_stock_price = sc.inverse_transform(predicted_stock_price)
```

Output:

```
In [18]: plt.plot(real_stock_price, color = 'black', label = 'TATA Stock Price')
         plt.plot(predicted_stock_price, color = 'green', label = 'Predicted TATA Stock Price')
         plt.title('TATA Stock Price Prediction')
         plt.xlabel('Time')
         plt.ylabel('TATA Stock Price')
         plt.legend()
         plt.show()
```



Conclusion: So as we can see the green line is predicted Stock price and black one is actual stock price, we can conclude that our model is perfectly predicting the market for TATA stocks.

Practical 6

Aim : To do feature extraction using Linear factor model

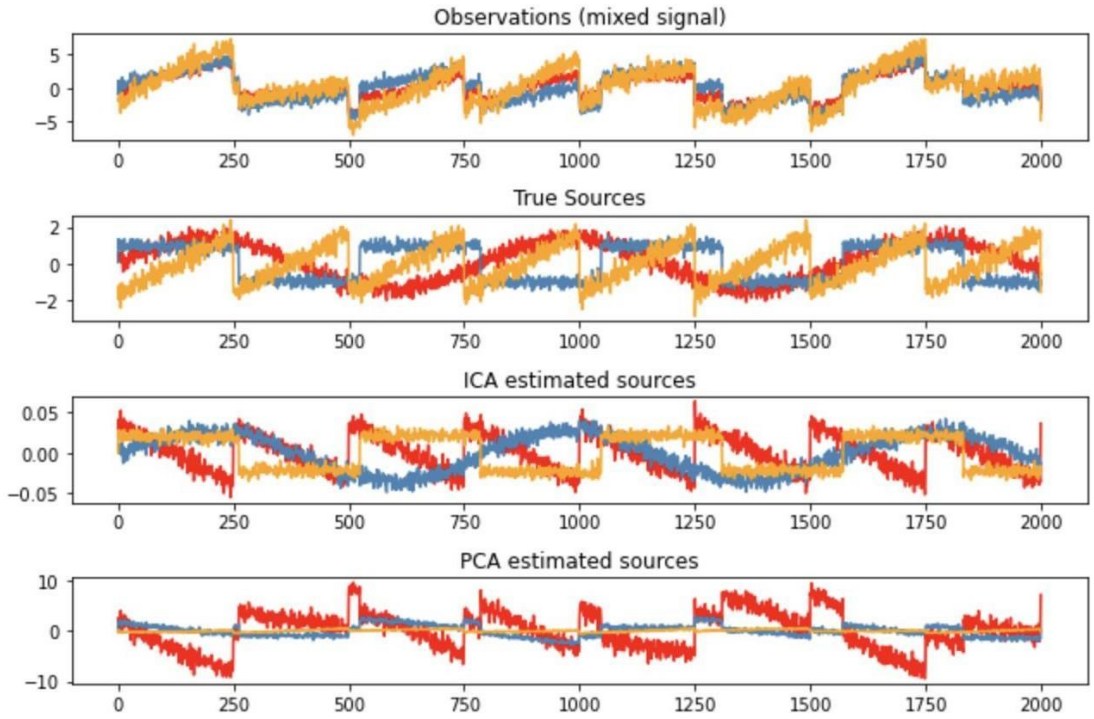
Theory : We are going to create sinusoidal signals and then by help of ica and pca we are going to extract features from the signal that we created.

Input:

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

from sklearn.decomposition import FastICA, PCA

np.random.seed(0)  # set seed for reproducible results
n_samples = 2000
time = np.linspace(0, 8, n_samples)

s1 = np.sin(2 * time)  # Signal 1 : sinusoidal signal
s2 = np.sign(np.sin(3 * time))  # Signal 2 : square signal
s3 = signal.sawtooth(2 * np.pi * time)  # Signal 3: sawtooth signal

S = np.c_[s1, s2, s3]
S += 0.2 * np.random.normal(size=S.shape)  # Add noise

S /= S.std(axis=0)  # Standardize data
# Mix data
A = np.array([[1, 1, 1], [0.5, 2, 1.0], [1.5, 1.0, 2.0]])  # Mixing matrix
X = np.dot(S, A.T)  # Generate observations
```

In [2]:
```
# compute ICA
ica = FastICA(n_components=3)
S_ = ica.fit_transform(X)  # Get the estimated sources
A_ = ica.mixing_  # Get estimated mixing matrix

# compute PCA
pca = PCA(n_components=3)
H = pca.fit_transform(X)  # estimate PCA sources

plt.figure(figsize=(9, 6))

models = [X, S, S_, H]
names = ['Observations (mixed signal)',
         'True Sources',
         'ICA estimated sources',
         'PCA estimated sources']
colors = ['red', 'steelblue', 'orange']

for ii, (model, name) in enumerate(zip(models, names), 1):
    plt.subplot(4, 1, ii)
    plt.title(name)
    for sig, color in zip(model.T, colors):
        plt.plot(sig, color=color)

plt.tight_layout()
```

Output:



Conclusion : As we can see in above output the first diagram is a mixed signal, in the second diagram we can see that the first signal is a mixture of three signals yellow, red and blue.
And by using ICA and PCA we can exactly extract the signal from mixed signal and can identify them clearly

