

Name: Jash Mavani

Roll No: 19BCE123

Batch: EL2

Course Name & Course Code: 2CSDE93 & Blockchain Technology

Practical-2

Aim: - To create a blockchain and implement replay attacks on blockchain

```
In [1]: import os
import time
from hashlib import sha256
import pandas as pd
```

Block Class

```
In [2]: class Block:
def __init__(self, index, timestamp, data, prevhash):
    self.index = str(index)
    self.timestamp = str(timestamp)
    self.data = str(data)
    self.prevhash = str(prevhash)

def computehash(self):
    self.hash = sha256((self.index + '\t' + self.data + '\t' + self.timestamp + '\t' + self.prevhash).encode())

def __str__(self) -> str:
    return f' HASH --- {self.hash}, DATA --- {self.data}, TIMESTAMP --- {self.timestamp}, PREVHASH --- {self.prevhash}'

block = Block(0, time.time(), 'hello', None)
```

Blockchain Class

```
In [3]: class Blockchain:
def __init__(self):
    self.pointer = Block(0, time.time(), "GENESIS", None)
    self.pointer.computehash()

def add(self, block):
    block.prevhash = str(self.pointer.hash)
    block.computehash()
    self.pointer = block
```

Initializing Chain

```
In [4]:
```

```
blockChain = BlockChain()
```

Adding Blocks

```
In [5]: blocks = [Block(1, time.time(), '1', None),
                  Block(2, time.time(), '2', None),
                  Block(3, time.time(), '3', None),
                  Block(4, time.time(), '4', None),
                  Block(5, time.time(), '5', None),
                  Block(6, time.time(), '6', None),
                  Block(7, time.time(), '7', None),
                  Block(8, time.time(), '8', None),
                  Block(9, time.time(), '9', None),
                  Block(10, time.time(), '10', None),
                  ]
values = []
```

```
In [6]: def validate(blocks):
        print('validating')
        last = len(blocks) - 1
        for i in range(last-1, 0, -1):
            print(blocks[i].hash, '--', blocks[last].prevhash)
            print(last, i)
            if (hash(blocks[i].hash) ^ hash(blocks[last].prevhash))!=0:
                return False
            last -= 1
        return True
```

```
In [7]: for i in blocks:
        blockChain.add(i)
        values.append([i.index, i.data, str(i.hash), i.prevhash])
        validate(blocks)
```

validating

```
b"oT4\x17\xf5\xc0\x03\n=\xf8\xa2G\xcdD\xd5\xc5\x88\x1dH'\xe7P\x1fBJ\xee\xa8\x11m\x13
\x8a\x80" -- b"oT4\x17\xf5\xc0\x03\n=\xf8\xa2G\xcdD\xd5\xc5\x88\x1dH'\xe7P\x1fBJ\xee
\xa8\x11m\x13\x8a\x80"
```

9 8

```
b'j\xaa"\xb1\xb4c^\xb5\xfa\x8b`B\x069F\x07q6\x0b\xb6\xfe\xd4\xb3#\x015\x89^\xf8\xb5
\xd3)' -- b'j\xaa"\xb1\xb4c^\xb5\xfa\x8b`B\x069F\x07q6\x0b\xb6\xfe\xd4\xb3#\x015\x89
^\xf8\xb5\xd3)'
```

8 7

```
b'x\xe9\x1aW\xfe\x9ak\x8bw\xee|\x0e\x93\x8ah\xc5_\xfa\xcd\xad\x82\x95\xb8J\xfc\xc4\x
f3\n\xcc\x1a\x01\xf3' -- b'x\xe9\x1aW\xfe\x9ak\x8bw\xee|\x0e\x93\x8ah\xc5_\xfa\xcd\x
ad\x82\x95\xb8J\xfc\xc4\xf3\n\xcc\x1a\x01\xf3'
```

