FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING
Department of Computer Engineering

1. **Course , Subject & Experiment Details**

Academic Year	2022-23	Estimated Time	02 - Hours
Course & Semester	B.E. (CMPN)- Sem VII	Subject Name & Code	BCT - (CSDC7022)
Chapter No.	05	Chapter Title	Private Blockchain

Practical No:	8
Title:	Implementation of PAXOS Algorithm
Date of Performance:	03/10/2022
Date of Submission:	10/10/2022
Roll No:	8953
Name of the Student:	Brendan Lucas

Evaluation:

Sr. No	Rubric	Grade
1	On time submission Or completion (2)	
2	Preparedness(2)	
3	Skill (4)	
4	Output (2)	

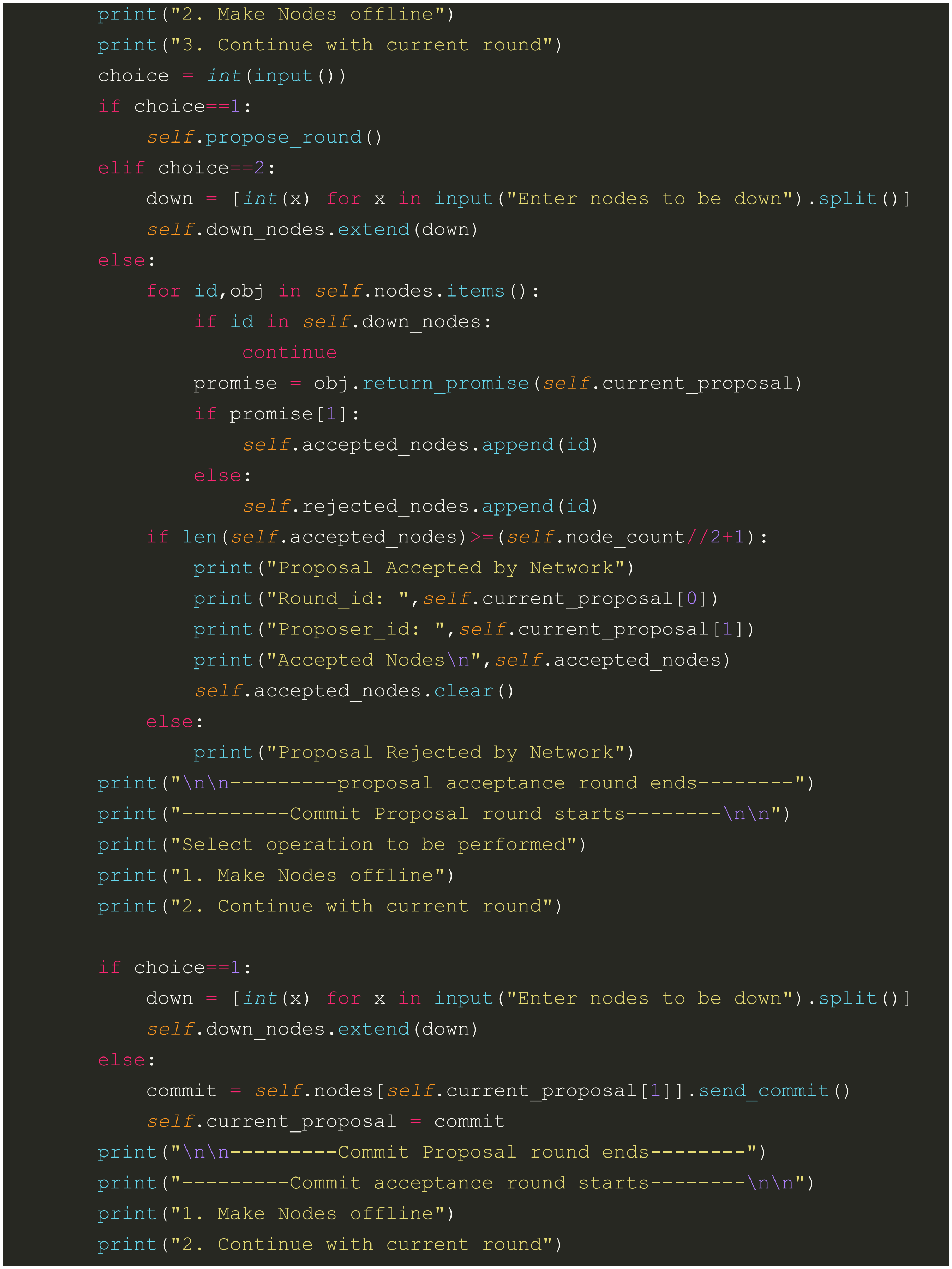
Signature of the Teacher:

Date:

