INDEX

PRACTICAL:1

G. A simple client class that generates the private and public keys by usingthe built-in Python RS Algorithm and test it.

```
importbinascii

importCrypto
fromCrypto.PublicKeyimportRSA
fromCrypto.SignatureimportPKCS1_v1_5


classClient:
    def____init____(self):
        random=Crypto.Random.new().readself._private_key=
        RSA.generate(1024, random) self._public_key =
        self._private_key.publickey()self._signer=
        PKCS1_v1_5.new(self._private_key)

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


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

Output:

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

Public Key:  30819f300d06092a864886f70d010101050003818d00308189028181 00adcc265
040fdf19988db8eabc5e73fbc2d4527f95af6f3b9305377b0182d61fc44441af11dc1c8537c06d
452718289d83e92245c1af7373bf3d45e95c78383d0a82edb026f63d4fa805366017b991bc9ac8
6391f59935bf6559f8a23d89aa915a9e2f4c3e0113f9d9b9b5e071e2c4f780fff35fb0c9506c7c
b596a0128fe5f230203010001
```

A) Atransactionclassto sendand receivemoneyandtestit.

```
importbinasciiimport
collectionsimportdatetim
e
fromclientimportClientfrom
Crypto.Hash import SHA
fromCrypto.SignatureimportPKCS1_v1_5


classTransaction:
    def____init____(self,sender,recipient,value):self.sender=
        sender
        self.recipient=recipientself.value=value
        self.time = datetime.datetime.now()

    defto_dict(self):
        identity="Genesis"ifself.sender="Genesis"elseself.sender.identity
        returncollections.OrderedDict(
            {
                "sender": identity, "recipient":
                self.recipient,"value":   self.value,
                "time": self.time,
            }
        )

    defsign_transaction(self):
        private_key=self.sender._private_keysigner=
        PKCS1_v1_5.new(private_key)
        h=SHA.new(str(self.to_dict()).encode("utf8"))
        returnbinascii.hexlify(signer.sign(h)).decode("ascii")


Dinesh = Client()
Ramesh=Client()


t = Transaction(Dinesh, Ramesh.identity, 5.0) print("\nTransaction
Recipient:\n",t.recipient)#print("\nTransactionSender:\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/Loca
l/Programs/Python/Python39/python.exe c:/Users/Achsah/Documents/MScIT/sem4/blockchain_practi
cal/prac1/prac1b.py

Transaction Recipient:  30819f300d06092a864886f70d010101050003818d0030818902818100c308b9261d
2397e09dffcf67981240735cb2e3e0f4f510d29e21a70335503f142005e5f09e9db9091b263e73b6a32cd909fdc7
7a616bd4a5e09d044bf63c7906a98b791021ee41dbfb83d5022fb2423185262689e31287543b0863385d7325e30b
cf8bc722907bfa0b4a39495f6a2ac2d6bf5e50e77d2b52d6efcafd3a062a9f0203010001

Transaction Value:  5.0

Signature:  b3a8342acd21883671ff67dde74172f31f094935a2775765ec6e20f5ba910627eb9450b14d721933
ea2ecca46d7a14e38d8b1e3e2382b9132c09ea94077b31c4f4a7cdf33b0f3ec4e0378fb6f53e8ba450b79572737b
440f8584bc79c3fe3360ac75d23655d81e2c8f1dbe1435a2735100a3738d05522aeaadeee7f5bba6fff2
```

B) Createmultipletransactionsanddisplaythem.

fromclientimportClient
fromtransaction_classimportTransaction

Dinesh = Client()
Ramesh=Client()

t = Transaction(Dinesh, Ramesh.identity, 5.0) print("\nTransaction Recipient:\n",t.recipient)#print("\nTransactionSender:\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)

fortransactionintransactions:Transaction.display_transaction(transaction)print("–
————————————————————")

Output:

```
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100c123f94a104b17803a5fb728b6
a4e3abb26f2554e5652b5be5df08cf3f56efef5a36196fe4eebbb8fe7f299d1fbe153031bce451e3c45ef2680237
5c49f3474b9d23312534badccf3a8ecf4c238dc593a8a488eeaf155b347fda86b5548de80a96b3e1543eb20d4867
03574d6c28a67cc04797c247e457fc233a6074f5e1c0cb0203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100cc47acc592a9c8ec78b211e
bda5ef91f40518e9c23338e0c99824892012b533656c8872d512994269e79d58a54e9fd8548141f204b26a3d89e6
36468c81171b2147a2ca0c5745d66822b19d826f235afa2cab4a9f4b1623895019db6fdbcd752ff6a3dbc709d76c
dd64df5e12ae674a5c896c09b632ab0b6b19c731c4d9004b30203010001
-----
value: 6.0
-----
time: 2023-04-22 22:13:48.783100
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100c551eccbd6e7624223f4a51741
4b122ae738153aa00dd11951cf58e7f3cd436e639cc89fd84d34a93892450966378401babe918f186401a514162e
de7fcab891df9023dc6604d1bfea1df2e83e9a3a985cdfcb00a9e2e55ba4364b48a1200c5ed6d163e4e7e8e39d3d
e67272f63b04e559872fec9719fc7870b308581761fec10203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100ae7406d1f27b484dc241f33
a48b66df19d6e5f3b732fefda2622ee726bb49dcfea390ff1f5a11c651f7a96fd888f9e901630645da2bfe9d8987
69a859481a10eff8f977a40e59701f43e278992741af99bb77aed08bb6fa5297ed2116441300469e73ec347e0bb8
e790c960948b7872e6a60060581caf4b78d1624b0a45848610203010001
-----
value: 2.0
-----
time: 2023-04-22 22:13:48.784604
--------------
sender: 30819f300d06092a864886f70d010101050003818d0030818902818100cc47acc592a9c8ec78b211ebda
5ef91f40518e9c23338e0c99824892012b533656c8872d512994269e79d58a54e9fd8548141f204b26a3d89e6364
68c81171b2147a2ca0c5745d66822b19d826f235afa2cab4a9f4b1623895019db6fdbcd752ff6a3dbc709d76cdd6
4df5e12ae674a5c896c09b632ab0b6b19c731c4d9004b30203010001
-----
recipient: 30819f300d06092a864886f70d010101050003818d0030818902818100c551eccbd6e7624223f4a51
7414b122ae738153aa00dd11951cf58e7f3cd436e639cc89fd84d34a93892450966378401babe918f186401a5141
62ede7fcab891df9023dc6604d1bfea1df2e83e9a3a985cdfcb00a9e2e55ba4364b48a1200c5ed6d163e4e7e8e39
d3de67272f63b04e559872fec9719fc7870b308581761fec10203010001
-----
value: 4.0
-----
time: 2023-04-22 22:13:48.787805
```

C) Createablockchain,agenesisblockandexecuteit.

```
fromclientimportClient
fromtransaction_classimportTransaction


classBlock:
    def____init____(self, client):
        self.verified_transactions = []
        self.previous_block_hash=""self.Nonce= ""
        self.client=client


defdump_blockchain(blocks):
    print(f"\nNumberofblocksinthechain:{len(blocks)}")

    fori,blockinenumerate(blocks):print(f"block#
        {i}")
        fortransactionin 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
3bd1b733f2eb7027fff0920abac3204b093247fcee235a5a90203010001
-----
value: 500.0
-----
time: 2023-04-22 22:40:58.531260
-----
--------------
====================================
```

D) Createaminingfunctionandtestit.

```
importhashlib


defsha256(message):
    returnhashlib.sha256(message.encode("ascii")).hexdigest()


def mine(message, difficulty=1):assert
    difficulty÷1prefix="1"*difficulty for i
    in range(1000):
        digest=sha256(str(hash(message))+str(i))if
        digest.startswith(prefix):
            print(f"after{str(i)}iterationsfoundnonce:{digest}")#returnprint(digest) mine("test

message", 2)
```

Output:

```
C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\prac1>C:/Users/Achsah/AppData/Loca
l/Programs/Python/Python39/python.exe c:/Users/Achsah/Documents/MScIT/sem4/blockchain_practi
cal/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] Addblockstothemineranddumptheblockchain.

```
importdatetime
import hashlib

#Createaclasswithtwofunctions


classBlock:
        def____init____(self,data,previous_hash):
                self.timestamp=datetime.datetime.now(datetime.timezone.utc)self.data=data
                self.previous_hash = previous_hashself.hash =
                self.calc_hash()

        defcalc_hash(self):
                sha=hashlib.sha256()
                hash_str=self.data.encode("utf-8")sha.update(hash_str)
                returnsha.hexdigest()


#Instantiatetheclass

blockchain=[Block("Firstblock","0")]

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

# Dumping the blockchain

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

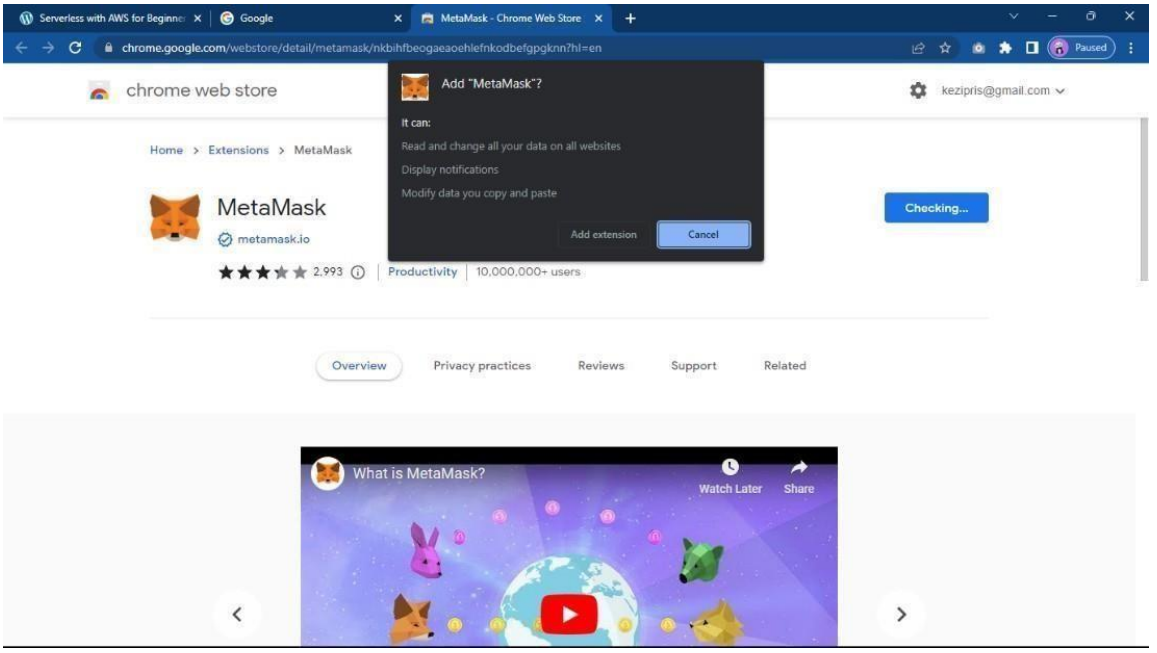
Output:

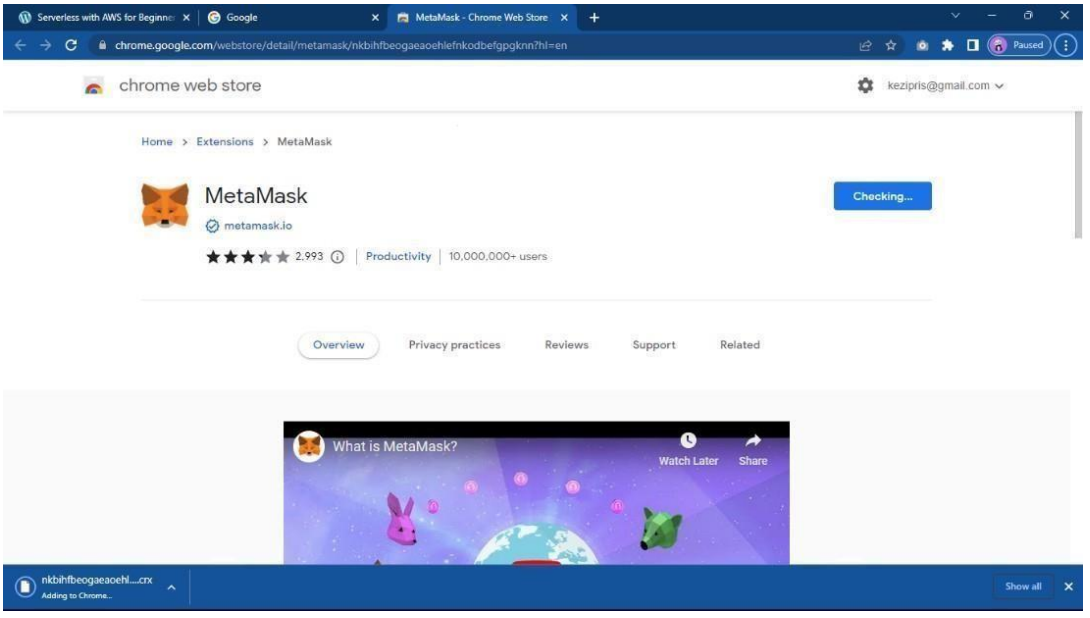
```
Timestamp: 2023-04-22 17:41:07.240201+00:00
Data: First block
Previous Hash: 0
Hash: 876fb923a443ba6afe5fb32dd79961e85be2b582cf74c233842b630ae16fe4d9

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

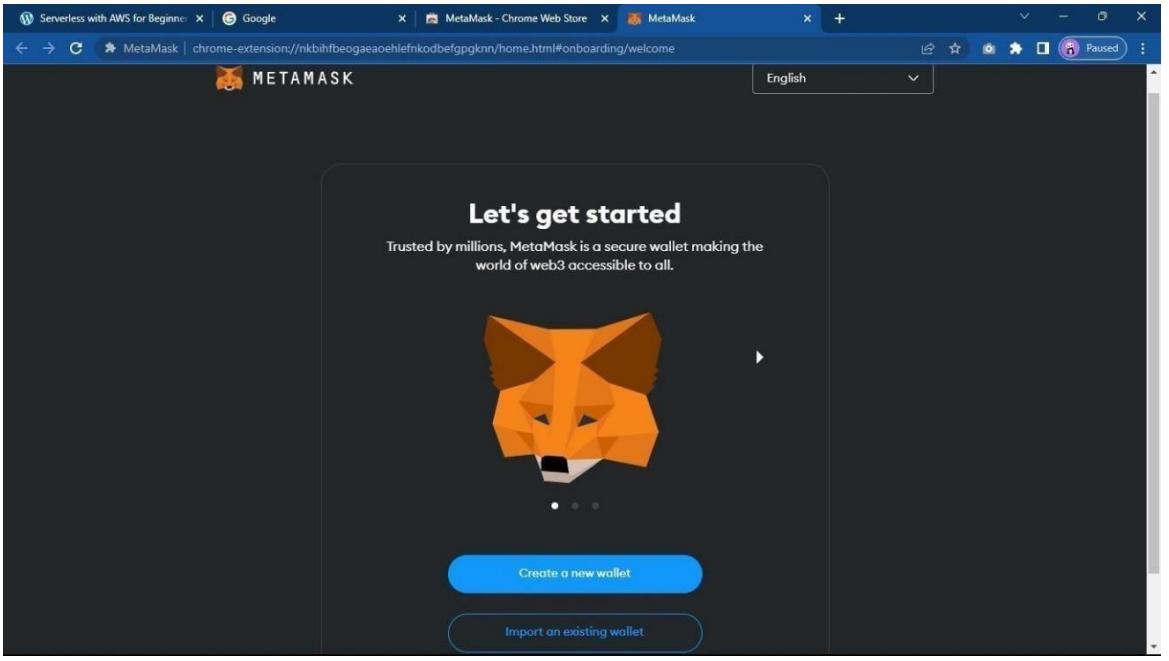
Aim:Install and configure go Ethereum and themist browser.develop and test asample application (MetaMask& remix)

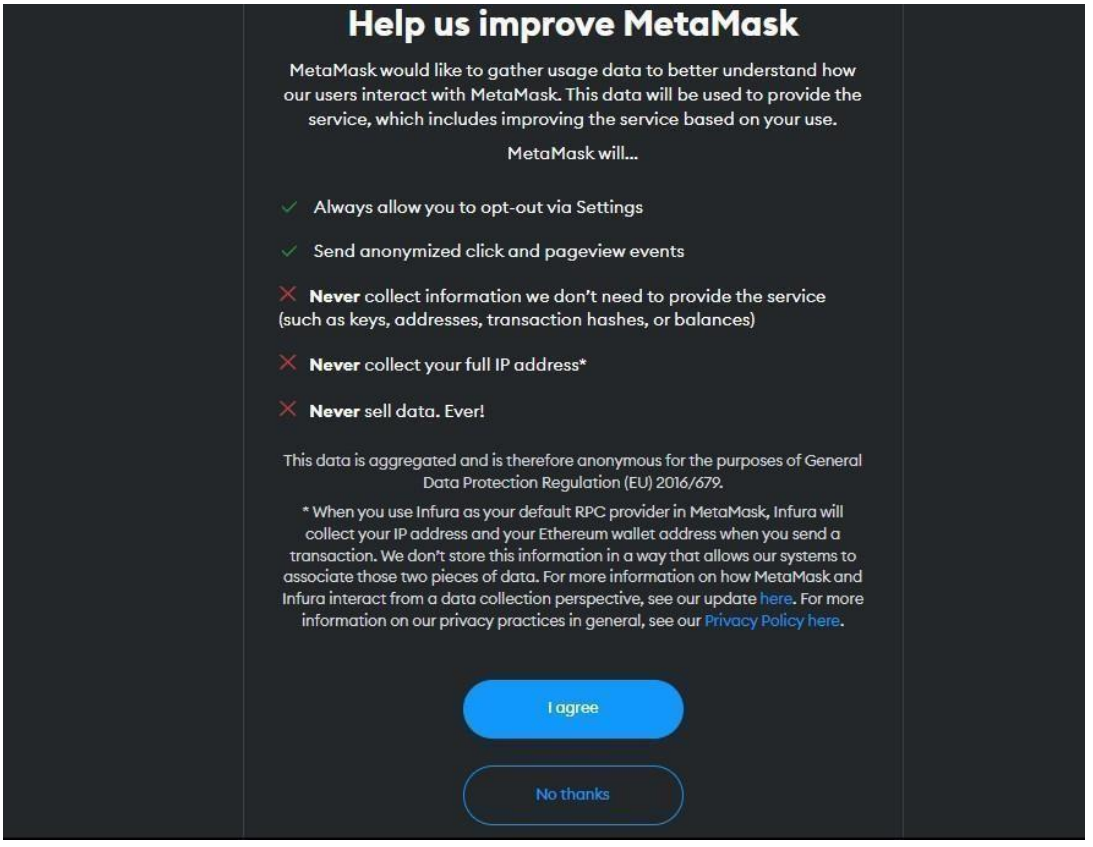
Step1-> InstallMetaMaskextensionforchromefromChromeWebStore



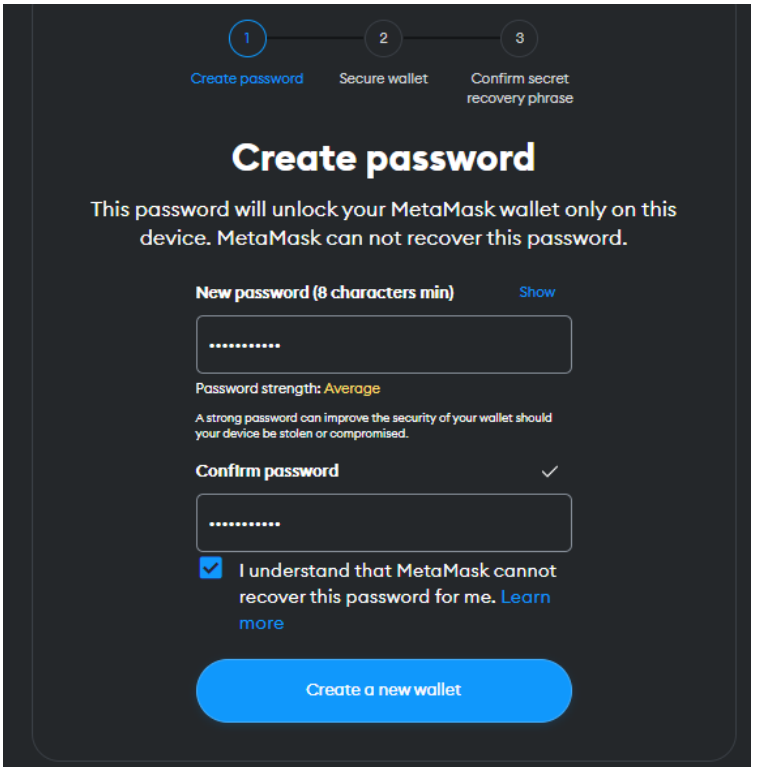


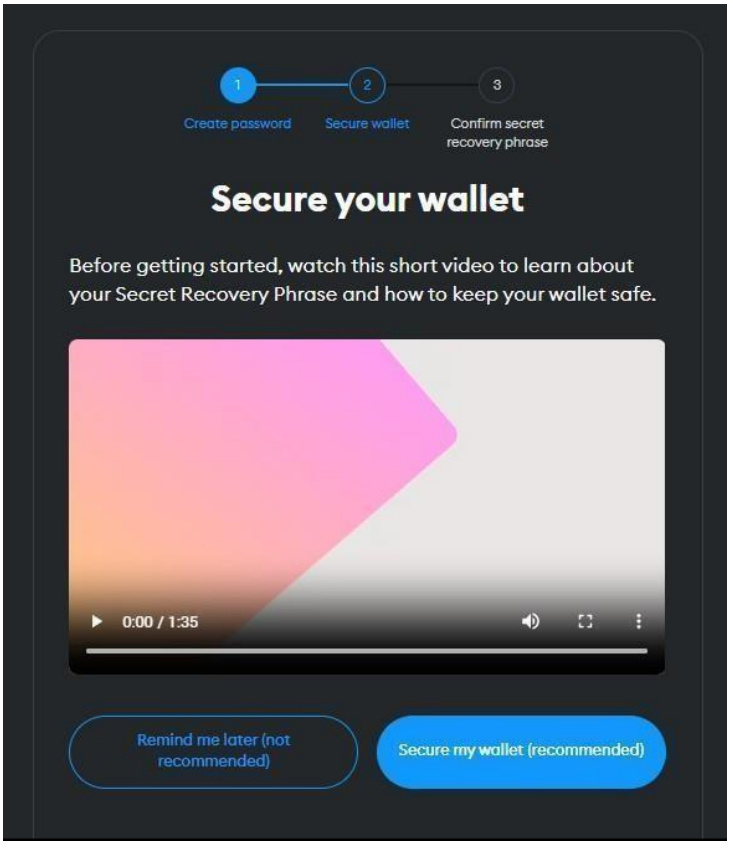
Step 2-> Click on MetamaskExtension in Extensions. Below page will open in anewtab.Clickon Create a New Wallet. Click on I agree.



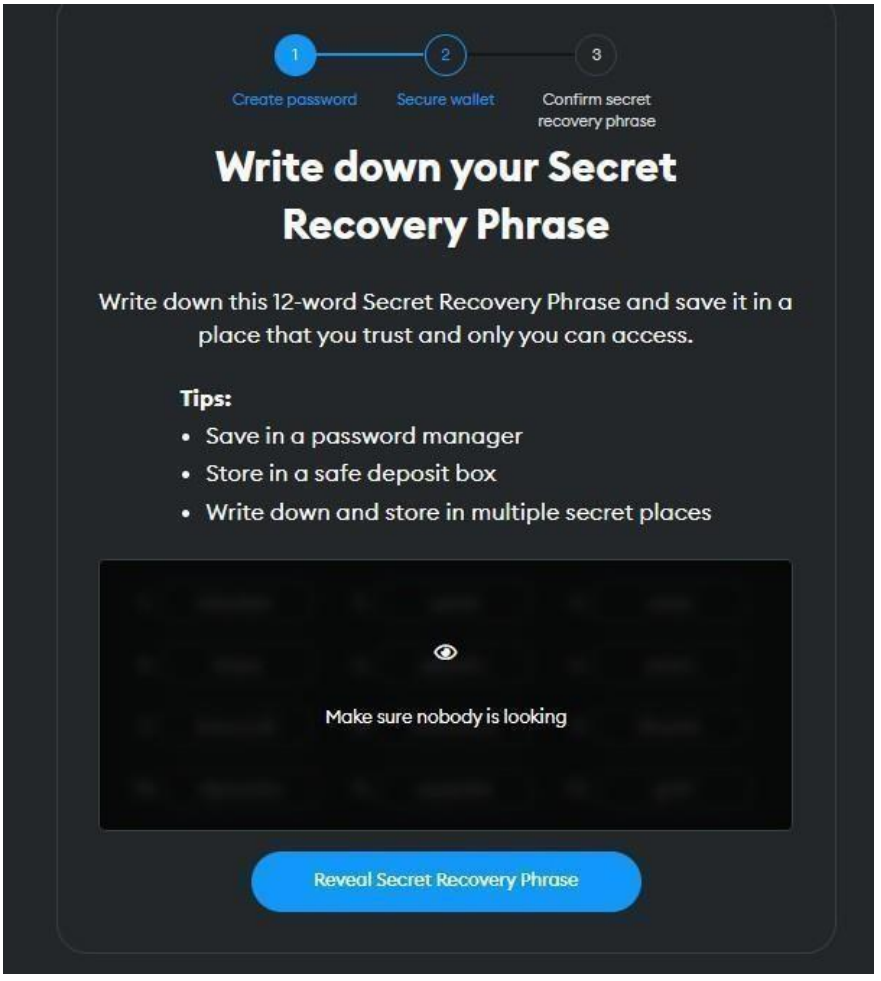


Step 3->Createapassword.Thispasswordcanbeusedonlyonthedeviceitwas created on. Create a Strong password and click on Create a new Wallet button

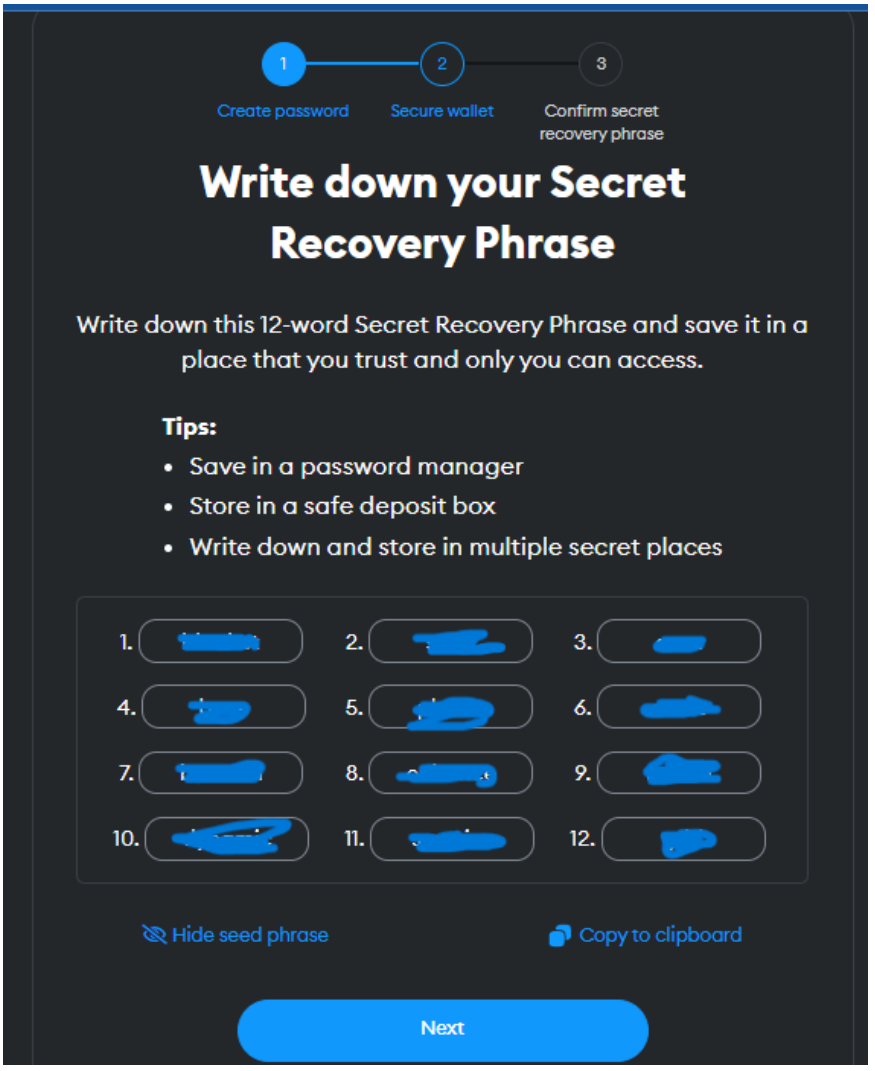




Step4-> ClickonSecuremywalletbutton,followingwindowwillappear

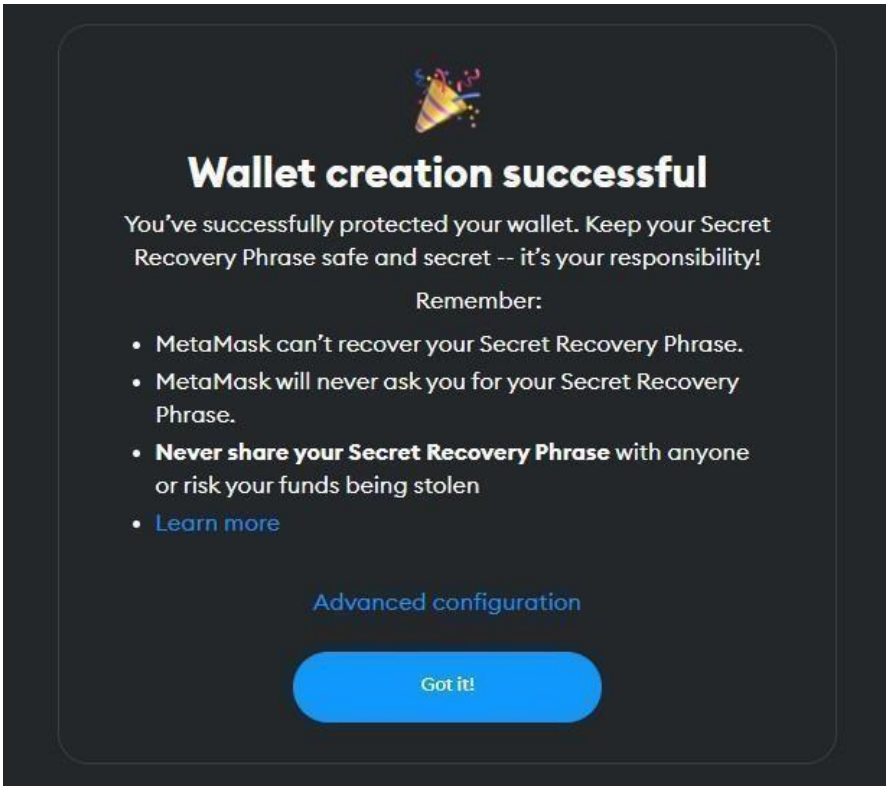


Step5-> ClickonRevealSecretRecoveryPhrasebuttonandsavethewordsinthesamesequence

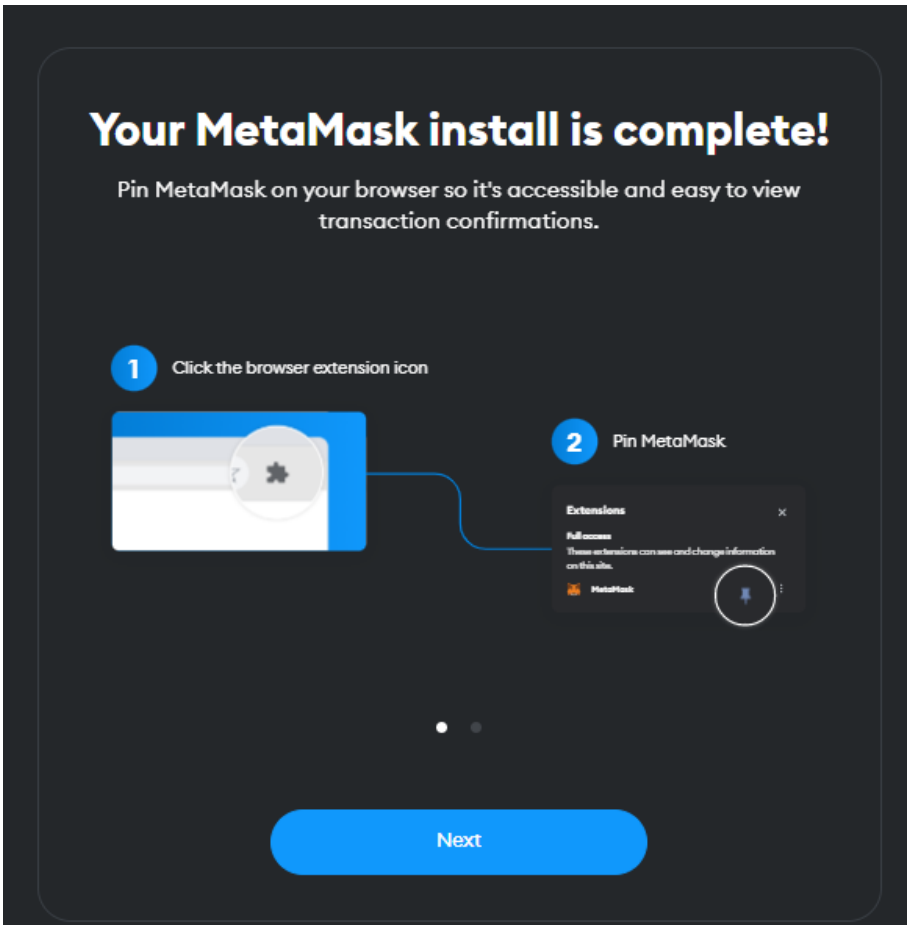


Step6-> Entertherespectivewordsintheemptypositionsandclick**Confirm**.

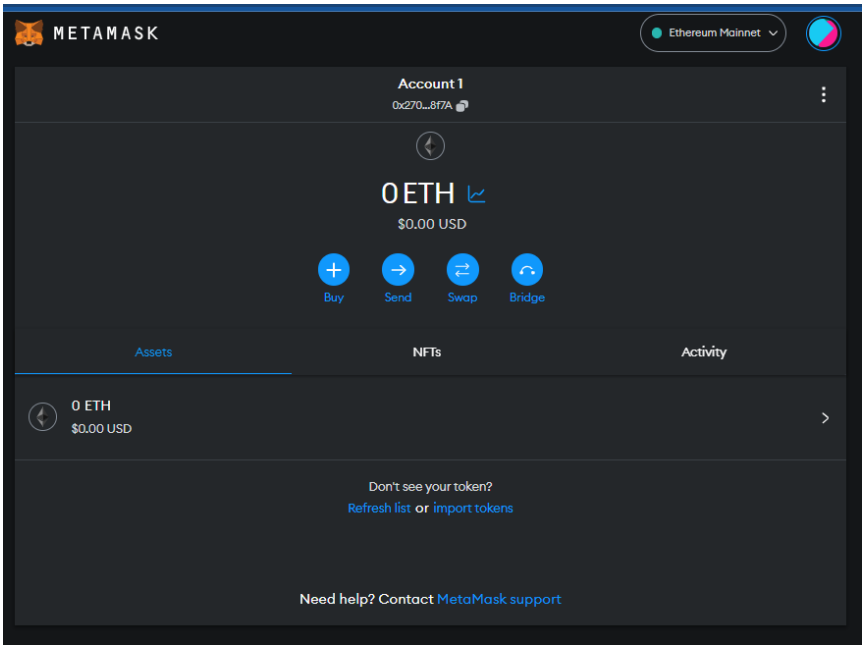