Practical 7

Aim : To do auto encoding of the images

Theory : We are going to take famous mnist digit dataset to perform this task,

Input:

```
import numpy as np
from keras.layers import Input, Dense
from keras.models import Model
from keras.datasets import mnist
import matplotlib.pyplot as plt
```

```
# this is the size of our encoded representations
encoding_dim = 32 # 32 floats -> compression of factor 24.5, assuming the input is 784 floats

# this is our input placeholder
input_img = Input(shape=(784,))
# "encoded" is the encoded representation of the input
encoded = Dense(encoding_dim, activation='relu')(input_img)
# "decoded" is the lossy reconstruction of the input
decoded = Dense(784, activation='sigmoid')(encoded)
# this model maps an input to its reconstruction
autoencoder = Model(input_img, decoded)
# this model maps an input to its encoded representation
encoder = Model(input_img, encoded)
# create a placeholder for an encoded (32-dimensional) input
encoded_input = Input(shape=(encoding_dim,))
# retrieve the last layer of the autoencoder model
decoder_layer = autoencoder.layers[-1]
# create the decoder model
decoder = Model(encoded_input, decoder_layer(encoded_input))
# configure our model to use a per-pixel binary crossentropy loss, and the Adadelta optimizer:
autoencoder.compile(optimizer='adadelta', loss='binary_crossentropy')
```

```
[ ] (x_train, _), (x_test, _) = mnist.load_data()
    # normalize all values between 0 and 1 and we will flatten the 28x28 images into vectors of size 784.
    x_train = x_train.astype('float32') / 255.
    x_test = x_test.astype('float32') / 255.
    x_train = x_train.reshape((len(x_train), np.prod(x_train.shape[1:])))
    x_test = x_test.reshape((len(x_test), np.prod(x_test.shape[1:])))
    print (x_train.shape)
    print (x_test.shape)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step
(60000, 784)
(10000, 784)
```

```
autoencoder.fit(x_train, x_train,
epochs=50,
batch_size=256,
shuffle=True,
validation_data=(x_test, x_test))
# encode and decode some digits
# note that we take them from the *test* set
encoded_imgs = encoder.predict(x_test)
decoded_imgs = decoder.predict(encoded_imgs)
```

```
235/235 [==============================] - 3s 11ms/step - loss: 0.6885 - val_loss: 0.6882
Epoch 22/50
235/235 [==============================] - 2s 10ms/step - loss: 0.6882 - val_loss: 0.6880
Epoch 23/50
235/235 [==============================] - 2s 10ms/step - loss: 0.6879 - val_loss: 0.6877
Epoch 24/50
```

Output:

```
n = 20 # how many digits we will display
plt.figure(figsize=(20, 4))
for i in range(n):
# display original
  ax = plt.subplot(2, n, i + 1)
  plt.imshow(x_test[i].reshape(28, 28))
  plt.gray()
  ax.get_xaxis().set_visible(False)
  ax.get_yaxis().set_visible(False)



# display reconstruction
ax = plt.subplot(2, n, i + 1 + n)
plt.imshow(decoded_imgs[i].reshape(28, 28))
plt.gray()
ax.get_xaxis().set_visible(False)
ax.get_yaxis().set_visible(False)
plt.show()
```

Conclusion : As we can see the encoding of the image to the bottleneck and then decoding of the digits is done perfectly and both the autoencoders are working pretty well.

Practical 8

Aim : To convert speech (audio format) into text format by neural network

Theory : First we have done audio preprocessing on the input audio data we have done various operations on the audio data and then we passed the audio file to the CNN model, and our CNN model is taking care of all the attributes and learning the patterns from the audio perfectly

Input:

In [1]:
```
import os
import librosa   #for audio processing
import IPython.display as ipd
import matplotlib.pyplot as plt
import numpy as np
from scipy.io import wavfile #for audio processing
import warnings

train_audio_path = librosa.load('input.wav', sr = 16000)
samples, sample_rate = train_audio_path
```

In [2]:
```
ipd.Audio(samples, rate=sample_rate)
print(sample_rate)
```

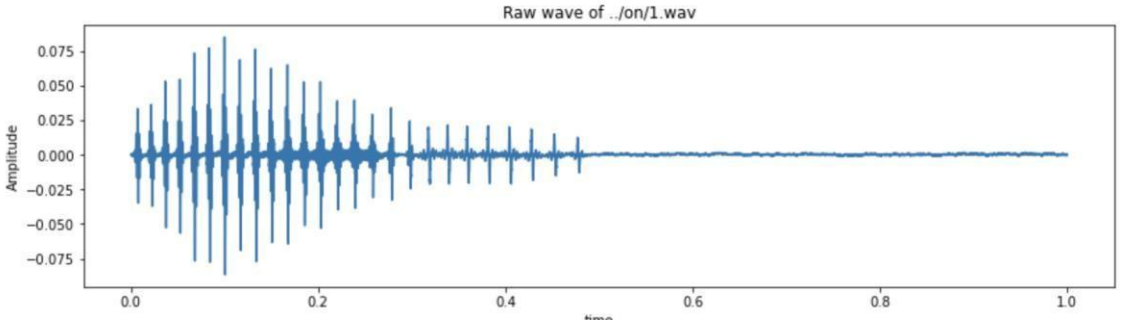
```
16000
```

In [3]:
```
os.listdir('input')
```

Out[3]:
```
['bed',
 'bird',
 'cat',
 'dog',
 'down',
 'eight',
 'five',
 'four',
 'go',
 'happy',
```

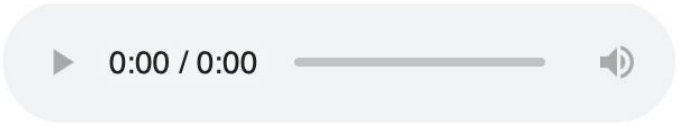
In [4]:
```
train_audio_path = 'input'
samples, sample_rate = librosa.load(train_audio_path  + '/on/1.wav', sr = 16000)
fig = plt.figure(figsize=(14, 8))
ax1 = fig.add_subplot(211)
ax1.set_title('Raw wave of ' + '../on/1.wav')
ax1.set_xlabel('time')
ax1.set_ylabel('Amplitude')
ax1.plot(np.linspace(0, sample_rate/len(samples), sample_rate), samples)
```

Out[4]: [<matplotlib.lines.Line2D at 0x2670a8a81c0>]



In [5]:
```
ipd.Audio(samples, rate=sample_rate)
```

Out[5]:



In [6]:
```
print(sample_rate)
```

```
16000
```

In [7]:
```
samples = librosa.resample(samples, sample_rate, 8000)
ipd.Audio(samples, rate=8000)
```

Out[7]:

In [8]:
```
labels=os.listdir(train_audio_path)
```

In [13]:
```
labels=["bed", "bird", "cat", "dog", "down", "eight", "five", "four",  "go","happy","house","left","marvel","nine","no"
```

In [15]:
```
train_audio_path = 'input'
all_wave = []
all_label = []
for label in labels:
    print(label)
    waves = [f for f in os.listdir(train_audio_path + '/'+ label) if f.end
    for wav in waves:
        samples, sample_rate = librosa.load(train_audio_path + '/' + label
        samples = librosa.resample(samples, sample_rate, 8000)
        if(len(samples)== 8000) :
            all_wave.append(samples)
            all_label.append(label)
```

```
bed
bird
cat
dog
down
eight
five
four
go
```

In [32]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y=le.fit_transform(all_label)
classes= list(le.classes_)
```

In [33]:
```
from keras.utils import np_utils
y=np_utils.to_categorical(y, num_classes=len(labels))
```

In [34]:
```
all_wave = np.array(all_wave).reshape(-1,8000,1)
```

In [35]:
```
from sklearn.model_selection import train_test_split
x_tr, x_val, y_tr, y_val = train_test_split(np.array(all_wave),np.array(y)
```

In [42]:
```
from keras.layers import Dense, Dropout, Flatten, Conv1D, Input, MaxPoolin
from keras.models import Model
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras import backend as K
K.clear_session()

inputs =  Input(shape=(8000,1))
```

```
#First Conv1D layer
conv = Conv1D(8,13, padding='valid', activation='relu', strides=1)(inputs)
conv = MaxPooling1D(3)(conv)
conv = Dropout(0.3)(conv)

#Second Conv1D layer
conv = Conv1D(16, 11, padding='valid', activation='relu', strides=1)(conv)
conv = MaxPooling1D(3)(conv)
conv = Dropout(0.3)(conv)

#Third Conv1D layer
conv = Conv1D(32, 9, padding='valid', activation='relu', strides=1)(conv)
conv = MaxPooling1D(3)(conv)
conv = Dropout(0.3)(conv)
```

```
#Fourth Conv1D layer
conv = Conv1D(64, 7, padding='valid', activation='relu', strides=1)(conv)
conv = MaxPooling1D(3)(conv)
conv = Dropout(0.3)(conv)

#Flatten layer
conv = Flatten()(conv)

#Dense Layer 1
conv = Dense(256, activation='relu')(conv)
conv = Dropout(0.3)(conv)
```

```
#Dense Layer 2
conv = Dense(128, activation='relu')(conv)
conv = Dropout(0.3)(conv)

outputs = Dense(len(labels), activation='softmax')(conv)

model = Model(inputs, outputs)
model.summary()
```

In [43]:
```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['ac
```

In [44]:
```
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=10,
mc = ModelCheckpoint('best_model.hdf5', monitor='val_acc', verbose=1, save_
```

In [45]:
```
history=model.fit(x_tr, y_tr ,epochs=30, callbacks=[es,mc], batch_size=32,
```

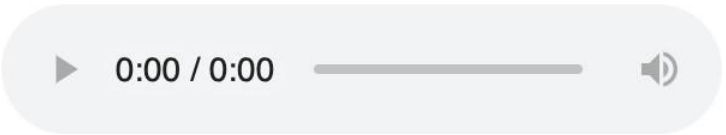
```
Epoch 1/30
1047/1047 [==============================] - 190s 179ms/step - loss: 2.92
18 - accuracy: 0.1606 - val_loss: 2.1934 - val_accuracy: 0.3624
WARNING:tensorflow:Can save best model only with val_acc available, skipp
ing.
Epoch 2/30
1047/1047 [==============================] - 187s 179ms/step - loss: 1.77
27 - accuracy: 0.4425 - val_loss: 1.1613 - val_accuracy: 0.6352
WARNING:tensorflow:Can save best model only with val_acc available, skipp
ing.
Epoch 3/30
1047/1047 [==============================] - 189s 180ms/step - loss: 1.25
61 - accuracy: 0.5911 - val_loss: 0.8985 - val_accuracy: 0.7091
WARNING:tensorflow:Can save best model only with val_acc available, skipp
ing.
Epoch 4/30
1047/1047 [==============================] - 190s 182ms/step - loss: 0.99
49 - accuracy: 0.6684 - val_loss: 0.6496 - val_accuracy: 0.7849
WARNING:tensorflow:Can save best model only with val_acc available, skipp
```

In [48]:
```
def predict(audio):
    prob=model.predict(audio.reshape(1,8000,1))
    index=np.argmax(prob[0])
    return classes[index]
```

In [49]:
```
import random
index=random.randint(0,len(x_val)-1)
samples=x_val[index].ravel()
print("Audio:",classes[np.argmax(y_val[index])])
ipd.Audio(samples, rate=8000)
```

```
Audio: six
```

Out[49]:



Output :

In [54]:
```
print("Text:",predict(samples))
```

```
Text: six
```

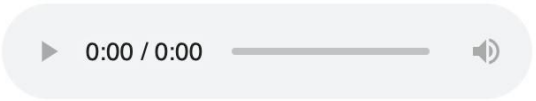
In [59]:
```
import sounddevice as sd
import soundfile as sf

samplerate = 16000
duration = 1 # seconds
filename = 'yes.wav'
print("start")
mydata = sd.rec(int(samplerate * duration), samplerate=samplerate,
    channels=1, blocking=True)
print("end")
sd.wait()
sf.write(filename, mydata, samplerate)
```