7 6

```
b'\xc4U\xdcNy0\xa0\xfbWj\xb6^\x1c\x07:\xcb> h\xd5&\xd3\xc3\x9a\xcf\xba\xb2\x16\xf3\x
95\xe3&' -- b'\xc4U\xdcNy0\xa0\xfbWj\xb6^\x1c\x07:\xcb> h\xd5&\xd3\xc3\x9a\xcf\xba\x
b2\x16\xf3\x95\xe3&'
```

6 5

```
b'\t\x0c\xe5_\xf03R\xda\x83\xac\xce\x90|I\x8b\xd0\xc8\xc1\x8e}\xde\xa2\xbf\x96Q|Z\x
f9\xb4M\xed^' -- b'\t\x0c\xe5_\xf03R\xda\x83\xac\xce\x90|I\x8b\xd0\xc8\xc1\x8e}\xde\x
a2\xbf\x96Q|Z\xf9\xb4M\xed^'
```

5 4

```
b'\x94Ss\x00n\x89\x8a\x02$\xf1\xdf#MF\x12\xe1w\xc9\xa8Y\x8e\x03j\xae\xc6\xcd0\xb08N
\x1d\xf4' -- b'\x94Ss\x00n\x89\x8a\x02$\xf1\xdf#MF\x12\xe1w\xc9\xa8Y\x8e\x03j\xae\x
c6\xcd0\xb08N\x1d\xf4'
```

4 3

```

b'\xb0ZO\x83\xdc\x9f9i\xa0bVz:q5#\xe2Y\x12\xe1\xb1\x15\x89&\x8c)\x85`"\xc6\xc0]\r' --
b'\xb0ZO\x83\xdc\x9f9i\xa0bVz:q5#\xe2Y\x12\xe1\xb1\x15\x89&\x8c)\x85`"\xc6\xc0]\r'
3 2
b'g\x94U*\xdbD\xb4\xbe\x17D\x13\xb0\xe8\xa0\x07\x82\xf1\xc8I\xf7\x8c\x178\xc5U\xcf\x
04~\xd0\xe6\xc8\x1a' -- b'g\x94U*\xdbD\xb4\xbe\x17D\x13\xb0\xe8\xa0\x07\x82\xf1\xc8I
\xf7\x8c\x178\xc5U\xcf\x04~\xd0\xe6\xc8\x1a'
2 1
True

```

Out[7]:

Creating dataframe for easy visualization

In [8]:

```
df = pd.DataFrame(values, columns=['Index', 'Data', 'Current Hash', 'Prev Hash'])
```

In [9]:

```
df
```

Out[9]:

	Index	Data	Current Hash
0	1	1	b"\xfe\xdd\xabB\xbazkm"\xc4\x9eK\xf2\xf9%\xa1\...
1	2	2	b'g\x94U*\xdbD\xb4\xbe\x17D\x13\xb0\xe8\xa0\x0...
2	3	3	b'\xb0ZO\x83\xdc\x9f9i\xa0bVz:q5#\xe2Y\x12\xe1\...
3	4	4	b'\x94Ss\x00n\x89\x8a\x02\$\xf1\xdf#MF\x12\xe1w...
4	5	5	b'\t\x0c\xe5_\xf03R\xda\x83\xac\xce\x90 I\x8b\...
5	6	6	b'\xc4U\xdcNy0\xa0\xfbWj\xb6^\x1c\x07:\xcb> h\...
6	7	7	b'x\xe9\x1aW\xfe\x9ak\x8bw\xee \x0e\x93\x8ah\x...
7	8	8	b'j\xaa"\xb1\xb4c^\xb5\xfa\x8b`B\x069F\x07q6\x...
8	9	9	b"oT4\x17\xf5\xc0\x03\n=\xf8\xa2G\xcd\xd5\xc5...
9	10	10	b'\x1fZs\x12fe\x05\xdb\xed\xa0N\x8c8^\xcc\xd8\...

In []: