

University of Mumbai

**Practical Journal of
Blockchain, Natural Language
Processing & Deep Learning**

M.Sc.(Information Technology) Part-II

Submitted by

Shaikh Obed S.A

Seat No: 1172743



**DEPARTMENT OF INFORMATION TECHNOLOGY
PILLAI HOC COLLEGE OF ARTS, SCIENCE & COMMERCE,
RASAYANI
(Affiliated to Mumbai University)
RASAYANI, 410207 MAHARASHTRA
2023-2024**

Mahatma Education Society's
Pillai Hoc College of Arts, Science & Commerce, Rasayani
(Affiliated to Mumbai University)
RASAYANI-MAHARASHTRA-410207

**DEPARTMENT OF
INFORMATION TECHNOLOGY**



CERTIFICATE

This is to certify that the experiment work entered in this journal is as per the syllabus in **M.Sc. (Information Technology) Part-II, Semester-IV**; class prescribed by University of Mumbai for the subject **Blockchain** was done in computer lab of Mahatma Education Society's Pillai HOC College of Arts, Science & Commerce, Rasayani by **SHAIKH OBED S.A** during Academic year 2023-2024.

Exam Seat No: 1172743

Subject In-Charge

Coordinator

External Examiner

Principal

Date:

College Seal

BLOCKCHAIN

INDEX

Sr. No.	PracticalName	Page No.
1	<p>Write the following programs for Blockchainin Python:</p> <ul style="list-style-type: none"> A. Asimpleclientclassthatgeneratestheprivateandpublickeysby using the built-in Python RS Aalgorithm 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. 	1
2	Install and configure Go Ethereumand the Mistbrowser. Develop and test a sample application. (Meta Mask & Remix)	8
3	<p>Implement and demonstrate the use of the following in Solidity:</p> <ul style="list-style-type: none"> A. Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs, Mappings, Conversions, Ether Units, Special Variables. B. Functions, Function Modifiers, View functions, Pure Functions, Fall back Function, Function Overloading, Mathematical functions, Cryptographic functions. 	20
4	<p>Implement and demonstrate the use of the following in Solidity:</p> <ul style="list-style-type: none"> A. Withdrawal Pattern, Restricted Access. B. Contracts, Inheritance, Constructors, Abstract Contracts, Interfaces. C. Libraries, Assembly, Events, Error handling. 	45
5	Write a program to demonstrate mining of Ether.	64
6	Demonstrate the running of the blockchain node.	66
7	Create your own blockchain and demonstrate its use.	69

PRACTICAL:1

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

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

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

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

Dinesh = Client()
print("\nPublic Key:", Dinesh.identity)
```

Output:

```
C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\prac1>C:/Users/Achsah/AppData/Local/Programs/Python/Python39/python.exe c:/Users/Achsah/Documents/MScIT/sem4/blockchain_practical/prac1/prac1a.py

Public Key: 30819f300d06092a864886f70d010101050003818d0030818902818100adcc265
040fdf19988db8eabc5e73fb2d4527f95af6f3b9305377b0182d61fc44441af11dc1c8537c06d
452718289d83e92245c1af7373bf3d45e95c78383d0a82edb026f63d4fa805366017b991bc9ac8
6391f59935bf6559f8a23d89aa915a9e2f4c3e0113f9d9b9b5e071e2c4f780fff35fb0c9506c7c
b596a0128fe5f230203010001
```

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

```

import binascii
import collections
import datetime
from client import Client
from Crypto.Hash import SHA
from Crypto.Signature import PKCS1_v1_5

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

    def to_dict(self):
        identity = "Genesis" if self.sender == "Genesis" else 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)
print("\nTransaction Recipient:\n", t.recipient)
# print("\nTransaction Sender:\n", t.sender)
```

```

print("\nTransactionValue:\n",t.value)

signature=t.sign_transaction() print("\nSignature:\n",signature)

```

Output:

```

C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\prac1>C:/Users/Achsah/AppData/Local/Programs/Python/Python39/python.exe c:/Users/Achsah/Documents/MScIT/sem4/blockchain_practical/prac1/prac1b.py

Transaction Recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100c308b9261d2397e09dffcf67981240735cb2e3e0f4f510d29e21a70335503f142005e5f09e9db9091b263e73b6a32cd909fdc77a616bd4a5e09d044bf63c7906a98b791021ee41dbfb83d5022fb2423185262689e31287543b0863385d7325e30bcf8bc722907bfa0b4a39495f6a2ac2d6bf5e50e77d2b52d6efcaf3a062a9f0203010001

Transaction Value: 5.0

Signature: b3a8342acd21883671ff67dde74172f31f094935a2775765ec6e20f5ba910627eb9450b14d721933ea2ecc46d7a14e38d8b1e3e2382b9132c09ea94077b31c4f4a7cdf33b0f3ec4e0378fb6f53e8ba450b79572737b440f8584bc79c3fe3360ac75d23655d81e2c8f1dbe1435a2735100a3738d05522aeaadeee7f5bba6ffff2

```

C) Create multiple transactions and display them.

```
from client import Client
```

```
from transaction_class import Transaction
```

```
Dinesh = Client()
```

```
Ramesh=Client()
```

```
t = Transaction(Dinesh, Ramesh.identity, 5.0) print("\nTransaction Recipient:\n",t.recipient)#print("\nTransaction Sender:\n",t.sender) print("\nTransaction Value:\n", t.value)
```

```
signature=t.sign_transaction() print("\nSignature:\n",signature)
```

```
Dinesh = Client()
```

```
Ramesh = Client()
```

```
Seema=Client()Vijay
```

```
=Client()
```

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

```
transactions = [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:
    Transaction.display_transaction(transaction)
print("-----")
```

Output:

```
-----  

sender: 30819f300d06092a864886f70d010101050003818d0030818902818100c123f94a104b17803a5fb728b6  

a4e3abb26f2554e5652b5b5e5df08cf3f56efef5a36196fe4eebb8fe7f299d1fbe153031bce451e3c45ef2680237  

5c49f3474b9d23312534badccf3a8ecf4c238dc593a8a488eeaf155b347fda86b5548de80a96b3e1543eb20d4867  

03574d6c28a67cc04797c247e457fc233a6074f5e1c0cb0203010001  

-----  

recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100cc47acc592a9c8ec78b211e  

bda5ef91f40518e9c23338e0c99824892012b533656c8872d512994269e79d58a54e9fd8548141f204b26a3d89e6  

36468c81171b2147a2ca0c5745d66822b19d826f235afa2cab4a9f4b1623895019db6fdbcd752ff6a3dbc709d76c  

dd64df5e12ae674a5c896c09b632ab0b6b19c731c4d9004b30203010001  

-----  

value: 6.0  

-----  

time: 2023-04-22 22:13:48.783100  

-----  

sender: 30819f300d06092a864886f70d010101050003818d0030818902818100c551eccbd6e7624223f4a51741  

4b122ae738153aa00dd11951cf58e7f3cd436e639cc89fd84d34a93892450966378401bab918f186401a514162e  

de7fcab891df9023dc6604d1bfea1df2e83e9a3a985cdfcb00a9e2e55ba4364b48a1200c5ed6d163e4e7e8e39d3d  

e67272f63b04e559872fec9719fc7870b308581761fec10203010001  

-----  

recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100ae7406d1f27b484dc241f33  

a48b66df19d6e5f3b732fefda2622ee726bb49dcfea390f1f5a11c651f7a96fd888f9e901630645da2bfe9d8987  

69a859481a10eff8f977a40e59701f43e278992741af99b77aed08bb6fa5297ed2116441300469e73ec347e0bb8  

e790c960948b7872e6a60060581caf4b78d1624b0a45848610203010001  

-----  

value: 2.0  

-----  

time: 2023-04-22 22:13:48.784604  

-----  

sender: 30819f300d06092a864886f70d010101050003818d0030818902818100cc47acc592a9c8ec78b211ebda  

5ef91f40518e9c23338e0c99824892012b533656c8872d512994269e79d58a54e9fd8548141f204b26a3d89e6364  

68c81171b2147a2ca0c5745d66822b19d826f235afa2cab4a9f4b1623895019db6fdbcd752ff6a3dbc709d76cdd6  

4df5e12ae674a5c896c09b632ab0b6b19c731c4d9004b30203010001  

-----  

recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100c551eccbd6e7624223f4a51  

7414b122ae738153aa00dd11951cf58e7f3cd436e639cc89fd84d34a93892450966378401bab918f186401a5141  

62ede7fcab891df9023dc6604d1bfea1df2e83e9a3a985cdfcb00a9e2e55ba4364b48a1200c5ed6d163e4e7e8e39  

d3de67272f63b04e559872fec9719fc7870b308581761fec10203010001  

-----  

value: 4.0  

-----  

time: 2023-04-22 22:13:48.787805
```

D) Create a blockchain, a genesis block and execute it.

```

from client import Client
from transaction_class import Transaction

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

def dump_blockchain(blocks):
    print(f"\nNumber of blocks in the chain: {len(blocks)}")

    for i, block in enumerate(blocks):
        print(f"block# {i}")
        for transaction in block.verified_transactions:
            Transaction.display_transaction(transaction)
            print("-----")
        print("=====")

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

block0 = Block(Dinesh)
block0.previous_block_hash = "" NONCE = None

block0.verified_transactions.append(t0)
digest = hash(block0)
last_block_hash = digest

TPCoins = [block0]
dump_blockchain(TPCoins)

```

Output

```

Number of blocks in the chain: 1
block # 0
sender: Genesis
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100b6dbe8af2c6f079fc7bdf8a
5f00cf97738460294c2cb1d968cd6e59961afb3a39c96e132ada370ac2802aa8a58bf2d6ef13d39c95f744b31af0
0467c883980d7e825fc83fcf6a4d925be93c50d3cd1691d58495bd07aded1ef8c05d9b5606dcef55dd85721d4804
3bd1b733f2eb7027ffff0920abac3204b093247fcee235a5a90203010001
-----
value: 500.0
-----
time: 2023-04-22 22:40:58.531260
-----
=====
=====
```

E) Create a mining function and test it.

import hashlib

```
def sha256(message):
```

```
    return hashlib.sha256(message.encode("ascii")).hexdigest()
```

```
def mine(message, difficulty=1): assert
```

```
    difficulty // 1 prefix = "1" * difficulty
    for i in range(1000):
        digest = sha256(str(hash(message)) + str(i))
        if digest.startswith(prefix):
```

```
            print(f"after {str(i)} iterations found nonce: {digest}")
            #return print(digest)
            mine("test message", 2)
```

Output:

```
C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\prac1>C:/Users/Achsah/AppData/Local/Programs/Python/Python39/python.exe c:/Users/Achsah/Documents/MScIT/sem4/blockchain_practical/prac1/prac1e.py

After 119 iterations found nonce: 11a90de765a93c9fd75b5da05644bf4ef06059ac26b95d283270b3527
4c50050

After 146 iterations found nonce: 11e7b37a2c393112e7190f748400462e8fd3eec0afbbbc16c28e92faa
19b19bf

After 350 iterations found nonce: 11eeaf6cacc8cc0fb4cc8f0a32a5ad6702e74702e8c745e996945b6c4
9b4dae8

After 464 iterations found nonce: 11c5bf9e6a861f4e9ac8bd60af865e19f2d7460cf46a0a79bae84ab85
e47b911
```

F] Add blocks to the miner and dump the blockchain.

```

import datetime
import hashlib

#Create a class with two functions

class Block:
    def __init__(self,data,previous_hash):
        self.timestamp=datetime.datetime.now(datetime.timezone.utc)
        self.data=data
        self.previous_hash=previous_hash
        self.hash=
        self.calc_hash()

    def calc_hash(self):
        sha=hashlib.sha256()
        hash_str=self.data.encode("utf-8")
        sha.update(hash_str)
        return sha.hexdigest()

#Instantiate the class

blockchain=[Block("First block","0")]

blockchain.append(Block("Second block", blockchain[0].hash))
blockchain.append(Block("Third block", blockchain[1].hash))

# Dumping the blockchain

for block in blockchain:
    print(
        f"Timestamp:{block.timestamp}\nData:{block.data}\nPreviousHash:
{block.previous_hash}\nHash:{block.hash}\n")

```

Output:

```

Timestamp: 2023-04-22 17:41:07.240201+00:00
Data: First block
Previous Hash: 0
Hash: 876fb923a443ba6afe5fb32dd79961e85be2b582cf74c233842b630ae16fe4d9

Timestamp: 2023-04-22 17:41:07.240201+00:00
Data: Second block
Previous Hash: 876fb923a443ba6afe5fb32dd79961e85be2b582cf74c233842b630ae16fe4d9
Hash: 8e2fb9e02898feb024dff05ee0b27fd5ea0a448e252d975e6ec5f7b0a252a6cd