Step7-> Click**Gotit**!



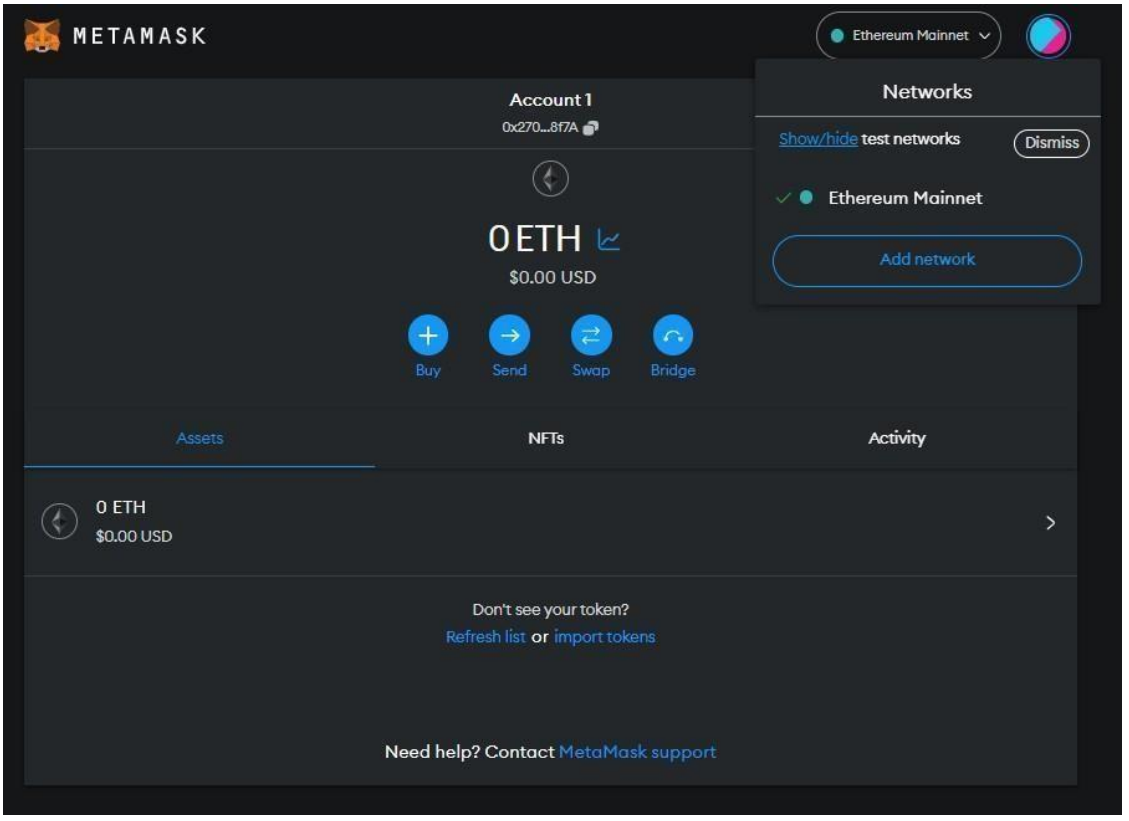
Step8-> ClickonNext

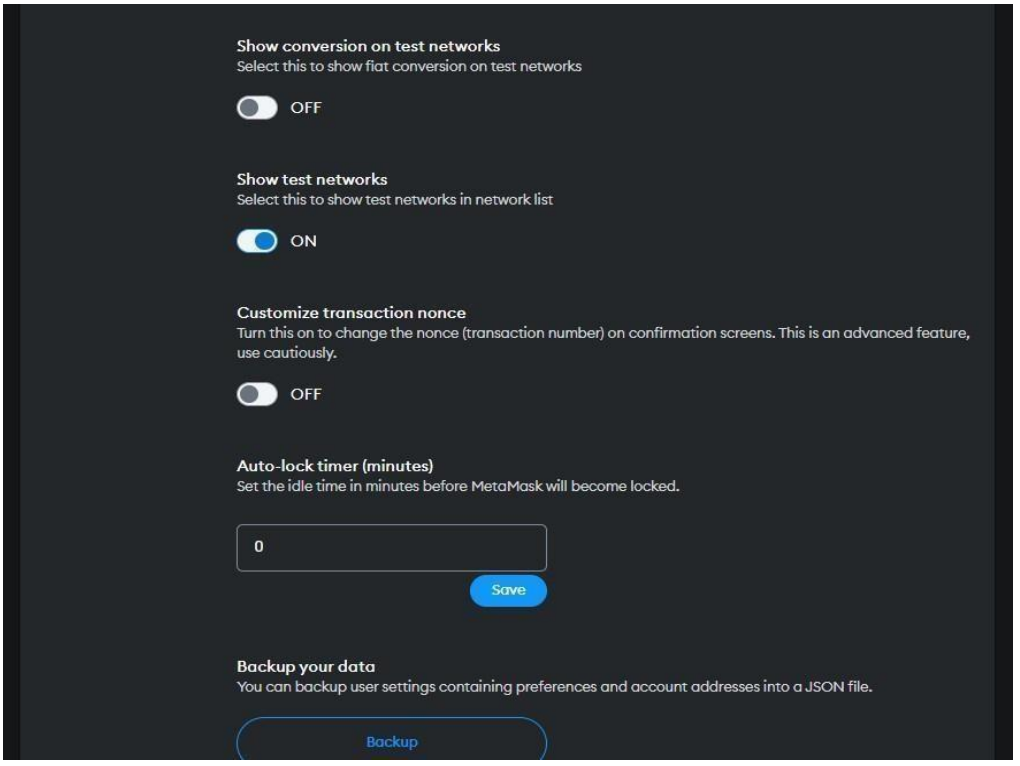


Step9-> FollowingwillbetheDashboard

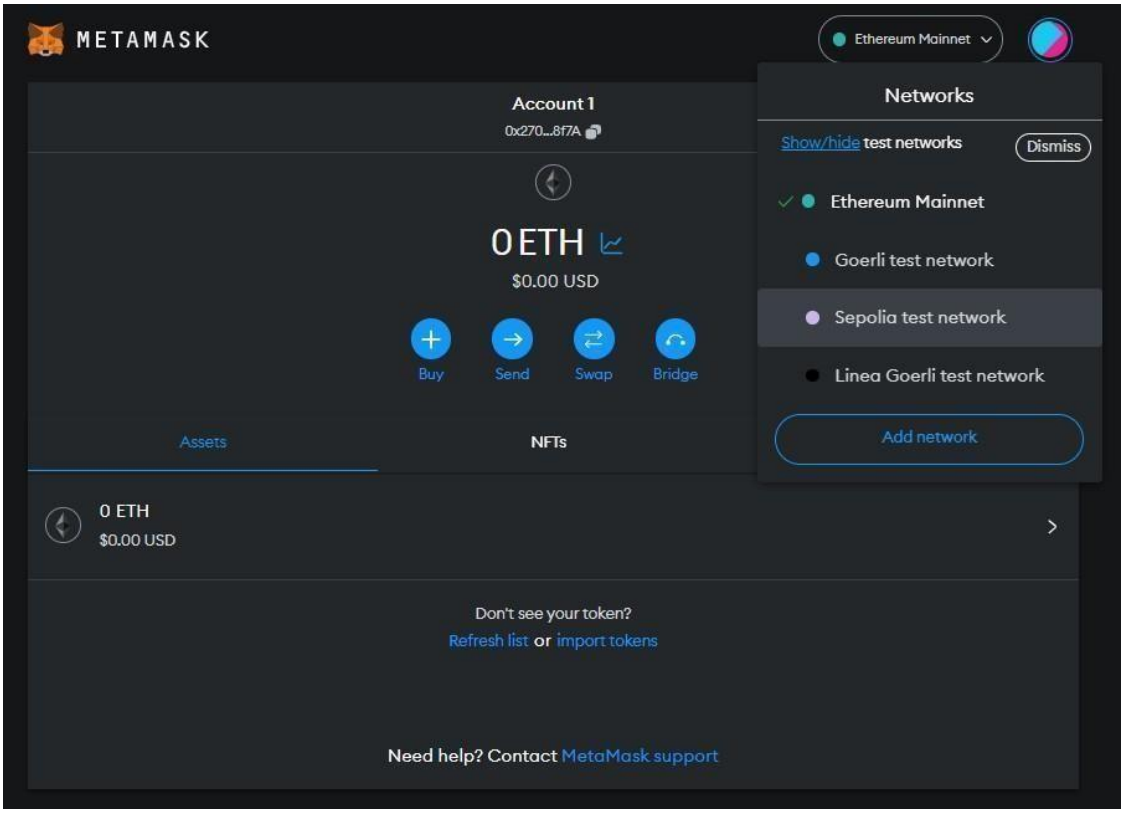


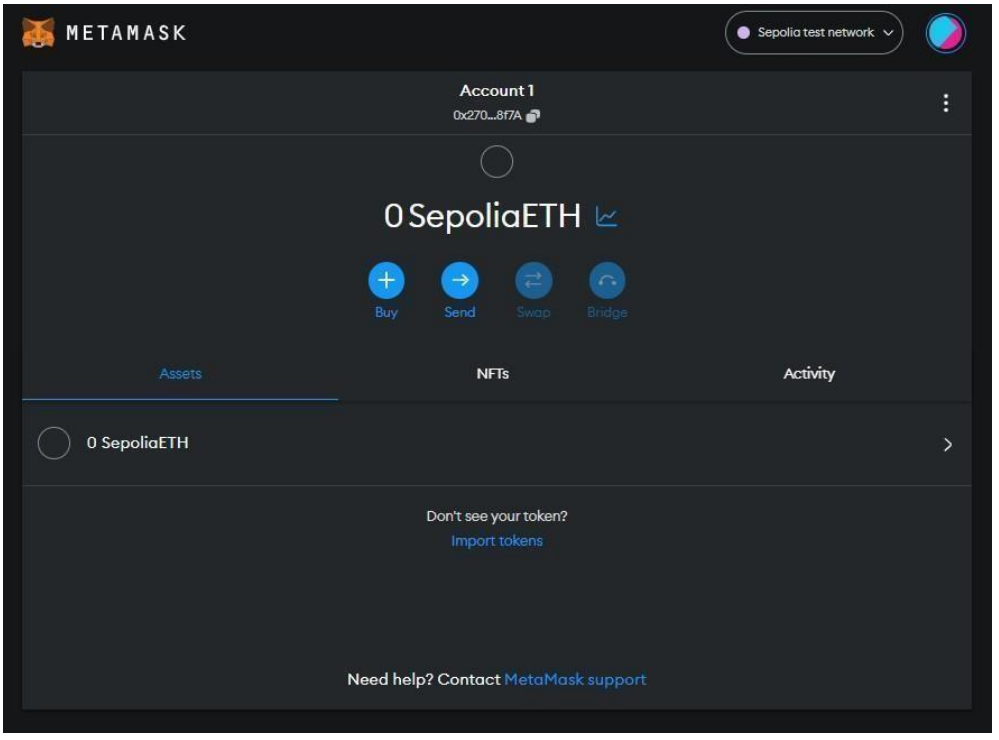
Step10->ClickonEthereumMainnet button.NextclickonShow/hidetestnetworks.



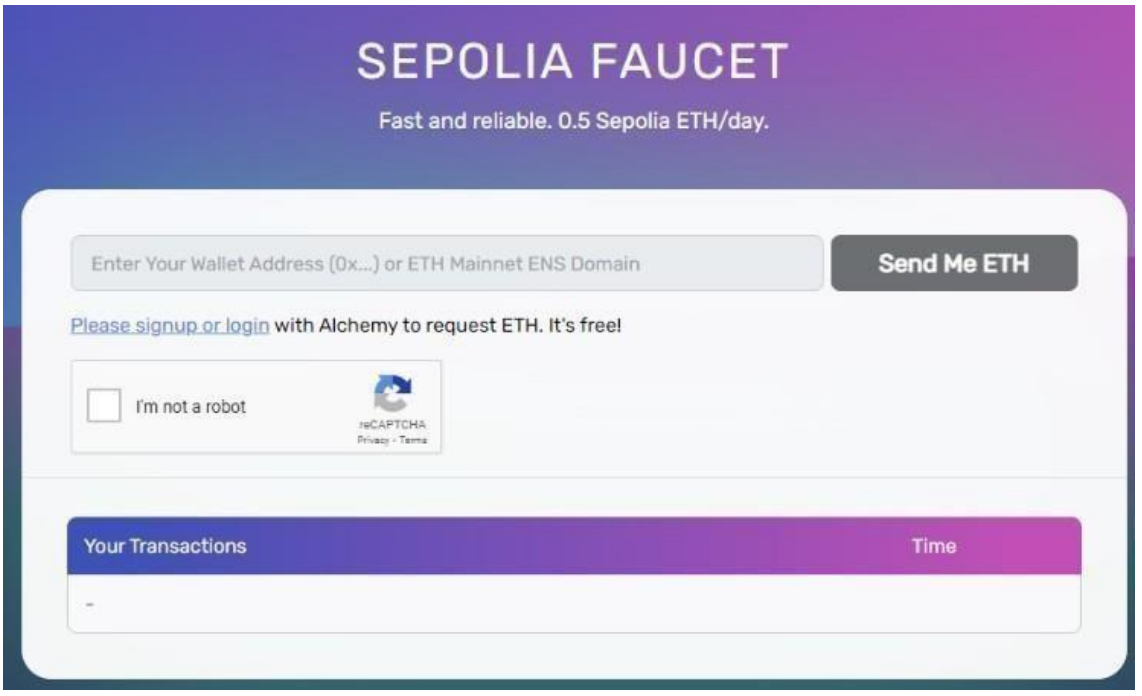


Step 11-> Check if tesnets are shown byclickingonEtherumMainnetbutton.ClickonSepoliatest network.



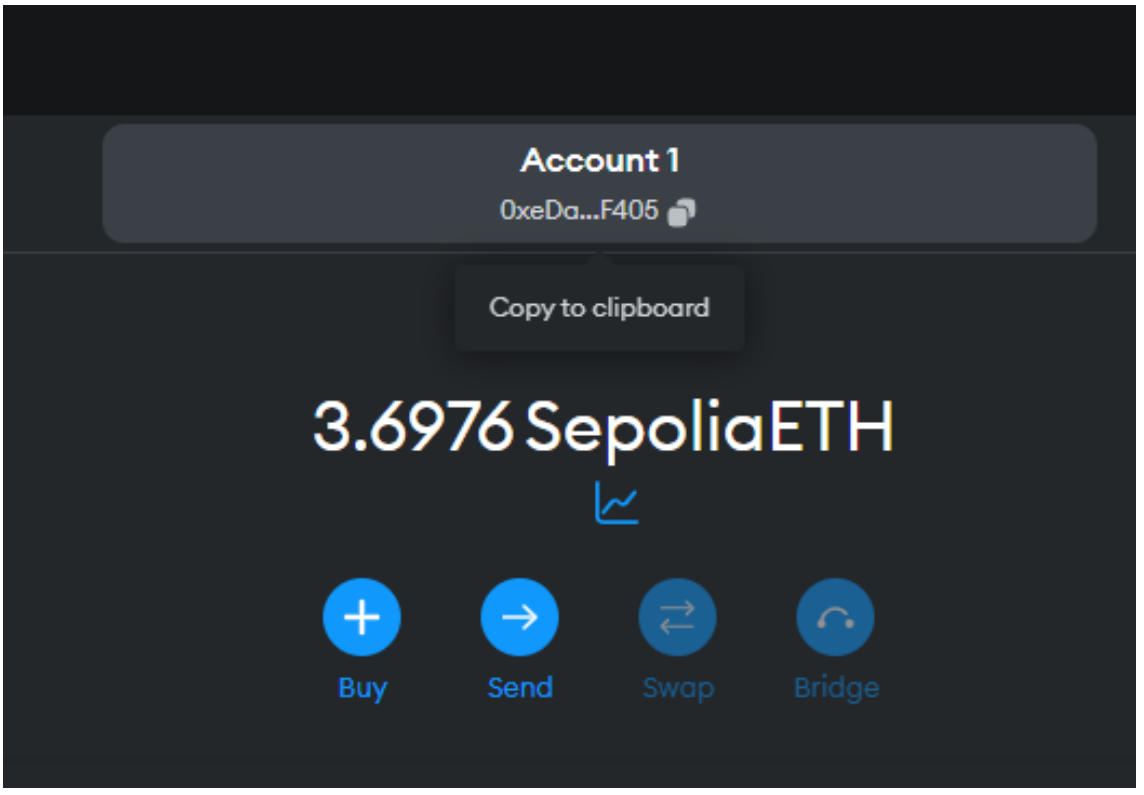


Step12->Gotohttps://sepoliafaucet.com/andClickonAlchemyLoginbutton.

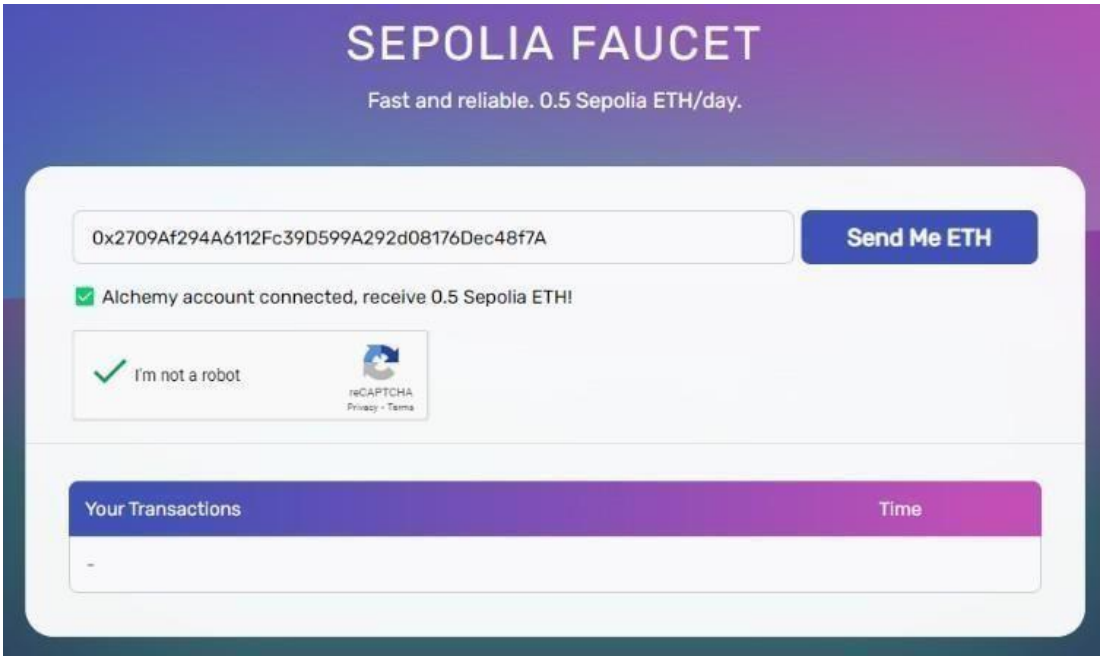


Step13->Loginto agmailaccountin anotherbrowsertabandclickonSignin withGoogle

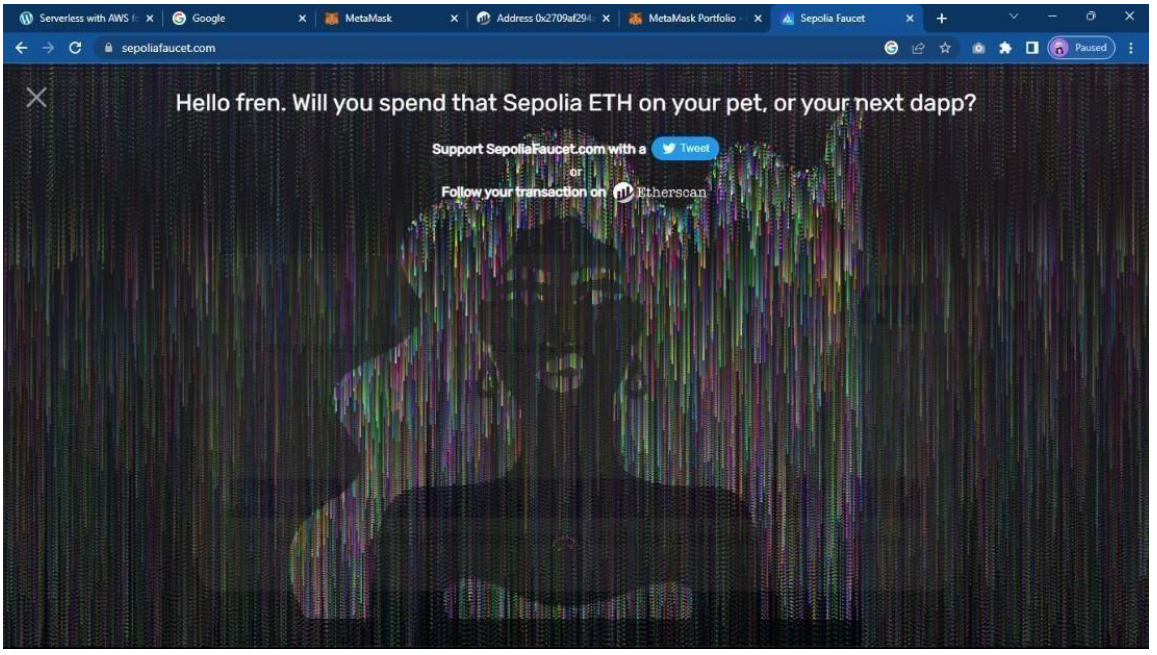
Step14->NowgotoMetaMaskandcopytheaccountaddress.



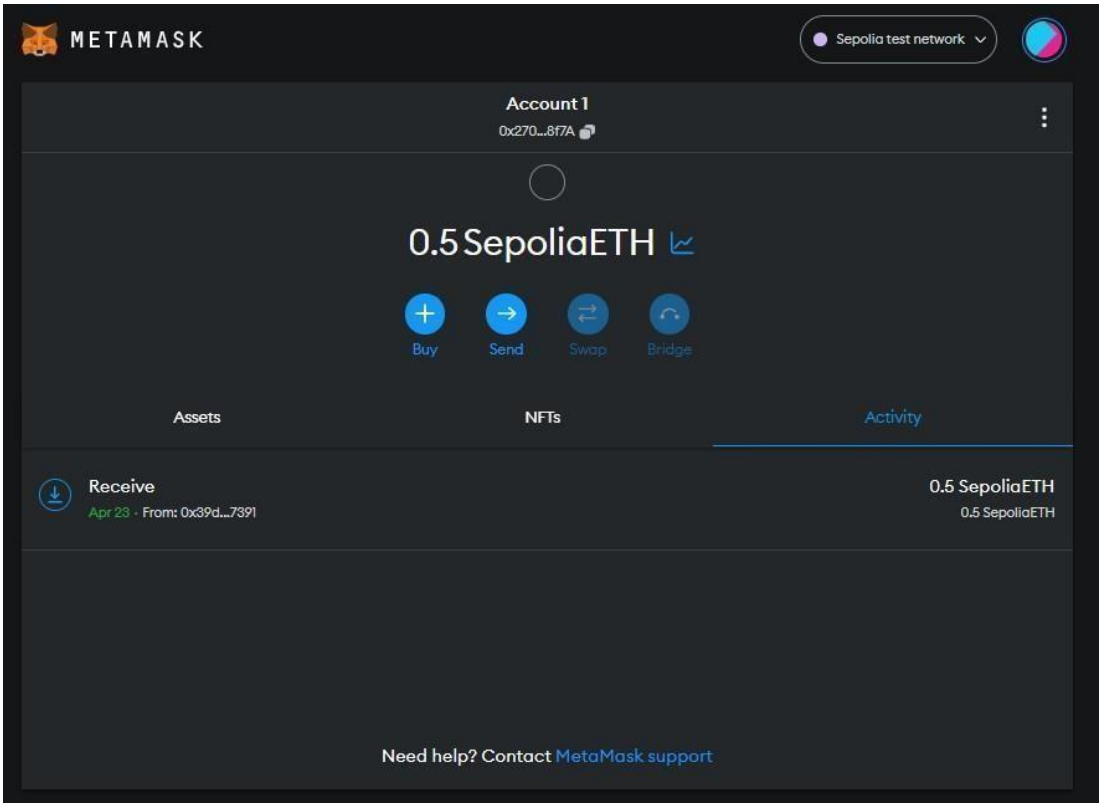
Step15->PastetheaddressandclickonSendMeETH.



Step16->YourETHtransferissuccesfull.Youshouldseeasimilaranimation.



Step17->CheckyourMetaMaskaccountforSepoliatestnetwork.0.5ETHwillbeadded.



PRACTICAL-3

Aim:Implementanddemonstratetheuseofthefollowinginsolidity\

1. TOEXECUTESOLIDITYSCRIPTSGOTO->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. DEPLOYTHESCRIPTSTOEXECUTECODE

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

1. **Variable**