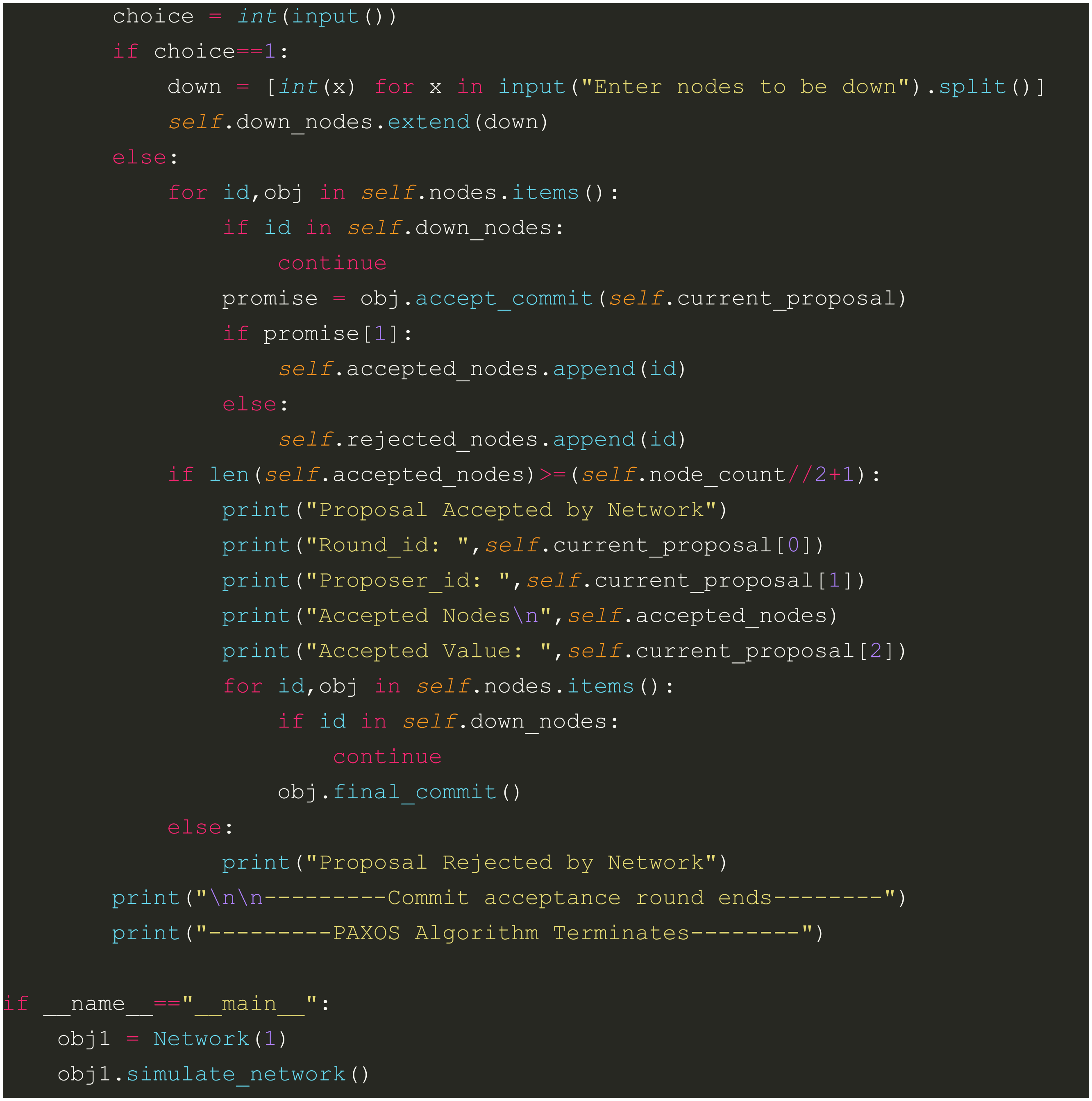
Code:

```
class Node:
    def __init__(self,id):
        self.id=id
        self.status=0
        self.proposed_round=0
        self.proposed_proposer_id=0
        self.proposed_value=0
        self.round=0
        self.proposer_id=0
        self.value=0

    def send_proposal(self):
        key = int(input("Enter Key for proposal: "))
        self.status = 1
        self.proposed_round = key
        self.proposed_proposer_id = self.id
        return (key,self.id)

    def return_promise(self,proposal):
        if self.status==0:
            if proposal[0]>self.proposed_round:
                self.proposed_key=proposal[0]
                self.proposed_proposer_id = proposal[1]
                return [self.id,True,proposal[0],proposal[1]]
            else:
                return [self.id,False,proposal[0],proposal[1]]
        else:
            return [self.id,True,proposal[0],proposal[1]]

    def send_commit(self):
        value = input("Enter the value to be comitted: ")
        self.proposed_value=value
        return (self.proposed_round,self.id,self.proposed_value)

    def accept_commit(self,proposal):
        if self.proposed_round==proposal[0]:
            self.proposed_value = proposal[2]
            return [self.id,True,proposal[0],proposal[1],proposal[2]]
        elif proposal[0]>self.proposed_round:
            return
```

```
[self.id,True,proposal[0],self.proposed_round,self.proposed_proposer_id,self.p
roposed_value]

    def final_commit(self):
        self.value = self.proposed_value
        self.round = self.proposed_round
        self.proposer_id = self.proposed_proposer_id

class Network:
    def __init__(self,ids):
        self.network_id = ids
        self.node_count=int(input("Enter the Number of Nodes: "))
        self.nodes={}
        self.current_proposal = []
        for i in range(1,self.node_count+1):
            self.nodes[i]=Node(i)

        self.down_nodes = []
        self.accepted_nodes=[]
        self.rejected_nodes=[]

    def propose_round(self):
        node_id=int(input("Enter the node ID for proposer: "))
        pro_node = self.nodes[node_id]
        self.current_proposal = pro_node.send_proposal()

    def simulate_network(self):
        print("Select operation to be performed")
        print("1. Select Proposer")
        print("2. Make Nodes offline")
        choice = int(input())
        if choice==1:
            self.propose_round()
        else:
            down = [int(x) for x in input("Enter nodes to be down").split()]
            self.down_nodes.extend(down)

        print("\n\n---------proposal round done--------")
        print("---------proposal acceptance round starts--------\n\n")
        print("Select operation to be performed")
        print("1. Select Proposer")
```



```
        print("2. Make Nodes offline")
        print("3. Continue with current round")
        choice = int(input())
        if choice==1:
            self.propose_round()
        elif choice==2:
            down = [int(x) for x in input("Enter nodes to be down").split()]
            self.down_nodes.extend(down)
        else:
            for id,obj in self.nodes.items():
                if id in self.down_nodes:
                    continue
                promise = obj.return_promise(self.current_proposal)
                if promise[1]:
                    self.accepted_nodes.append(id)
                else:
                    self.rejected_nodes.append(id)
            if len(self.accepted_nodes)>=(self.node_count//2+1):
                print("Proposal Accepted by Network")
                print("Round_id: ",self.current_proposal[0])
                print("Proposer_id: ",self.current_proposal[1])
                print("Accepted Nodes\n",self.accepted_nodes)
                self.accepted_nodes.clear()
            else:
                print("Proposal Rejected by Network")
        print("\n\n---------proposal acceptance round ends--------")
        print("---------Commit Proposal round starts--------\n\n")
        print("Select operation to be performed")
        print("1. Make Nodes offline")
        print("2. Continue with current round")

        if choice==1:
            down = [int(x) for x in input("Enter nodes to be down").split()]
            self.down_nodes.extend(down)
        else:
            commit = self.nodes[self.current_proposal[1]].send_commit()
            self.current_proposal = commit
        print("\n\n---------Commit Proposal round ends--------")
        print("---------Commit acceptance round starts--------\n\n")
        print("1. Make Nodes offline")
        print("2. Continue with current round")
```



```python
        choice = int(input())
        if choice==1:
            down = [int(x) for x in input("Enter nodes to be down").split()]
            self.down_nodes.extend(down)
        else:
            for id,obj in self.nodes.items():
                if id in self.down_nodes:
                    continue
                promise = obj.accept_commit(self.current_proposal)
                if promise[1]:
                    self.accepted_nodes.append(id)
                else:
                    self.rejected_nodes.append(id)
            if len(self.accepted_nodes)>=(self.node_count//2+1):
                print("Proposal Accepted by Network")
                print("Round_id: ",self.current_proposal[0])
                print("Proposer_id: ",self.current_proposal[1])
                print("Accepted Nodes\n",self.accepted_nodes)
                print("Accepted Value: ",self.current_proposal[2])
                for id,obj in self.nodes.items():
                    if id in self.down_nodes:
                        continue
                    obj.final_commit()
            else:
                print("Proposal Rejected by Network")
        print("\n\n---------Commit acceptance round ends--------")
        print("---------PAXOS Algorithm Terminates--------")

if __name__=="__main__":
    obj1 = Network(1)
    obj1.simulate_network()
```

Output:

```
Enter the Number of Nodes: 16
Select operation to be performed
1. Select Proposer
2. Make Nodes offline
1
Enter the node ID for proposer: 6
Enter Key for proposal: 167


---------proposal round done--------
---------proposal acceptance round starts--------


Select operation to be performed
1. Select Proposer
2. Make Nodes offline
3. Continue with current round
3
Proposal Accepted by Network
Round_id:  167
Proposer_id:  6
Accepted Nodes
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]


---------proposal acceptance round ends--------
---------Commit Proposal round starts--------


Select operation to be performed
1. Make Nodes offline
2. Continue with current round
Enter the value to be comitted: Sunday is a holiday


---------Commit Proposal round ends--------
---------Commit acceptance round starts--------


1. Make Nodes offline
2. Continue with current round
2
Proposal Accepted by Network
Round_id:  167
Proposer_id:  6
Accepted Nodes
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
Accepted Value:  Sunday is a holiday


---------Commit acceptance round ends--------
---------PAXOS Algorithm Terminates--------
```