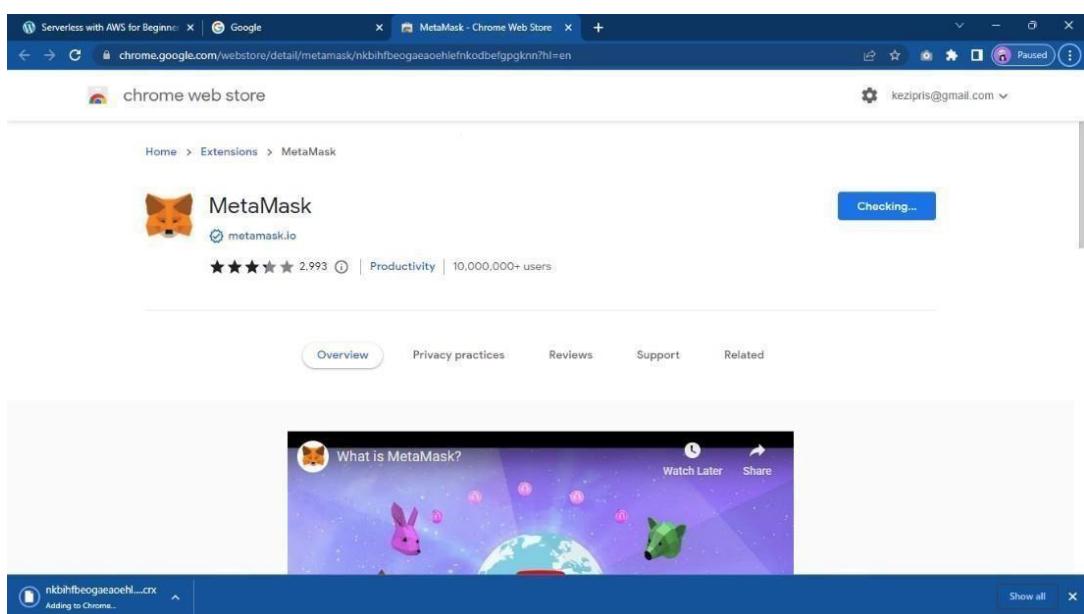
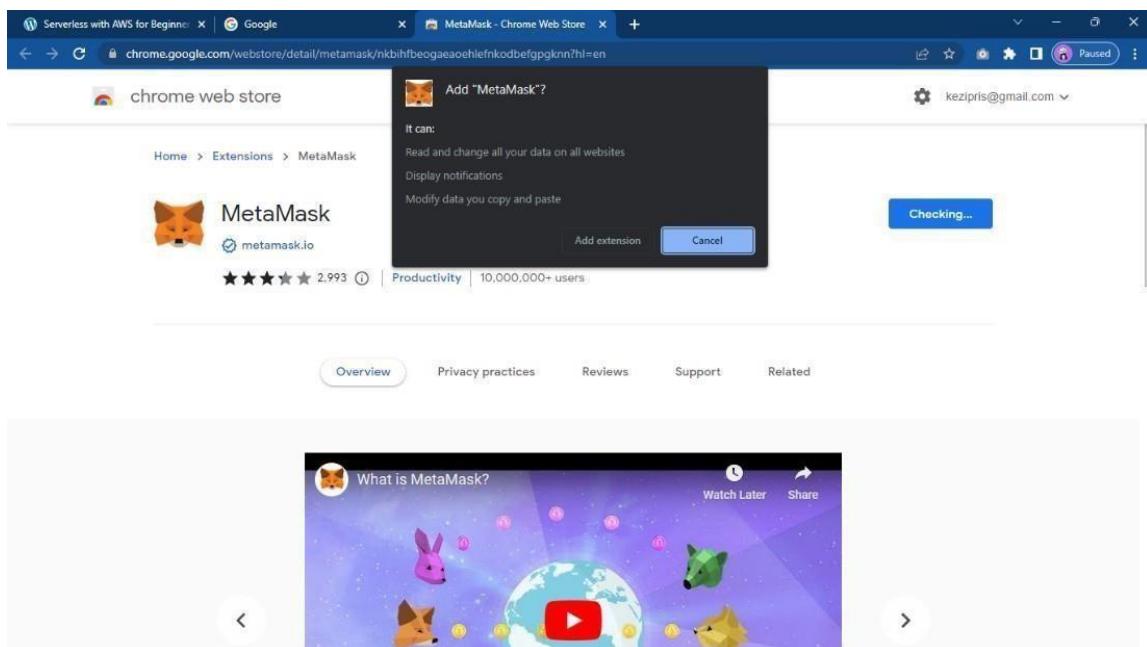
Timestamp: 2023-04-22 17:41:07.240201+00:00
Data: Third block
Previous Hash: 8e2fb9e02898feb024dff05ee0b27fd5ea0a448e252d975e6ec5f7b0a252a6cd
Hash: 06e369fbfbe5362a8115a5c6f3e2d3ec7292cc4272052dcc3280898e3206208d

```

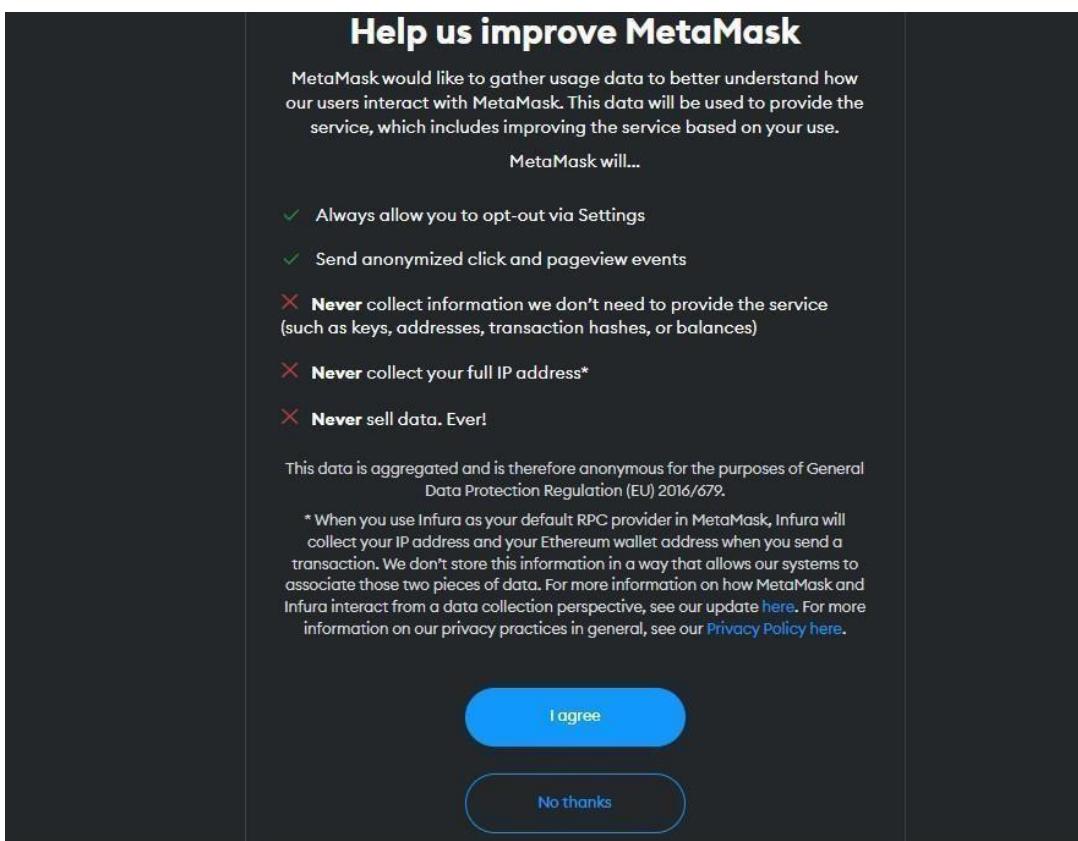
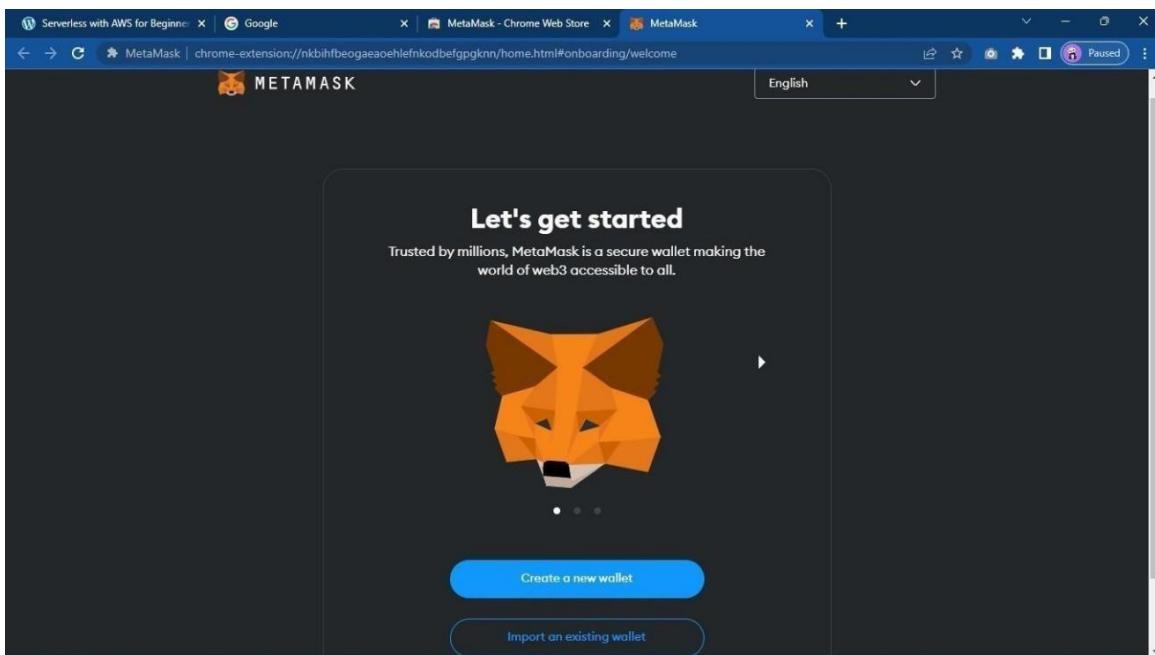
PRACTICAL-2

Aim:Install and configure go Ethereum and themist browser.develop and test a sample application (MetaMask& remix)

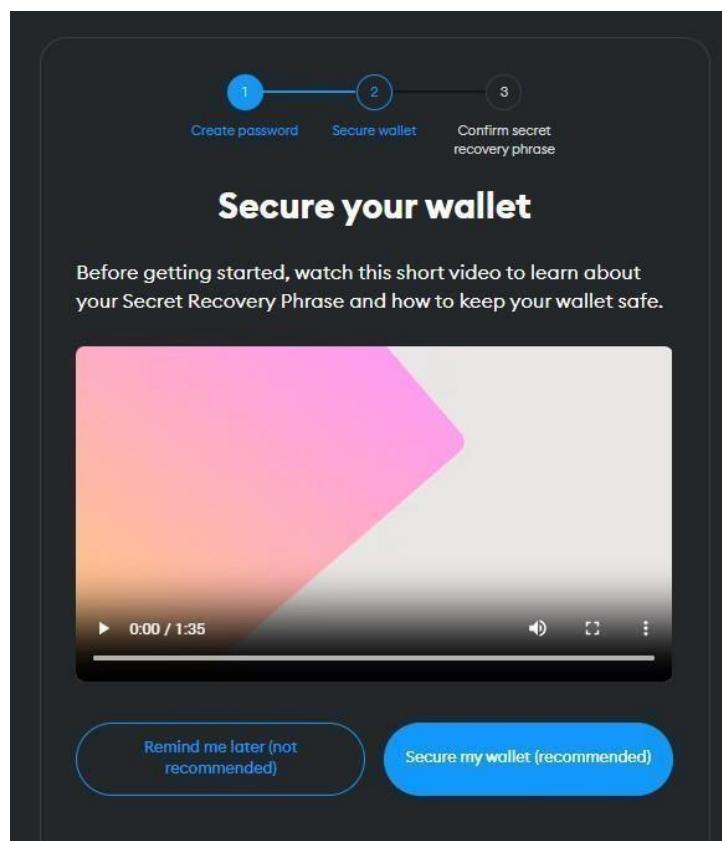
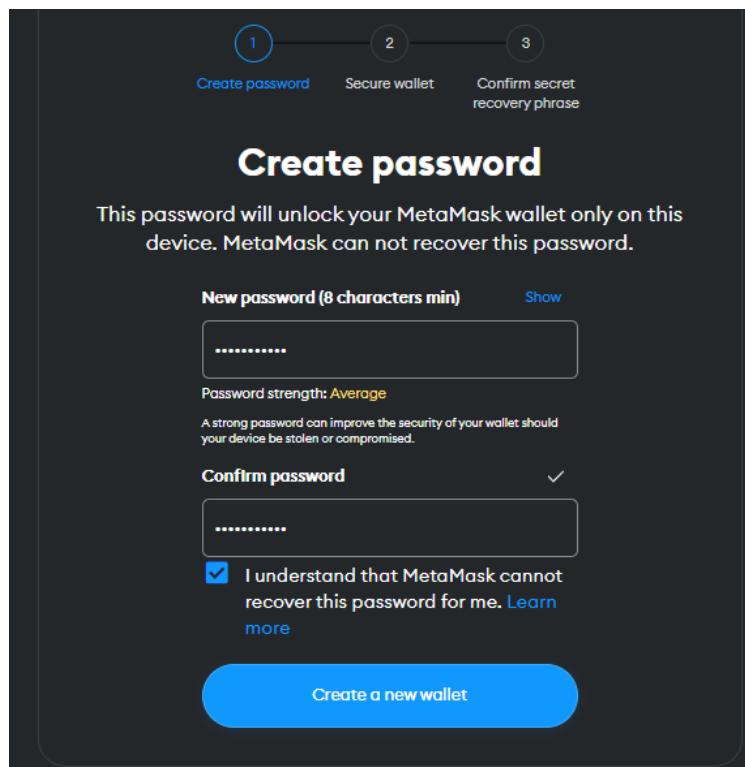
Step1-> InstallMetaMaskextensionforchromefromChromeWebStore



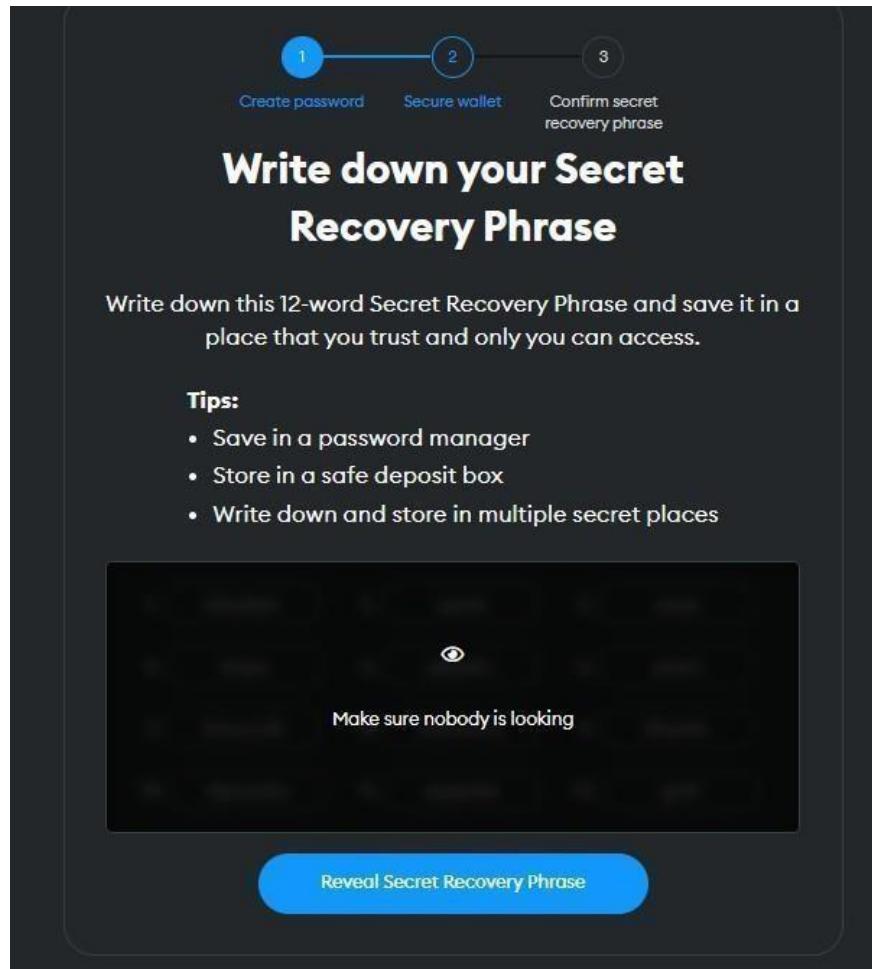
Step 2-> Click on MetamaskExtension in Extensions. Below page will open in anewtab.Clickon Create a New Wallet. Click on I agree.



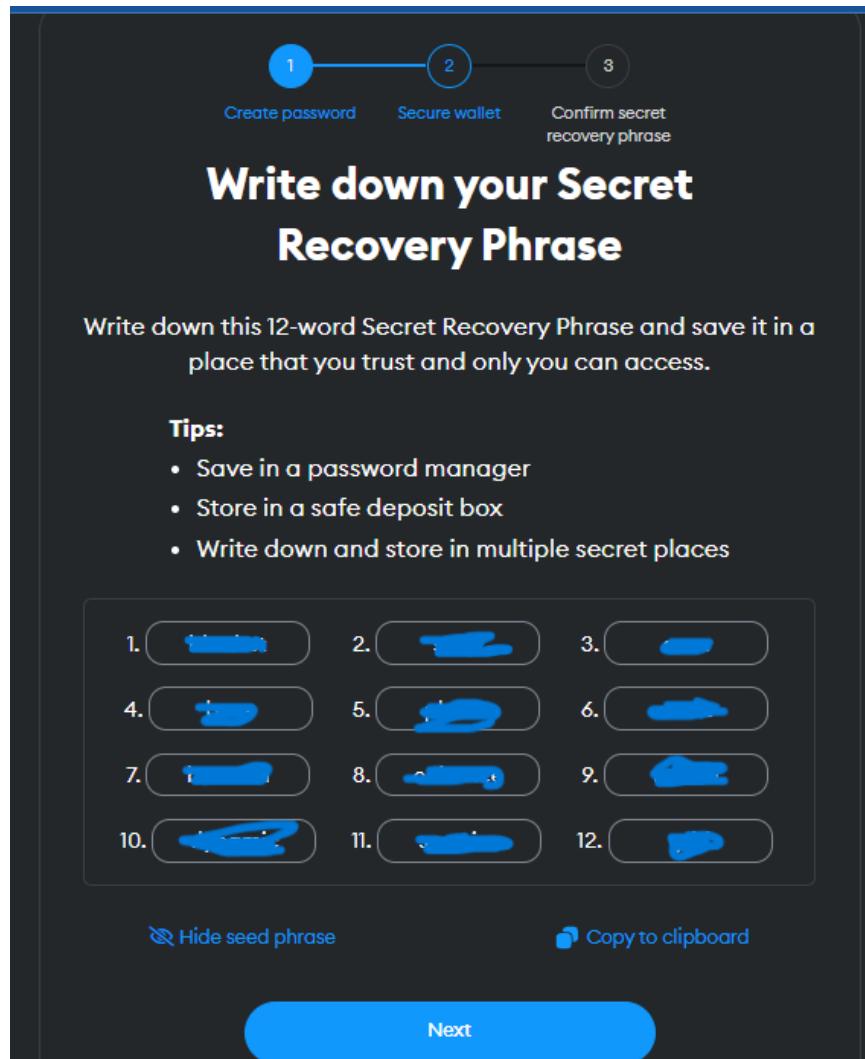
Step 3->Create a password. This password can be used only on the device it was created on. Create a Strong password and click on Create a new Wallet button



Step4-> Click on Secure my wallet button, following window will appear



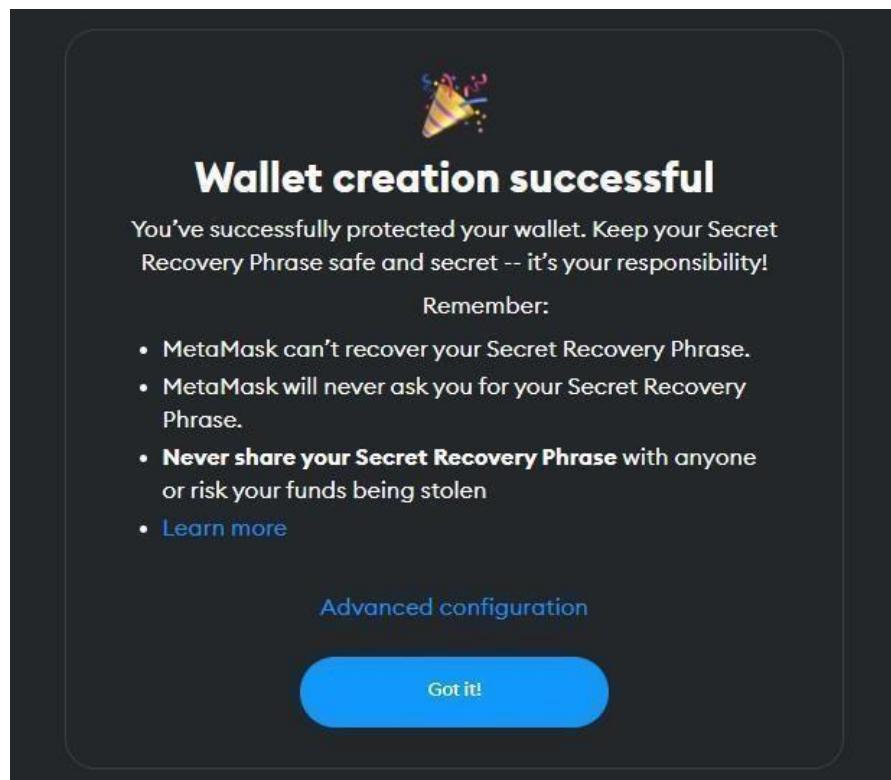
Step5-> Click on Reveal Secret Recovery Phrase button and save the words in the same sequence



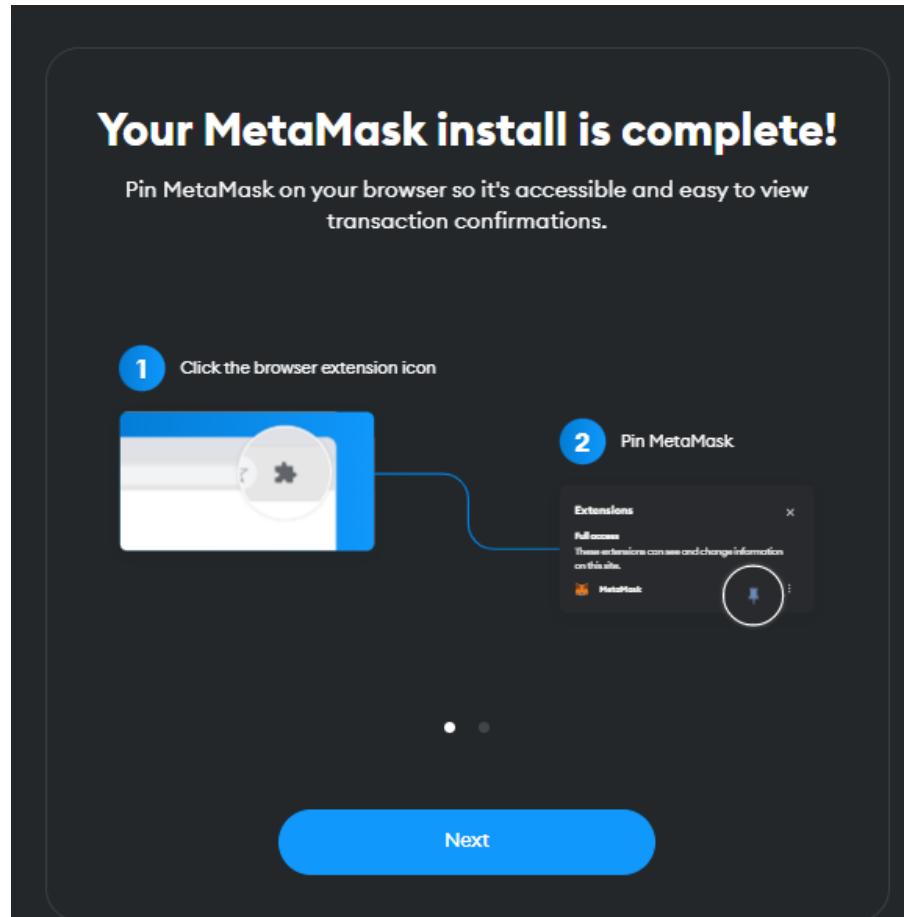
Step6-> Enter the respective words in the empty positions and click Confirm.



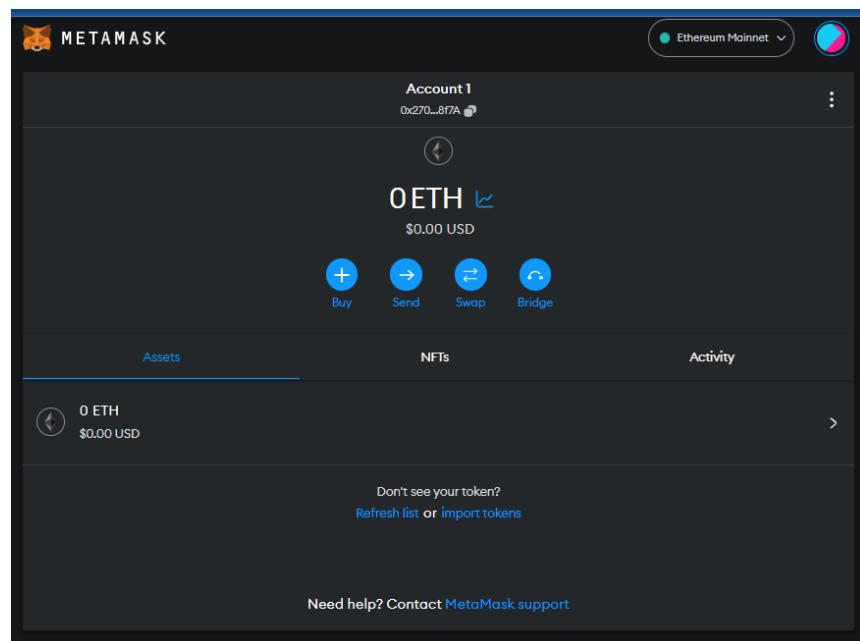
Step7-> Click Gotit!



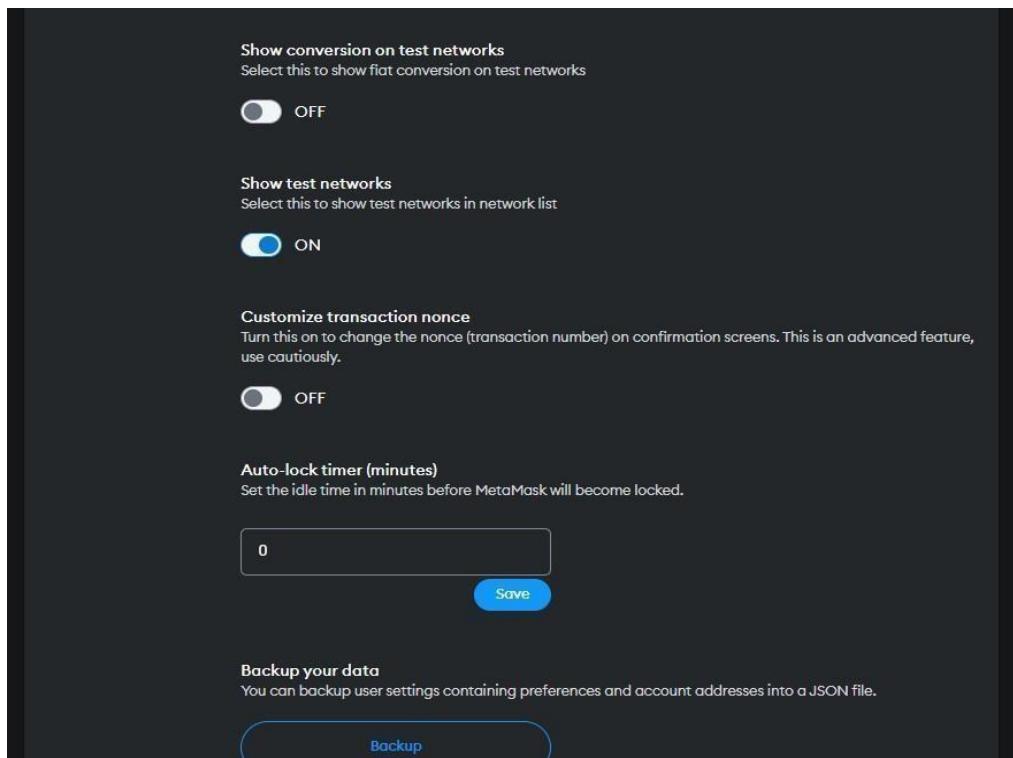
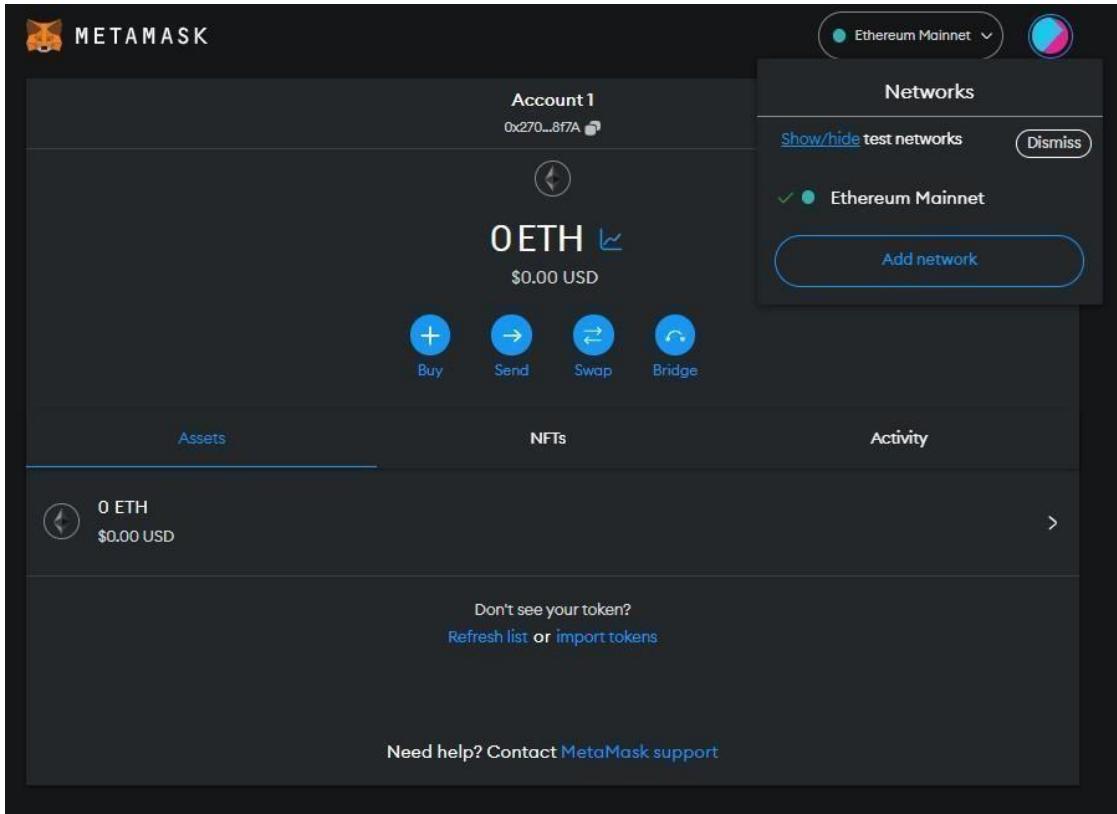
Step8-> ClickonNext



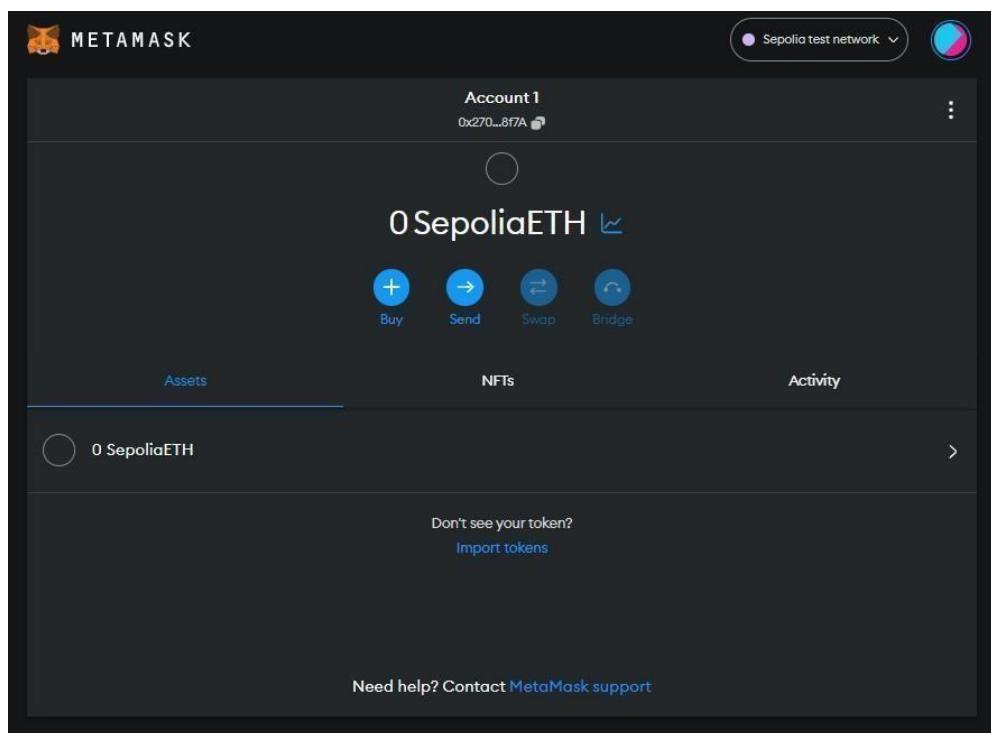
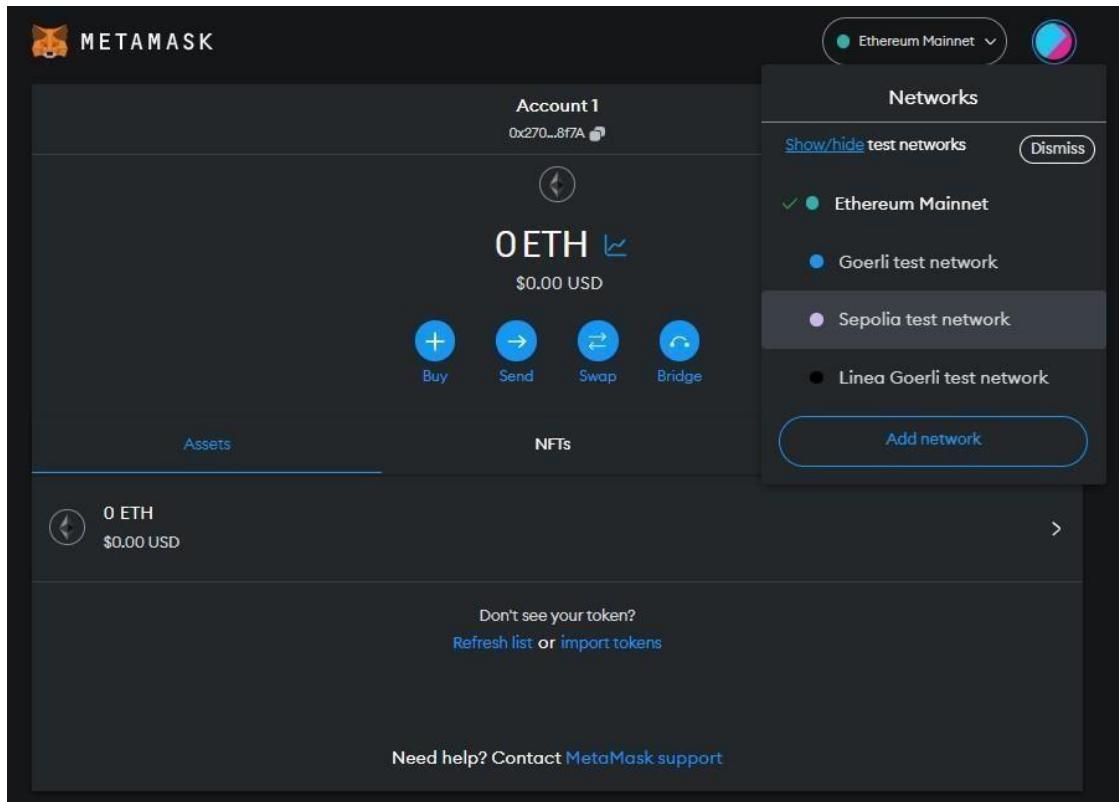
Step9-> Followingwillbethedashboard



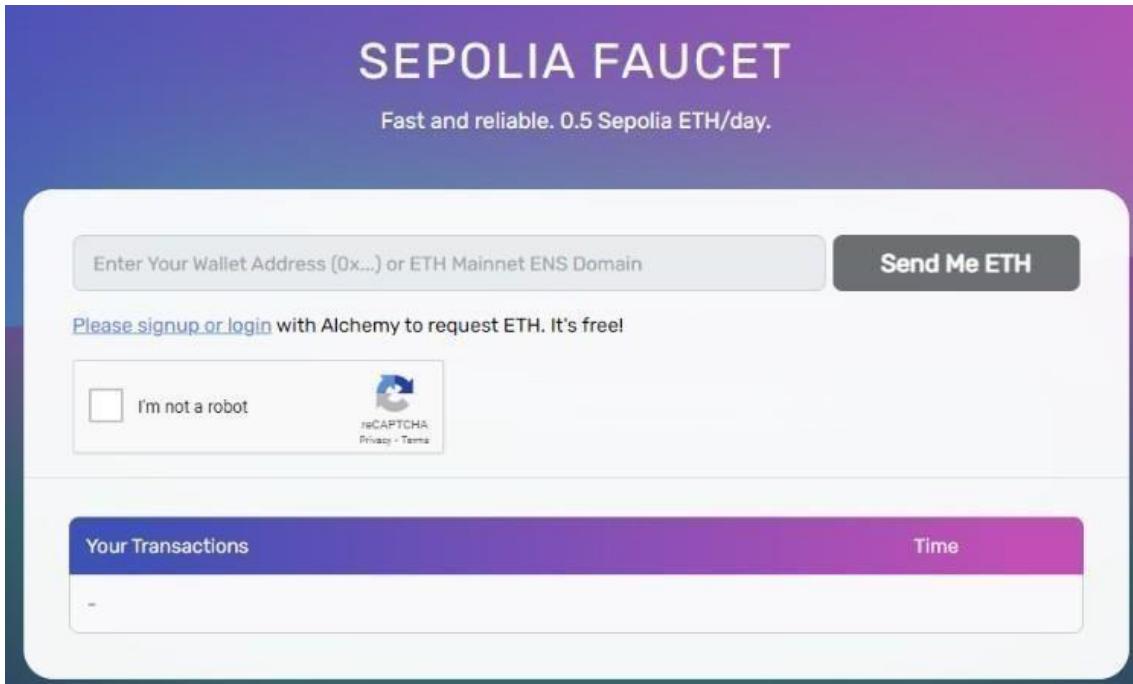
Step10->ClickonEthereumMainnet button.NextclickonShow/hidetestnetworks.



Step 11-> Check if tesnets are shown by clicking on Ethereum Mainnet button. Click on Sepolia test network.



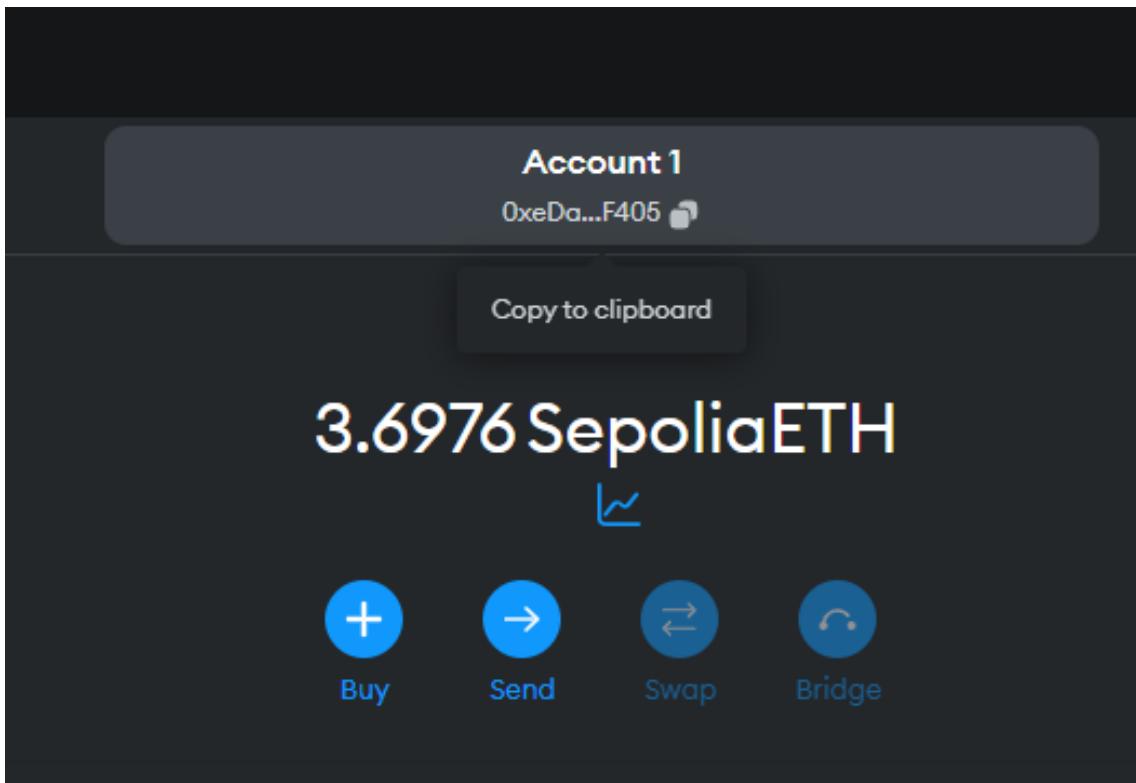
Step12->Goto <https://sepoliafaucet.com/> and Click on Alchemy Login button.



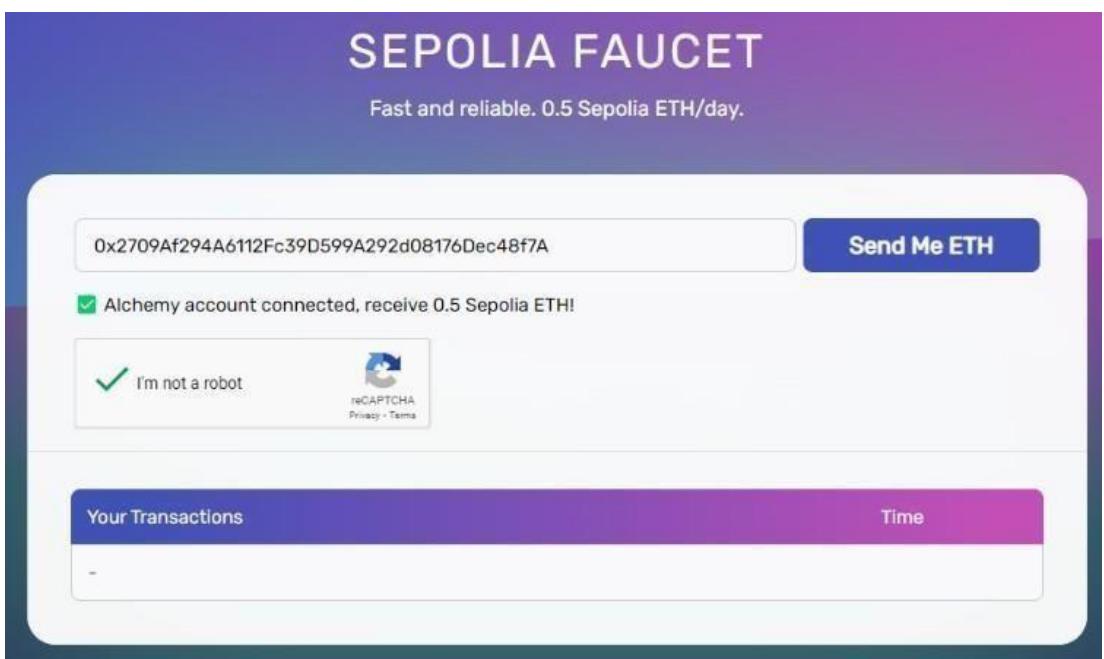
Step13->Log into agmail account in another browser tab and click on Signin with Google



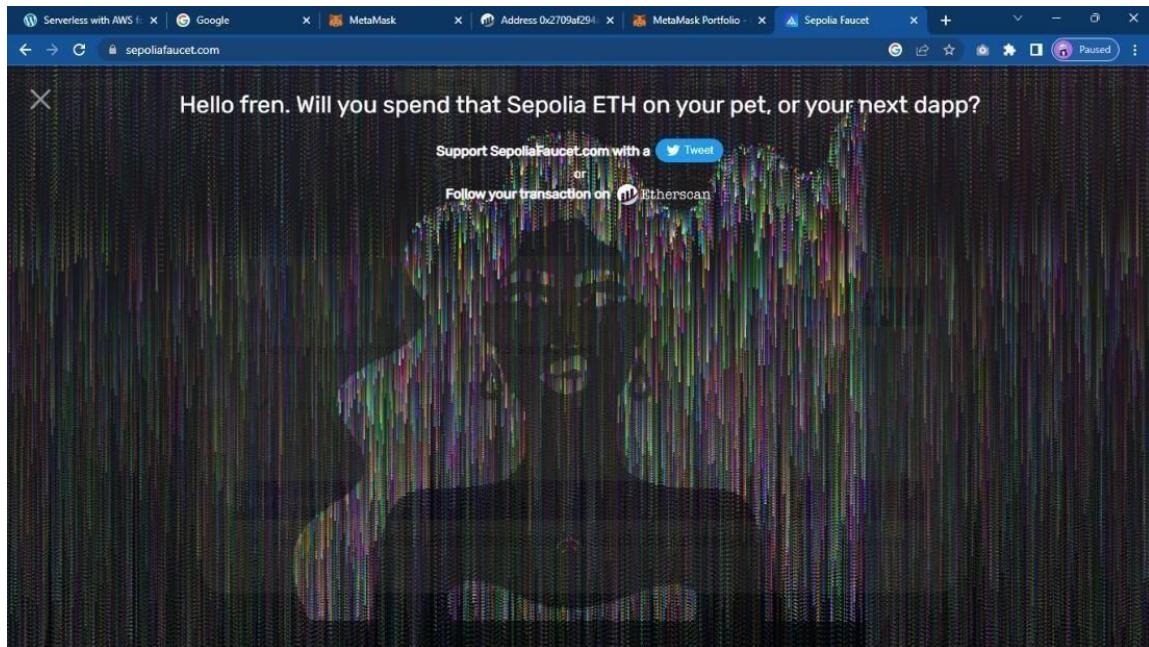
Step14->Now goto MetaMask and copy the account address.



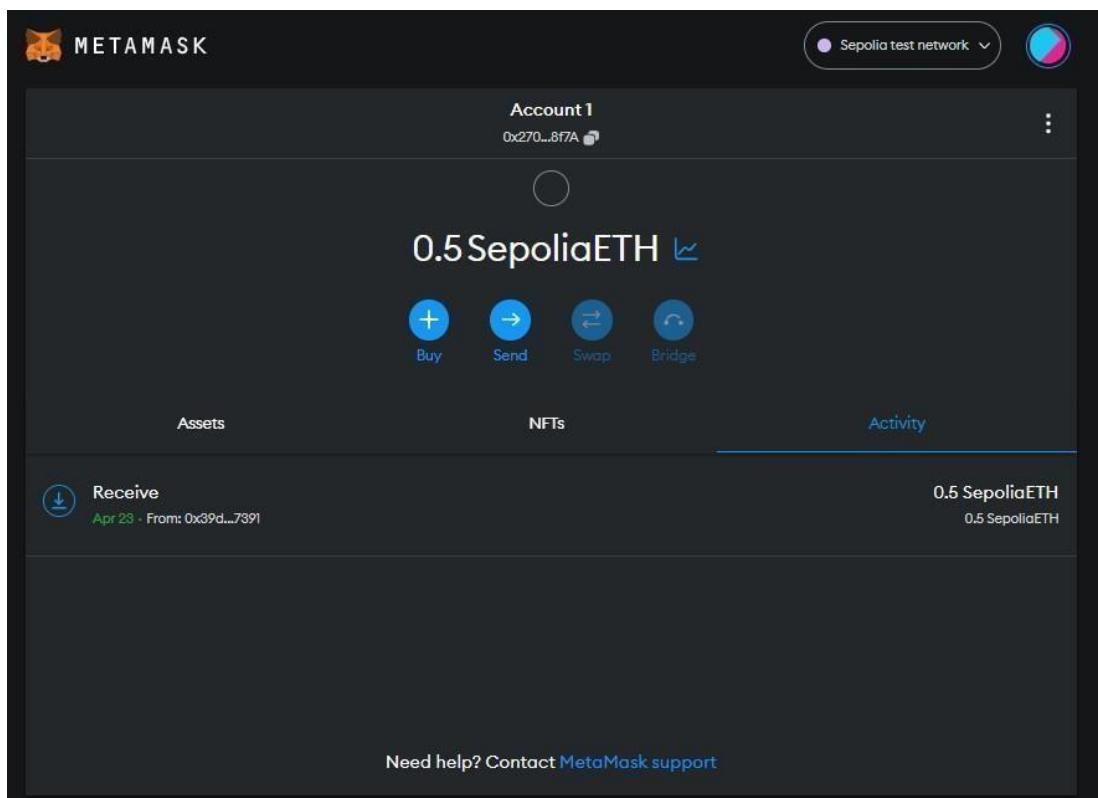
Step15->Paste the address and click on Send Me ETH.



Step16->YourETHtransferissuccesfull.Youshouldseeasimilaranimation.



Step17->CheckyourMetaMaskaccountforSepoliatestnetwork.0.5ETHwillbeadded.



PRACTICAL-3

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

1. TO EXECUTE SOLIDITY SCRIPTS GO TO -> HTTPS://REMIX.ETHEREUM.ORG/
2. OPEN CONTRACTS FOLDER AND STARTING WRITING SCRIPTS. THE SCRIPTS ARE COMPILED USING SOLIDITY COMPILER.
3. THE FOLLOWING SCRIPTS WERE COMPILED USING 0.5.0+COMMIT.1D4F565A SOLIDITY COMPILER
4. DEPLOY THE SCRIPTS TO EXECUTE CODE

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

1. Variable

```
pragma solidity^0.5.0;

contract variable_demo{uint256 s
    um=4;
    //statevariable
    int256 x;address a
    ;
    strings="welcome";

    function add(uint256)public{uint256 y=
        2;//local variable
        sum=sum+x+y:sum=sum+x+y;
    }

    function display()publicviewreturns(uint256)
    {return sum;
}
```

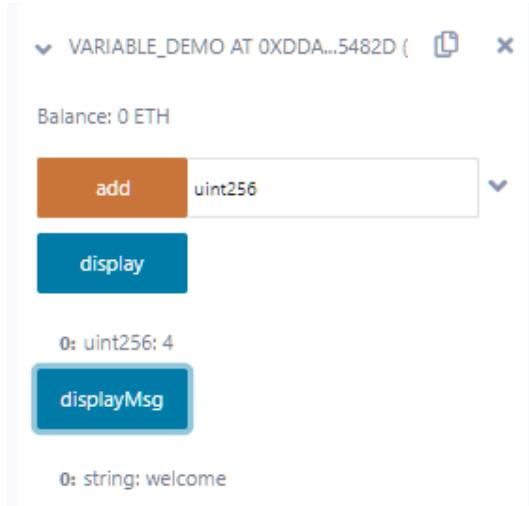


Figure 1-Displaying variable value

2. Strings

```
pragmasolidity^0.5.0;

contractLearningStrings{string
    ext;

    functiongetText()publicviewreturns(stringmemory){returntex
        t;
    }

    functionsetText()public{text
        t="hello";
    }

    functionsetTextByPassing(stringmemorymessage)public{text=m
        essage;
    }
}
```

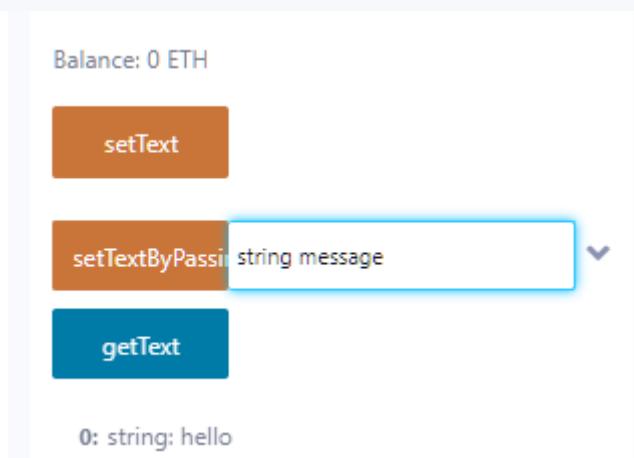


Figure2-Beforesettingnewstringvalue

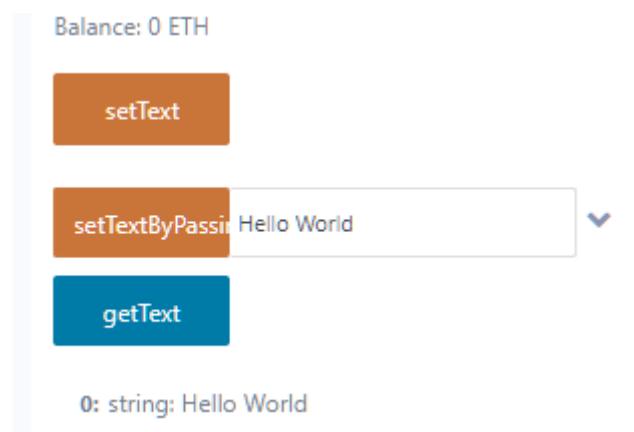


Figure3-Aftersettingstringvalue

3. Operators

```
pragmasolidity^0.5.0;

contractSolidityTest{uint
    16publica=20;uint16pu
    blicb=10;
    uint256publicsum=a+b;uint256pub
    licdiff=a-
    b;uint256publicmul=a*b;uint256p
    ublicdiv=a/b;uint256publicmod=a
    %b;uint256publicdec= --
    b;uint256publicinc= ++a;
}
```



Figure4-Alloperatorsofsoliditydisplayed

4. Array

```
pragmasolidity^0.5.0;contract  
arraydemo  
{  
    //StaticArray  
    uint[6]arr2=[10,20,30];  
  
    functiondispstaticarray()publicviewreturns(uint[6]memory)  
    {  
        returnarr2;  
    }  
  
    //DynamicArray  
    uint x=5;  
    uint[] arr1;  
  
    functionarrayDemo()public  
    {  
        while(x>0)  
        {  
            arr1.push(x);x=x-  
            1;  
        }  
    }  
  
    functiondispdynamicarray()publicviewreturns(uint[]memory)  
    {  
        returnarr1;  
    }  
}
```

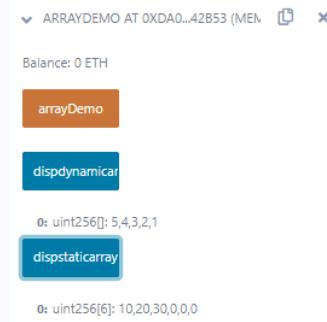


Figure5-Arraydisplayed

5. DecisionMaking

IfElse

```
pragmasolidity^0.5.0;contract
ifelsedemo
{
    uinti=10;
    functiondecision_making()publicviewreturns(stringmemory)
    {
        if(i%2==0)
        {
            return"even";
        }
        else
        {
            return"Odd";
        }
    }
}
```

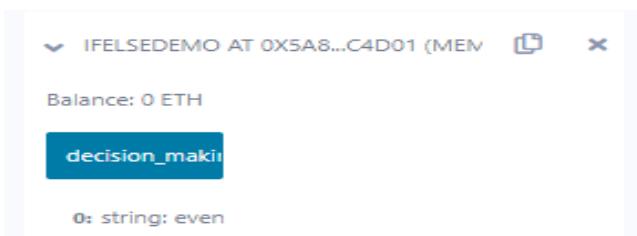


Figure6-Ifelseoutput

6. Loops

ForLoop

For Loop

```
pragmasolidity^0.5.0;contract
loopDemo
{
    uint[] data;

    functionforDemo()publicreturns(uint[]memory)
    {
        for(uinti=0;i<10;i++){
            data.push(i);
        }
        returndata;
    }

    functiondisp()publicviewreturns(uint[]memory)
    {
        returndata;
    }
}
```

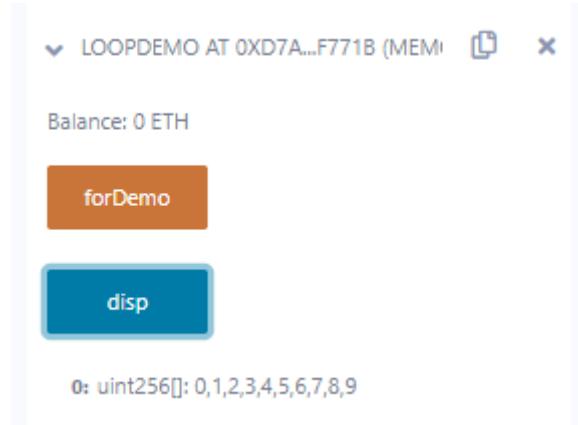


Figure 7-Appending values to array using for loop

WhileLoop

```
pragmasolidity^0.5.0;contract
```

```
whiledemo
```

```
{
```

```
    uint[]data;uintx=0;
```

```
functionwhileLoopDemo()public
```

```
{
```

```
    while(x<5)
```

```
{
```

```
        data.push(x);
```

```
        x=x+1;
```

```
}
```

```
}
```

```
functiondispwhileloop()publicviewreturns(uint[]memory)
```

```
{
```

```
    returndata;
```

```
}
```

```
}
```

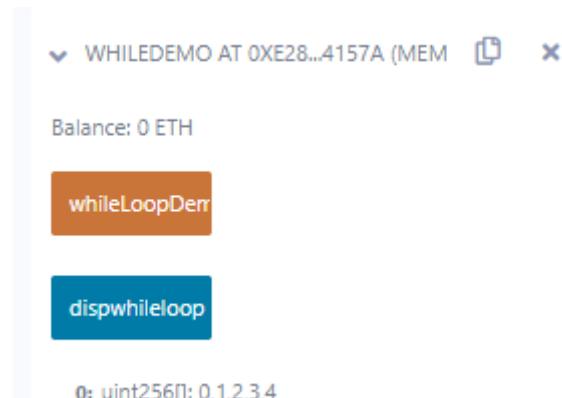


Figure8-Appendingvaluesetoarrayusingwhileloop

Do While

```
pragmasolidity^0.5.0;

//Creatingacontractcontra
ctDoWhile{
    //Declaringadynamicarrayuint256
    []data;

    //Declaringstatevariable
    uint8j=0;

    //Definingfunctiontodemonstrate
    // 'Do-Whileloop'
    functionloop()publicreturns(uint256[]memory){do{
        j++;
        data.push(j);
    }while(j<5);return
    ndata;
}
functiondisplay()publicviewreturns(uint256[]memory){return
    data;
}
}
```

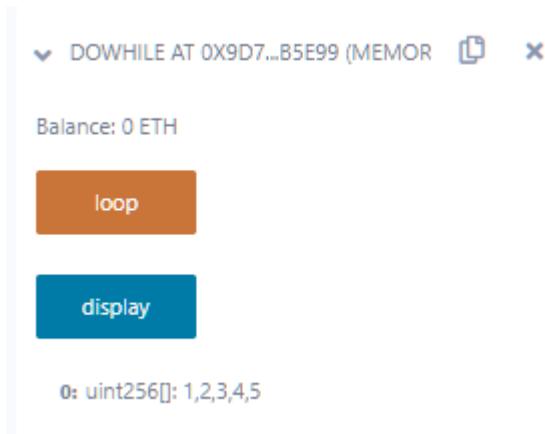


Figure9Appendingvaluestoarrayusingdowhileloop

7. Enums

```

pragmasolidity^0.5.0;

contractenumdemo{enumW
    eek_days{
        Monday,Tuesday
        ,Wednesday,Thu
        rsday,Friday,S
        aturday,Sunday
    }

    week_days
    week;week_dayschoice;
    week_daysconstantdefault_value=week_days.Sunday;

    functionset_value()public{choic
        e=week_days.Tuesday;
    }

    functionget_choice()publicviewreturns(week_days){returncho
        ice;
    }

    functionget_defaultvalue()publicviewreturns(week_days){returnd
        efault_value;
    }
}

```

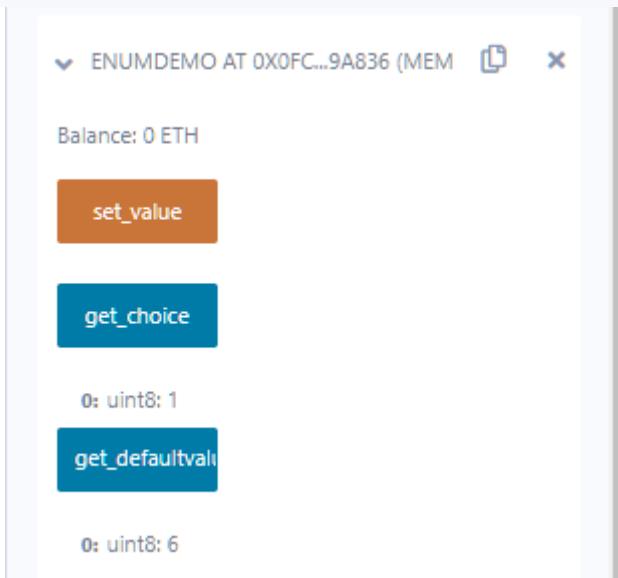


Figure10-Accessingenumvalues

8. Structs

```

pragmasolidity^0.5.0;

contractstructdemo{struct
    Book{
        stringname;string
        author;uint256id;
        boolavailability;
    }
    Bookbook2;
    Bookbook1=Book("ALittleLife","HanyaYanagihara",2,false);

    functionset_details()public{
        book2=Book("Almond","Sohnwon-pyung",1,true);
    }

    functionbook_info()p
        ublic
        viewreturn
        ns(
            stringmemory,s
            tringmemory,ui
            nt256,
            bool
        )
    {
        return(book1.name,book1.author,book1.id,book1.availability);
    }

    functionget_details()public
        viewreturn
        s(
            stringmemory,stringmemory,uint256,bool
        )
    {
        return(book2.name,book2.author,book2.id,book2.availability);
    }
}

```

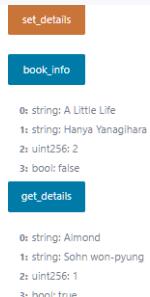


Figure 11-Structureddatatypeinsolidity

9. Mappings

```
pragmasolidity^0.5.0;

contractLedgerBalance{
    mapping(address=>uint256)publicbalances;

    functionupdateBalance(uint256newBalance)public{balances[msg.sender] =newBalance;
}
}

contractUpdater{
    functionupdateBalance()publicreturns(uint256)
    {LedgerBalanceledgerBalance=newLedgerBalance();return ledgerBalance.balances(address(this));
}
}
```

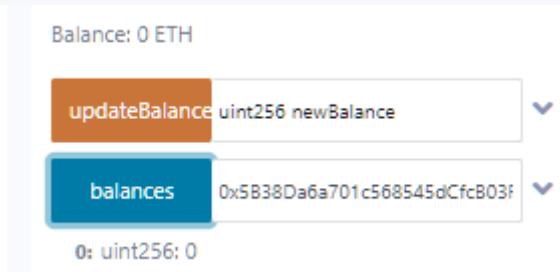


Figure12-Beforeupdatingbalance

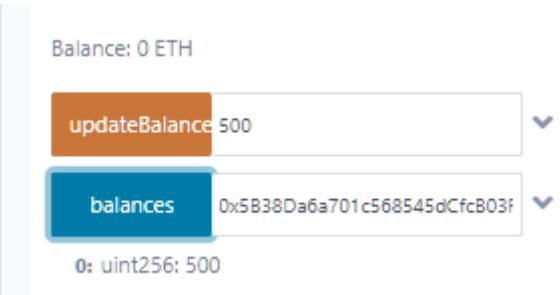


Figure13-Afterupdatingbalance

10. Conversions

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

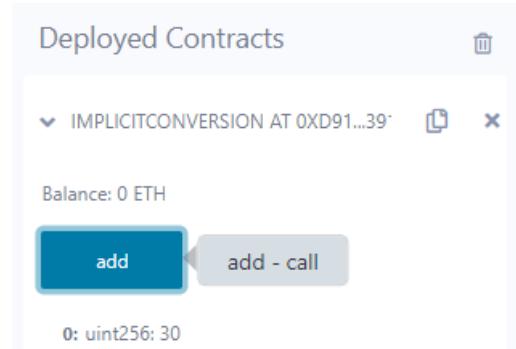
contract ImplicitConversion{
    function add() public pure returns(uint256) {uint256 a=10
        ;
        uint256 b=20; return
        a+b;
    }
}

contract ExplicitConversion{
    function convert() public pure returns(bytes memory){string memo
        str="Hello World";
        bytes memory b=bytes(str); return b
        ;
    }
}
```