```
pragmasolidity^0.5.0;

contractvariable_demo{uint256s
    um=4;
    //statevariableu
    int256x;addressa
    ;
    strings="welcome";

    functionadd(uint256)public{uint256y=
        2;//local variable
        sum=sum+x+y:sum=sum+x+y;
    }

    functiondisplay()publicviewreturns(uint256)
        {returnsum;
    }
```

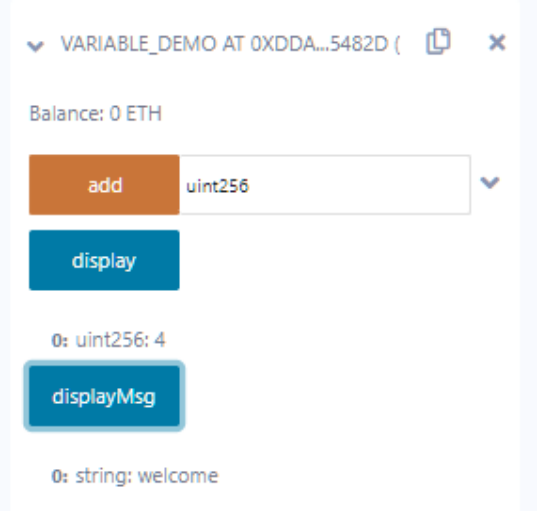


Figure1-Displayingvariablevalue

2. **Strings**

```
pragmasolidity^0.5.0;

contractLearningStrings{stringt
    ext;

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

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

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

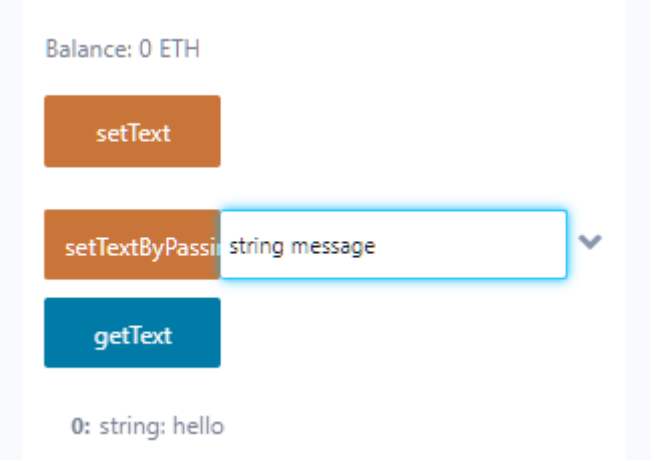


Figure2-Beforesettingnewstringvalue

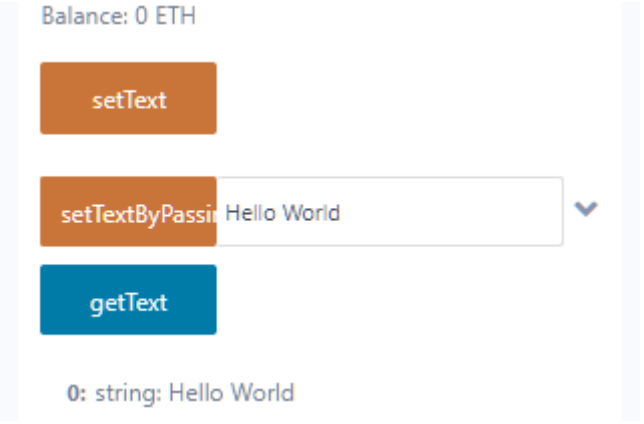


Figure3-Aftersettingstringvalue

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;
}
```

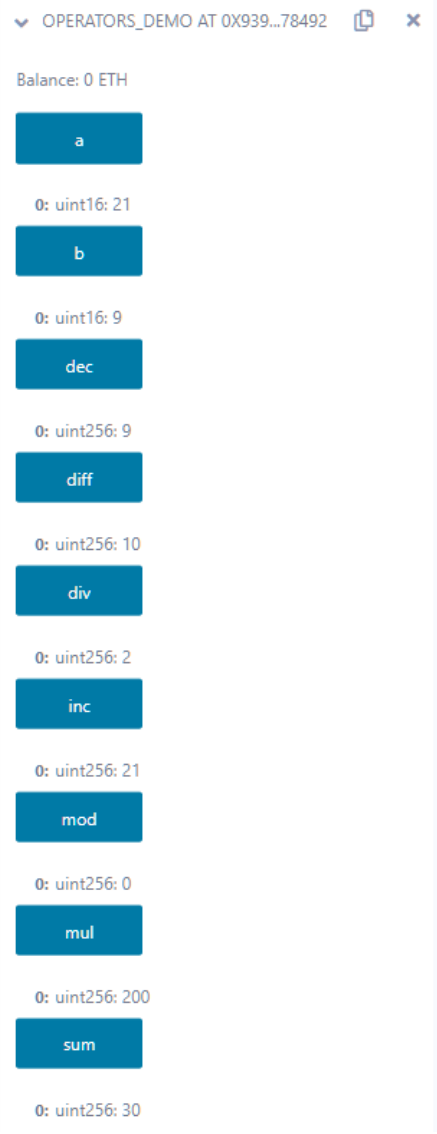


Figure4-Alloperatorsofsoliditydisplayed

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;

	}

}
```

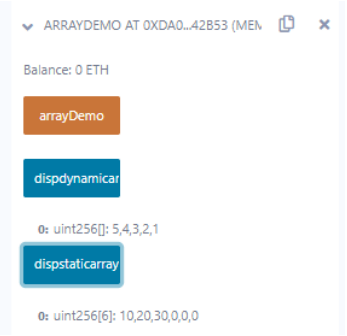


Figure5-Arraydisplayed

5. DecisionMaking

IfElse

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

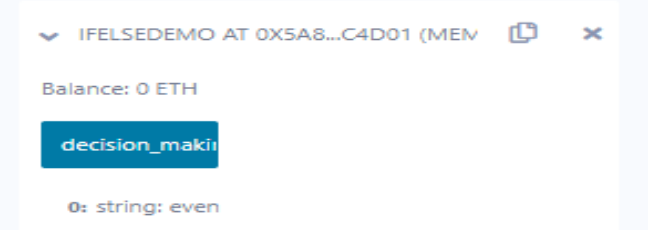


Figure6-Ifelseoutput

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;
    }
}
```

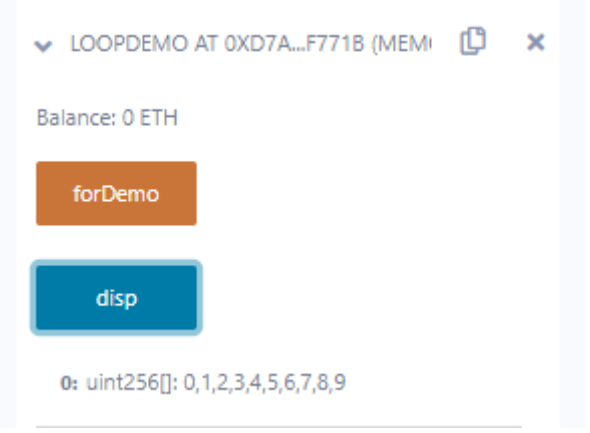


Figure7-Appendingvaluestoarrayusingforloop

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;
    }
}
```

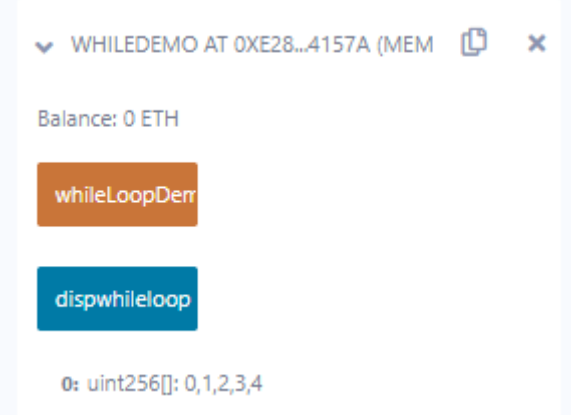


Figure8-Appendingvaluestoarrayusingwhileloop

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);retur
        ndata;
    }
    functiondisplay()publicviewreturns(uint256[]memory){return
        data;
    }
}
```

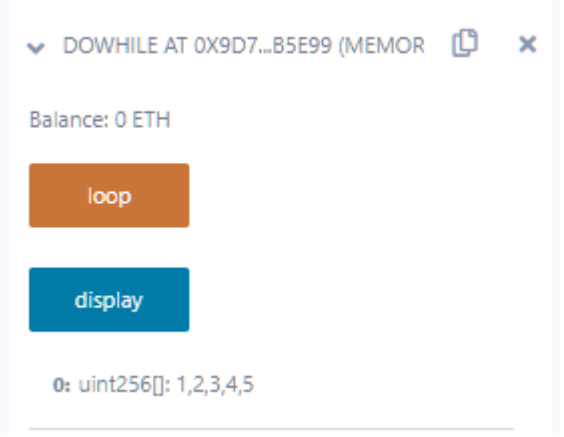


Figure9Appendingvaluestoarrayusingdowhileloop

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;
    }
}
```

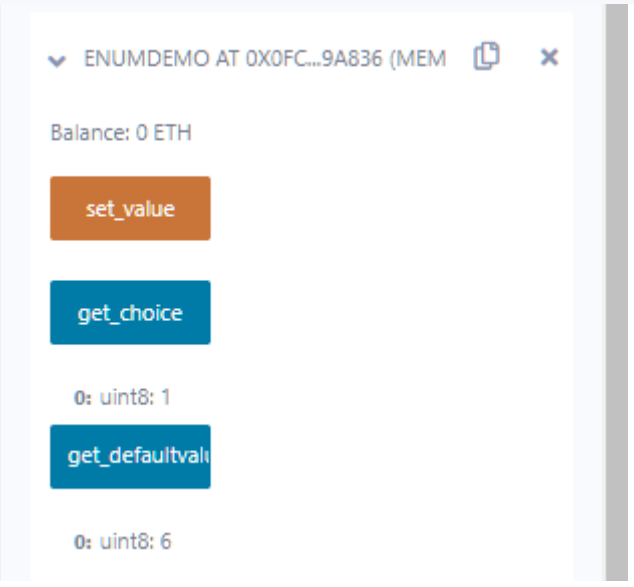


Figure10-Accessingenumvalues

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
        viewretur
        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);
    }
}
```

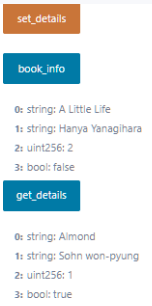


Figure 11-Structuredatatypeinsolidity

9. Mappings

```
pragmasolidity^0.5.0;

contractLedgerBalance{
    mapping(address=>uint256)publicbalances;

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

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

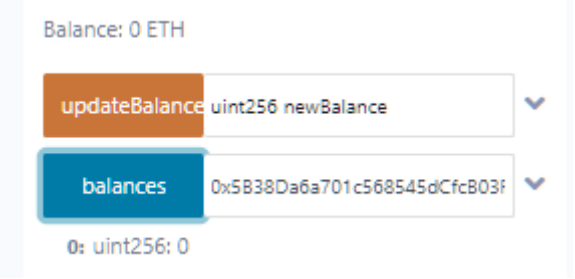


Figure12-Beforeupdatingbalance

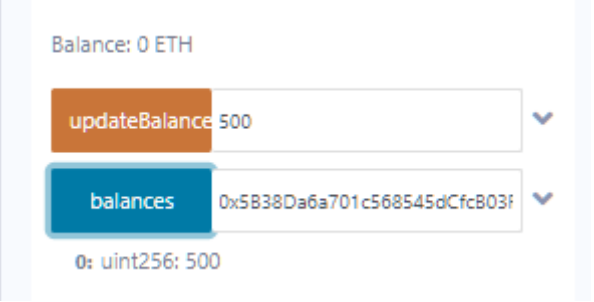
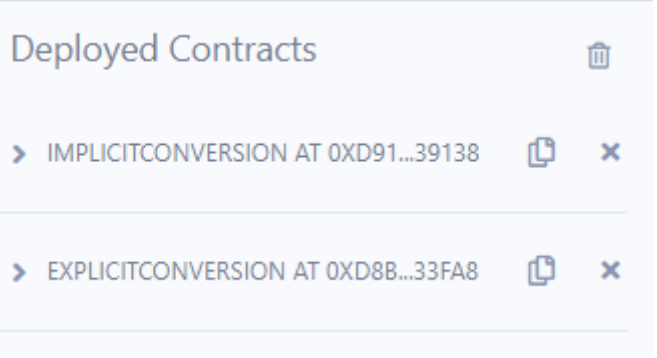


Figure13-Afterupdatingbalance

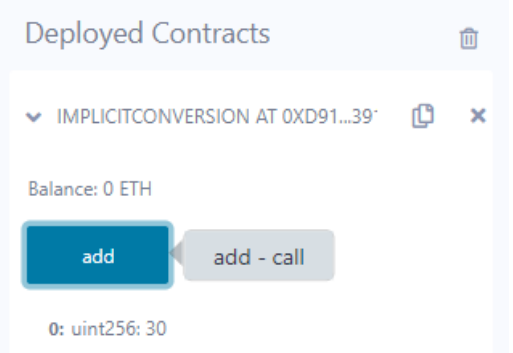
10. Conversions

```
//SPDX-License-
Identifier:MITpragma
solidity^0.8.0;

contractImplicitConversion{
    functionadd()publicpurereturns(uint256){uint256a=10
        ;
        uint256b=20;retur
        na+b;
    }
}

contractExplicitConversion{
    functionconvert()publicpurereturns(bytesmemory){stringmemo
        rystr="Hello World";
        bytesmemoryb=bytes(str);returnb
        ;
    }
}
```

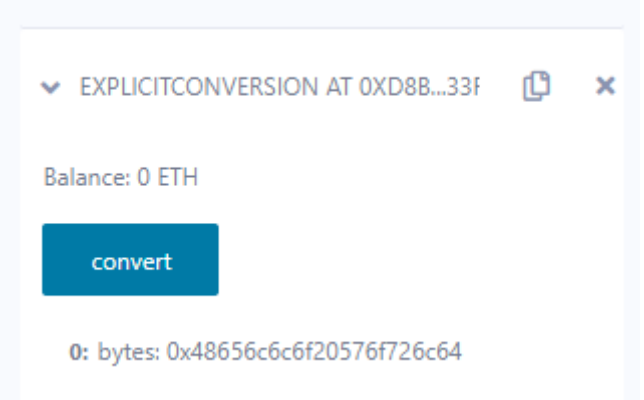
Step1-> Deploybothcontracts



Step2-> OpenImplicitConversionandclickonaddbuttontosumanddisplayvalue



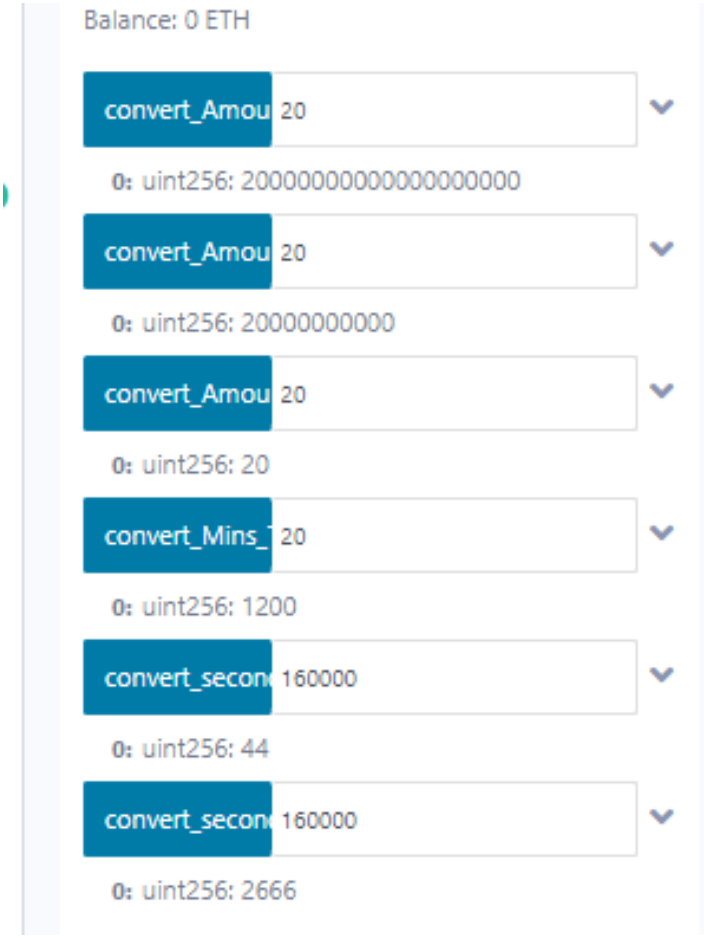
Step3-> OpenExplicitConversionandclickonconvertbutton