```
start
end
```

In [76]:
```
filepath='input'
```

In [75]:
```
samples, sample_rate = librosa.load(filepath +'/happy/2.wav', sr = 16000)
samples = librosa.resample(samples, sample_rate, 8000)
ipd.Audio(samples,rate=8000)
```

Out[75]:



In [77]:
```
predict(samples)
```

Out[77]:
```
'happy'
```

Conclusion : As we can see in above output we have passed the audio file in which a person is saying ‘happy’ word and our model is predicting the output as ‘happy’ accurately.

Practical 9

Aim : To predict the perfect emoji for the given text

Theory : We are first converting a training emoji dataset and testing emoji dataset to dataframe then we are passing all the training data to RNN model recurrent neural network will help to sustain the information and it performs very well in classification of emoji on emoji dataset.

Input:

```
import numpy as np
import pandas as pd
import emoji

from keras.models import Sequential
from keras.layers import Dense, Input, Dropout, SimpleRNN,LSTM, Activation
from keras.utils import np_utils

import matplotlib.pyplot as plt
```

```
# train_emoji data and test emoji data is Attached in the repo
train = pd.read_csv('\train_emoji.csv',header=None)
test = pd.read_csv('\test_emoji.csv',header=None)
```

```
# checking the data by printing first 5 entries
train.head()
```

	0	1	2	3
0	never talk to me again	3	NaN	NaN
1	I am proud of your achievements	2	NaN	NaN
2	It is the worst day in my life	3	NaN	NaN
3	Miss you so much	0	NaN	[0]
4	food is life	4	NaN	NaN

```
# checking the data by printing first 5 entries
test.head()
```

	0	1
0	I want to eat\t	4
1	he did not answer\t	3
2	he got a raise\t	2
3	she got me a present\t	0
4	ha ha ha it was so funny\t	2

```
# Creating the Dictionary of some emoji's with key a number and value is emoji
emoji_dict = { 0 : ":heart:", 1 : ":baseball:", 2:":smile:", 3 : ":disappointed:", 4 : ":fork_and_knife:"}
```

```
# printing the emoji icon by emojifying each emoji
for ix in emoji_dict.keys():
    print(ix,end=" ")
    print (emoji.emojize(emoji_dict[ix], use_aliases=True))
```

```
0 ❤
1 ⚾
2 😊
3 😞
4 🍴
```

```
# Creating the training and testing data

X_train = train[0]
Y_train = train[1]

X_test = test[0]
Y_test = test[1]

print(X_train.shape, Y_train.shape, X_test.shape, Y_test.shape)
print(("-------------------------"))
print(X_train[0],Y_train[0])
```

```
(132,) (132,) (56,) (56,)
-------------------------
never talk to me again 3
```

```
# Splitting the train data from sentences to list of words
for ix in range(X_train.shape[0]):
    X_train[ix] = X_train[ix].split()

# Splitting the test data from sentences to list of words
for ix in range(X_test.shape[0]):
    X_test[ix] = X_test[ix].split()

# Converting the labels into categorical Form
Y_train = np_utils.to_categorical(Y_train)
```

```
print(X_train[0],Y_train[0])
type(X_train)
```

```
['never', 'talk', 'to', 'me', 'again'] [0. 0. 0. 1. 0.]
pandas.core.series.Series
```

```
# To check what's the maximum length exist in the training data
np.unique(np.array([len(ix) for ix in X_train]) , return_counts=True)
```

```
(array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),
 array([ 4,  5, 26, 35, 20, 21, 11,  5,  1,  4], dtype=int64))
```

```
# Creating the Embedding dictionary with key = word and value = list of words
embeddings_index = {}
file = open('glove.6B.50d.txt', encoding="utf16")
file = open('glove.6B.50d.txt', errors="ignore")

try:
    for line in file:
        values = line.split()
#     print(values)
        word = values[0]
        coefs = np.asarray(values[1:], dtype='float32')
        embeddings_index[word] = coefs
    file.close()
except:
    file.__next__()
```

```
# Checking the length of each vector
embeddings_index["i"].shape
```

```
(50,)
```

```
# Using Scipy library to import the embedding matrix
from scipy import spatial

# Checking the cosine similarity of happy and sad
spatial.distance.cosine(embeddings_index["happy"], embeddings_index["sad"])
```

```
0.31093674898147583
```

```
# Checking the cosine similarity of india and delhi
spatial.distance.cosine(embeddings_index["india"], embeddings_index["delhi"])
```

```
0.18572336435317993
```

```
# Checking the cosine similarity of france and paris
spatial.distance.cosine(embeddings_index["france"], embeddings_index["paris"])
```

```
0.19746702909469604
```

```
# Filling the Embedding Matrix

embedding_matrix_train = np.zeros((X_train.shape[0], 10, 50))
embedding_matrix_test = np.zeros((X_test.shape[0], 10, 50))
try:
    for ix in range(X_train.shape[0]):
        for ij in range(len(X_train[ix])):
            embedding_matrix_train[ix][ij] = embeddings_index[X_train[ix][ij].lower()]

    for ix in range(X_test.shape[0]):
        for ij in range(len(X_test[ix])):
            embedding_matrix_test[ix][ij] = embeddings_index[X_test[ix][ij].lower()]
except:
         X_train.shape[0]
```

```
print(embedding_matrix_train.shape, embedding_matrix_test.shape)
```

```
(132, 10, 50) (56, 10, 50)
```

```
model = Sequential()
model.add(SimpleRNN(64, input_shape=(10,50), return_sequences=True))
model.add(Dropout(0.5))
model.add(SimpleRNN(64, return_sequences=False))
model.add(Dropout(0.5))
model.add(Dense(5))
model.add(Activation('softmax'))
```