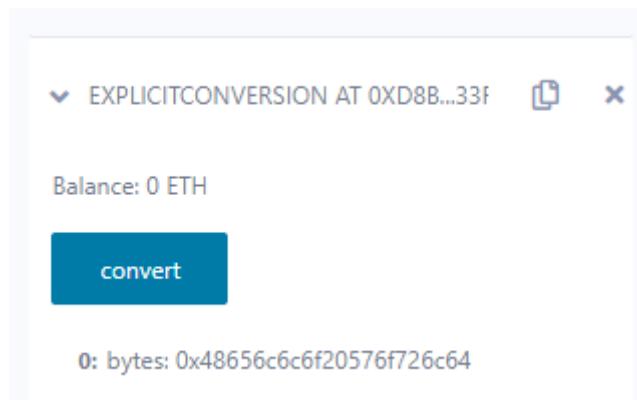
Step1-> Deploy both contracts



Step2-> Open ImplicitConversion and click on add button to sum and display value



Step3-> OpenExplicitConversionandclickonconvertbutton



11. EtherUnits

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

contract SolidityTest{
    function convert_Amount_to_Wei(uint256 Amount) public
        pure
        returns(uint256)
    {
        return Amount*1wei;
    }

    function convert_Amount_To_Ether(uint256 Amount) public
        pure
        returns(uint256)
    {
        return Amount*1ether;
    }

    function convert_Amount_To_Gwei(uint256 Amount) public
        pure
        returns(uint256)
    {
        return Amount*1gwei;
    }

    function convert_seconds_To_mins(uint256 seconds) public
        pure
        returns(uint256)
    {
        return seconds/60;
    }
}
```

```
}

function convert_seconds_To_Hours(uint256_seconds)public
{
    pure
    returns(uint256)
{
    return_seconds/3600;
}

function convert_Mins_To_Seconds(uint256_mins)public
{
    pure
    returns(uint256)
{
    return_mins*60;
}
}
```

Balance: 0 ETH

The screenshot shows a blockchain transaction interface with the following details:

- Function: convert_Amount (Value: 20)
 - Output 0: uint256: 20000000000000000000000000000000
- Function: convert_Amount (Value: 20)
 - Output 0: uint256: 200000000000
- Function: convert_Amount (Value: 20)
 - Output 0: uint256: 20
- Function: convert_Mins_To_Seconds (Value: 20)
 - Output 0: uint256: 1200
- Function: convert_seconds_To_Hours (Value: 160000)
 - Output 0: uint256: 44
- Function: convert_seconds_To_Hours (Value: 160000)
 - Output 0: uint256: 2666

Step1-> Provide values to each function and click on them

SOLIDITYTEST AT 0xD7A...F771B (ME)

Balance: 0 ETH

convert_Amount uint256 Amount

convert_Amount uint256 Amount

convert_Amount uint256 Amount

convert_Mins_ uint256 _mins

convert_seconds uint256 _seconds

convert_seconds uint256 _seconds

Balance: 0 ETH

convert_Amount 20

0: uint256: 20000000000000000000000000

convert_Amount 20

0: uint256: 200000000000

convert_Amount 20

0: uint256: 20

convert_Mins_ 20

0: uint256: 1200

convert_seconds 16000

0: uint256: 4

convert_seconds 160000

0: uint256: 2666

12. SpecialVariables

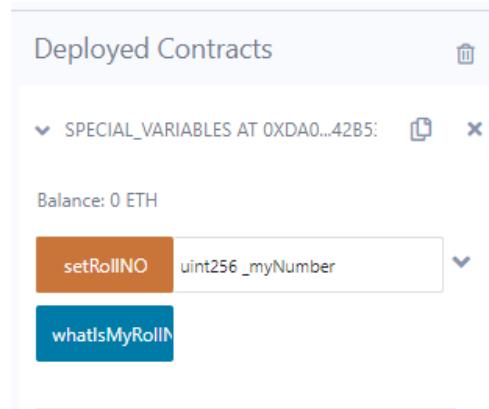
```
// SPDX-License-Identifier: MIT
pragma solidity^0.8.0;

contract Special_Variables{mapping(address=>uint256)rollNo;

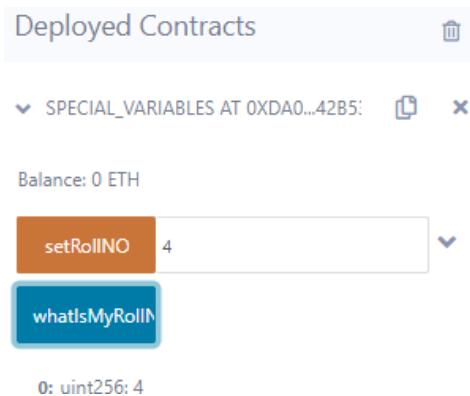
    function setRollNO(uint256 _myNumber)public{rollNo[msg.sender] =_myNumber;
}

    function whatIsMyRollNumber()public view returns(uint256){return rollNo[msg.sender];
}
}
```

Step1-> Deploy contract SpecialVariables



Step 2-> Input a number for setRollNO function and click on it & whatIsMyRollNumber button



B) Functions, FunctionModifiers, Viewfunctions, PureFunctions, FallbackFunction, Function Overloading, Mathematical functions, Cryptographic functions

1. ViewFunctions

```
pragmasolidity^0.5.0;

contract
    view_demo{uint256
    num1
    =2;uint256
    num2=4;

    functiongetResult()publicviewreturns(uint256product,uint256sum)
    {product=num1*num2;
    sum=num1+num2;
    }
```

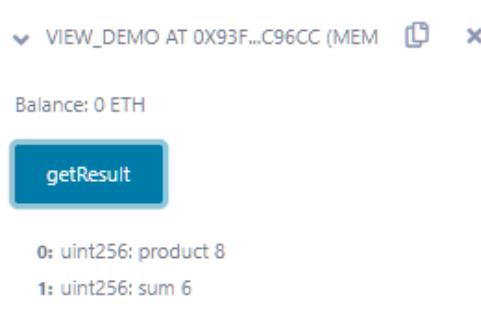


Figure14-Viewfunctiondemo

2. PureFunctions

```
pragmasolidity^0.5.0;

contractpure_demo{
    functiongetResult()publicpurereturns(uint256product,uint256sum)
    {uint256num1=2;
    uint256num2=4;product=n
    um1*num2;sum=num1
    +num2;
    }
}
```

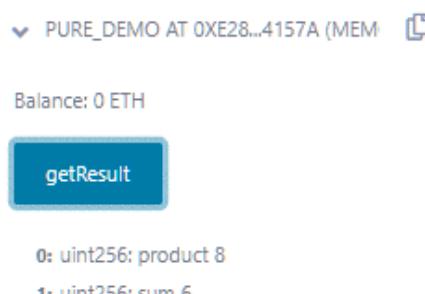


Figure15-Purefunctionoutput

3. Mathematical Functions

```
pragmasolidity^0.5.0;

contract Test{

    functionCallAddMod()publicpurereturns(uint){return

        addmod(7,3,3);

    }

    functionCallMulMod()publicpurereturns(uint){return

        mulmod(7,3,3);

    }

}
```

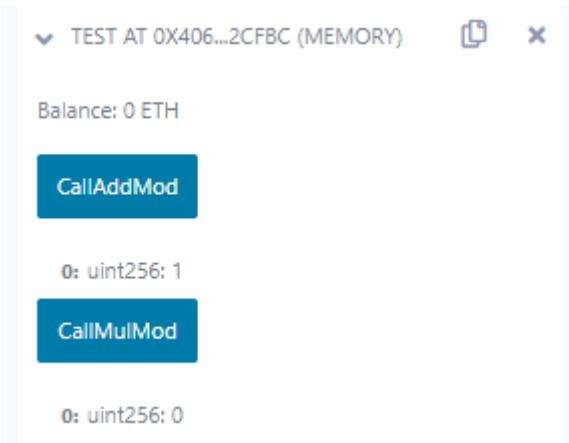


Figure 16-Mathematical functions in solidity

4. Cryptographic Functions

pragmasolidity^0.5.0;contract

Test{

```
functioncallKeccak256()publicpurereturns(bytes32result){return
    keccak256("BLOCKCHAIN");
}

functioncallsha256()publicpurereturns(bytes32result){return sha256("BLOCKCHAIN");
}

functioncallripemd()publicpurereturns(bytes20result){return
    ripemd160("BLOCKCHAIN");
}

}
```

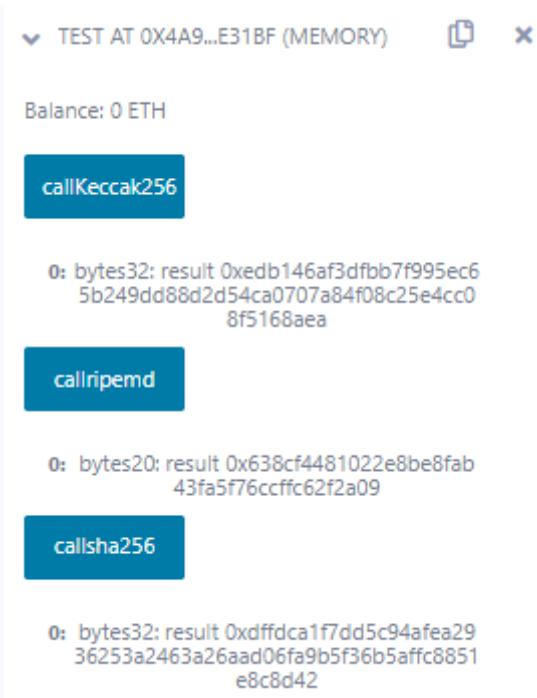
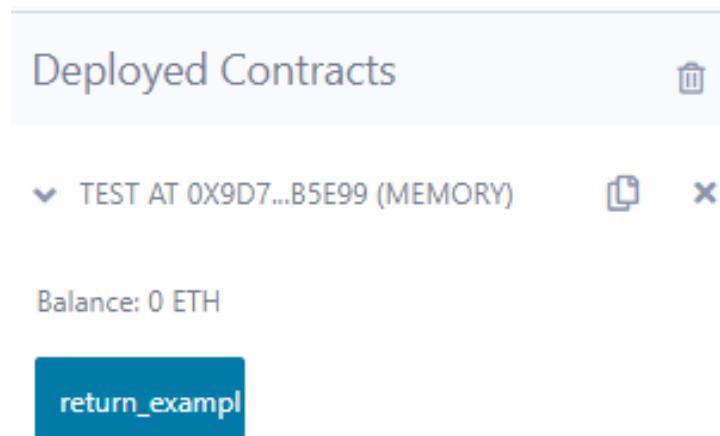


Figure17-Cryptographyalgorithmsinsolidity

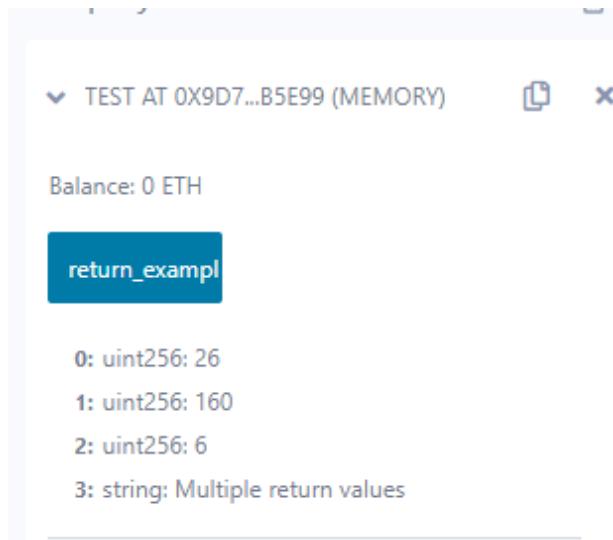
5. Functions

```
// SPDX-License-Identifier:  
MITpragmasolidity>=0.4.22<0.9.0;  
  
contract Test{  
    function return_example() public  
        view  
        pure returns  
        string memory  
    {  
        uint256 num1=10; uint256  
        num2=16;  
        uint256 sum=num1+num2; uint25  
        6 prod=num1*num2; uint256 diff  
        =num2-num1;  
        string memory message="Multiple return values"; return(sum,  
        prod,diff,message);  
    }  
}
```

Step1-> DeployTestContract



Step2-> Clickonreturn_examplebuttontodisplayallvalues



6. FallbackFunction

```
// SPDX-License-Identifier: MIT
pragma solidity^0.5.12;

contract A{
    uint256 n;

    function set(uint256 value) external {n=value; }

    function() external payable {n=0; }
}

contract example{
    function callA(address a) public returns(bool){
        (bool success,) = address(a).call(abi.encodeWithSignature("setter()")); require(success);
        address payable payableA = address(uint160(address(a))); return
        (payableA.send(2ether));
    }
}
```

Step1-> Deploy both A & example contracts

Deployed Contracts

- A AT 0X838...2A4DC (MEMORY)
- EXAMPLE AT 0X9A2...BD189 (MEMORY)

Step2-> Provide values to both deployed contracts accordingly (use any address)

A AT 0X838...2A4DC (MEMORY)

Balance: 0 ETH

set 4000

Low level interactions

CALldata

Transact

EXAMPLE AT 0X9A2...BD189 (MEMORY)

Balance: 0 ETH

callA 0x5B38Da6a701c568545dCfcB03F

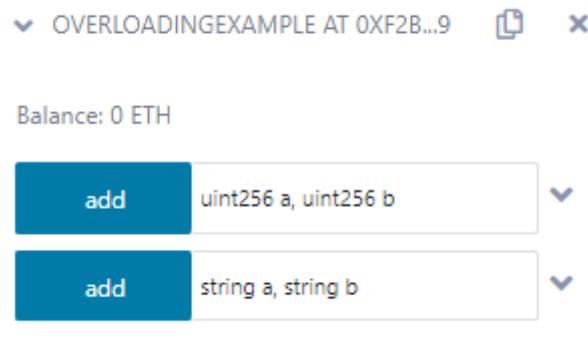
7. FunctionOverloading

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

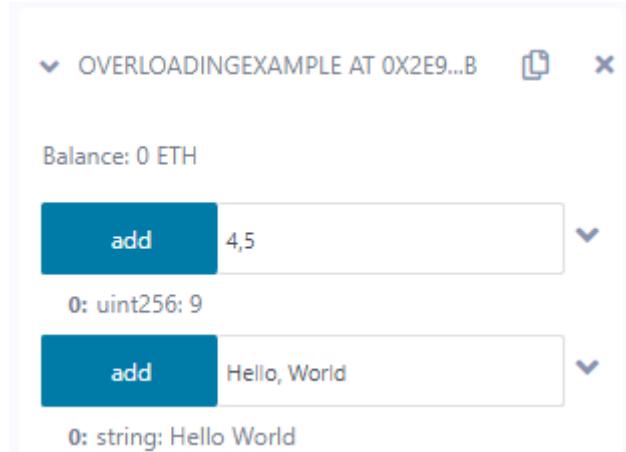
contract OverloadingExample{
    function add(uint256 a, uint256 b) public pure returns(uint256){return a+b; }

    function add(string memory a, string memory b) public
        pure
        returns(string memory)
    {
        return string(abi.encodePacked(a,b));
    }
}
```

Step1-> DeployOverloadingExamplecontract

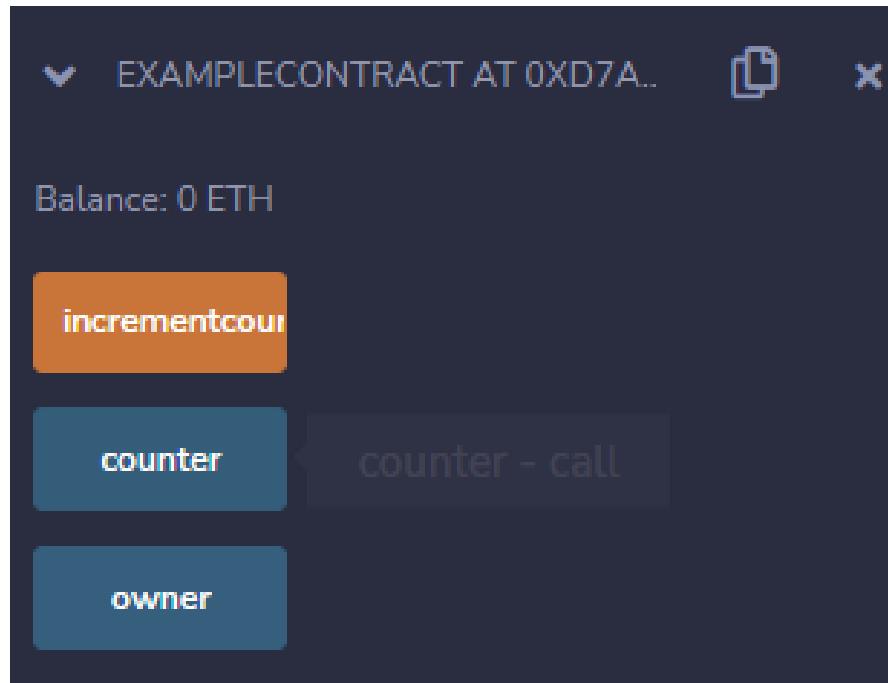


Step2-> Give integer and string values to both add functions as below

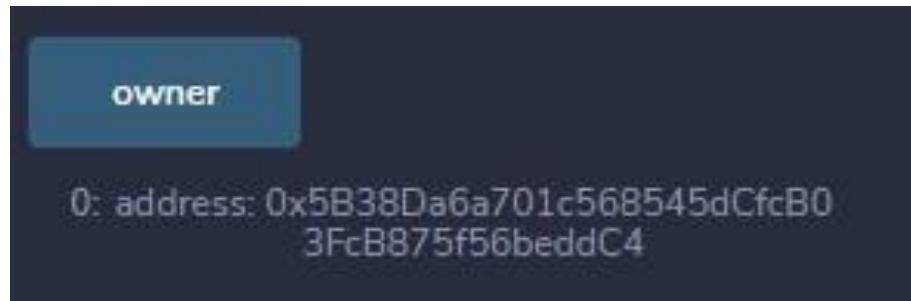


8. Functionmodifiers

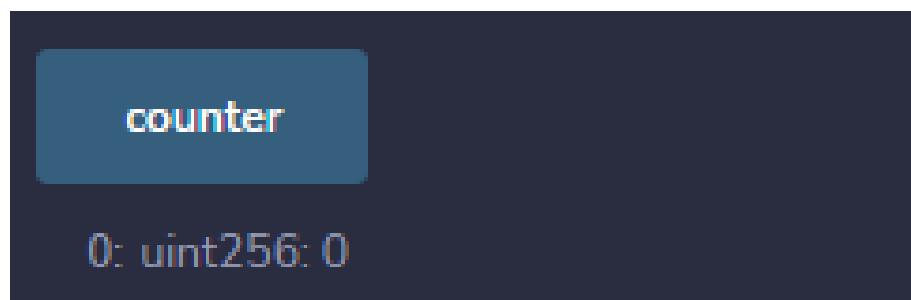
```
// SPDX-License-  
Identifier:MIT  
pragma  
solidity^0.5.0;  
  
contract ExampleContract{  
    address public owner=0x5B38Da6a701c568545dCfcB03FcB875f56beddC4; uint256 public  
    liccounter;  
  
    modifier onlyowner(){  
        require(msg.sender==owner, "Only the contract owner can call");  
        _;  
    }  
  
    function incrementcounter() public onlyowner{counter++  
        ;  
    }  
}
```



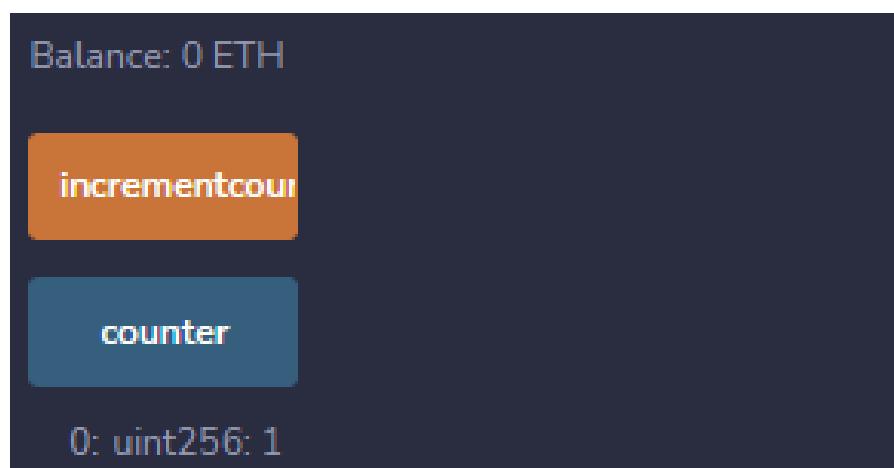
Step1-> Clickonownerbutton



Step2-> Clickoncounterbuttoninitiallyitis 0.



Step 3-> Then click on increment counter button and again clickoncounterbutton,thecounterhas been increased



PRACTICAL-4

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

A) Withdrawal Pattern, Restricted Access

1) Withdrawal Pattern

```
// SPDX-License-Identifier: MIT
pragma solidity 0.8.18;

contract WithdrawalPattern {
    address owner;
    uint256 lockedBalance;
    uint256 withdrawableBalance;

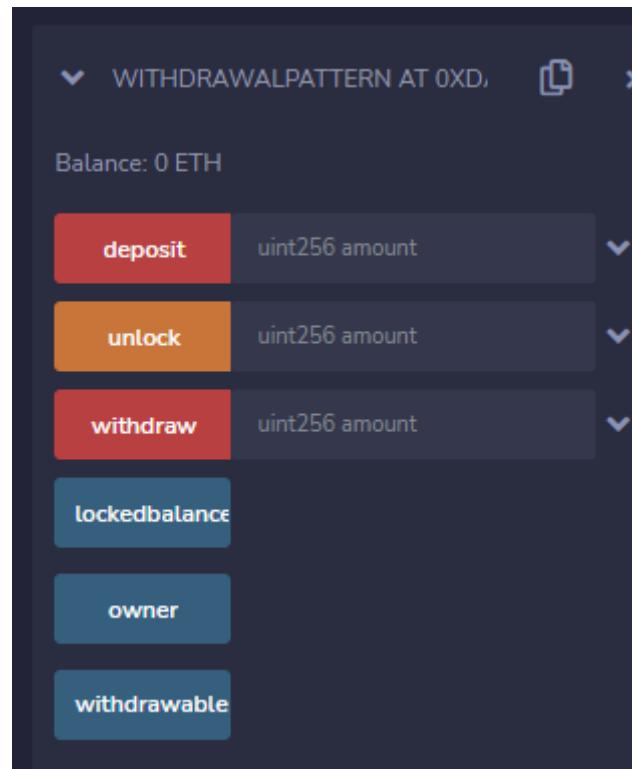
    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Only the owner can call this function");
    }

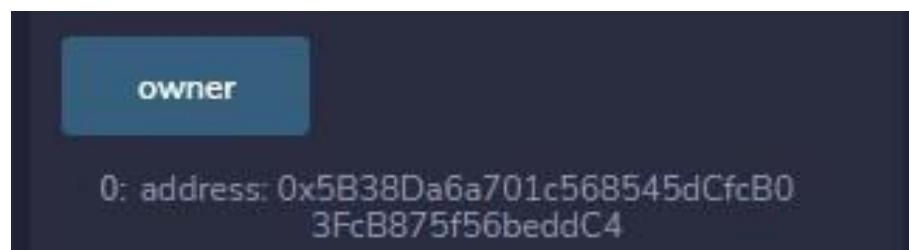
    function deposit(uint256 amount) public payable {
        require(amount > 0, "Amount must be greater than zero");
        lockedBalance += amount;
    }

    function withdraw(uint256 amount) public payable onlyOwner {
        require(amount <= withdrawableBalance, "Insufficient withdrawable balance");
        withdrawableBalance -= amount;
        payable(msg.sender).transfer(amount);
    }

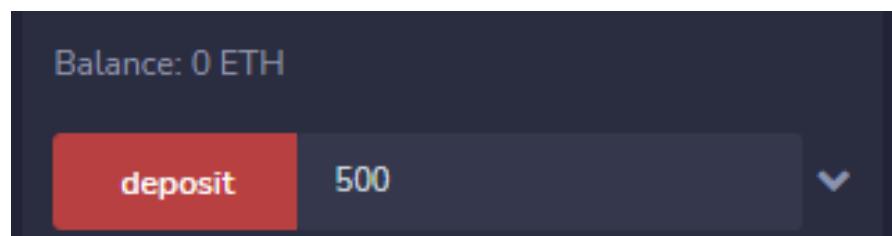
    function unlock(uint256 amount) public onlyOwner {
        require(amount <= lockedBalance, "Insufficient locked balance");
        lockedBalance -= amount;
        withdrawableBalance += amount;
    }
}
```

Output:**Flow of execution**

Step1-> Click on owner



Step2-> Enter an amount and click on deposit



Step3-> Clickonlockedbalancebuttontodisplaythelockedamountintheaccount



Step4-> Clickonwithdrawablebalancebutton

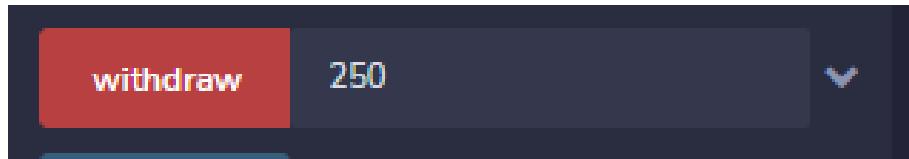


Step 5-> Clickonunlockbuttonandenteranyamounttotransferamounttowithdrawable
balance.Check locked balance and withdrawable balance.



Step6-> Enter any amount you want to withdraw and Click the withdraw button.

You should get an error and the transaction should be reverted.



```
CALL [call] from: 0x5838Da6a701c568545dCfcB03FcB875f56beddC4 to: WithdrawalPattern.withdrawablebalance() data: 0xd11...c9cb7
transact to WithdrawalPattern.withdraw pending ...

transact to WithdrawalPattern.withdraw errored: VM error: revert.

revert
    The transaction has been reverted to the initial state.
    Note: The called function should be payable if you send value and the value you send should be less than your current balance.
    Debug the transaction to get more information.

✖ [vm] from: 0x583...eddC4 to: WithdrawalPattern.withdraw(uint256) 0xdda...5482d value: 0 wei data: 0x2e1...000fa logs: 0 hash: 0x128...c475c
transact to WithdrawalPattern.withdraw pending ...

transact to WithdrawalPattern.withdraw errored: VM error: revert.

revert
    The transaction has been reverted to the initial state.
    Note: The called function should be payable if you send value and the value you send should be less than your current balance.
    Debug the transaction to get more information.

✖ [vm] from: 0x583...eddC4 to: WithdrawalPattern.withdraw(uint256) 0xdda...5482d value: 0 wei data: 0x2e1...000fa logs: 0 hash: 0x3e3...0937c
```

2) RestrictedAccess

```
// SPDX-License-
Identifier: MIT
pragma
solidity^0.8.18;

contract RestrictedAccess{
    address public owner=msg.sender;
    uint256 public creationTime=block.timestamp;

    modifier onlyBy(address _account){
        require(msg.sender==_account, "Sendernotauthorized!");
        _;
    }

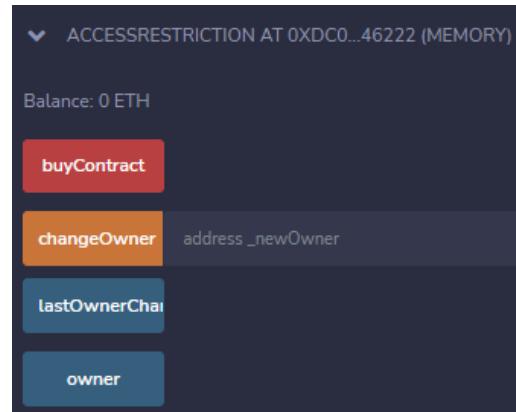
    modifier onlyAfter(uint256 _time){
        require(block.timestamp>=_time, "Functionwascalledtooearly!");
        _;
    }

    modifier costs(uint256 _amount){
        require(msg.value>=_amount, "NotenoughEtherprovided!");
        _;
    }

    function forceOwnerChange(address _newOwner) public
        payable
        costs(200
        ether)
    {
        owner=_newOwner;
    }

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

    function disown() public
        onlyBy(owner)
        onlyAfter(creationTime+3weeks)
    {
        delete owner;
    }
}
```

Output:**Flow of execution**

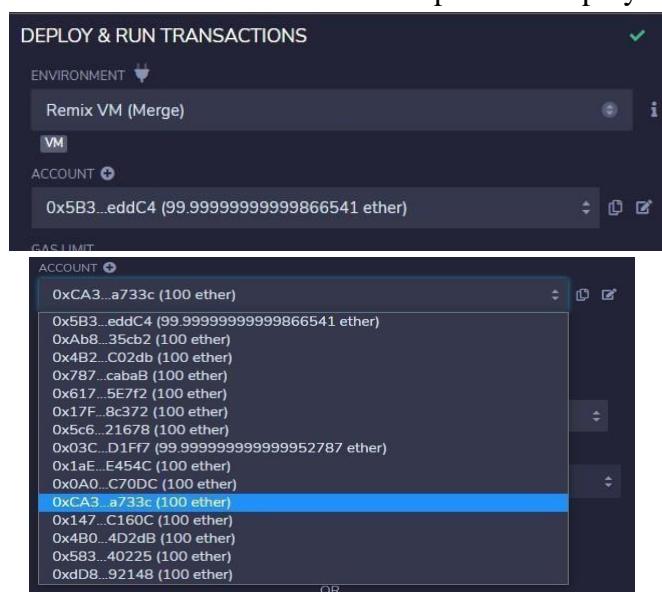
Step1-> Click on owner to create an owner object



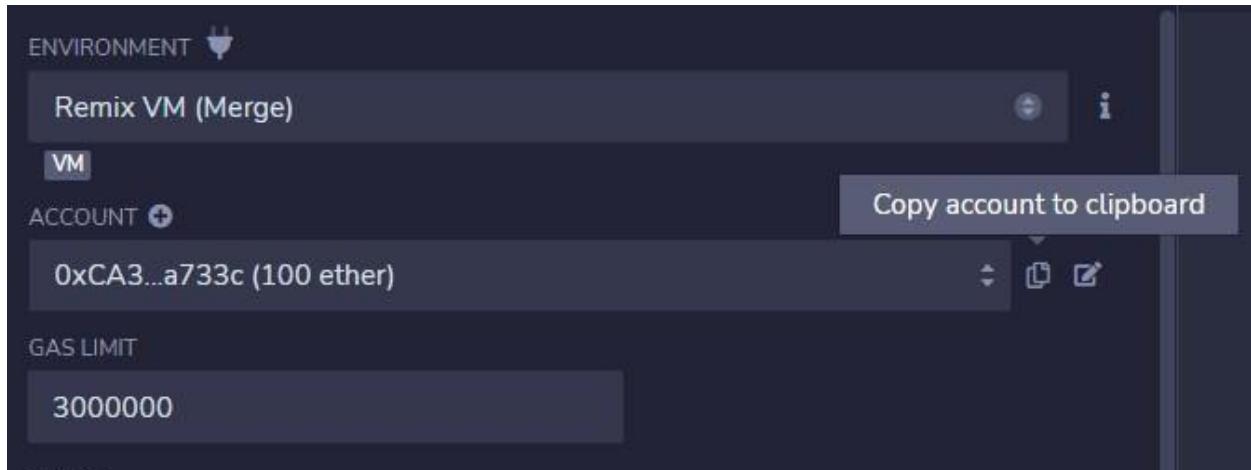
Step2-> Click on lastOwnerChange button



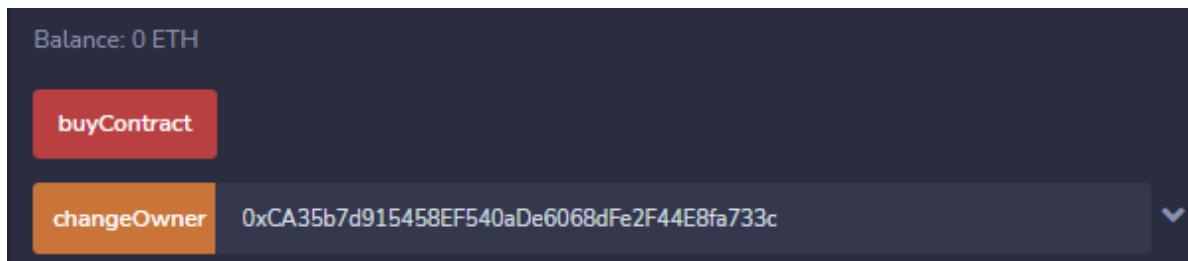
Step3-> Change the address of the account from Account dropdown in Deploy tab of Remix IDE.



Step4-> Copytheaddress



Step5-> PastetheaddressinchangeOwnerinputandclickonchangeOwner.



Step6-> Youshouldgetanerrorasfollowing



Step7-> Ifyouclickonbuycontractitshouldgiveanerrorasfollows



Step 8-> Now,pastetheactualaddressoftheaccountinthechangeownerinputand click on changeowner

```
[vm] from: 0xCA3...a733c to: AccessRestriction.changeOwner(address) 0x0fC...9A836 value: 0 wei data: 0xa6f...eddc4 logs: 0
hash: 0xd88...cc14a
transact to AccessRestriction.changeOwner pending ...

transact to AccessRestriction.changeOwner errored: VM error: revert.

revert
The transaction has been reverted to the initial state.
Note: The called function should be payable if you send value and the value you send should be less than your current balance.
Debug the transaction to get more information.

[vm] from: 0xCA3...a733c to: AccessRestriction.changeOwner(address) 0x0fC...9A836 value: 0 wei data: 0xa6f...eddc4 logs: 0
hash: 0x3cf...85a41
```

B) Contracts,Inheritance,Constructors,AbstractContracts,Interfaces

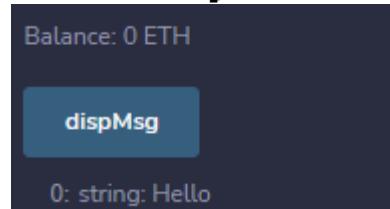
1) Contract

```
pragmasolidity^0.5.0;

contractContract_demo{string m
essage="Hello";

functiondispMsg()publicviewreturns(stringmemory){returnmes
sage;
}
}
```

Output



2) Inheritance

```
pragmasolidity>=0.4.22<0.6.0;

contractParent{uint256in
    ternalsum;

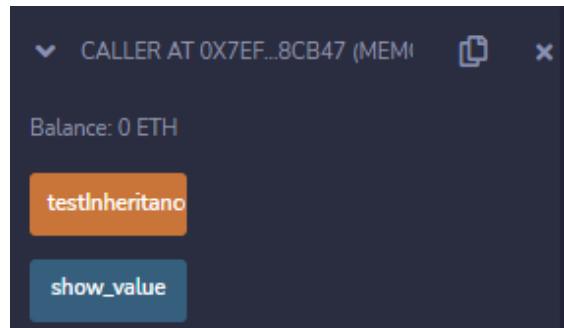
    functionsetValue()external{uint256a=
        10;
        uint256b=20;sum=a
        +b;
    }
}

contractchildisParent{
    functiongetValue()externalviewreturns(uint256){return
        sum;
    }
}

contractcaller{
    childcc=newchild();

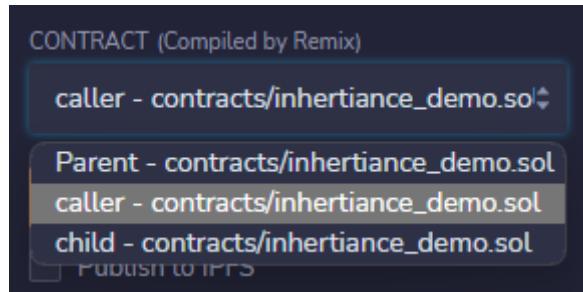
    functiontestInheritance()publicreturns(uint256){
        cc.setValue();
        returncc.getValue();
    }

    functionshow_value()publicviewreturns(uint256){returncc
        .getValue();
    }
}
```

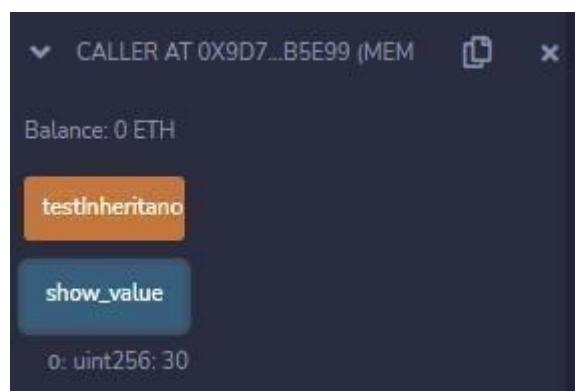
Output:

Flow of execution

Step1-> Select caller contract to deploy in Contract and deploy



Step2-> Click testInheritance and then click on show_value to view value



3) Abstract Contracts

```
// SPDX-License-Identifier: MIT
pragma solidity^0.5.17;

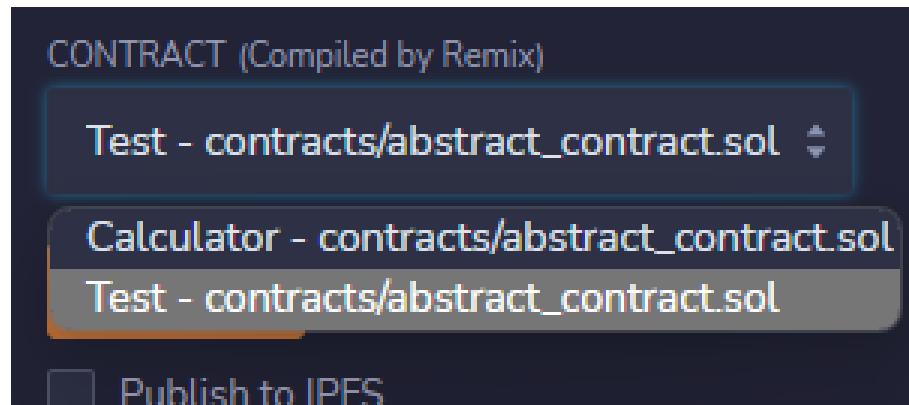
contract Calculator{
    function getResult() external view returns(uint256);
}

contract TestCalculator{constructor() public{

    function getResult() external view returns(uint256){uint256 a=1;
        ;
        uint256 b=2;
        uint256 result=a+b; return
        result;
    }
}}
```

Outputs:**Flowofexecution**

Step1-> SelectTestcontractanddeploy



Step2-> Thecontactwilldeployasbelow



Step3-> ClickongetResulttogetsumofa+b



1) Constructors

```
// SPDX-License-  
Identifier: MIT  
pragma  
solidity^0.5.0;  
  
// Creating a  
contract contractconstructorEx  
ample{  
    stringstr;  
  
    constructor()public{  
        str="GeeksForGeeks";  
    }  
  
    functiongetValue()publicviewreturns(stringmemory){returnst  
        r;  
    }  
}
```

Outputs**Flow of execution**

Step1-> Click on getValue to print strin



2) Interfaces

```
pragmasolidity^0.5.0;

interfaceCalculator{
    functiongetResult()externalviewreturns(uint);
}

contractTestisCalculator{constr
    uctor()public{}
    functiongetResult()externalviewreturns(uint){uinta=1
        ;
        uintb=2;
        uintresult=a+b;returnresu
        lt;
    }
}
```

Outputs:**Flowofexecution**

Step1-> Clickon getResulttодisplaysum



C) Libraries,Assembly,Events,Errorhandling.

1) Libraries

myLib.sol Code

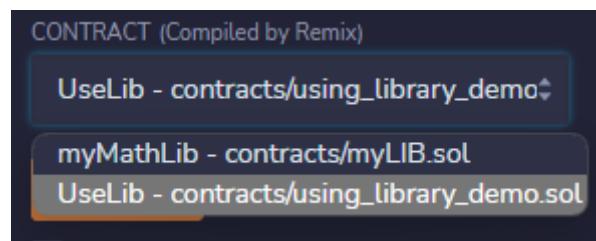
```
// SPDX-License-
Identifier:MIT

```

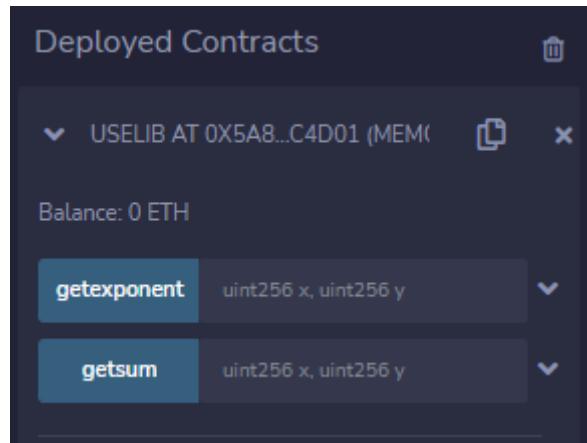
Outputs:

Flow of execution

Step1-> Change contract to UseLib and deploy.



Step2-> The deployed contract should be same as below



Step3-> Input values to both getexponent and getsum functions as below



Step4-> Execute both functions. You will get below output



2) Assembly

```

// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16<0.9.0;

contract InlineAssembly{
    //Defining function
    function add(uint256 a) public view returns(uint256 b){assembly{
        let c:=add(a,16) msto
        re(0x80,c)
        {
            let d:=add(sload(c),12)b :=d
        }
        b:=add(b,c)
    }
}
}

```

Outputs



Flow of execution

Step1-> Input a number for add function



Step2-> Clickaddtooutputsum



3) Events

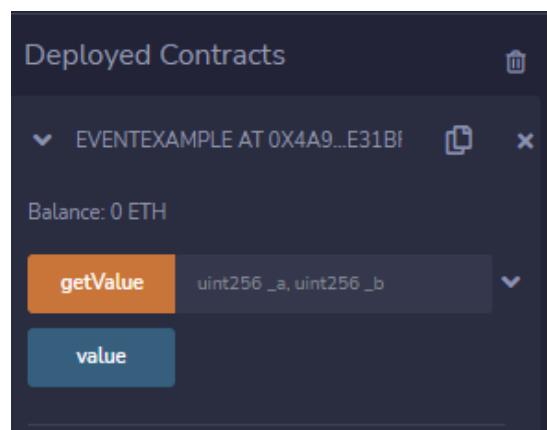
```
// SPDX-License-
Identifier:MIT
pragma
solidity^0.5.0;

// Creating a
contract contracteventExam
ple{
    //Declaringstatevariablesuint25
6publicvalue=0;

    //Declaringanevent
    event Increment(addressowner);

    //Definingafunctionforloggingevent
    functiongetValue(uint256_a,uint256_b)public{emitInc
        rement(msg.sender);
        value=_a+_b;
    }
}
```

Output



Flowofexecution

Step1-> Provide values to get Value function and click on it.



Step2-> In the terminal check for logs

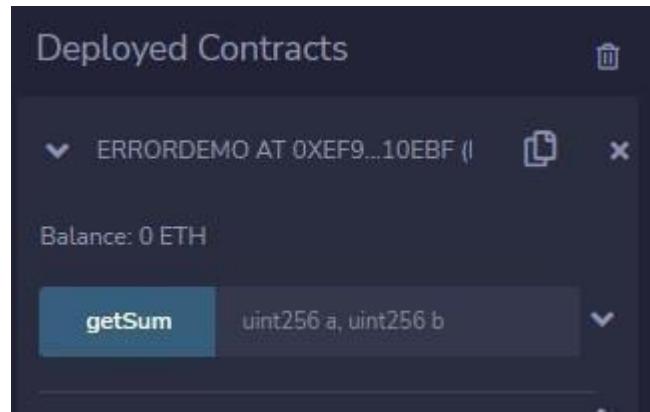
```
logs
[{"from": "0x4a9C121080f6D9250Fc0143f41B595fD172E31bf",
 "topic": "0xfc3a67c9f0b5967ae4041ed898b05ec1fa49d2a3c22336247201d71be6f97120",
 "event": "Increment",
 "args": {"0": "0x5838Da6a701c568545dCfcB03FcB875f56beddC4",
          "owner": "0x5838Da6a701c568545dCfcB03FcB875f56beddC4"}]
```

4) Error Handling

```
// SPDX-License-Identifier: MIT
pragma solidity^0.5.17;

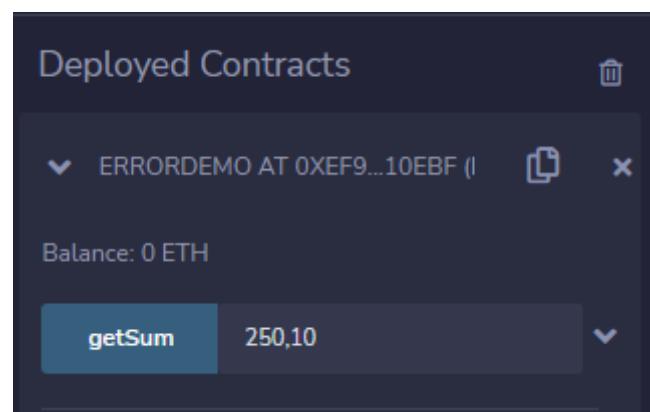
contract ErrorDemo{
    function getSum(uint256 a, uint256 b) public pure returns (uint256) {
        uint256 sum = a + b;
        // require(sum < 255, "Invalid");
        assert(sum < 255);
        return sum;
    }
}
```

Output



Flow of execution

Step1-> Provides some values and press on getSum



Step2-> Check terminal panel

```
creation of ErrorDemo pending...

[vm] from: 0x5B3...eddC4 to: ErrorDemo.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0x6b3...56a6f
call to ErrorDemo.getSum

call [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: ErrorDemo.getSum(uint256,uint256) data: 0x8e8...0000a
call to ErrorDemo.getSum errored: VM error: invalid opcode.

invalid opcode

The execution might have thrown.

Debug the transaction to get more information.
```

PRACTICAL-5

Aim: Write a program to demonstrate mining of ether

```
constWeb3=require('web3');

constweb3=newWeb3(new
Web3.providers.HttpProvider('http:127.0.0.1:7545'));ReplacewithyourGanacheHTTPprovider

asyncfunctionmine(){
    constaccounts=awaitweb3.eth.getAccounts();constcoinbaseacc1=
    accounts[0];
    constcoinbaseacc2=accounts[1];
    console.log(`MiningetheronGanachewithcoinbaseaddress:
${coinbaseacc1}`);
    while(true){try{
        awaitweb3.eth.sendTransaction({from:
            coinbaseacc1,
            to:coinbaseacc2,value:
            50,
        });
        console.log(`Minedanewblock!`);
    }catch(err){console.error(err);
    }
}
}

mine();
```

Output:

```
C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\prac6>npm install web3
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated uglify-es@3.3.9: support for ECMAScript is superseded by 'uglify-js' as of v3.13.0

added 651 packages, and audited 1097 packages in 1m

85 packages are looking for funding
  run `npm fund` for details

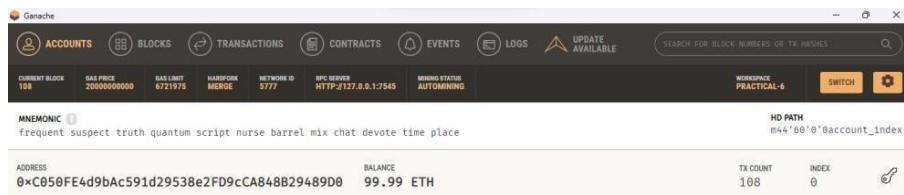
19 vulnerabilities (9 moderate, 10 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

```
C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\prac6>node ethermine.js
Mining ether on Ganache with coinbase address: 0xC050FE4d9bAc591d29538e2FD9cCA848B29489D0
Mined a new block!
```



PRACTICAL-6

Aim:Demonstrate the running of the blockchain node

Step1->Create a folder named ethermine and a JSON file named genesis.json and write the following lines in it.

```
{  
  "config":{  
    "chainId":3792,  
    "homesteadBlock":0,  
    "eip150Block":0,  
    "eip155Block":0,  
    "eip158Block":0  
  },  
  "difficulty":"2000",  
  "gasLimit":"2100000","alloc":  
  {  
    "0x0b6C4c81f58B8d692A7B46AD1e16a1147c25299F": {"balance":  
      "9000000000000000000000000000000"  
    }  
  }  
}
```

```
genesis.json ethnode_steps.txt  
1  {  
2   "config": {  
3     "chainId": 3792,  
4     "homesteadBlock": 0,  
5     "eip150Block": 0,  
6     "eip155Block": 0,  
7     "eip158Block": 0  
8   },  
9   "difficulty": "2000",  
10  "gasLimit": "2100000",  
11  "alloc": {  
12    "0x3A7b442afa94ba96396DF86336172947Fa9C48BE":  
13      {  
14        "balance": "9000000000000000000000000000000"  
15      }  
16    }  
17  }
```

Step 2->Run command geth account new –

datadirC:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\etherminetestnet-blockchain

```
C:\Users\Achsah>geth account new --datadir C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\ethermine
INFO [04-20|20:03:09.337] Maximum peer count           ETH=50 LES=0 total=50
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key: 0x77CB2BdBC0f1743bC73E92fla8b1AB80BEDB35AE
Path of the secret key file: C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\ethermine\key
store\UTC--2023-04-20T14-33-26.959134300Z--77cb2bdbc0f1743bc73e92fla8b1ab80bedb35ae

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!
```

Step 3-> Run command geth account new --

**datadirC:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\eth
ermine**

```
C:\Users\Achsah>geth --datadir C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\ethermine init C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\ethermine\genesis.json
Fatal: invalid genesis file: math/big: cannot unmarshal "\"3792\""" into a *big.Int

C:\Users\Achsah>geth --datadir C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\ethermine init C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\ethermine\genesis.json
INFO [04-20|20:23:47.707] Maximum peer count           ETH=50 LES=0 total=50
INFO [04-20|20:23:47.717] Set global gas cap          cap=50,000,000
INFO [04-20|20:23:47.720] Using leveldb as the backing database
INFO [04-20|20:23:47.720] Allocated cache and file handles   database=C:\Users\Achsah\Document
s\MScIT\sem4\blockchain_practical\ethermine\geth\chaindata cache=16.00MiB handles=16
INFO [04-20|20:23:47.741] Using LevelDB as the backing database
INFO [04-20|20:23:47.765] Opened ancient database        database=C:\Users\Achsah\Document
s\MScIT\sem4\blockchain_practical\ethermine\geth\chaindata\ancient\chain readonly=false
INFO [04-20|20:23:47.767] Writing custom genesis block
INFO [04-20|20:23:47.773] Persisted trie from memory database   nodes=1 size=147.00B time="636.4μ
```

**Step4->Run command geth --identity "localB" --http --http.port "8280" --http.corsdomain "*" --
http.api "db,eth,net,web3" --datadir "C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\ethermine" --port "30303" -
nodiscover --networkid 5777 console. This command will enable geth console.**

```
C:\Users\Achsah>geth --identity "localB" --http --http.port "8280" --http.corsdomain "*" --http.api
"db,eth,net,web3" --datadir "C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\ethermine" --
port "30303" --nodiscover --networkid 5777 console
INFO [04-20|20:29:41.383] Maximum peer count           ETH=50 LES=0 total=50
INFO [04-20|20:29:41.389] Set global gas cap          cap=50,000,000
INFO [04-20|20:29:41.392] Allocated trie memory caches   clean=154.00MiB dirty=256.00MiB
INFO [04-20|20:29:41.396] Using leveldb as the backing database
INFO [04-20|20:29:41.396] Allocated cache and file handles   database=C:\Users\Achsah\Document
s\MScIT\sem4\blockchain_practical\ethermine\geth\chaindata cache=512.00MiB handles=8192
INFO [04-20|20:29:41.412] Using LevelDB as the backing database
INFO [04-20|20:29:41.420] Opened ancient database        database=C:\Users\Achsah\Document
s\MScIT\sem4\blockchain_practical\ethermine\geth\chaindata\ancient\chain readonly=false
INFO [04-20|20:29:41.423] Disk storage enabled for ethash caches   dir=C:\Users\Achsah\Documents\MSc
IT\sem4\blockchain_practical\ethermine\geth\ethash count=3
INFO [04-20|20:29:41.424] Disk storage enabled for ethash DAGs    dir=C:\Users\Achsah\AppData\Local
\Ethash count=2
INFO [04-20|20:29:41.426] Initialising Ethereum protocol   network=5777 dbversion=<nil>
INFO [04-20|20:29:41.427]
INFO [04-20|20:29:41.430] -----
```

Step5-> Run the command

```
miner.setEtherbase('0xC050FE4d9bAc591d29538e2FD9cCA848B29489D0')inthegethconsole
```

Step6-> Run the command `miner.start()` to start mining

```
To exit, press ctrl-d or type exit
> INFO [04-20|20:29:45.021] Mapped network port           proto=tcp extport=30303 intport=3030
NP IGDv1-IP1"

>
> miner.setEtherbase('0xC050FE4d9bAc591d29538e2FD9cCA848B29489D0')
true
> miner.start()
INFO [04-20|20:34:45.673] Updated mining threads          threads=4
INFO [04-20|20:34:45.674] Transaction pool price threshold updated price=1,000,000,000
null
> INFO [04-20|20:34:45.683] Commit new sealing work       number=1 sealhash=2e6f57..6db9c6 uncl
=0 fees=0 elapsed=7.571ms
INFO [04-20|20:34:45.686] Commit new sealing work       number=1 sealhash=2e6f57..6db9c6 uncle
=0 fees=0 elapsed=9.940ms
INFO [04-20|20:34:47.975] Generating DAG in progress    epoch=0 percentage=0 elapsed=1.636s
INFO [04-20|20:34:49.873] Generating DAG in progress    epoch=0 percentage=1 elapsed=3.534s
```

Step7-> Below screenshots are the mining processes running on your local machine.

<pre>INFO [04-20 20:38:42.556] Generating DAG in progress 6.216s INFO [04-20 20:38:46.897] Generating DAG in progress .557s INFO [04-20 20:38:46.901] Generated ethash verification cache INFO [04-20 20:38:48.755] Successfully sealed new block hash=ccf3e9..10adff elapsed=4m3.071s INFO [04-20 20:38:48.765] "⛏️ mined potential block" INFO [04-20 20:38:48.756] Commit new sealing work uncles=0 txs=0 gas=0 fees=0 elapsed="504.9µs" INFO [04-20 20:38:48.770] Commit new sealing work uncles=0 txs=0 gas=0 fees=0 elapsed=14.488ms INFO [04-20 20:38:49.389] Successfully sealed new block hash=4c7137..a04b67 elapsed=632.526ms</pre>	<pre>epoch=0 percentage=98 elapsed=3m5 epoch=0 percentage=99 elapsed=4m0 epoch=0 elapsed=4m0.561s number=1 sealhash=2e6f57..6db9c6 number=1 hash=ccf3e9..10adff number=2 sealhash=cb4ba0..84eldd number=2 sealhash=cb4ba0..84eldd number=2 sealhash=cb4ba0..84eldd</pre>
--	--

Step8-> To stop the mining press **Ctrl+D**

```
INFO [04-20|20:39:21.980] Commit new sealing work          number=17 sealhash=923697..cb5b4d
uncles=0 txs=0 gas=0 fees=0 elapsed=117.201ms
INFO [04-20|20:39:21.984] Ethereum protocol stopped
INFO [04-20|20:39:22.046] Transaction pool stopped
INFO [04-20|20:39:22.047] Writing cached state to disk
=0c083a..cddeff
INFO [04-20|20:39:22.081] Persisted trie from memory database
gcnodes=0 gcsiz=0.00B gctime=0s livenodes=31 livesize=3.83KiB
INFO [04-20|20:39:22.087] Writing cached state to disk
=903c8d..6038c0
INFO [04-20|20:39:22.089] Persisted trie from memory database
gcnodes=0 gcsiz=0.00B gctime=0s livenodes=29 livesize=3.58KiB
INFO [04-20|20:39:22.098] Writing snapshot state to disk
INFO [04-20|20:39:22.130] Persisted trie from memory database
gcnodes=0 gcsiz=0.00B gctime=0s livenodes=29 livesize=3.58KiB
INFO [04-20|20:39:22.135] Writing clean trie cache to disk
cIT\sem4\blockchain_practical\ethermine\geth\triecache threads=4
INFO [04-20|20:39:22.323] Persisted the clean trie cache
cIT\sem4\blockchain_practical\ethermine\geth\triecache elapsed=143.729ms
INFO [04-20|20:39:22.490] Blockchain stopped
```

PRACTICAL-7

Aim: Create your own blockchain and demonstrate its use

Create a javascript folder with the following code in any folder of your choice.

JavaScript Code

```
const SHA256 = require("crypto-js/sha256");
class Block {
    constructor(index, timestamp, data, previousHash = "") {
        this.index = index;
        this.timestamp = timestamp;
        this.data = data;
        this.previousHash = previousHash;
        this.hash = this.calculateHash();
    }

    calculateHash() {
        return SHA256(
            this.index +
            this.previousHash +
            this.timestamp +
            JSON.stringify(this.data)
        ).toString();
    }
}

class Blockchain {
    constructor() {
        this.chain = [this.createGenesisBlock()];
    }

    createGenesisBlock() {
        return new Block(0, "21/04/2023", "Genesis Block", "0");
    }

    getLatestBlock() {
        return this.chain[this.chain.length - 1];
    }

    addBlock(newBlock) {
        newBlock.previousHash = this.getLatestBlock().hash;
    }
}
```

```
    newBlock.hash=newBlock.calculateHash();this.chain.push(newBlock);
}

isChainValid(){
    for(let i=1;i<this.chain.length;i++){ const currentBlock= this.chain[i];
        const previousBlock=this.chain[i-1];

        if(currentBlock.hash !== previousBlock.calculateHash()) {return false;
    }

    if(currentBlock.previousHash !== previousBlock.hash) { return
        false;
    }
}

return true;
}
}
```

BlockchainImplementation

```
let myCoin=newBlockchain();
myCoin.addBlock(newBlock(1,"22/04/2023",{amount:4}));myCoin.addBlock(newBlock(2,"22/04/2023",
{amount:8}));
console.log('Isblockchaininvalid?'+myCoin.isChainValid());console.log(JSON.stringify(myCoin,null, 4));
```

Output

Flowofexecution

Step1-> Makesureyouhaveinstallednodejsinyoursystem

```
C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\prac9>node -v
v14.17.5
```

Step2-> Weneedcrypto-jsnodemoduletomakeourownblockchain. Soinstallitas following

```
C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\prac9>npm install crypto-js
npm WARN @react-native-community/geolocation@2.0.2 requires a peer of react@* but none is installed. You may need to install it manually.
npm WARN @react-native-community/geolocation@2.0.2 requires a peer of react-native@* but none is installed.
npm WARN Achsah No description
npm WARN Achsah No repository field.
npm WARN Achsah No license field.

+ crypto-js@4.1.1
added 1 package from 1 contributor and audited 161 packages in 1.383s

5 packages are looking for funding
  run `npm fund` for details

found 8 vulnerabilities (2 moderate, 6 high)
  run `npm audit fix` to fix them, or `npm audit` for details
```

Step3-> Runtheabovecodeincommandlineusingcommand:nodomain.js

```
C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\prac9>node main.js
{
  "chain": [
    {
      "index": 0,
      "timestamp": "21/04/2023",
      "data": "Genesis Block",
      "previousHash": "0",
      "hash": "32dd10ad547e8e81623998bdffa2d8e9e3863fd252f5c3ea1cbea4ae26f54b1c"
    },
    {
      "index": 1,
      "timestamp": "22/04/2023",
      "data": {
        "amount": 4
      },
      "previousHash": "32dd10ad547e8e81623998bdffa2d8e9e3863fd252f5c3ea1cbea4ae26f54b1c",
      "hash": "eb78a02763c37fc2b1c4e331df64ca34733e47e017ef320d92ae89b148de5a3"
    },
    {
      "index": 2,
      "timestamp": "22/04/2023",
      "data": {
        "amount": 8
      },
      "previousHash": "eb78a02763c37fc2b1c4e331df64ca34733e47e017ef320d92ae89b148de5a3",
      "hash": "946b1f95d7761daee4f0c5d33a671c003ef5682333fd9a2d182a73104e9aea88"
    }
  ]
}
```

University of Mumbai

**Practical Journal of
Blockchain, Natural Language
Processing & Deep Learning**

M.Sc.(Information Technology) Part-II

Submitted by

Shaikh Obed S.A

Seat No: 1172743



**DEPARTMENT OF INFORMATION TECHNOLOGY
PILLAI HOC COLLEGE OF ARTS, SCIENCE & COMMERCE,
RASAYANI**
(Affiliated to Mumbai University)
**RASAYANI, 410207
MAHARASHTRA
2023-2024**

**Mahatma Education Society's
Pillai Hoc College of Arts, Science & Commerce,
Rasayani**

(Affiliated to Mumbai University)
RASAYANI-MAHARASHTRA-410207

**DEPARTMENT OF
INFORMATIONTECHNOLOGY**



CERTIFICATE

This is to certify that the experiment work entered in this journal is as per the syllabus in **M.Sc. (Information Technology) Part-II, Semester-IV**; class prescribed by University of Mumbai for the subject **Natural Language Processing** was done in computer lab of Mahatma Education Society's Pillai HOC College of Arts, Science& Commerce, Rasayani by **SHAIKH OBED S.A** during Academic year 2023-2024.

Exam Seat No: 1172743

Subject In-Charge

Coordinator

External Examiner

Principal

Date:

College Seal

Natural Language Processing

INDEX

Sr. No.	Practical Name	Page No.
1	A. Install NLTK B. Convert the given text to speech. C. Convert audio file Speech to Text.	1
2	A. Study of various Corpus–Brown, Inaugural, Reuters, udhr with various method like fields, raw, words, sents, categories. B. Study Condition a frequency distributions C. Studio f tagged corpora with methods like tagged sents, tagged_words. D. Map Words to Properties Using Python Dictionaries.	6
3	A. Study of Word net Dictionary with methods assents,definitions, examples, antonyms B. Study lemmas, hyponyms, hypernyms. C. Write a program using python to find synonym andantonym of word "active" using Word net.	11
4	A. Tokenization using Python's split() function B. Tokenization using Regular Expressions (Reg Ex) C. Tokenization using NLTK D. Tokenization using the spa clyibrary	13
5	Import NLPL libraries for Indian Language sand perform word tokenizationin Hindi	15
6	Illustrate part of speech tagging. A. Part of speech Tagging and chunking of user deined text. B. Named Entity recognition of user defined text. C. Named Entity recognition with diagram using NLTK corpus– Treebank	16
7	Finitestate automata A. Define grammar using nltk. Analyze a sentence using the same. B. Accept the input string with Regular expression of Finite Automaton: 101+.	18

	C.Accept the input string with Regular expression of FA: $(a+b)^*bba$.	
8	Study Porter Stemmer, Lancaster Stemmer, Regexp Stemmer, Snowball Stemmer Study Word Net Lemmatizer	21
9	Implement Naive Bayes classifier	22
10	A. Speech tagging using nktl B. Usage of Give and Gave in the Penn Treebank sample C. Probabilistic parser D. Malt parsing: Parse a sentence and draw a tree using malt parsing.	24
11	A. Multi word Expressions in NLP B. Normalized Web Distance and Word Similarity C. Word Sense Disambiguation	29

PRACTICAL: 1

1A]Install NLTK

Python3.9.2InstallationonWindows

InstallNLTK

Python3.9.2InstallationonWindows

Step1)Gotolink<https://www.python.org/downloads/>,andselectthe latestversionforwindows.



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

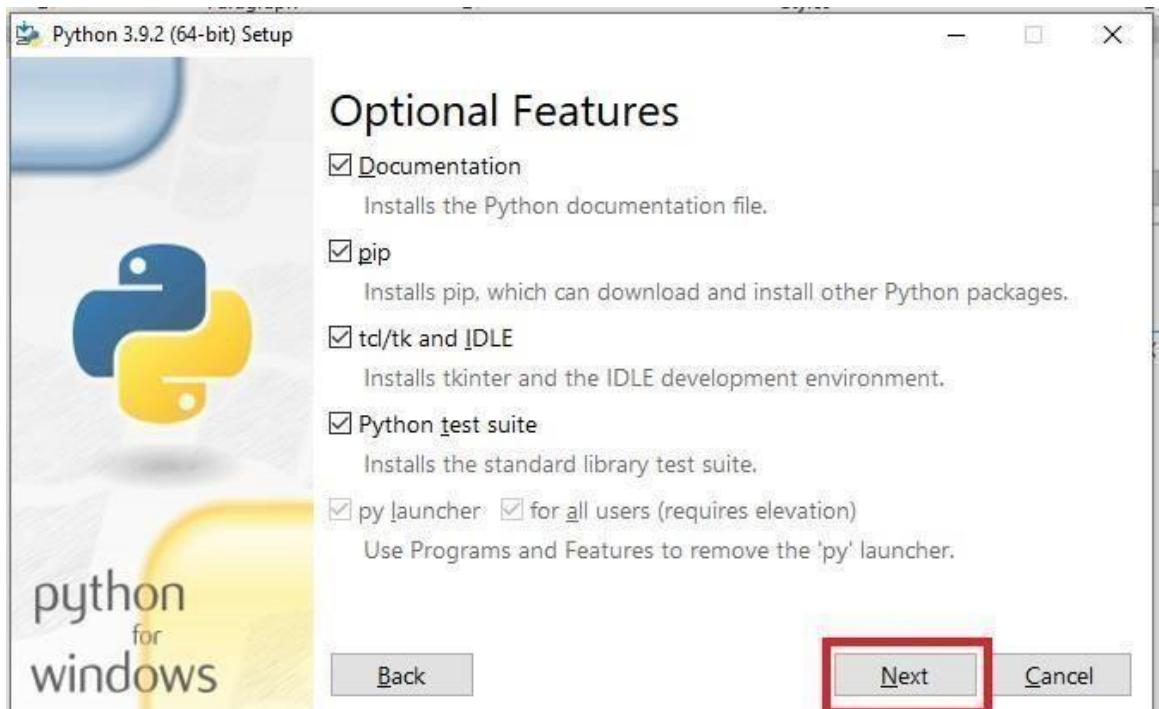
Files						
Version	Operating System	Description	MD5 Sum	File Size	GPG	
Gzipped source tarball	Source release		8cf053206beeca72c7ee531817dc24c7	25399571	SIG	
XZ compressed source tarball	Source release		f0dc9000112abeb16de4eccce9a870ab	18889164	SIG	
macOS 64-bit Intel installer	Mac OS X	for macOS 10.9 and later	a6fbfb297fa3be07a34ba89d13d54	29845662	SIG	
macOS 64-bit universal2	Mac OS X	for macOS 10.9 and later; including macOS 11 Big Sur on Apple Silicon (experimental)	fc8d028618c376d044916950c73e263	37618901	SIG	
Windows embedable package (32-bit)	Windows		cde7d9bfbd87b7777d7ff0ba4b0cd4506d	7578904	SIG	
Windows embedable package (64-bit)	Windows		bd4903eb930cf1747be01e6b8ddcd28a	8408823	SIG	
Windows help file	Windows		e2308d543374e671ff0e0344d3f36062	8844275	SIG	
Windows installer (32-bit)	Windows	Recommended	81294c31bd7e2d4470658721b2887ed5	27202848	SIG	
Windows installer (64-bit)	Windows		efb20aa1b648a2bddd59c142d9eb06	28287512	SIG	

Step2)ClickontheWindowsinstaller(64bit)

Step3>SelectCustomizeInstallation

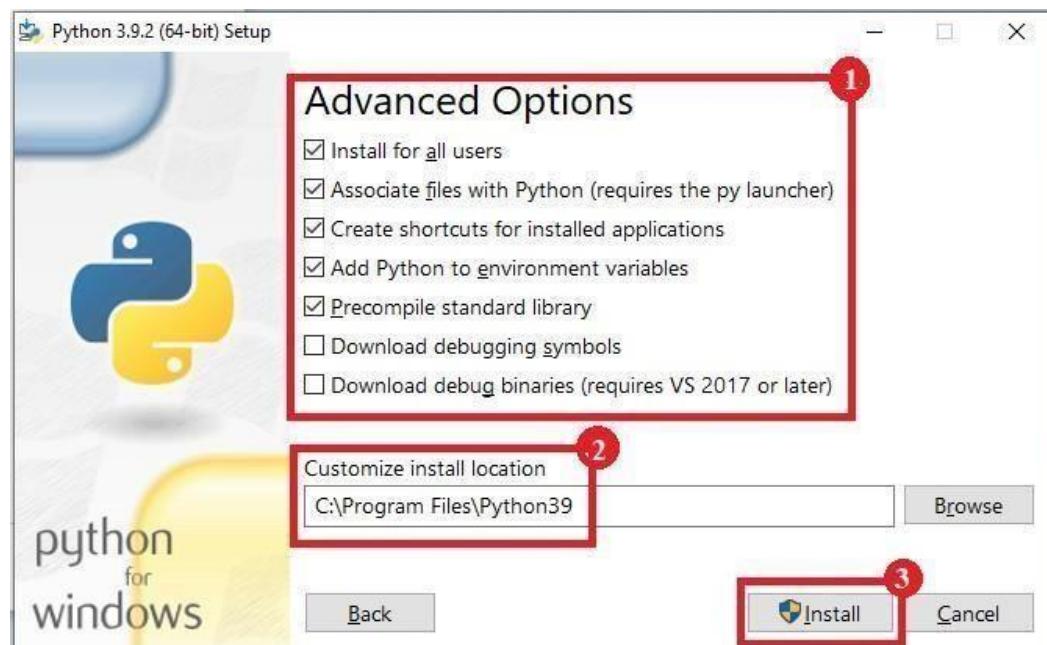


Step4) ClickNEXT



Step5) In next screen

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



Step6) Click Close button once install is done.

Step7) open command prompt window and run the following commands:
 C:\Users\BeenaKapadia>pip install --upgrade pip
 C:\Users\Beena Kapadia> pip install --user -U nltk
 C:\Users\BeenaKapadia>>pip install --user-U numpy
 C:\Users\BeenaKapadia>python
 >>>import nltk
 >>>

```
Command Prompt - python
C:\Users\Beena Kapadia>pip install --user -U nltk
Collecting nltk
  Using cached nltk-3.6.2-py3-none-any.whl (1.5 MB)
Requirement already satisfied: joblib in c:\users\beena kapadia\appdata\roaming\python\python39\site-packages (from nltk)
(1.0.1)
Requirement already satisfied: tqdm in c:\users\beena kapadia\appdata\roaming\python\python39\site-packages (from nltk)
(4.60.0)
Requirement already satisfied: regex in c:\users\beena kapadia\appdata\roaming\python\python39\site-packages (from nltk)
(2021.4.4)
Requirement already satisfied: click in c:\users\beena kapadia\appdata\roaming\python\python39\site-packages (from nltk)
(7.1.2)
Installing collected packages: nltk
  WARNING: The script nltk.exe is installed in 'C:\Users\Beena Kapadia\AppData\Roaming\Python\Python39\Scripts' which is
not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed nltk-3.6.2

C:\Users\Beena Kapadia>pip install --user -U numpy
Collecting numpy
  Using cached numpy-1.20.3-cp39-cp39-win_amd64.whl (13.7 MB)
Installing collected packages: numpy
  WARNING: The script f2py.exe is installed in 'C:\Users\Beena Kapadia\AppData\Roaming\Python\Python39\Scripts' which is
not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-1.20.3

C:\Users\Beena Kapadia>python
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.
>>> import nltk
```

1B] Convert the given text to speech.

```

✓ [1] # text to speech

✓ [2] # pip install gtts

✓ [3] # pip install playsound

✓ [4] pip install playsound
      Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
      Collecting playsound
        Downloading playsound-1.3.0.tar.gz (7.7 kB)
          Preparing metadata (setup.py) ... done
        Building wheels for collected packages: playsound
          Building wheel for playsound (setup.py) ... done
            Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=7019 sha256=c65d0c3feafffb97c8b575b7eadcf78f9dc89a0f023ed6a9227b67ce790e37d1
            Stored in directory: /root/.cache/pip/wheels/90/89/ed/2d643f4226fc8c7c9156fc28abd8051e2d2c0de37ae51ac45c
          Successfully built playsound
        Installing collected packages: playsound
          Successfully installed playsound-1.3.0

✓ [5] from playsound import playsound
      WARNING:playsound:playsound is relying on another python subprocess. Please use `pip install pygobject` if you want playsound to run more efficiently.

✓ [6] # import required for text to speech conversion

✓ [7] pip install gtts
      Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
      Collecting gtts
        Downloading gTTS-2.3.2-py3-none-any.whl (28 kB)
          Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python3.10/dist-packages (from gtts) (2.27.1)
          Requirement already satisfied: click<8.2,>=7.1 in /usr/local/lib/python3.10/dist-packages (from gtts) (8.1.3)
          Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->gtts) (1.26.15)
          Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->gtts) (2022.12.7)
          Requirement already satisfied: charset-normalizer>~2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->gtts) (2.0.12)
          Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->gtts) (3.4)
        Installing collected packages: gtts
          Successfully installed gtts-2.3.2

✓ [8] from gtts import gTTS
      from playsound import playsound

✓ [9] import pygame
      pygame 2.3.0 (SDL 2.24.2, Python 3.10.12)
      Hello from the pygame community. https://www pygame org/contribute.html

Activate Window:
Go to Settings to activate

✓ [10] mytext = "Welcome to Natural Language programming"
      language = "en"
      myobj = gTTS(text=mytext, lang=language, slow=False)
      myobj.save("myfile.mp3")

✓ [11] import pygame

✓ [12] pip install pydub
      Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
      Collecting pydub
        Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
      Installing collected packages: pydub
      Successfully installed pydub-0.25.1

1s  ➔ from pydub import AudioSegment
      from pydub.playback import play

      audio_file = AudioSegment.from_file("/content/myfile.mp3", format="mp3")
      play(audio_file)

```

1C]Convert audiofile Speech to Text.

```
✓ [1] !pip install SpeechRecognition pydub
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting SpeechRecognition
  Downloading SpeechRecognition-3.10.0-py2.py3-none-any.whl (32.8 MB)
    32.8/32.8 MB 42.7 MB/s eta 0:00:00
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Requirement already satisfied: requests>=2.26.0 in /usr/local/lib/python3.10/dist-packages (from SpeechRecognition) (2.27.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->SpeechRecognition) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->SpeechRecognition) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->SpeechRecognition) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->SpeechRecognition) (3.4)
Installing collected packages: pydub, SpeechRecognition
Successfully installed SpeechRecognition-3.10.0 pydub-0.25.1
```

The screenshot shows a Jupyter Notebook cell in Google Colab. The code in the cell is:

```
import speech_recognition as sr
from google.colab import files

# Upload the audio file
uploaded = files.upload()

# Get the filename of the uploaded audio file
filename = next(iter(uploaded))

# 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)
```

Below the code cell, there is a file upload interface with the following message:

Choose File about_time.wav
• about_time.wav(audio/wav) - 29026 bytes, last modified: 6/20/2023 - 100% done
Saving about_time.wav to about_time.wav
well it's about time you got here

PRACTICAL:2

2A] Study of various Corpus—Brown, Inaugural, Reuters, udhr with various method like fileids, raw, words, sents, categories.

```

✓ 3s ① import nltk
   from nltk.corpus import brown
   nltk.download('brown')
   print ('File ids of brown corpus\n',brown.fileids())

↳ [nltk_data] Downloading package brown to /root/nltk_data...
[nltk_data]  Unzipping corpora/brown.zip.
File ids of brown corpus
['ca01', 'ca02', 'ca03', 'ca04', 'ca05', 'ca06', 'ca07', 'ca08', 'ca09', 'ca10', 'ca11', 'ca12', 'ca13', 'ca14', 'ca15', 'ca16', 'ca17', 'ca18', 'ca19', 'ca20',
 ↳

✓ 0s [2] ca01 = brown.words('ca01')

✓ 0s [3] print ('\nca01 has following words:\n',ca01)

ca01 has following words:
['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]

✓ 0s [4] print ('\nca01 has',len(ca01),'words')

ca01 has 2242 words


✓ 0s ② print ('\n\nCategories or file in brown corpus:\n')
print (brown.categories())

↳
Categories or file in brown corpus:
['adventure', 'belles_lettres', 'editorial', 'fiction', 'government', 'hobbies', 'humor', 'learned', 'lore', 'mystery', 'news', 'religion', 'reviews', 'romance', ...]

✓ 0s [6] 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))

 ↳

Statistics for each text:
AvgWordLen      AvgSentenceLen  no.ofTimesEachWordAppearsOnAvg      FileName

✓ 0s [7] num_words = len(brown.words(fileid))

✓ 0s [8] num_sents = len(brown.sents(fileid))

 ↳

✓ 0s [9] num_vocab = len(set([w.lower() for w in brown.words(fileid)]))

✓ 0s [9] print (int(num_chars/num_words),'\t\t\t', int(num_words/num_sents),'\t\t\t', int(num_vocab/num_words),'\t\t\t', fileid)
 ↳

8          23          2          cr09

```

2C]StudyConditionalfrequency distributions

```

✓ 4s ① pip install nltk
    □ Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
      Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
      Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.3)
      Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.2.0)
      Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2022.10.31)
      Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.65.0)

✓ 5s [2] #process a sequence of pairs
    text = ['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
    pairs = [('news', 'The'), ('news', 'Fulton'), ('news', 'County'), ...]