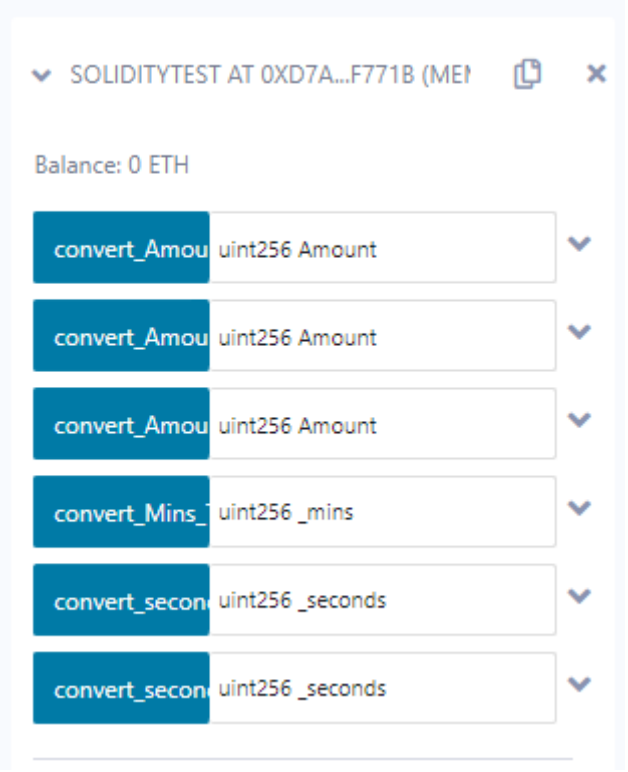
11. EtherUnits

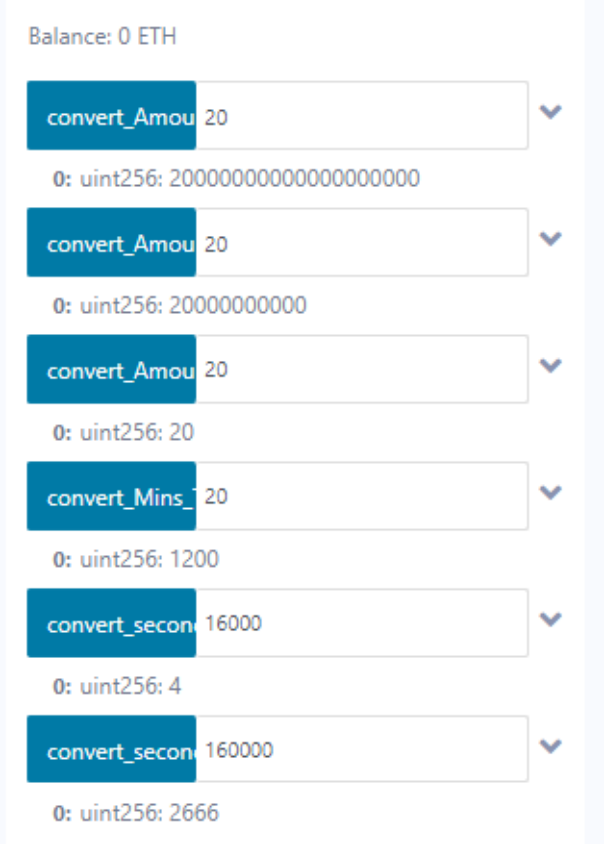
```
//SPDX-License-
Identifier:MITpragma
solidity^0.8.0;

contractSolidityTest{
    functionconvert_Amount_to_Wei(uint256Amount)public
        pure
        returns(uint256)
    {
        returnAmount*1wei;
    }

    functionconvert_Amount_To_Ether(uint256Amount)publi
        c
        pure
        returns(uint256)
    {
        returnAmount*1ether;
    }

    functionconvert_Amount_To_Gwei(uint256Amount)public
        pure
        returns(uint256)
    {
        returnAmount*1gwei;
    }

    functionconvert_seconds_To_mins(uint256_seconds)pub
        lic
        pure
        returns(uint256)
    {
        return_seconds/60;
```

```
    }

    functionconvert_seconds_To_Hours(uint256_seconds)pu
        blic
        pure
        returns(uint256)
    {
        return_seconds/3600;
    }

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



Step1-> Providevaluestoeachfunctionandclickonthem

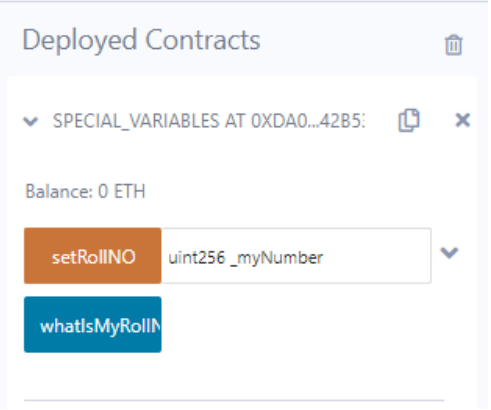




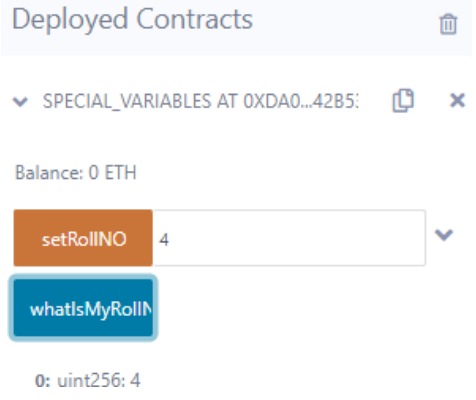
12. SpecialVariables

```
//SPDX-License-
Identifier:MITpragma
solidity^0.8.0;

contractSpecial_Variables{mapping(addre
    ss=>uint256)rollNo;

    functionsetRollNO(uint256_myNumber)public{rollNo[msg.sende
        r] =_myNumber;
    }

    functionwhatIsMyRollNumber()publicviewreturns(uint256){returnr
        ollNo[msg.sender];
    }
}
```

Step1-> DeploycontractSpecialVariables



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;
    }
```

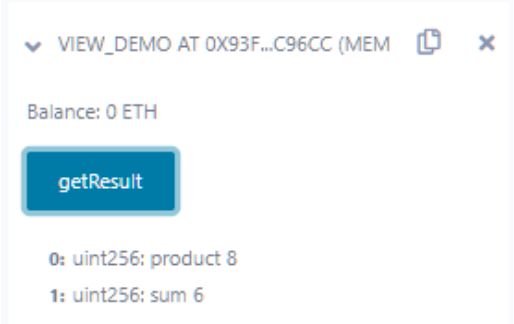


Figure14-Viewfunctiondemo

2. **PureFunctions**

```
pragmasolidity^0.5.0;

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

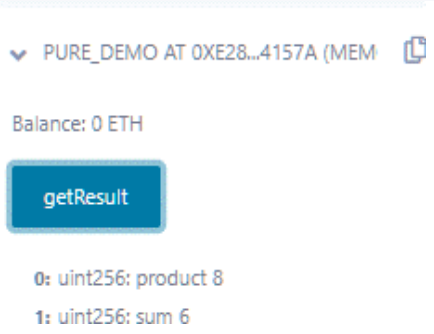


Figure15-Purefunctionoutput

3. MathematicalFunctions

```
pragmasolidity^0.5.0;

contract Test{

    functionCallAddMod()publicpurereturns(uint){return

        addmod(7,3,3);

    }

    functionCallMulMod()publicpurereturns(uint){return

        mulmod(7,3,3);

    }

}
```

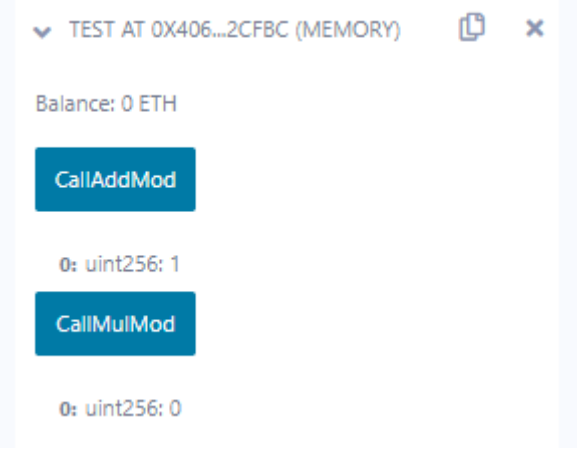


Figure16-Mathematicalfunctionsinsolidity

4. CryptographicFunctions

```
pragmasolidity^0.5.0;contract

Test{

    functioncallKeccak256()publicpurereturns(bytes32result){return

        keccak256("BLOCKCHAIN");

    }

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

    }

    functioncallripemd()publicpurereturns(bytes20result){return

        ripemd160("BLOCKCHAIN");

    }

}
```

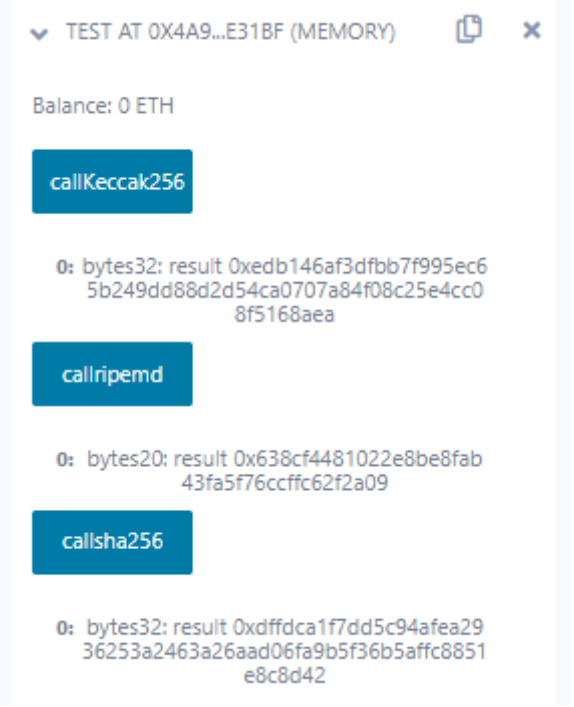
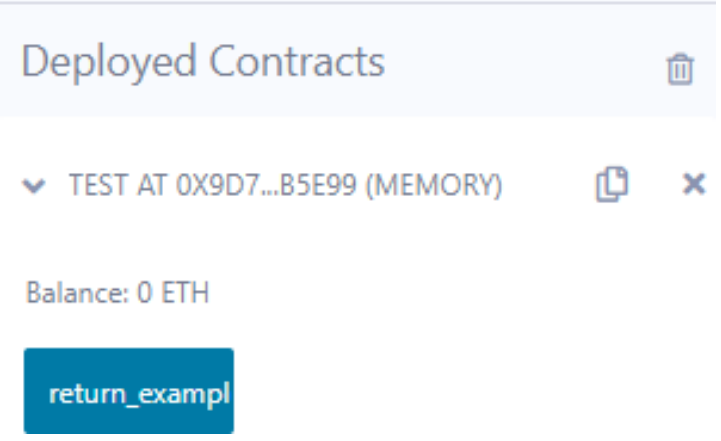


Figure17-Cryptographyalgorithmsinsolidity

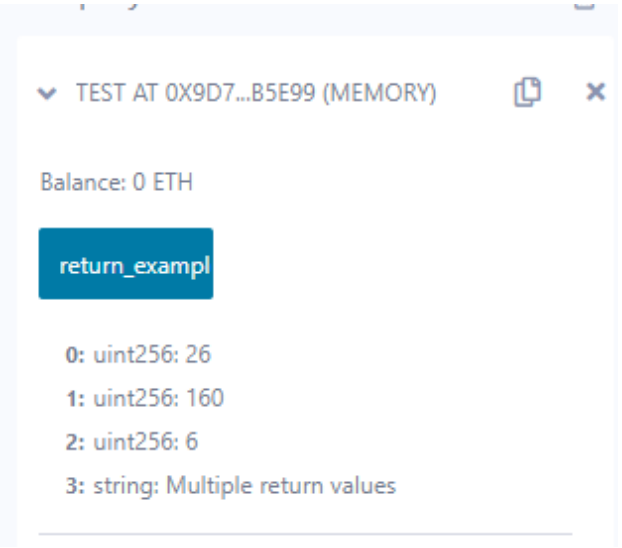
5. **Functions**

```
// SPDX-License-Identifier:
MITpragmasolidity>=0.4.22<0.9.0;

contractTest{
    functionreturn_example()pub
        lic
        purereturn
        s(
            uint256,uint2
            56,uint256,st
            ringmemory
        )
    {
        uint256num1=10;uint256n
        um2=16;
        uint256sum=num1+num2;uint25
        6prod=num1*num2;uint256diff
        =num2-num1;
        stringmemorymessage="Multiplereturnvalues";return(sum,
        prod,diff,message);
    }
}
```

Step1-> DeployTestContract



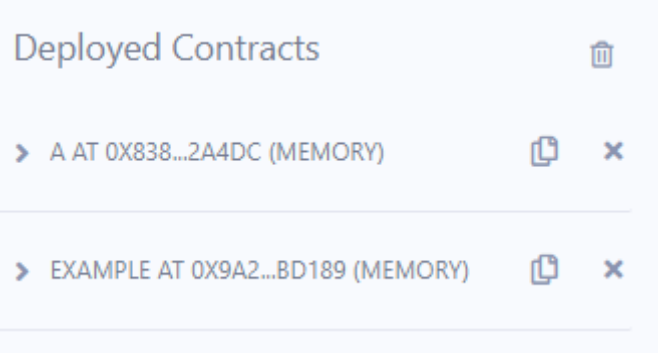
Step2-> Clickonreturn_examplebuttontodisplayallvalues



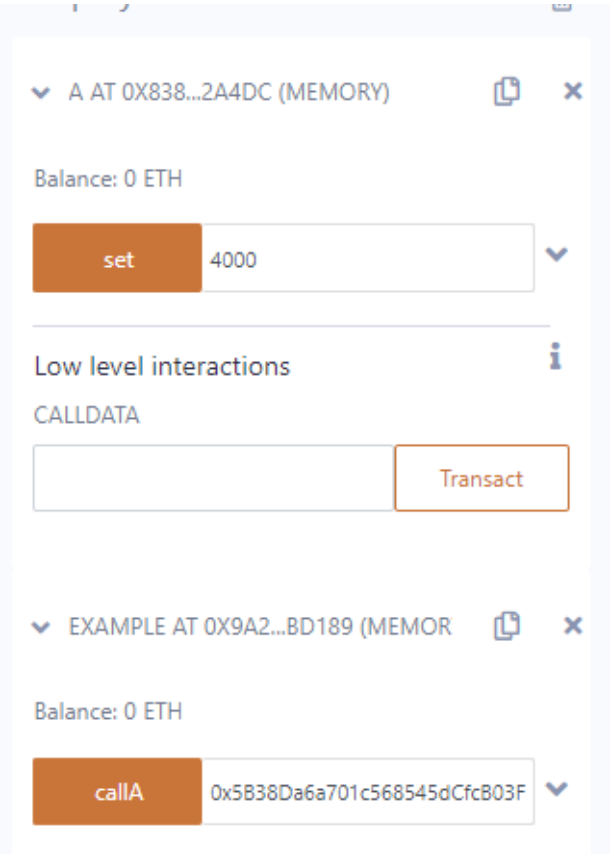
6. **FallbackFunction**

```
//SPDX-License-
Identifier:MITpragma
solidity^0.5.12;

contractA{
    uint256n;

    functionset(uint256value)external{n=va
        lue;
    }

    function()externalpayable{n=0
        ;
    }
}

contractexample{
    functioncallA(Aa)publicreturns(bool){
        (boolsuccess,)=address(a).call(abi.encodeWithSignature("setter()"));require(success
        );
        addresspayablepayableA=address(uint160(address(a)));return
        (payableA.send(2ether));
    }
}
```

Step1-> DeploybothA&examplecontracts



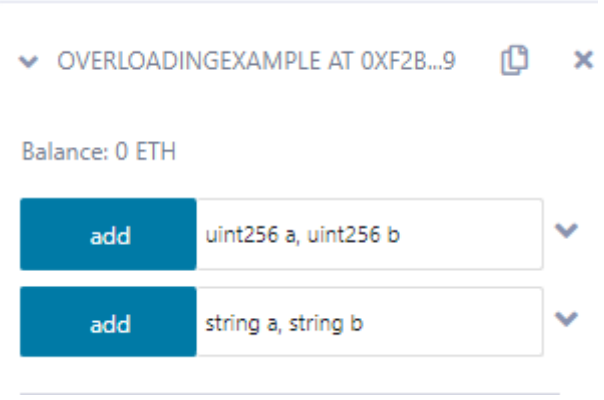
Step2-> Providevaluestobothdeployedcontractsaccordingly(useanyaddress)



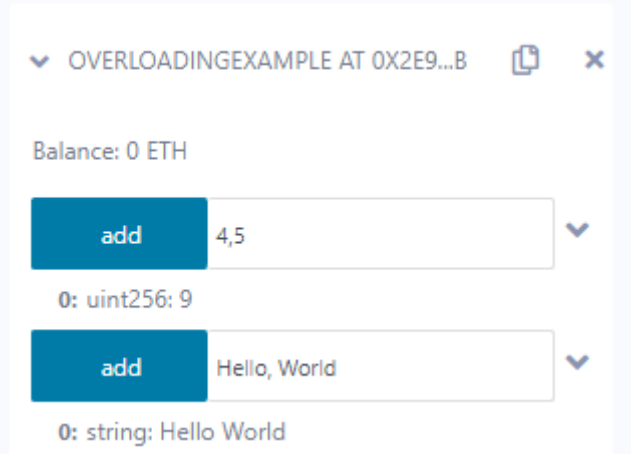
7. FunctionOverloading

```
//SPDX-License-
Identifier:MITpragma
solidity^0.8.0;

contractOverloadingExample{
    functionadd(uint256a,uint256b)publicpurereturns(uint256){returna+b;
    }

    functionadd(stringmemorya,stringmemoryb)public
        pure
        returns(stringmemory)
    {
        returnstring(abi.encodePacked(a,b));
    }
}
```

Step1-> DeployOverloadingExamplecontract

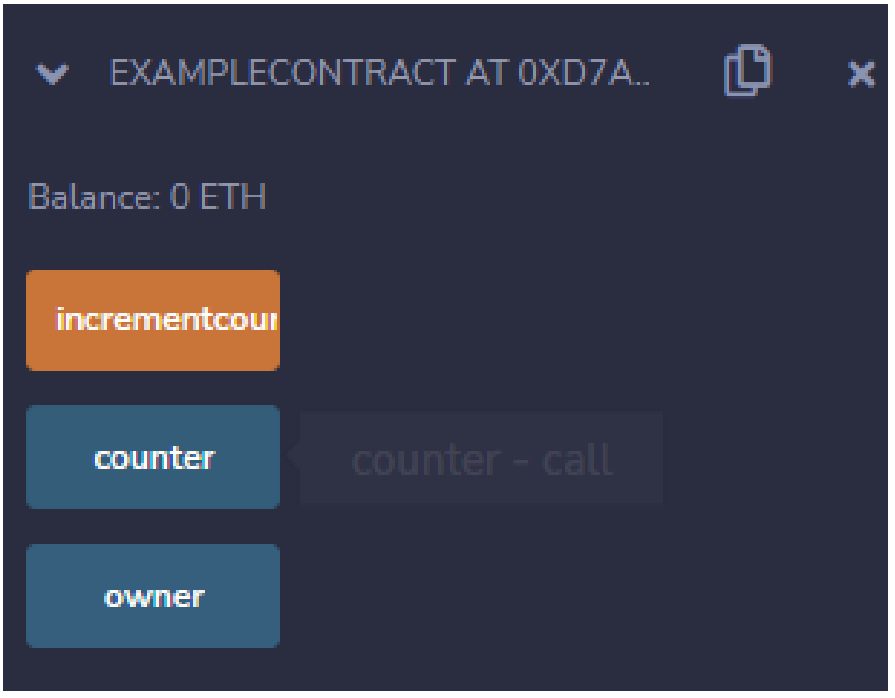


Step2-> Giveintegerandstringvaluestobothaddfunctionsasbelow



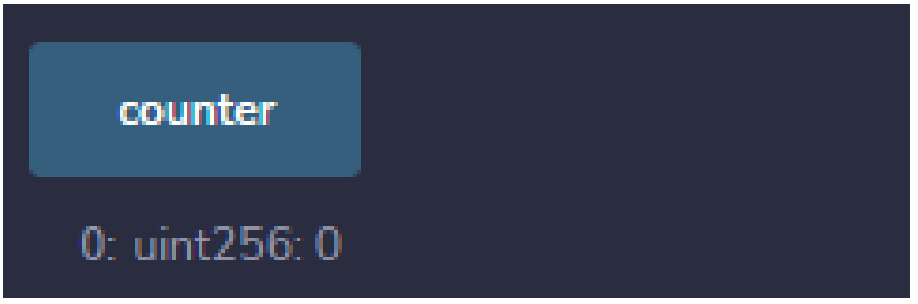
8. Functionmodifiers

```
//SPDX-License-
Identifier:MITpragma
solidity^0.5.0;

contractExampleContract{
    addresspublicowner=0x5B38Da6a701c568545dCfcB03FcB875f56beddC4;uint256pub
    liccounter;

    modifieronlyowner(){
        require(msg.sender==owner,"Onlythecontractownercancall");
        _;
    }

    functionincrementcounter()publiconlyowner{counter++
        ;
    }
}
```

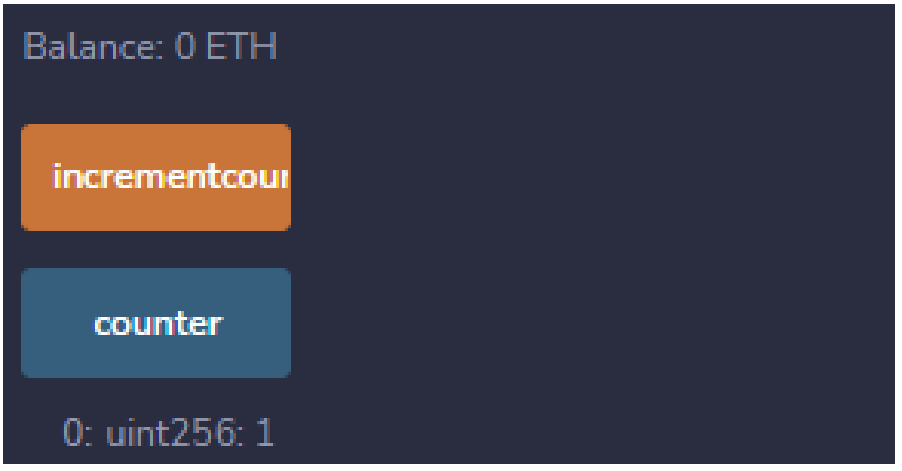


Step1-> Clickonownerbutton

Step2-> Clickoncounterbuttoninitiallyitis 0.



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



PRACTICAL-4

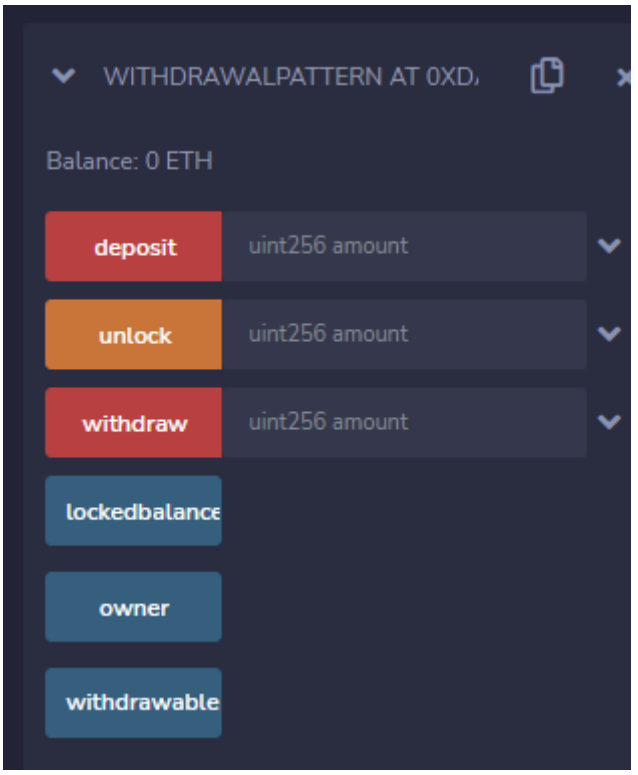
Aim:Implementanddemonstratetheuseofthefollowinginsolidity

A) WithdrawalPattern,RestrictedAccess

1) WithdrawalPattern

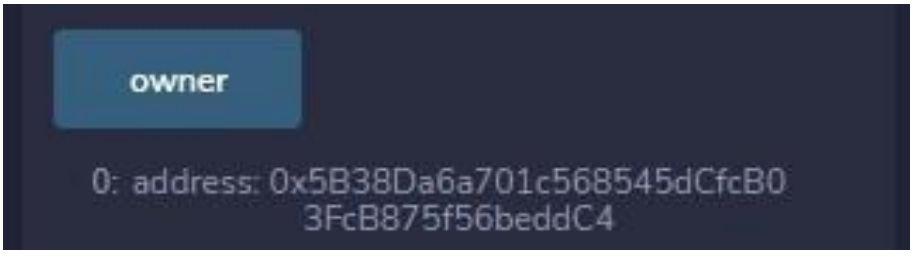
```
//SPDX-License-
Identifier:MITpragma
solidity0.8.18;

contractWithdrawalPattern{addre
    sspublicowner;
    uint256publiclockedbalance;uint256p
    ublicwithdrawablebalance;

    constructor(){
        owner=msg.sender;
    }

    modifieronlyowner(){
        require(msg.sender==owner,"Onlytheownercancallthisfunction");
        _;
    }

    functiondeposit(uint256amount)public
        payable{require(amount>0,"Amountmustbegreaterthanzero");lo
        ckedbalance+=amount;
    }

    functionwithdraw(uint256amount)publicpayableonlyowner{require(
            amount<=withdrawablebalance,"Insufficient
            withdrawablebalance"
        );
        withdrawablebalance-
        =amount;payable(msg.sender).transfer(amount);
    }

    functionunlock(uint256amount)publiconlyowner{
        require(amount<=lockedbalance,"Insufficientlockedbalance");lockedbal
        ance-=amount;
        withdrawablebalance+=amount;
    }
}
```

Output:

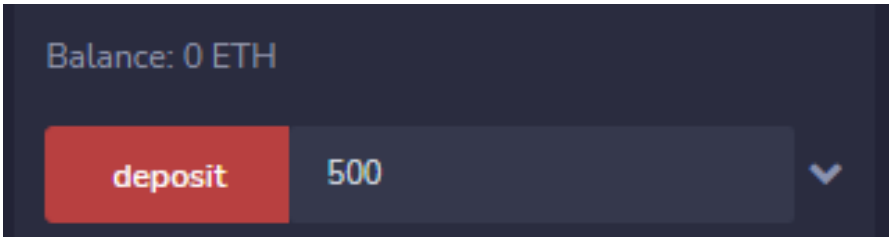


Flowofexecution

Step1-> Clickonowner



Step2-> Enteranamountandclickondeposit

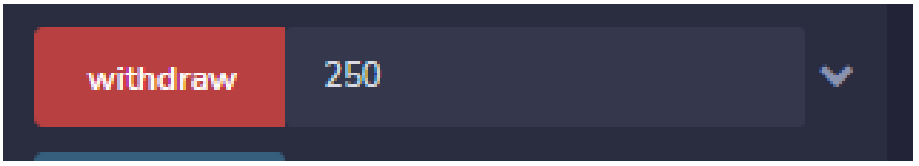


Step3-> Clickonlockedbalancebuttontodisplaythelockedamountintheaccount

Step4-> Clickonwithdrawablebalancebutton

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

Step6-> Enteranyamountyouwanttowithdrawand Clickthewithdrawbutton. Youshouldgetanerrorandthetransactionshouldbereverted.

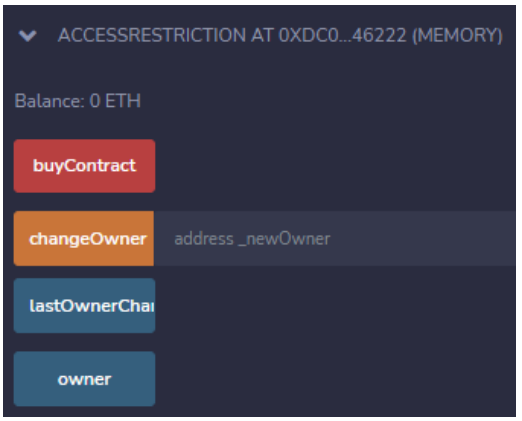



```
CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: WithdrawalPattern.withdrawablebalance() data: 0xd11...c9cb7

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: 0x5B3...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: 0x5B3...eddC4 to: WithdrawalPattern.withdraw(uint256) 0xdda...5482d value: 0 wei data: 0x2e1...000fa logs: 0 hash: 0x3e3...0937c
```

2) RestrictedAccess

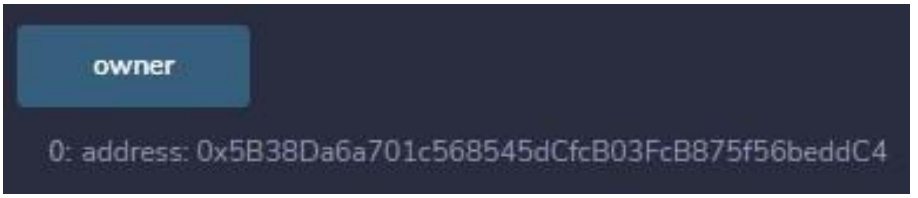
```
//SPDX-License-
Identifier:MITpragma
solidity^0.8.18;

contractRestrictedAccess{
    addresspublicowner=msg.sender;
    uint256publiccreationTime=block.timestamp;

    modifieronlyBy(address_account){
        require(msg.sender==_account,"Sendernotauthorized!");
        _;
    }

    modifieronlyAfter(uint256_time){
        require(block.timestamp>=_time,"Functionwascalledtooearly!");
        _;
    }

    modifiercosts(uint256_amount){
        require(msg.value>=_amount,"NotenoughEtherprovided!");
        _;
    }

    functionforceOwnerChange(address_newOwner)public
        payablecosts(200
        ether)
    {
        owner=_newOwner;
    }

    functionchangeOwner(address_owner)publiconlyBy(owner){owner=_o
        wner;
    }

    functiondisown()publiconlyBy(owner)onlyAfter(creationTime+3weeks){deleteowner;
    }
}
```

Output:



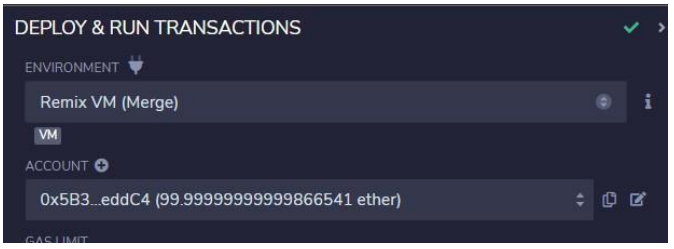
Flowofexecution

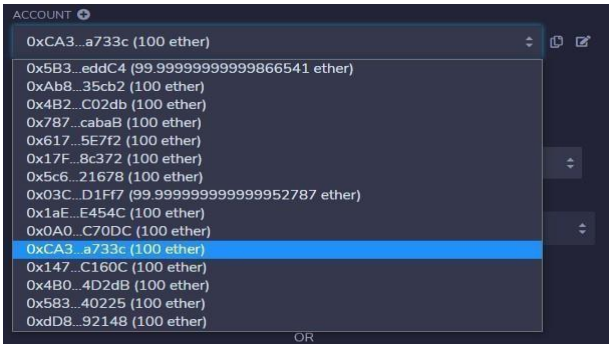
Step1-> Clickonownertocreateanownerobject



Step2-> ClickonlastOwnerChangebutton

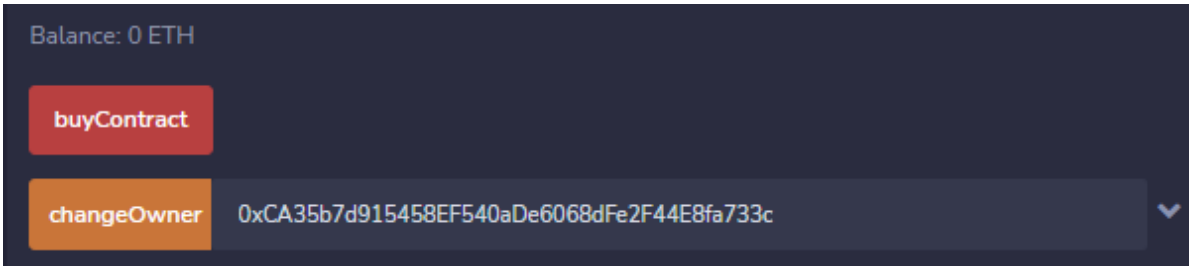
Step3-> ChangetheaddressoftheaccountfromAccountdropdowninDeploytabofRemix IDE.





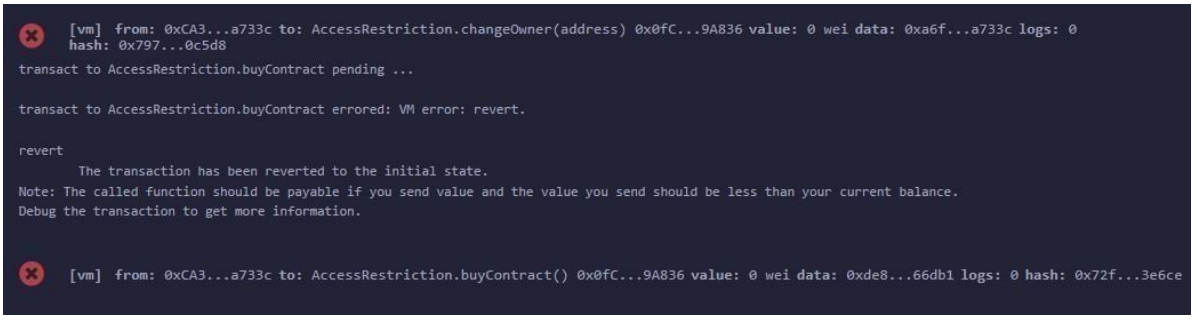
Step4-> Copytheaddress

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{stringm
    essage="Hello";

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

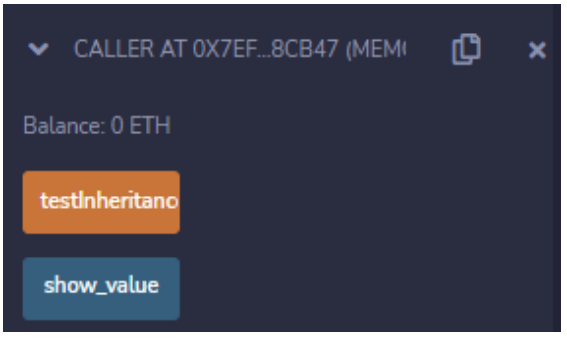
Output

```
Balance: 0 ETH

dispMsg

0: string: Hello
```

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){returnc
        c.getValue();
    }
}
```

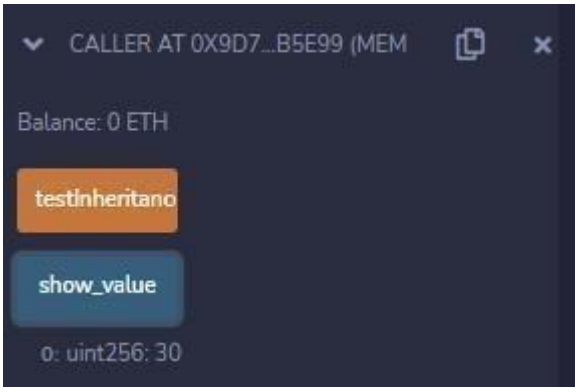
Output:



Flowofexecution

Step1-> SelectcallercontracttodeployinContractanddeploy

Step2-> ClicktestInheritanceandthenclickonshow_valuetoviewvalue



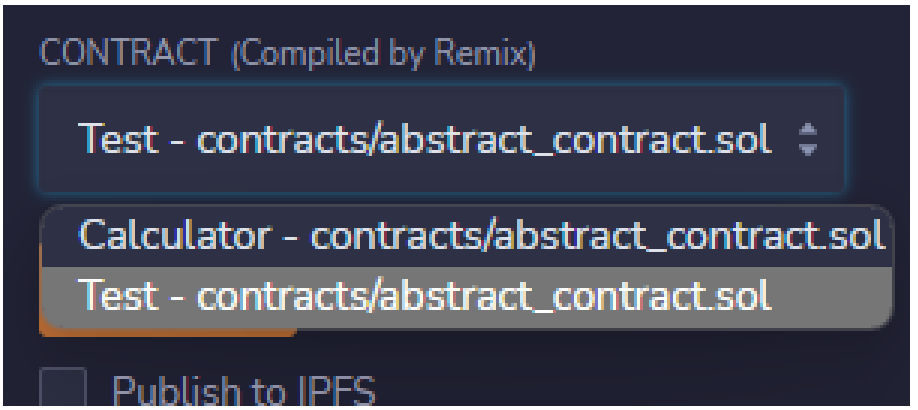
3) AbstractContracts

```
//SPDX-License-
Identifier:MITpragma
solidity^0.5.17;

contractCalculator{
    functiongetResult()externalviewreturns(uint256);
}

contractTestisCalculator{constr
    uctor()public{}

    functiongetResult()externalviewreturns(uint256){uint256a=1
        ;
        uint256b=2;
        uint256result=a+b;retur
        nresult;
    }
}
```

Outputs:

Flowofexecution

Step1-> SelectTestcontractanddeploy



Step2-> Thecontactwilldeployasbelow

Step3-> ClickongetResulttogetsumofa+b

1) Constructors

```
//SPDX-License-
Identifier:MITpragma
solidity^0.5.0;

// Creating a
contractcontractconstructorEx
ample{
    stringstr;

    constructor()public{
        str="GeeksForGeeks";
    }

    functiongetValue()publicviewreturns(stringmemory){returnst
        r;
    }
```

Outputs

Flowofexecution

Step1-> ClickongetValuetoprintstrin

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-> ClickongetResulttodisplaysum

C) Libraries,Assembly,Events,Errorhandling.

1) Libraries

myLib.sol Code

```
//SPDX-License-
Identifier:MITpragmasolidity>=0
.7.0<0.9.0;

librarymyMathLib{
    functionsum(uint256a,uint256b)publicpurereturns(uint256){returna+b;
    }

    functionexponent(uint256a,uint256b)publicpurereturns(uint256){returna**b
        ;
    }
```

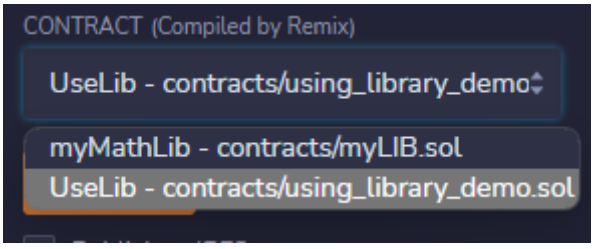
using_library.solCode

```
//SPDX-License-
Identifier:MITpragmasolidity>=0
.7.0<0.9.0;

import"contracts/myLIB.sol";con

tractUseLib{
    functiongetsum(uint256x,uint256y)publicpurereturns(uint256){
        returnmyMathLib.sum(x,y);
    }

    functiongetexponent(uint256x,uint256y)publicpurereturns(uint256){returnmyMathLi
        b.exponent(x,y);
    }
}
```

Outputs:

Flowofexecution

Step1-> ChangecontracttoUseLibanddeploy.



Step2-> Thedeployedcontractshouldbesameasbelow

Step3-> Inputvaluestobothgetexponentandgetsumfunctionsasbelow

Step4-> Executebothfunctions.Youwillgetbelowoutpu

2) Assembly

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

contractInlineAssembly{
    //Definingfunction
    functionadd(uint256a)publicviewreturns(uint256b){assembly{
            letc:=add(a,16)msto
            re(0x80,c)
            {
                letd:=add(sload(c),12)b :=
                d
            }

            b:=add(b,c)
        }
    }
}
```

Outputs

Flowofexecution

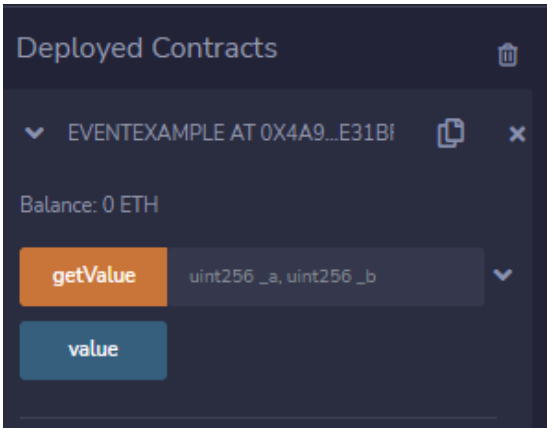
Step1-> Inputanumberforaddfunction

Step2-> Clickaddtooutputsum

3) Events

```
//SPDX-License-
Identifier:MITpragma
solidity^0.5.0;

// Creating a
contractcontracteventExam
ple{
    //Declaringstatevariablesuint25
    6publicvalue=0;

    //Declaringanevent
    eventIncrement(addressowner);

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

Output



Flowofexecution

Step1-> ProvidevaluestogetValuefunctionandclickonit.

Step2-> Intheterminalcheckforlogs


```
logs                    [
                            {
                                "from": "0x4a9C121080f6D9250Fc0143f41B595fD172E31bf",
                                "topic": "0xfc3a67c9f0b5967ae4041ed898b05ec1fa49d2a3c22336247201d71be6f97120",
                                "event": "Increment",
                                "args": {
                                    "0": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                    "owner": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4"
                                }
                            }
```

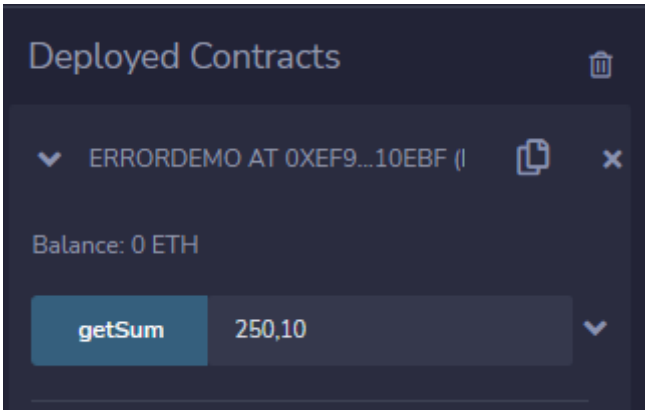

4) ErrorHandling

```
//SPDX-License-Identifier:MITpragma
solidity^0.5.17;

contractErrorDemo{
    functiongetSum(uint256a,uint256b)publicpurereturns(uint256){uint256sum=a+b;
        //require(sum<255,"Invalid");assert(sum<255);
        returnsum;
    }
}
```

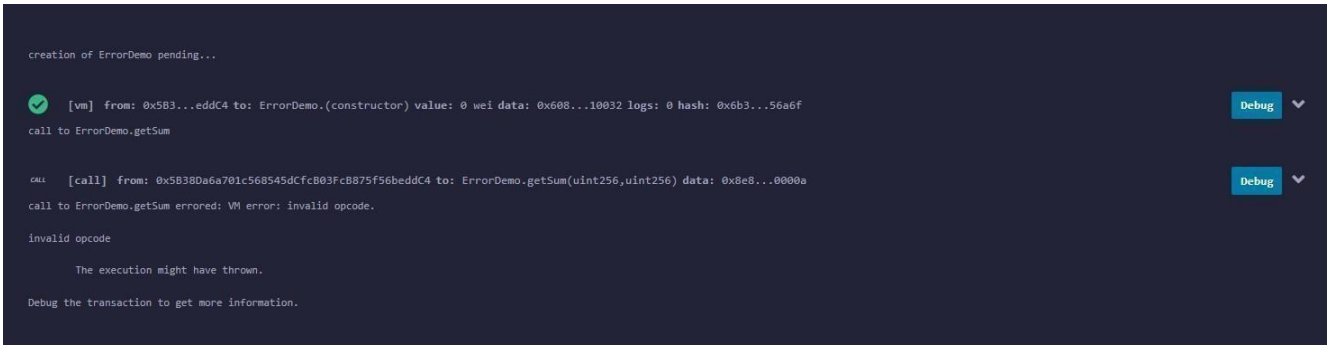
Output

Flowofexecution

Step1-> Providesomevaluesandpressongetsum



Step2-> Checkterminalpanel



PRACTICAL-5

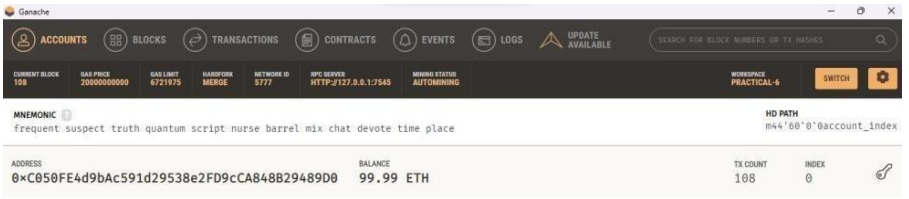
Aim:Writeaprogramtodemonstrateminingofether

```
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!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
Mined a new block!
```



PRACTICAL-6

Aim:Demonstratetherunningoftheblockchainnode

Step1->Create a folder named ethermine and a JSON file named genesis.jsonandwritethefollowinglines in it.

```
{
    "config":{
        "chainId":3792,
        "homesteadBlock":0,
        "eip150Block":0,
        "eip155Block":0,
        "eip158Block":0
    },
    "difficulty":"2000",
    "gasLimit":"2100000","alloc":
    {
        "0×0b6C4c81f58B8d692A7B46AD1e16a1147c25299F": {"balance":
            "9000000000000000000"
        }
    }
}
```

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" : "9000000000000000000"
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:   0x77CB2BdBC0f1743bC73E92f1a8b1AB80BEDB35AE
Path of the secret key file: C:\Users\Achsah\Documents\MScIT\sem4\blockchain_practical\ethermine\key
store\UTC--2023-04-20T14-33-26.959134300Z--77cb2bdbc0f1743bc73e92f1a8b1ab80bedb35ae

- 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 i
nit 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 i
nit 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->Runcommand**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 willenablegeth 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-> Runthecommand**miner.start()**tostartmining

```
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 unc
=0 fees=0 elapsed=7.571ms
INFO [04-20|20:34:45.686] Commit new sealing work                number=1 sealhash=2e6f57..6db9c6 uncle
 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-> Belowscreenshotsaretheminingprocessesrunningonyourlocalmachine.

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

Step8-> TostoptheminingpressCtrl+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           block=16 hash=f09f60..c23237 root
=0c083a..cddeff
INFO [04-20|20:39:22.081] Persisted trie from memory database    nodes=3 size=408.00B time=1.574lm
s gcnodes=0 gcsize=0.00B gctime=0s livenodes=31 livesize=3.83KiB
INFO [04-20|20:39:22.087] Writing cached state to disk           block=15 hash=d73b6d..f4a2cf root
=903c8d..6038c0
INFO [04-20|20:39:22.089] Persisted trie from memory database    nodes=2 size=262.00B time=0s
  gcnodes=0 gcsize=0.00B gctime=0s livenodes=29 livesize=3.58KiB
INFO [04-20|20:39:22.098] Writing snapshot state to disk         root=d56154..abe42a
INFO [04-20|20:39:22.130] Persisted trie from memory database    nodes=0 size=0.00B   time=0s
  gcnodes=0 gcsize=0.00B gctime=0s livenodes=29 livesize=3.58KiB
INFO [04-20|20:39:22.135] Writing clean trie cache to disk       path=C:\Users\Achsah\Documents\MS
cIT\sem4\blockchain_practical\ethermine\geth\triecache threads=4
INFO [04-20|20:39:22.323] Persisted the clean trie cache         path=C:\Users\Achsah\Documents\MS
cIT\sem4\blockchain_practical\ethermine\geth\triecache elapsed=143.729ms
INFO [04-20|20:39:22.490] Blockchain stopped
```

PRACTICAL-7

Aim:Createyourownblockchainanddemonstrateitsuse

Createajavascriptfolderwiththefollowingcodeinanyfolderofyourchoice.

JavaScript Code

```
constSHA256=require("crypto-js/sha256");classBlock{
    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();
    }
}

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

    createGenesisBlock(){
        returnnewBlock(0,"21/04/2023","GenesisBlock","0");
    }

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

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

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

    isChainValid(){
        for(leti=1;i<this.chain.length;i+){constcurrentBlock= this.chain[i];
            constpreviousBlock=this.chain[i-1];

            if(currentBlock.hash                    currentBlock.calculateHash()){returnfalse;
            }

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

        returntrue;
    }
}
```

BlockchainImplementation

```
letmyCoin=newBlockchain();
myCoin.addBlock(newBlock(1,"22/04/2023",{amount:4}));myCoin.addBlock(newBlock(2,"22/04/2023",
{amount:8}));
  console.log('Isblockchainvalid?'+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->Weneed**crypto–js**nodemoduletomakeourownblockchain.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 in
npm WARN @react-native-community/geolocation@2.0.2 requires a peer of react-native@* but non
elf.
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
```

```
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": "eb78a02763c37cfc2b1c4e331df64ca34733e47e017ef320d92ae89b148de5a3"
        },
        {
            "index": 2,
            "timestamp": "22/04/2023",
            "data": {
                "amount": 8
            },
            "previousHash": "eb78a02763c37cfc2b1c4e331df64ca34733e47e017ef320d92ae89b148de5a3",
            "hash": "946b1f95d7761daee4f0c5d33a671c003ef5682333fd9a2d182a73104e9aea88"
        }
    ]
}
```

Step3-> Runtheabovecodeincommandlineusingcomman