```
# Setting Loss, Optimizer of the Model

model.compile(loss='categorical_crossentropy', optimizer='adam')
```

```
hist = model.fit(embedding_matrix_train,Y_train,
                 epochs = 50, batch_size=32,shuffle=True
                 )
```

```
5/5 [==============================] - 0s 8ms/step - loss: 1.4612
Epoch 22/50
5/5 [==============================] - 0s 9ms/step - loss: 1.5177
Epoch 23/50
5/5 [==============================] - 0s 7ms/step - loss: 1.4679
Epoch 24/50
5/5 [==============================] - 0s 8ms/step - loss: 1.5209
Epoch 25/50
5/5 [==============================] - 0s 8ms/step - loss: 1.4869
Epoch 26/50
5/5 [==============================] - 0s 7ms/step - loss: 1.4846
Epoch 27/50
5/5 [==============================] - 0s 7ms/step - loss: 1.4558
Epoch 28/50
5/5 [==============================] - 0s 8ms/step - loss: 1.5145
Epoch 29/50
5/5 [==============================] - 0s 8ms/step - loss: 1.4911
Epoch 30/50
5/5 [==============================] - 0s 7ms/step - loss: 1.4886
```

```
# printing the sentences with the predicted emoji and the labelled emoji

for ix in range(embedding_matrix_test.shape[0]):

    if pred[ix] != Y_test[ix]:
        print(ix)
        print(test[0][ix],end=" ")
        print(emoji.emojize(emoji_dict[pred[ix]], use_aliases=True),end=" ")
        print(emoji.emojize(emoji_dict[Y_test[ix]], use_aliases=True))
```

Output :

['I', 'want', 'to', 'eat'] 😄 🍴
1
['he', 'did', 'not', 'answer'] 😄 😞
3
['she', 'got', 'me', 'a', 'present'] 😄 ❤
5
['he', 'is', 'a', 'good', 'friend'] 😄 ❤
6
['I', 'am', 'upset'] 😄 ❤
7
['We', 'had', 'such', 'a', 'lovely', 'dinner', 'tonight'] 😄 ❤
8
['where', 'is', 'the', 'food'] 😄 🍴
10
['where', 'is', 'the', 'ball'] 😄 ⚾
11
['work', 'is', 'hard'] 😄 😞
12
['This', 'girl', 'is', 'messing', 'with', 'me'] 😄 😞
14
['Let', 'us', 'go', 'play', 'baseball'] 😄 ⚾
15
['This', 'stupid', 'grader', 'is', 'not', 'working'] 😄 😞
16
['work', 'is', 'horrible'] 😄 😞
18
['stop', 'messing', 'around'] 😄 😞
19
['any', 'suggestions', 'for', 'dinner'] 😄 🍴
20
['I', 'love', 'taking', 'breaks'] 😄 ❤
22
['I', 'boiled', 'rice'] 😄 🍴
23
['she', 'is', 'a', 'bully'] 😄 😞
24
['Why', 'are', 'you', 'feeling', 'bad'] 😄 😞
25
['I', 'am', 'upset'] 😄 😞
26
['I', 'worked', 'during', 'my', 'birthday'] 😄 😞
27
['My', 'grandmother', 'is', 'the', 'love', 'of', 'my', 'life'] 😄 ❤
29
['valentine', 'day', 'is', 'near'] 😄 ❤
30
['I', 'miss', 'you', 'so', 'much'] 😄 ❤

Conclusion : As we can see the model is perfectly classifying the perfect emoji for the respective text in the first example of ‘i want to eat’ the model is accurately showing the emoji of spoon and fork as it is classifying the eat word accurately.

Practical 10

Aim : To remove the noise from the image using autoencoders.

Theory : We are first taking the perfect image and adding the noise to that image using CV2 library and after that we have created autoencoders by using the CNN model and we will remove the noise from the data which is having noise and create a new fresh image which not contain noise.

Input :

```
[4]  import cv2
     from matplotlib import pyplot as plt
     from skimage.util import random_noise
     imig = cv2.imread("/content/drive/My Drive/DL/plant/images/Train_1439.jpg")
     img = cv2.resize(imig,(80, 80))
     img = img /255
     #blur = cv2.GaussianBlur(img,(5,5),4)
     blur = random_noise(img, mode='speckle', mean=0, var=0.05, clip=True)
```

```
import re
import os
import cv2
from matplotlib import pyplot as plt
from skimage.util import random_noise
from tqdm import tqdm
def get_images():

    images = []
    images_noise = []
    count = 0
    for root, dirnames, filenames in os.walk(r"/content/drive/My Drive/DL/plant/images/"):
        for filename in tqdm(filenames):
            if re.search("\.(jpg|jpeg|JPEG|png|bmp|tiff|jfif)$", filename):
                  filepath = os.path.join(root, filename)
                  image = cv2.imread(filepath)
                  image_resized = cv2.resize(image, (80,80))
                  img = image_resized.astype('float32') / 255
                  images.append(img)
                  img_noise =  random_noise(img, mode='s&p', salt_vs_pepper=0.5, clip=True)
                  #img_noise = random_noise(img, mode='gaussian', mean=0.2, var=0.15, clip=True)
                  images_noise.append(img_noise)
                  count = count +1
                  if count == 500:
                    break
    return images,images_noise
```

```
[28] img , img_noise = get_images()
```

```
  9%|▉         | 339/3642 [00:12<02:06, 26.02it/s]
  9%|▉         | 342/3642 [00:13<02:05, 26.34it/s]
  9%|▉         | 345/3642 [00:13<02:04, 26.49it/s]
 10%|▉         | 348/3642 [00:13<02:04, 26.40it/s]
 10%|▉         | 351/3642 [00:13<02:02, 26.82it/s]
 10%|▉         | 354/3642 [00:13<02:05, 26.26it/s]
 10%|▉         | 357/3642 [00:13<02:04, 26.38it/s]
 10%|▉         | 360/3642 [00:13<02:04, 26.34it/s]
 10%|▉         | 363/3642 [00:13<02:04, 26.33it/s]
 10%|▉         | 366/3642 [00:13<02:03, 26.50it/s]
 10%|▉         | 369/3642 [00:14<02:03, 26.40it/s]
 10%|▉         | 372/3642 [00:14<02:07, 25.71it/s]
 10%|▉         | 375/3642 [00:14<02:04, 26.16it/s]
```

```
import numpy as np
from sklearn.model_selection import train_test_split
#x_train,x_test = train_test_split( img, test_size=0.33, random_state=42)

x_train_noise,y_test_noise , x_train_img,y_test_img = train_test_split(img_noise, img, test_size=0.33, random_state=42)
```

```
import numpy as np
x_train_noise = np.array(x_train_noise)
x_train_img = np.array(x_train_img)

y_test_noise = np.array(y_test_noise)
y_test_img = np.array(y_test_img)


print(x_train_noise.shape)
print(y_test_img.shape)
```

```
(335, 80, 80, 3)
(165, 80, 80, 3)
```

```
    import keras
    from keras.layers import *
    from keras.layers import *
    from keras.models import *

    Input_img = Input(shape=(80, 80, 3))

    #encoding architecture
    x1 = Conv2D(64, (3, 3), activation='relu', padding='same',)(Input_img)
    x1 = MaxPool2D( (2, 2), padding='same')(x1)
    x2 = Conv2D(32, (3, 3), activation='relu', padding='same')(x1)
    x2 = MaxPool2D( (2, 2), padding='same')(x2)
    x3 = Conv2D(16, (3, 3), activation='relu', padding='same')(x2)
    x3=   MaxPool2D( (2, 2), padding='same')(x3)
    x4= Conv2D(16, (3, 3), activation='relu', padding='same')(x3)
    encoded    = MaxPool2D( (2, 2), padding='same')(x3)

#decoder
    x = Conv2DTranspose(16, (3, 3), activation='relu',strides=2,padding='same')(encoded)
    x = Conv2DTranspose(16, (3, 3), activation='relu',strides=2,padding='same')(x)
    x = Conv2DTranspose(32, (3, 3), activation='relu',strides=2,padding='same')(x)
    x = Conv2DTranspose(64, (3, 3), activation='relu',strides=2,padding='same')(x)
    decoded = Conv2DTranspose(3, (3, 3),padding='same', activation='sigmoid')(x)


    autoencoder = Model(Input_img, decoded)
    adam = keras.optimizers.Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, amsgrad=False)
    autoencoder.compile(optimizer=adam, loss='mse')
```

Loading...

```
autoencoder.summary()
```

```
Model: "model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 80, 80, 3)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 80, 80, 64)        1792
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 40, 40, 64)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 40, 40, 32)        18464
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 20, 20, 32)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 20, 20, 16)        4624
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 10, 10, 16)        0
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 5, 5, 16)          0
_________________________________________________________________
conv2d_transpose_6 (Conv2DTr (None, 10, 10, 16)        2320
_________________________________________________________________
conv2d_transpose_7 (Conv2DTr (None, 20, 20, 16)        2320
```

```
callback = keras.callbacks.callbacks.EarlyStopping(monitor='loss', min_delta=0, patience=4, verbose=0, mode='auto')
autoencoder.fit(x_train_noise, x_train_img,
                epochs=1000,
                batch_size=10,
                shuffle=True,callbacks=[ Loading... ,
                validation_data=(y_test_noise,y_test_img))
```

```
Epoch 21/1000
335/335 [==============================] - 0s 1ms/step - loss: 0.0175 - val_loss: 0.0182
Epoch 22/1000
335/335 [==============================] - 0s 1ms/step - loss: 0.0175 - val_loss: 0.0183
Epoch 23/1000
335/335 [==============================] - 0s 1ms/step - loss: 0.0175 - val_loss: 0.0183
Epoch 24/1000
335/335 [==============================] - 0s 1ms/step - loss: 0.0175 - val_loss: 0.0182
```

```
def compare_outputs(y_train_noise, decoded_imgs=None, n=10):
    plt.figure(figsize=(22, 5))
    for i in range(n):
        ax = plt.subplot(2, n, i+1)
        plt.imshow(y_train_noise[i].reshape(80,80,3))

        ax.get_xaxis().set_visible(False)
        ax.get_yaxis().set_visible(False)

        if decoded_imgs is not None:
            ax = plt.subplot(2, n, i+ 1 +n)
            plt.imshow(decoded_imgs[i].reshape(80,80,3))

            ax.get_xaxis().set_visible(False)
            ax.get_yaxis().set_visible(False)
    plt.show()

decoded_imgs = autoencoder.predict(y_train_noise)
print('Upper row: Input image provided \nBottom row: Decoded output generated')

compare_outputs(y_train_noise, decoded_imgs)
```

Output :

Conclusion : As we can see the model is accurately getting rid of all the noise giving exact output.

Practical 11

Aim : To detect the object from image accurately.

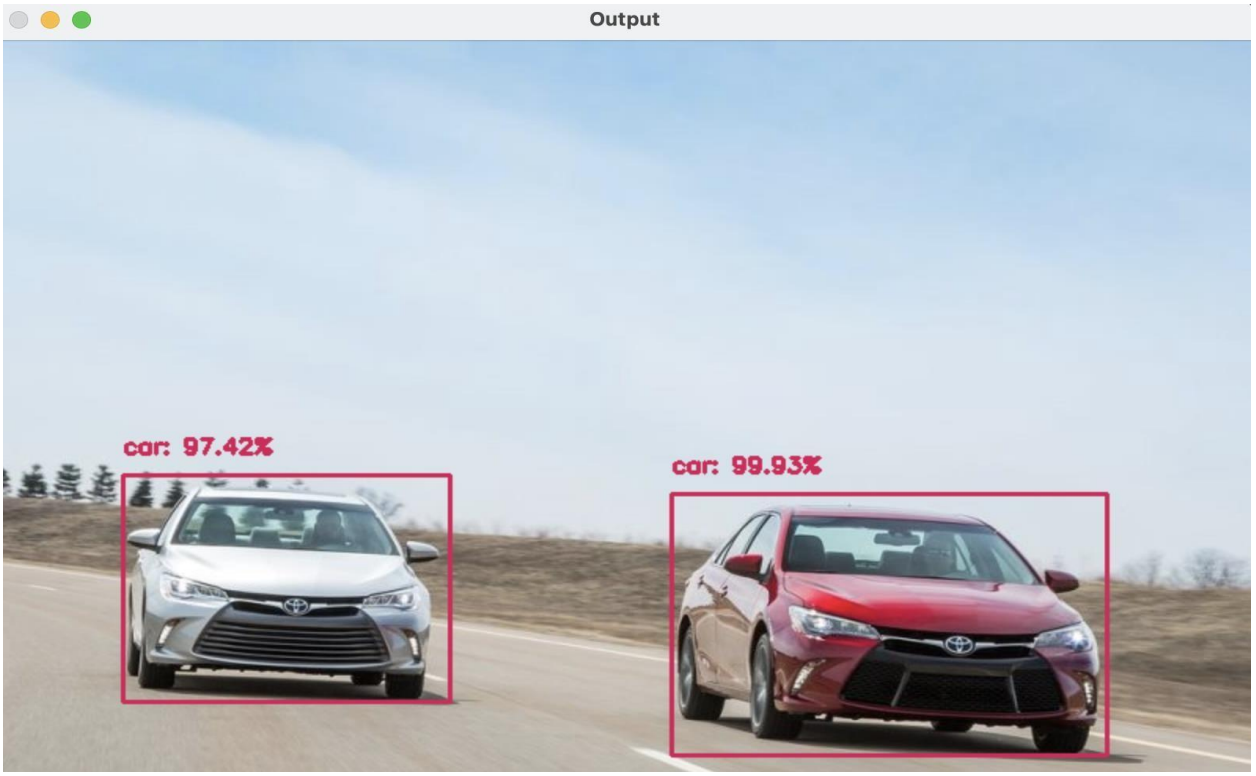
Theory : We are first going to provide some utility packages in our code and use them for the object detection task, then we are going to create some classes of the objection aeroplane, car, etc and we will pass it all to our model and see for the output accuracy.

Input :

```
# import the necessary packages
import numpy as np
import argparse
import cv2

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
	help="path to input image")
ap.add_argument("-p", "--prototxt", required=True,
	help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
	help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
	help="minimum probability to filter weak detections")
args = vars(ap.parse_args())

# initialize the list of class labels MobileNet SSD was trained to
# detect, then generate a set of bounding box colors for each class
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
	"bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
	"dog", "horse", "motorbike", "person", "pottedplant", "sheep",
	"sofa", "train", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))
```

```
# load our serialized model from disk
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])

# load the input image and construct an input blob for the image
# by resizing to a fixed 300x300 pixels and then normalizing it
# (note: normalization is done via the authors of the MobileNet SSD
# implementation)
image = cv2.imread(args["image"])
(h, w) = image.shape[:2]
blob = cv2.dnn.blobFromImage(cv2.resize(image, (300, 300)), 0.007843, (300, 300), 127.5)

# pass the blob through the network and obtain the detections and
# predictions
print("[INFO] computing object detections...")
net.setInput(blob)
detections = net.forward()

# loop over the detections
for i in np.arange(0, detections.shape[2]):
    # extract the confidence (i.e., probability) associated with the
    # prediction
    confidence = detections[0, 0, i, 2]

    # filter out weak detections by ensuring the `confidence` is
    # greater than the minimum confidence
    if confidence > args["confidence"]:
        # extract the index of the class label from the `detections`,
        # then compute the (x, y)-coordinates of the bounding box for
        # the object
        idx = int(detections[0, 0, i, 1])
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        # display the prediction
        label = "{}: {:.2f}%".format(CLASSES[idx], confidence * 100)
        print("[INFO] {}".format(label))
        cv2.rectangle(image, (startX, startY), (endX, endY),
            COLORS[idx], 2)
        y = startY - 15 if startY - 15 > 15 else startY + 15
        cv2.putText(image, label, (startX, y),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)

# show the output image
cv2.imshow("Output", image)
cv2.waitKey(0)
```

Output :



Conclusion : As we can see the accuracy of the model while detecting the object is very nice and it is working fantastic.

Practical 12

Aim : To classify the disease type by using CNN model

Theory : We first took the dataset of Brain tumor and we applied encoding on it and printed various parts of the brain on which brain tumor can happen. Then we are training our CNN model of sequential layers and will pass all the data in it, it seems that it will perform perfectly and give accurate results.

Input:

```
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D, Conv3D, BatchNormalization, Activation
from keras import backend as K
import os
from PIL import Image
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
import matplotlib.pyplot as plt
from matplotlib.pyplot import imshow
import pandas as pd
```

```
classes = os.listdir('\Brain-Tumor-Classification-Data\Training')
```

```
enc = OneHotEncoder()
enc.fit([[0], [1], [2], [3]])
def names(number):
    if(number == 0):
        return classes[0]
    elif(number == 1):
        return classes[1]
    elif(number == 2):
        return classes[2]
    elif(number == 3):
        return classes[3]
```

```
trainData = []
trainLabel = []
dim = (150, 150)
trainPath = "\Brain-Tumor-Classification-DataSet\Training"
index = 0
for dir in os.listdir(trainPath):
    filePaths = []
    subDir = os.path.join(trainPath, dir)
    for file in os.listdir(subDir):
        imgFullPath = os.path.join(subDir, file)
        filePaths.append(imgFullPath)
        img = Image.open(imgFullPath)
        x = img.resize(dim)
        x = np.array(x)
        trainData.append(np.array(x))
        trainLabel.append(enc.transform([[index]]).toarray())
    print(names(index))
    print(str(dir))
    index += 1

trainData = np.array(trainData)
trainLabel = np.array(trainLabel).reshape(2870, 4)
print(trainData.shape)
print(trainLabel.shape)
```

```
glioma_tumor
glioma_tumor
meningioma_tumor
meningioma_tumor
no_tumor
no_tumor
pituitary_tumor
pituitary_tumor
(2870, 150, 150, 3)
(2870, 4)
```

```
testData = []
testLabel = []
dim = (150, 150)
testPath = "\Brain-Tumor-Classification-Data\Testing"
index = 0
for dir in os.listdir(testPath):
    filePaths = []
    subDir = os.path.join(testPath, dir)
    for file in os.listdir(subDir):
        imgFullPath = os.path.join(subDir, file)
        filePaths.append(imgFullPath)
        img = Image.open(imgFullPath)
        x = img.resize(dim)
        x = np.array(x)
        testData.append(np.array(x))
        testLabel.append(enc.transform([[index]]).toarray())
    print(names(index))
    print(str(dir))
    index += 1
testData = np.array(testData)
testLabel = np.array(testLabel).reshape(394, 4)
print(testData.shape)
print(testLabel.shape)
```

```
glioma_tumor
glioma_tumor
meningioma_tumor
meningioma_tumor
no_tumor
no_tumor
pituitary_tumor
pituitary_tumor
(394, 150, 150, 3)
(394, 4)
```

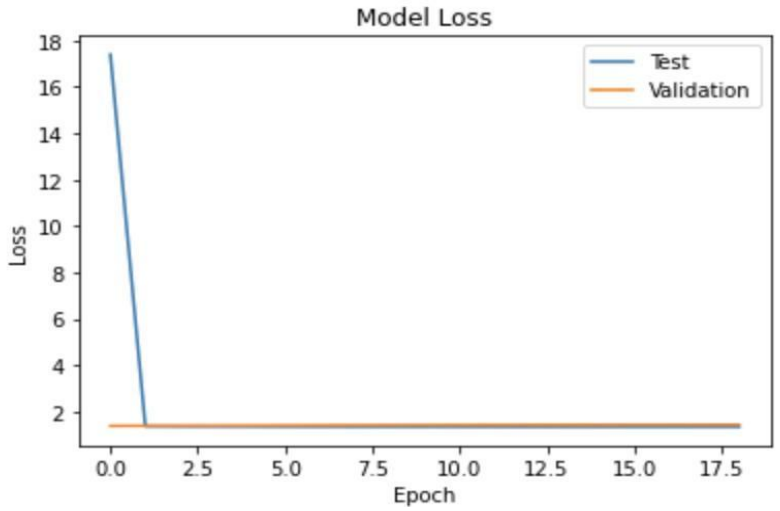
```
model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=(150,150,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(32))
model.add(Activation('relu'))
model.add(Dropout(0.25))
model.add(Dense(4))
model.add(Activation('softmax'))

model.compile(loss = "categorical_crossentropy", optimizer='adam')
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 148, 148, 32)      896
_________________________________________________________________
activation (Activation)      (None, 148, 148, 32)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 72, 72, 32)        9248
_________________________________________________________________
activation_1 (Activation)    (None, 72, 72, 32)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 36, 36, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 41472)             0
_________________________________________________________________
dense (Dense)                (None, 32)                1327136
_________________________________________________________________
activation_2 (Activation)    (None, 32)                0
_________________________________________________________________
dropout (Dropout)            (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 132
_________________________________________________________________
activation_3 (Activation)    (None, 4)                 0
=================================================================
Total params: 1,337,412
Trainable params: 1,337,412
Non-trainable params: 0
```

```
history = model.fit(trainData, trainLabel,batch_size = 32, epochs = 19, verbose=1,validation_data=(testData, testLabel))
```

```
Epoch 1/19
90/90 [==============================] - 39s 424ms/step - loss: 60.9387 - val_loss: 1.3870
Epoch 2/19
90/90 [==============================] - 38s 427ms/step - loss: 1.3772 - val_loss: 1.3899
Epoch 3/19
90/90 [==============================] - 37s 414ms/step - loss: 1.3677 - val_loss: 1.3940
```

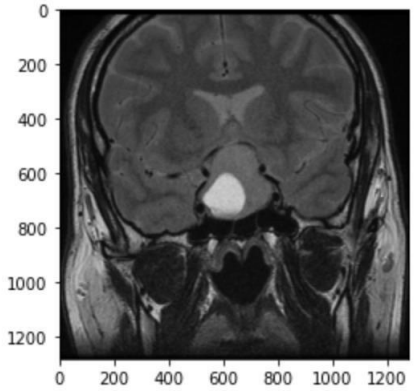
```
# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Test', 'Validation'], loc='upper right')
plt.show()
```



Output:

```
img = Image.open('pituitary_tumor\image.jpg')
x = np.array(img.resize(dim))
x = x.reshape(1,150,150,3)
answ = model.predict_on_batch(x)
classification = np.where(answ == np.amax(answ))[1][0]
imshow(img)
print(str(answ[0][classification]*100) + '% Confidence This Is ' + names(classification))
```

```
28.728872537612915% Confidence This Is pituitary_tumor
```



Conclusion : As we can see the CNN model is perfectly classifying the area perfectly in which tumor is detected in this case it is a pituitary type of tumor.