✓ 6s [3] import nltk

✓ 7s [4] from nltk.corpus import brown

✓ 8s [5] nltk.download('brown')
[nltk_data] Downloading package brown to /root/nltk_data...
[nltk_data]  Unzipping corpora/brown.zip.
True

```

```

✓ 3s ① fd = nltk.ConditionalFreqDist(
    (genre, word)
    for genre in brown.categories()
    for word in brown.words(categories=genre))

✓ 4s [7] genre_word = [(genre, word)
    for genre in ['news', 'romance']
    for word in brown.words(categories=genre)]

✓ 5s [8] print(len(genre_word))
print(genre_word[:4])
print(genre_word[-4:])
cfд = nltk.ConditionalFreqDist(genre_word)

170576
[('news', 'The'), ('news', 'Fulton'), ('news', 'County'), ('news', 'Grand')]
[('romance', 'afraid'), ('romance', 'not'), ('romance', '''), ('romance', '.')]

✓ 6s [10] print(cfд)
print(cfд.conditions())
print(cfд['news'])
print(cfд['romance'])
print(list(cfд['romance']))

```

```
+ Code + Text
✓ 0s ① print(cfd)
   print(cfd.conditions())
   print(cfd['news'])
   print(cfd['romance'])
   print(list(cfd['romance']))

  □ <ConditionalFreqDist with 2 conditions>
    ['news', 'romance']
    <FreqDist with 14394 samples and 100554 outcomes>
    <FreqDist with 8452 samples and 70022 outcomes>
    [',', '.', 'the', 'and', 'to', 'a', 'of', ',', "'", 'was', 'I', 'in', 'he', 'had', '?', 'her', 'that', 'it', 'his', 'she', 'with', 'you', 'for', 'at', 'He', 'or'
    ↵

✓ [11] from nltk.corpus import inaugural

✓ [12] nltk.download('inaugural')

[nltk_data] Downloading package inaugural to /root/nltk_data...
[nltk_data]  Unzipping corpora/inaugural.zip.
True

✓ [13] cfd = nltk.ConditionalFreqDist(
    (target, fileid[:4])
    for fileid in inaugural.fileids()
    for w in inaugural.words(fileid)
    for target in ['romance', 'citizen'])

✓ [14] for w in inaugural.words(fileid)
    for target in ['america', 'citizen']
    if w.lower().startswith(target))

✓ [14] nltk.download('udhr')

  □ [nltk_data] Downloading package udhr to /root/nltk_data...
  [nltk_data]  Unzipping corpora/udhr.zip.
True

✓ [15] from nltk.corpus import udhr

✓ [16] languages = ['Chickasaw', 'English', 'German_Deutsch',
   'Greenlandic_Inuktitut', 'Hungarian_Magyar', 'Ibibio_Efik']
   cfd = nltk.ConditionalFreqDist(
     (lang, len(word))
     for lang in languages
     for word in udhr.words(lang + '-Latin1'))

  □ 0s cfd.tabulate(conditions=['English', 'German_Deutsch'],
    samples=range(10), cumulative=True)

      0   1   2   3   4   5   6   7   8   9
English  0  185  525  883  997 1166 1283 1440 1558 1638
German_Deutsch  0  171  263  614  717  894 1013 1110 1213 1275
```

2D] Study of tagged corpora with methods like tagged_sents, tagged_words.

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
+ Code + Text ✓ RAM Disk ▾ ▴
```

```
✓ 3s [1] import nltk
  from nltk import tokenize
  nltk.download('punkt')
  nltk.download('words')

  ↗ [nltk_data] Downloading package punkt to /root/nltk_data...
  [nltk_data]  Unzipping tokenizers/punkt.zip.
  [nltk_data] Downloading package words to /root/nltk_data...
  [nltk_data]  Unzipping corpora/words.zip.
  True

✓ 0s [2] para = "Hello! My name is Beena Kapadia. Today you'll be learning NLTK."
  sents = tokenize.sent_tokenize(para)
  print("\nsentence tokenization\n-----\n",sents)

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

✓ 0s [3] print("\nword tokenization\n-----\n")

  word tokenization
  -----
```

```
✓ 0s [1] print("\nword tokenization\n-----\n")

  ↗ word tokenization
  -----
```

```
✓ 0s [2] for index in range(len(sents)):
  word = tokenize.word_tokenize(sents[index])
  print(word)

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

2F]MapWordsto PropertiesUsingPython Dictionaries.

- ▼ creating and printing a dictionay by mapping word with its properties

```
✓ 4s  pip install nltk
```

```
↳ Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.3)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2022.10.31)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.65.0)
```

```
✓ 0s  thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
print(thisdict)
print(thisdict["brand"])
print(len(thisdict))
print(type(thisdict))
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
Ford
3
<class 'dict'>
```

PRACTICAL:3

3A] Study of Wordnet Dictionary with methods `synsets`, `definitions`, `examples`, `antonyms`

```

✓ 2s [1] import nltk
    nltk.download('wordnet')

    [nltk_data] Downloading package wordnet to /root/nltk_data...
    True

✓ 2s [2] import nltk
    from nltk.corpus import wordnet
    print(wordnet.synsets("computer"))

    [Synset('computer.n.01'), Synset('calculator.n.01')]

✓ 0s [3] print(wordnet.synset("computer.n.01").definition())

    a machine for performing calculations automatically

✓ 0s [4] print("Examples:", wordnet.synset("computer.n.01").examples())

    Examples: []

✓ 0s ⏪ print(wordnet.lemma('buy.v.01.buy').antonyms())

    [Lemma('sell.v.01.sell')]

```

3B] Study lemmas, hyponyms, hypernyms.

```

✓ 1s ⏪ import nltk
    nltk.download('wordnet')

    [nltk_data] Downloading package wordnet to /root/nltk_data...
    True

✓ 2s [2] from nltk.corpus import wordnet
    print(wordnet.synsets("computer"))
    print(wordnet.synset("computer.n.01").lemma_names())

    [Synset('computer.n.01'), Synset('calculator.n.01')]
    ['computer', 'computing_machine', 'computing_device', 'data_processor', 'electronic_computer', 'information_processing_system']

✓ 0s [3] #all lemmas for each synset.
    for e in wordnet.synsets("computer"):
        print(f'{e} --> {e.lemma_names()}')

    Synset('computer.n.01') --> ['computer', 'computing_machine', 'computing_device', 'data_processor', 'electronic_computer', 'information_processing_system']
    Synset('calculator.n.01') --> ['calculator', 'reckoner', 'figurer', 'estimator', 'computer']

✓ 0s [4] #print all lemmas for a given synset
    print(wordnet.synset("computer.n.01").lemmas())

    [Lemma('computer.n.01.computer'), Lemma('computer.n.01.computing_machine'), Lemma('computer.n.01.computing_device'), Lemma('computer.n.01.data_processor'), Lemma('computer.n.01.electronic_computer'), Lemma('computer.n.01.information_processing_system'), Lemma('computer.n.01.reckoner'), Lemma('computer.n.01.reckoning'), Lemma('computer.n.01.estimator'), Lemma('computer.n.01.calculator'), Lemma('computer.n.01.calculating')]

```

```

✓ [5] #get the synset corresponding to lemma
print(wordnet.lemma('computer.n.01.computing_device').synset())
Synset('computer.n.01')

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

✓ [7] #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())

<bound method _WordNetObject.hyponyms of Synset('computer.n.01')>

✓ [8] print([lemma.name() for synset in syn.hyponyms() for lemma in synset.lemmas()])
['analog_computer', 'analogue_computer', 'digital_computer', 'home_computer', 'node', 'client', 'guest', 'number_cruncher', 'pari-mutuel_machine', 'totalizer', 't
e

✓ [9] #the semantic similarity in WordNet
vehicle = wordnet.synset('vehicle.n.01')
car = wordnet.synset('car.n.01')
print(car.lowest_common_hypernyms(vehicle))

[Synset('vehicle.n.01')]

```

3C]Write a program using python to find synonym and antonym of word "active" using Wordnet.

```

+ Code + Text
✓ [1] import nltk
nltk.download('wordnet')

[nltk_data] Downloading package wordnet to /root/nltk_data...
True

✓ [2] from nltk.corpus import wordnet
print(wordnet.synsets("active"))

[Synset('active_agent.n.01'), Synset('active_voice.n.01'), Synset('active.n.03'), Synset('active.a.01'), Synset('active.s.02'), Synset('active.a.03'), Synset('act
e

✓ [3] print(wordnet.lemma('active.a.01.active').antonyms())
[Lemma('inactive.a.02.inactive')]

```

PRACTICAL:4

4A]TokenizationusingPython'ssplit() function

```

✓ 0s  [1]  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)

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

```

4B]TokenizationusingRegularExpressions(RegEx)

- Tokenization using Regular Expressions (RegEx)

```

✓ 0s  [1]  pip install nltk
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.3)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2022.10.31)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.65.0)

✓ 0s  [2]  import nltk

✓ 0s  [3]  from nltk.tokenize import RegexpTokenizer

✓ 0s  [4]  tk = RegexpTokenizer('\s+', gaps = True)

✓ 0s  [5]  str = "I love to study Natural Language Processing in Python"

✓ 0s  [6]  tokens = tk.tokenize(str)

✓ 0s  [7]  print(tokens)
['I', 'love', 'to', 'study', 'Natural', 'Language', 'Processing', 'in', 'Python']

```

Active
Go to S

4C]Tokenization usingNLTK

```
+ Code + Text

[1] import nltk
      from nltk.tokenize import word_tokenize

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

[3] import nltk
      nltk.download('punkt')

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]  Unzipping tokenizers/punkt.zip.
True

[4] print(word_tokenize(str))

['I', 'love', 'to', 'study', 'Natural', 'Language', 'Processing', 'in', 'Python']
```

4D]Tokenizationusingthe spaCylibrary

```
+ Code + Text

▼ Tokenization using the spaCy library

[1] import spacy
      nlp = spacy.blank("en")

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

[3] doc = nlp(str)

[4] words = [word.text for word in doc]
print(words)

['I', 'love', 'to', 'study', 'Natural', 'Language', 'Processing', 'in', 'Python']
```

PRACTICAL:5

ImportNLP Libraries for Indian Languages and perform 5A]

word tokenization in Hindi

```

✓ [1] !pip install torch
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.0.1+cu118)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch) (3.12.0)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from torch) (4.5.0)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch) (1.11.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch) (3.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch) (3.1.2)
Requirement already satisfied: triton==2.0.0 in /usr/local/lib/python3.10/dist-packages (from torch) (2.0.0)
Requirement already satisfied: cmake in /usr/local/lib/python3.10/dist-packages (from triton==2.0.0->torch) (3.25.2)
Requirement already satisfied: lit in /usr/local/lib/python3.10/dist-packages (from triton==2.0.0->torch) (16.0.5)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch) (2.1.2)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch) (1.3.0)

✓ 20s ① !pip install inltk
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy>=2.0.18->inltk) (2.0.7)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy>=2.0.18->inltk) (3.0.8)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-packages (from spacy>=2.0.18->inltk) (8.1.9)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy>=2.0.18->inltk) (1.1.1)
Requirement already satisfied: srslry<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from spacy>=2.0.18->inltk) (2.4.6)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (from spacy>=2.0.18->inltk) (2.0.8)
Requirement already satisfied: typer<0.8.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy>=2.0.18->inltk) (0.7.0)
Requirement already satisfied: pathy>=0.10.0 in /usr/local/lib/python3.10/dist-packages (from spacy>=2.0.18->inltk) (0.10.1). Go to Settings to activate Windows.
Requirement already satisfied: enumtree<1.0.0,>=0.9.7 in /usr/local/lib/python3.10/dist-packages (from spacy>=2.0.18->inltk) (0.9.7)

```

PRACTICAL:6

Illustrate part of speech tagging.

6A] Part of speech Tagging and chunking of user defined text.

- ▼ Part of speech Tagging and chunking of user defined text.

```

✓ [1] 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')

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]  Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]      /root/nltk_data...
[nltk_data]  Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]      /root/nltk_data...
[nltk_data]  Unzipping chunkers/maxent_ne_chunker.zip.
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]  Unzipping corpora/words.zip.
True

```

```

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

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

```

```

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

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

```

```

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

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'), Tree('ORGANIZATION', [(_('NLTK', 'NNP')]), ('.', '.'))], Tree('S', [(_

```

```

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

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

```

6B]NamedEntityrecognitionofuserdefinedtext.

- Named Entity recognition of user defined text.

```
[1] !pip install -U spacy
!python -m spacy download en_core_web_sm

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: spacy in /usr/local/lib/python3.10/dist-packages (3.5.2)
Collecting spacy
  Downloading spacy-3.5.3-cp310-cp310-manylinux2014_x86_64.whl (6.6 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.6/6.6 MB 78.0 MB/s eta 0:00:00
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.4)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.9)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.0.7)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.8)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-packages (from spacy) (8.1.9)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.1.1)
Requirement already satisfied: srslry<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.4.6)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.0.8)
Requirement already satisfied: typer<0.8.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (0.7.0)
Requirement already satisfied: pathy<0.10.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (0.10.1)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from spacy) (6.3.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (4.65.0)
Requirement already satisfied: numpy<1.15.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.22.4)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.27.1)
Requirement already satisfied: pydanticlt;1.8,>=1.8.1,<1.11.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.10.7)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.1.2)

[2] import spacy

[3] nlp = spacy.load("en_core_web_sm")

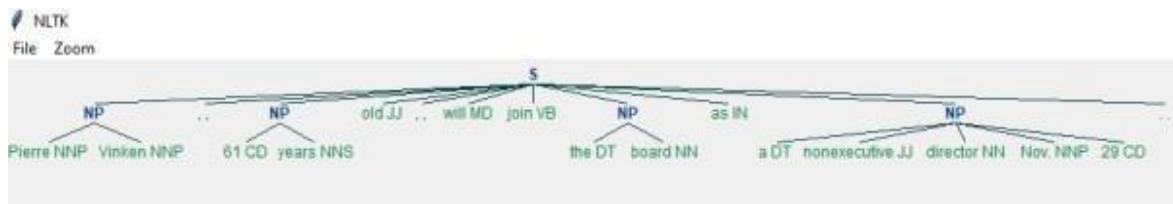
[4] text = ("When Sebastian Thrun started working on self-driving cars at "
           "Google in 2007, few people outside of the company took him "
           "seriously. "I can tell you very senior CEOs of major American "
           "car companies would shake my hand and turn away because I wasn't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

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

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

6C]Named Entity recognition with diagram using NLTK corpus – Treebank

```
import nltk
from nltk.corpus import treebank
treebank_chunk = treebank.chunk.tagged_sents()[0]
treebank_chunk.chunked_sents()[0]
treebank_chunk.chunked_sents()[0].draw()
```

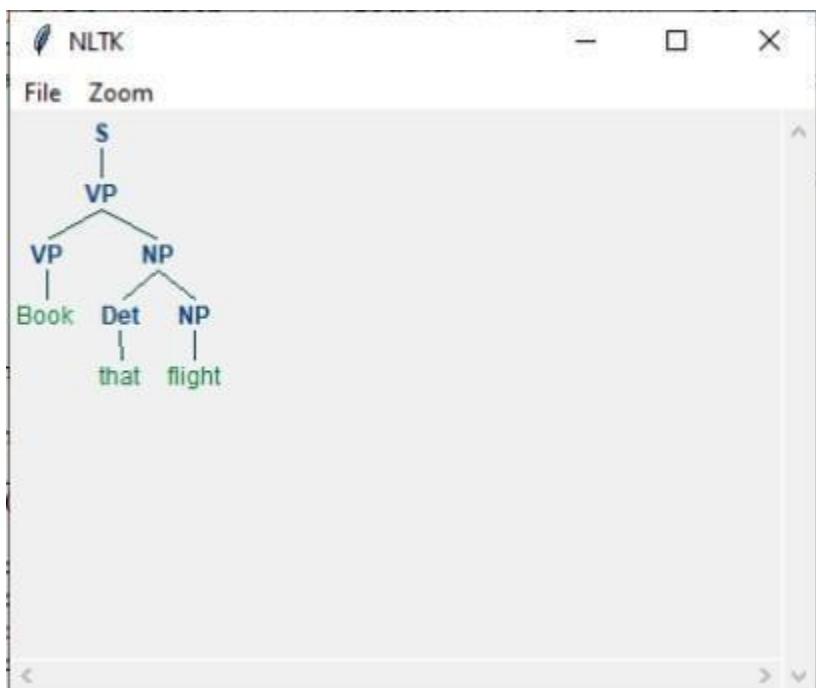


PRACTICAL:7

Finitestate automata

7A] Define grammar using nltk. Analyze a sentence using the same.

```
import nltk
from nltk import tokenize
grammar1=nltk.CFG.fromstring(""" S
-> VP
VP->VPNP
NP->DetNP
Det->'that'
NP->singularNoun 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()
```



7B]AccepttheinputstringwithRegularexpressionof Finite Automaton:101+.

```

defFA(s):
    #if the length is less than 3, it can't be accepted, so end the process if len(s) <
    3:
    return "Rejected"

    #first three characters are fixed, checking them using index
    if s[0] == '1':
        if s[1] == '0':
            if s[2] == '1':
                #After index 2, only "1" can appear, so 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 nested ifs are 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))

```

```

Rejected
Rejected
Accepted
Accepted
Rejected
Rejected
Rejected
Rejected
Accepted

```

7C]AccepttheinputstringwithRegularexpressionof FA:(a+b)*bba.

```

defFA(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 its length; it should be at least 3
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 3 ifs are true
    return "Rejected" # else of 3rd if return
    "Rejected" # else of 2nd if return
    "Rejected" # else of 1st if return
    "Rejected"
```

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

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

PRACTICAL:8

Study PorterStemmer, LancasterStemmer, RegexpStemmer, SnowballStemmer Study WordNetLemmatizer

- Study PorterStemmer, LancasterStemmer, RegexpStemmer, SnowballStemmer

Study WordNetLemmatizer

```

✓ 2s [1] import nltk
   from nltk.stem import PorterStemmer
   word_stemmer = PorterStemmer()
   print(word_stemmer.stem('writing'))

   write

✓ 0s [2] import nltk
   from nltk.stem import LancasterStemmer
   Lanc_stemmer = LancasterStemmer()
   print(Lanc_stemmer.stem('writing'))

   writ

✓ 0s [3] import nltk
   from nltk.stem import RegexpStemmer
   Reg_stemmer = RegexpStemmer('ing\$|s\$|e\$|able$', min=4)
   print(Reg_stemmer.stem('writing'))

   writ

```

```

✓ 0s [4] import nltk
   from nltk.stem import SnowballStemmer
   english_stemmer = SnowballStemmer('english')
   print(english_stemmer.stem ('writing'))

   write

```

```

✓ 0s [5] from nltk.stem import WordNetLemmatizer
   lemmatizer = WordNetLemmatizer()
   print("word :\tlemma")

   word : lemma

✓ 1s [6] import nltk
   nltk.download('wordnet')
   print("rocks :", lemmatizer.lemmatize("rocks"))
   print("corpora :", lemmatizer.lemmatize("corpora"))

[ nltk_data ] Downloading package wordnet to /root/nltk_data...
rocks : rock
corpora : corpus

```

```

✓ 0s ➜ import nltk
   nltk.download('wordnet')
   print("better :", lemmatizer.lemmatize("better", pos ="a"))

   better : good
[ nltk_data ] Downloading package wordnet to /root/nltk_data...
[ nltk_data ] Package wordnet is already up-to-date!

```

PRACTICAL:9**ImplementNaiveBayes classifier**

```

✓ [1] pip install pandas
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2022.7.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.22.4)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)

✓ [2] pip install sklearn
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting sklearn
  Downloading sklearn-0.0.post5.tar.gz (3.7 kB)
    Preparing metadata (setup.py) ... done
  Building wheels for collected packages: sklearn
    Building wheel for sklearn (setup.py) ... done
      Created wheel for sklearn: filename=sklearn-0.0.post5-py3-none-any.whl size=2950 sha256=6efd3412de321c02aa3c4985b8eb5124bea058dcba4acca9d439aced5e068682
      Stored in directory: /root/.cache/pip/wheels/38/1f/8d/4f812c590e074c1e928f5cec67bf5053b71f38e2648739403a
Successfully built sklearn
Installing collected packages: sklearn
Successfully installed sklearn-0.0.post5

✓ [3] import pandas as pd
      import numpy as np

```

```

✓ [6] sms_data = pd.read_csv("/content/drive/MyDrive/spam.csv", encoding='latin-1')

✓ [7] from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

✓ [8] import re
      import nltk
      from nltk.corpus import stopwords
      from nltk.stem.porter import PorterStemmer

✓ [9] nltk.download('stopwords')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]  Unzipping corpora/stopwords.zip.
True

✓ [10] 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()

```

Activate
Go to Setti

```

✓ [10] 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)

✓ [16] from sklearn.feature_extraction.text import CountVectorizer
countvectorizer =CountVectorizer()

✓ [17] x = countvectorizer.fit_transform(corpus).toarray()
print(x)

[[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]]

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

['ham' 'ham' 'spam' ... 'ham' 'ham' 'ham']

✓ [19] 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)

✓ [20] #Multinomial Naive Bayes.
from sklearn.naive_bayes import MultinomialNB
multinomialnb = MultinomialNB()
multinomialnb.fit(x_train,y_train)

+ MultinomialNB
MultinomialNB()

✓ [21] # Predicting on test data:
y_pred = multinomialnb.predict(x_test)
print(y_pred)

['ham' 'ham' 'ham' ... 'ham' 'ham' 'ham']

✓ [22] #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))

      precision    recall  f1-score   support

     ham       0.99      0.99      0.99     1448
     spam       0.92      0.93      0.92      224

  accuracy                           0.98     1672
   macro avg       0.95      0.96      0.96     1672
weighted avg       0.98      0.98      0.98     1672

accuracy_score:  0.979066985645933

```

PRACTICAL:10**10AII]Speech tagging using nltk**

▼ Speech tagging using nltk

```

✓ [7] import nltk
  from nltk.corpus import state_union
  from nltk.tokenize import PunktSentenceTokenizer

✓ [8] import nltk
  nltk.download('state_union')
  train_text = state_union.raw("2005-GWBush.txt")
  sample_text = state_union.raw("2006-GWBush.txt")

[nltk_data] Downloading package state_union to /root/nltk_data...
[nltk_data]   Package state_union is already up-to-date!

✓ [9] custom_sent_tokenizer = PunktSentenceTokenizer(train_text)

✓ [10] tokenized = custom_sent_tokenizer.tokenize(sample_text)

```

```

✓ [11] 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))

✓ import nltk
nltk.download('punkt')
import nltk
nltk.download('averaged_perceptron_tagger')
process_content()

[('PRESIDENT', 'NNP'), ('GEORGE', 'NNP'), ('W.', 'NNP'), ('BUSH', 'NNP'), ('S', 'POS'), ('ADDRESS', 'NNP'), ('BEFORE', 'IN'), ('A', 'NNP'), ('JOINT', 'NNP'), ('S
[('Mr.', 'NNP'), ('Speaker', 'NNP'), ('.', ','), ('Vice', 'NNP'), ('President', 'NNP'), ('Cheney', 'NNP'), (',', ','), ('members', 'NNS'), ('of', 'IN'), ('Congres
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]   /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]   date!

```

Activate Windows
Go to Settings to activate Windows.

10BI]Usageof Giveand Gavein the PennTreebanksample

```

✓ [1] import nltk
  import nltk.parse.viterbi
  import nltk.parse.pchart

✓ [2] 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())

✓ [3] def sent(t):
  return '.join(token for token in t.leaves() if token[0] not in '*-0')

✓ [4] 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)

2s  ● import nltk
nltk.download('treebank')
for tree in nltk.corpus.treebank.parsed_sents():
  for t in tree.subtrees(give):
    print_node(t, 72)

[nltk_data] Downloading package treebank to /root/nltk_data...
[nltk_data]  Unzipping corpora/treebank.zip.
gave NP: the chefs / NP: a standing ovation

```

10BII]probabilisticparser

```

✓ [1] import nltk
  from nltk import PCFG

✓ [2] 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]
  ''')

✓ [3] print(grammar)

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]

✓ [4] viterbi_parser = nltk.ViterbiParser(grammar)

```

```

✓ [5] token = "old men and women".split()
0s

✓ [6] obj = viterbi_parser.parse(token)
0s

✓ [7]  print("Output: ")
for x in obj:
    print(x)

Output:
(NP (JJ old) (NNS (NNS men) (CC and) (NNS women))) (p=0.000864)

```

10C]Malt parsing:**Parse a sentence and draw a tree using malt parsing.**

```

[1] !apt-get install openjdk-8-jdk-headless -qq
17s

Selecting previously unselected package libxtst6:amd64.
(Reading database ... 123069 files and directories currently installed.)
Preparing to unpack .../libxtst6_2%3a1.2.3-1_amd64.deb ...
Unpacking libxtst6:amd64 (2:1.2.3-1) ...
Selecting previously unselected package openjdk-8-jre-headless:amd64.
Preparing to unpack .../openjdk-8-jre-headless_8u372-ga~us1~0ubuntu1~20.04_amd64.deb ...
Unpacking openjdk-8-jre-headless:amd64 (8u372-ga~us1~0ubuntu1~20.04) ...
Selecting previously unselected package openjdk-8-jdk-headless:amd64.
Preparing to unpack .../openjdk-8-jdk-headless_8u372-ga~us1~0ubuntu1~20.04_amd64.deb ...
Unpacking openjdk-8-jdk-headless:amd64 (8u372-ga~us1~0ubuntu1~20.04) ...
Setting up libxtst6:amd64 (2:1.2.3-1) ...

[2] !wget https://www.maltparser.org/mco/english_parser/engmalt.linear-1.7.mco
!wget https://www.maltparser.org/mco/mco/english_parser/engmalt.linear-1.7.mco -P malt_parser

--2023-06-20 07:50:24-- https://www.maltparser.org/mco/english_parser/engmalt.linear-1.7.mco
Resolving www.maltparser.org (www.maltparser.org)... 109.235.174.4, 2a00:1968:0:1::16
Connecting to www.maltparser.org (www.maltparser.org)|109.235.174.4|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23065542 (22M)
Saving to: 'engmalt.linear-1.7.mco'
```

```
✓ 5s !pip install spacy

✖ Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: spacy in /usr/local/lib/python3.10/dist-packages (3.5.2)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.4)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.9)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.0.7)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.8)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-packages (from spacy) (8.1.9)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.1.1)

21s [4] !python -m spacy download en

2023-06-20 07:50:38.542916: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance mode. To enable the following instructions: AVX2, FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-06-20 07:50:40.687123: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
⚠ As of spacy v3.0, shortcuts like "en" are deprecated. Please use the full pipeline package name "en_core_web_sm" instead.
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting en-core-web-sm==3.5.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en\_core\_web\_sm-3.5.0/en\_core\_web\_sm-3.5.0-py3-none-any.whl (12.8 MB)
    12.8/12.8 MB 72.1 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.6.0,>=3.5.0 in /usr/local/lib/python3.10/dist-packages (from en-core-web-sm==3.5.0) (3.5.2)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-packages (from spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (1.0.4)
```

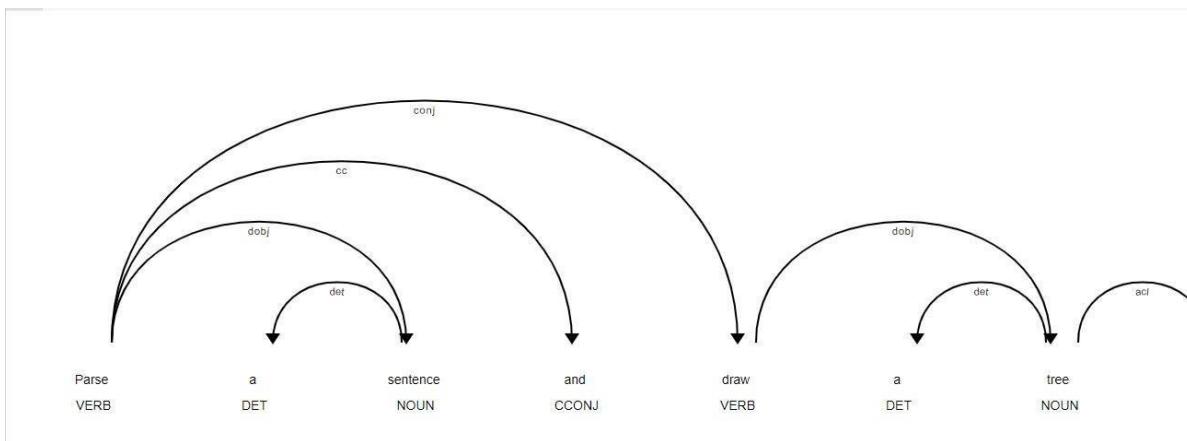
```
import spacy
from spacy import displacy

# Load the English language model
nlp = spacy.load("en_core_web_sm")

# Example sentence
sentence = "Parse a sentence and draw a tree using malt parsing."

# Process the sentence
doc = nlp(sentence)

# Visualize the dependency parse tree
displacy.render(doc, style="dep", jupyter=True)
```



PRACTICAL:11

11A]MultiwordExpressionsinNLP

▼ Multiword Expressions in NLP

```

✓ 3s  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='_')
    import nltk
    nltk.download('punkt')
    for sent in sent_tokenize(s):
        print(mwe.tokenize(word_tokenize(sent)))

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]  Unzipping tokenizers/punkt.zip.
['Good', 'cake', 'cost', 'Rs.1500\\kg', 'in', 'Mumbai', '.']
['Please', 'buy', 'me', 'one', 'of', 'them', '.']
['Thanks', '.']

```

11B]NormalizedWeb DistanceandWordSimilarity

```

import numpy as np
import re
import textdistance
from sklearn import
from sklearn.cluster import AgglomerativeClustering

texts =[

'Reliancesupermarket','Reliancehypermarket','Reliance','Reliance','Reliance',
'downtown', 'Reliance market',
'Mumbai','MumbaiHyper','Mumbaidxb','mumbai airport',
'k.mtrading','KMTrading','KMtrade', 'K.M.Trading', 'KM.Trading'
]

defnormalize(text):
    """Keeponlylower-casedtextandnumbers"""
    returnre.sub('[^a-z0-9]+', '', text.lower())

defgroup_texts(texts, threshold=0.4):
    """Replaceeachtextwiththerrepresentativeofitscluster"""
    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,
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))

```

```

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

```

11C]WordSenseDisambiguation

- ▼ Word Sense Disambiguation

```

✓  [1] from nltk.corpus import wordnet as wn

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

✓  [3] 
import nltk
nltk.download('wordnet')

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))

[nltk_data] Downloading package wordnet to /root/nltk_data...
<bound method Synset.name of Synset('bank.n.01')>: <bound method Synset.definition of Synset('bank.n.01')>
<bound method Synset.name of Synset('set.n.01')>: <bound method Synset.definition of Synset('set.n.01')>
<bound method Synset.name of Synset('put.v.01')>: <bound method Synset.definition of Synset('put.v.01')>

```

University of Mumbai

**Practical Journal of
Blockchain, Natural Language
Processing & Deep Learning**

M.Sc.(Information Technology) Part-II

Submitted by

Shaikh Obed S.A

Seat No: 1172743



**DEPARTMENT OF INFORMATION TECHNOLOGY
PILLAI HOC COLLEGE OF ARTS, SCIENCE & COMMERCE,
RASAYANI
(Affiliated to Mumbai University)
RASAYANI, 410207 MAHARASHTRA
2023-2024**

**Mahatma Education Society's
Pillai Hoc College of Arts, Science & Commerce, Rasayani**
(Affiliated to Mumbai University)
RASAYANI-MAHARASHTRA-410207

**DEPARTMENT OF
INFORMATION TECHNOLOGY**



CERTIFICATE

This is to certify that the experiment work entered in this journal is as per the syllabus in **M.Sc. (Information Technology) Part-II, Semester-IV**; class prescribed by University of Mumbai for the subject **Deep Learning** was done in computer lab of Mahatma Education Society's Pillai HOC College of Arts, Science & Commerce, Rasayani by **SHAIKH OBED S.A** during Academic year 2023-2024.

Exam Seat No: 1172743

Subject In-Charge

Coordinator

External Examiner

Principal

Date:

College Seal

Deep Learning

INDEX

Sr. No.	PracticalName	Page No.
1	Write a Program to demonstrate following operations. A. Create Vector, Matrix and Tensor B. Multiplication of two:Vector, Matrix and Tensor C. Addition of two:Vector,Matrix and Tensor D. Multiply Matrix with Vector E. Matrix Dot product and Matrix Inverse	1
2	Performing matrix multiplication and finding Eigen vectors andEigen values using Tensor Flow	5
3	Implementing deep neural network for performing binary classification task.	6
4	Using deep feed forward network with two hidden layers for performing classification and predicting the probability of class.	7
5	Evaluating feed for ward deep network for multiclass Classification using K-Fold cross-validation.	9
6	Implementation of convolution a neural network to predict numbers from number images.	10
7	Performing encoding and decoding of images using deep Auto encoder.	12
8	Denoising of images using Autoencoder.	14

PracticalNo:1

Aim: Write a Program to demonstrate following operations.

- A. Create Vector, Matrix and Tensor
- B. Multiplication of two: Vector, Matrix and Tensor
- C. Addition of two: Vector, Matrix and Tensor
- D. Multiply Matrix with Vector
- E. Matrix Dot product and Matrix Inverse

A) Create Vector, Matrix and Tensor

Code:

```
import numpy as np
import tensorflow as tf

x=np.array([1,2,3,4])
print("Create a Vector:",x)
print("\n")

A=np.array([[1,2],[3,4],[5,6]])
print("Create a Matrix:\n",A)
print("\n")

tensor_A=tf.constant([[1,2]],dtype=tf.int32)
print("Create a Tensor: ",tensor_A)
print("\n")
```

Output:

```
Create a Vector: [1 2 3 4]

Create a Matrix:
 [[1 2]
 [3 4]
 [5 6]]

Create a Tensor: tf.Tensor([[1 2]], shape=(1, 2), dtype=int32)
```

B) Multiplicationoftwo:Vector,MatrixandTensor

Code:

```
A=np.array([[1,2],[3,4],[5,6]])
print("A=",A)
print("\n")
```

```
B=np.array([[2,5],[7,4],[4,3]])
print("B=",B)
print("\n")
```

```
C=A*B
print("MultiplicationoftwoMatrix:\n",C)
print("\n")
```

```
x=np.array([1,2,3,4])
y=np.array([5,6,7,8])
```

```
z=x*y
print("MultiplicationoftwoVector:",z)
print("\n")
```

```
tensor_A=tf.constant([[4,2]],dtype=tf.int32)
print("A: ",tensor_A)
```

```
tensor_B=tf.constant([[7,4]],dtype=tf.int32)
print("B: ",tensor_B,"\\n")
```

```
tensor_multiply=tf.multiply(tensor_A,tensor_B)
print("MultiplicationoftwoTensor:",tensor_multiply)
print("\n")
```

Output:

```
A= [[1 2]
 [3 4]
 [5 6]]

B= [[2 5]
 [7 4]
 [4 3]]

Multiplication of two Matrix:
 [[ 2 10]
 [21 16]
 [20 18]]

Multiplication of two Vector:  [ 5 12 21 32]

A:  tf.Tensor([[4 2]], shape=(1, 2), dtype=int32)
B:  tf.Tensor([[7 4]], shape=(1, 2), dtype=int32)

Multiplication of two Tensor:  tf.Tensor([[28  8]], shape=(1, 2), dtype=int32)
```

C) Additionoftwo:Vector,MatrixandTensor

Code:

```
x=np.array([1,2,3,4])
y=np.array([5,6,7,8])

z=x+y
print("AdditionoftwoMatrix:",z)
print("\n")

A=np.array([[1,2],[3,4],[5,6]])
B=np.array([[2,5],[7,4],[4,3]])

C=A*B
print("AdditionoftwoVector:\n",C)
print("\n")

tensor_add=tf.add(tensor_A,tensor_B)
print("AdditionoftwoTensor:",tensor_add)
print("\n")
```

Output:

```
Addition of two Matrix: [ 6  8 10 12]

Addition of two Vector:
 [[ 2 10]
 [21 16]
 [20 18]]

Addition of two Tensor: tf.Tensor([[11  6]], shape=(1, 2), dtype=int32)
```

D) MultiplyMatrixwithVector**Code:**

```
x=np.array([1,2,3,7,3,5,2])
y=np.array([[1],[3],[5],[7],[8],[8],[2]])
c=x*y
print("MultiplicationofVectorandMatrix:\n",c)
print("\n")
```

Output:

```
Multiplication of Vector and Matrix:
[[ 1  2  3  7  3  5  2]
 [ 3  6  9  21  9  15  6]
 [ 5  10  15  35  15  25  10]
 [ 7  14  21  49  21  35  14]
 [ 8  16  24  56  24  40  16]
 [ 8  16  24  56  24  40  16]
 [ 2  4  6  14  6  10  4]]
```

E) MatrixDotproductandMatrixInverse**Code:**

```
U=[2,-3]
V=[1,3]
dotproduct=np.dot(U,V)
print("Matrixdotproduct:",dotproduct)
print("\n")
```

```
A=np.array([[6,1,1],
           [4,-2,5],
           [2,8,7]])
print("InverseofMatrix:\n",np.linalg.inv(A)) print("\n")
```

Output:

```
Matrix dot product: -7

Inverse of Matrix:
[[ 0.17647059 -0.00326797 -0.02287582]
 [ 0.05882353 -0.13071895  0.08496732]
 [-0.11764706  0.1503268   0.05228758]]
```

PracticalNo:2

Aim: Performing matrix multiplication and finding Eigenvectors and Eigen values using TensorFlow

Code:

```
import tensorflow as tf

x=tf.constant([1,2,3,4,5,6],shape=[2,3])
print(x)
y=tf.constant([7,8,9,10,11,12],shape=[3,2])
print(y)

z=tf.matmul(x,y)
print("\n")
print("Multiplying the matrices:")
print(z)
print("\n")

e_matrix_A=tf.random.uniform([2,2],minval=3,maxval=10,dtype=tf.float32,name="matrixA")
")
print("\n")
print("MatrixA:\n{ }\n\n".format(e_matrix_A))

eigen_values_A,eigen_vectors_A=tf.linalg.eigh(e_matrix_A)
print("Eigan Vectors: \n{ }\nEigan Values:
\n{ }\n\n".format(eigen_vectors_A,eigen_values_A))
```

Output:

```
tf.Tensor(
[[1 2 3]
 [4 5 6]], shape=(2, 3), dtype=int32)
tf.Tensor(
[[ 7  8]
 [ 9 10]
 [11 12]], shape=(3, 2), dtype=int32)

Multiplying the matrices:
tf.Tensor(
[[ 58  64]
 [139 154]], shape=(2, 2), dtype=int32)

Matrix A:
[[6.1428514 3.1488109]
 [5.0727997 3.0301542]]

Eigan Vectors:
[[-0.5944289  0.8041483]
 [ 0.8041483  0.5944289]]

Eigan Values:
[-0.7196745  9.892682 ]
```

PracticalNo:3

Aim:Implementingdeepneuralnetworkforperformingbinary classification task.

Code:

```

pip install scikeras
from pandas import read_csv
from keras.models import Sequential
from keras.layers import Dense
from scikeras.wrappers import KerasClassifier
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import StratifiedKFold
dataset = read_csv('sonar.csv', header=None)
dataset = dataset.values

X = dataset[:, 0:60].astype(float)
Y = dataset[:, 60]

encoder = LabelEncoder()
encoder.fit(Y)
encoded_y = encoder.transform(Y)

def create_baseline():
    model = Sequential()
    model.add(Dense(60, input_dim=60, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

estimator = KerasClassifier(model=create_baseline, epochs=10, batch_size=5, verbose=0)
KFold = StratifiedKFold(n_splits=10, shuffle=True)
results = cross_val_score(estimator, X, encoded_y, cv=KFold)

print("\n")
print("Baseline: %.2f%% (%.2f%%)" % (results.mean() * 100, results.std() * 100))

```

Output:

```

/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:81: UserWarning: Activity regularizer is not supported for this layer type. It will be ignored.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:81: UserWarning: Activity regularizer is not supported for this layer type. It will be ignored.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)

Baseline: 76.36% (8.19%)

```

PracticalNo:4

Aim:Using deep feedforward network with two hidden layers for performing classification and predicting the probability of class.

Code:

```

import numpy as np
from sklearn.datasets import load_wine
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder
from keras.layers import Dense, Input, concatenate, Dropout
from keras.models import Model
from tensorflow.keras import optimizers

optimizers.RMSprop()
mizers.Adam()

dataset = load_wine()
ensemble_num = 10
bootstrap_size = 0.8
training_size = 0.8

num_hidden_neurons = 10
dropout = 0.25

epochs = 100
batch = 10
temp = []
scaler = MinMaxScaler()
one_hot = OneHotEncoder()
dataset['data'] = scaler.fit_transform(dataset['data'])
dataset['target'] = one_hot.fit_transform(np.reshape(dataset['target'], (-1, 1))).toarray()
for i in range(len(dataset.data)):
    temp.append([dataset['data'][i], np.array(dataset['target'][i])])

temp = np.array(temp, dtype=object)
np.random.shuffle(temp)
# holdout training and test stop index
stop = int(training_size * len(dataset.data))

train_X = np.array([x for x in temp[:stop, 0]])
train_Y = np.array([x for x in temp[:stop, 1]])
test_X = np.array([x for x in temp[stop:, 0]])
test_Y = np.array([x for x in temp[stop:, 1]])

num_hidden_neurons = 64

sub_net_outputs = []
sub_net_inputs = []
for i in range(ensemble_num):
    # two hidden layers to keep it simple

```

```
#specifyinputshapetotheshapeofthetrainingset
net_input = Input(shape=(train_X.shape[1],))
sub_net_inputs.append(net_input)
Y=Dense(num_hidden_neurons)(net_input) Y
= Dense(num_hidden_neurons)(Y)
Y=Dropout(dropout)(Y)

#sub_netscontainstheoutputtensors
sub_net_outputs.append(Y)

Y=concatenate(sub_net_outputs)
Y=Dense(train_Y[0].shape[0],activation='softmax')(Y) model
= Model(inputs=sub_net_inputs, outputs=Y)
model.compile(optimizer='rmsprop',loss='categorical_crossentropy')
print('\n')
print("7_AdityaHadap")
print("Begintraining...")

model.fit([train_X]*ensemble_num,train_Y,validation_data=[[test_X]*ensemble_num,
test_Y], epochs=epochs, batch_size=batch)

print("Training complete...")
np.set_printoptions(precision=2, suppress=True)
for i in range(len(test_X)):
    print("Prediction:" + str(model.predict([test_X[i].reshape(1,test_X[i].shape[0])]*ensemble_num)) + " | True: " + str(test_Y[i]))
```

Output:

```
1/1 [=====] - 0s 59ms/step
Prediction: [[0. 0. 1.]] | True: [0. 0. 1.]
1/1 [=====] - 0s 53ms/step
Prediction: [[0. 1. 0.]] | True: [0. 1. 0.]
1/1 [=====] - 0s 82ms/step
Prediction: [[0.01 0.99 0.]] | True: [0. 1. 0.]
```

PracticalNo:5

Aim:Evaluating feedforward deep network for multiclass Classification using K-Fold cross-validation.

Code:

```
!pip install scikeras
!pip install np_utils

import pandas
# import np_utils
from keras.models import Sequential
from keras.layers import Dense
from scikeras.wrappers import KerasClassifier
from tensorflow.keras.utils import to_categorical #
from scikeras import np_utils
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline

dataframe = pandas.read_csv("iris.data", header=None)
dataset = dataframe.values
X = dataset[:, 0:4].astype(float)
Y = dataset[:, 4]

encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
dummy_y = to_categorical(encoded_Y)

def baseline_model():
    model = Sequential()
    model.add(Dense(8, input_dim=4, activation='relu'))
    model.add(Dense(3, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

estimator = KerasClassifier(build_fn=baseline_model, epochs=200, batch_size=5, verbose=0)
kfold = KFold(n_splits=10, shuffle=True)
results = cross_val_score(estimator, X, dummy_y, cv=kfold)
print("Accuracy: %.2f%% (%.2f%%)" % (results.mean() * 100, results.std() * 100))
```

Output:

```
super().__init__(activity_regularizer=activity)
/usr/local/lib/python3.10/dist-packages/scikeras
x, y = self._initialize(x, y)
/usr/local/lib/python3.10/dist-packages/keras/sr
super().__init__(activity_regularizer=activity)
Accuracy: 96.67%(3.33%)
```

PracticalNo:6

Aim:Implementationofconvolutionalneuralnetworktopredictnumbers from number images.

Code:

```
from keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Flatten
import matplotlib.pyplot as plt
import numpy as np
(X_train, Y_train), (X_test, Y_test) = mnist.load_data()

plt.imshow(X_train[0])
plt.show

print(X_train[0].shape)

X_train = X_train.reshape(60000, 28, 28, 1)
X_test = X_test.reshape(10000, 28, 28, 1)

Y_train = to_categorical(Y_train)
Y_test = to_categorical(Y_test)
Y_train[0]
print(Y_train[0])

model = Sequential()
model.add(Conv2D(64, kernel_size=3, activation='relu', input_shape=(28, 28, 1)))
model.add(Conv2D(32, kernel_size=3, activation='relu'))
model.add(Flatten())
model.add(Dense(10, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(X_train, Y_train, validation_data=(X_test, Y_test), epochs=1)

print(model.predict(X_test[:4]))
print(Y_test[:4])
```

Output:

```
1875/1875 [=====] - 182s 96ms/step -  
1/1 [=====] - 0s 128ms/step  
[[8.37227105e-07 2.23798462e-11 9.53875133e-06 6.72433962e-05  
9.96774052e-11 7.07288048e-11 1.33387364e-11 9.99921083e-01  
7.09786889e-07 6.57526357e-07]  
[9.79483275e-07 3.37575261e-06 9.99963760e-01 1.36108639e-07  
6.54944587e-09 2.82700041e-09 3.15081634e-05 2.01311337e-07  
1.43711674e-07 1.54105492e-10]  
[6.54704877e-07 9.99759853e-01 2.89732798e-05 6.00278383e-07  
2.82923993e-05 4.02716287e-05 3.63800063e-06 3.02873850e-05  
1.00186335e-04 7.15461238e-06]  
[9.97525275e-01 4.24347775e-07 1.15249306e-04 9.99267104e-06  
4.57059969e-06 2.09394238e-06 1.43698708e-03 1.23642039e-05  
4.25632024e-04 4.67542704e-04]]  
[[0. 0. 0. 0. 0. 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.]]]
```

PracticalNo:7

Aim:Performing encoding and decoding of images using deep autoencoder.

Code:

```

import keras
from keras import layers
from keras.datasets import mnist
import numpy as np
import matplotlib.pyplot as plt

_dim=32

input_img = keras.Input(shape=(784,))
encoded = layers.Dense(encoding_dim, activation='relu')(input_img)
decoded = layers.Dense(784, activation='sigmoid')(encoded)
autoencoder = keras.Model(input_img, decoded)

encoder = keras.Model(input_img, encoded)
encoded_input = keras.Input(shape=(encoding_dim,))
decoder_layer = autoencoder.layers[-1]
decoder = keras.Model(encoded_input, decoder_layer(encoded_input))

autoencoder.compile(optimizer='adam', loss='binary_crossentropy')

(x_train, _), (x_test, _) = mnist.load_data()
x_train = x_train.astype('float32') / 255.
x_test = x_test.astype('float32') / 25
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)

autoencoder.fit(x_train, x_train, epochs=13, batch_size=256, shuffle=True, validation_data=(x_test, x_test))
encoded_imgs = encoder.predict(x_test)
decoded_imgs = decoder.predict(encoded_imgs)

n=10
plt.figure(figsize=(20,4))
for i in range(n):
    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)

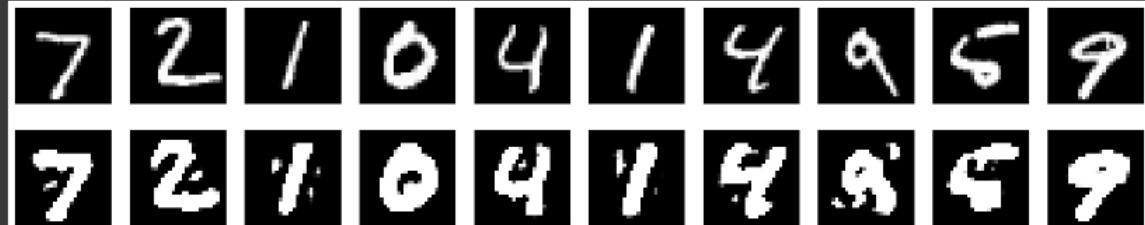
    ax=plt.subplot(2,n,i+1+n)
    plt.imshow(decoded_imgs[i].reshape(28,28))
    plt.gray()

```

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

Output:

```
Epoch 12/13
235/235 [=====] - 3s 13ms/step - loss: 0.0956 - val_loss: -7
Epoch 13/13
235/235 [=====] - 3s 13ms/step - loss: 0.0951 - val_loss: -7
313/313 [=====] - 1s 1ms/step
313/313 [=====] - 1s 2ms/step
```



PracticalNo:8

Aim:DenoisingofimagesusingAutoencoder.

Code:

```

import keras
from keras.datasets import mnist
from keras import layers
import numpy as np
from keras.callbacks import TensorBoard
import matplotlib.pyplot as plt

(X_train, _), (X_test, _) = mnist.load_data()

X_train = X_train.astype('float32') / 255.
X_test = X_test.astype('float32') / 255.
X_train = np.reshape(X_train, (len(X_train), 28, 28, 1))
X_test = np.reshape(X_test, (len(X_test), 28, 28, 1))
noise_factor = 0.5

X_train_noisy = X_train + noise_factor * \
    np.random.normal(loc=0.0, scale=1.0, size=X_train.shape)
X_test_noisy = X_test + noise_factor * \
    np.random.normal(loc=0.0, scale=1.0, size=X_test.shape)
X_train_noisy = np.clip(X_train_noisy, 0., 1.)
X_test_noisy = np.clip(X_test_noisy, 0., 1.)

n = 10
plt.figure(figsize=(20, 2))
for i in range(1, n+1):
    ax = plt.subplot(1, n, i)
    plt.imshow(X_test_noisy[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()

input_img = keras.Input(shape=(28, 28, 1))

x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(input_img)
x = layers.MaxPooling2D((2, 2), padding='same')(x)
x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(x)
encoded = layers.MaxPooling2D((2, 2), padding='same')(x)

x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(encoded)
x = layers.UpSampling2D((2, 2))(x)
x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(x)
x = layers.UpSampling2D((2, 2))(x)
decoded = layers.Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

```

```

autoencoder=keras.Model(input_img,decoded)
autoencoder.compile(optimizer='adam',loss='binary_crossentropy')
autoencoder.fit(X_train_noisy,X_train,
                epochs=3,
                batch_size=328,
                shuffle=True,
                validation_data=(X_test_noisy,X_test),
                callbacks=[TensorBoard(log_dir='/tmp/tb',histogram_freq=0,write_graph=False)])
predictions=autoencoder.predict(X_test_noisy)

```

```

m=10
plt.figure(figsize=(20,2))
for i in range(1,m+1):
    ax=plt.subplot(1,m,i)
    plt.imshow(predictions[i].reshape(28,28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()

```

Output:

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>
11490434/11490434 [=====] - 0s 0us/step



Epoch 1/3
183/183 [=====] - 138s 691ms/step - loss: 0.2161 - val_loss:
Epoch 2/3
183/183 [=====] - 119s 651ms/step - loss: 0.1254 - val_loss:
Epoch 3/3
183/183 [=====] - 118s 645ms/step - loss: 0.1177 - val_loss:
313/313 [=====] - 7s 21ms/step

