Name:Rhea Adhikari
DS Lab 7
Mutual Exclusion(Election Algorithm)

1)Bully Algorithm
(assumed nodes from 1 to N)

```
import sys
noOfNodes = int(sys.argv[1])
initiatorNode = int(sys.argv[2])


def bully_algorithm():
    print '----Bully Algorithm----'
    print 'Node %s notices the current coordinator %s has failed' \
        % (initiatorNode, noOfNodes)
    if initiatorNode >= noOfNodes:
        print 'INVALID'
    else:
        biggerNodes = []
        if initiatorNode == noOfNodes - 1:

            # no need of election

            biggerNodes.append(initiatorNode + 1)
        else:
            for i in range(initiatorNode + 1, noOfNodes + 1):
                print '%s sends ELECTION message to %s' \
                    % (initiatorNode, i)
                biggerNodes.append(i)
            for i in biggerNodes:
                print '%s sends OK message to %s' % (i, initiatorNode)
            while len(biggerNodes) > 1:
                i = biggerNodes[0]
                for j in range(i + 1, noOfNodes + 1):
                    print '%s sends ELECTION message to %s' % (i, j)
                for k in range(i + 1, noOfNodes + 1):
                    print '%s sends OK message to %s' % (k, i)
                biggerNodes.remove(i)

        newCoordinatorNode = biggerNodes[0]

        for i in range(1, newCoordinatorNode):
            print '%s sends COORDINATOR message to %s' \
                % (newCoordinatorNode, i)


if __name__ == '__main__':
    bully_algorithm()
```

```
student@dslab-12:~/Documents/190905156_DS/Lab7$ python3 q1.py 6 3
----Bully Algorithm----
Node 3 notices the current coordinator 6 has failed
3 sends ELECTION message to 4
3 sends ELECTION message to 5
3 sends ELECTION message to 6
4 sends OK message to 3
5 sends OK message to 3
6 sends OK message to 3
4 sends ELECTION message to 5
4 sends ELECTION message to 6
5 sends OK message to 4
6 sends OK message to 4
5 sends ELECTION message to 6
6 sends OK message to 5
6 sends COORDINATOR message to 1
6 sends COORDINATOR message to 2
6 sends COORDINATOR message to 3
6 sends COORDINATOR message to 4
6 sends COORDINATOR message to 5
student@dslab-12:~/Documents/190905156_DS/Lab7$ python3 q1.py 6 5
----Bully Algorithm----
Node 5 notices the current coordinator 6 has failed
6 sends COORDINATOR message to 1
6 sends COORDINATOR message to 2
6 sends COORDINATOR message to 3
6 sends COORDINATOR message to 4
6 sends COORDINATOR message to 5
student@dslab-12:~/Documents/190905156_DS/Lab7$
```

2)Ring Algorithm
(assumed nodes from 1 to N-1)

```
import sys
noOfNodes = int(sys.argv[1])
initiatorNode = int(sys.argv[2])


def ring_algorithm():
    print 'Node: ' + str(initiatorNode) + ' notices the coordinator: ' \
        + str(noOfNodes) + ' has failed'
    biggerNodes = []
    init = initiatorNode
    biggerNodes.append(init)
    print biggerNodes
    for i in range(0, noOfNodes):
        biggerNodes.append((i + init) % noOfNodes)
        print str((i + init) % noOfNodes) \
            + ' sent ELECTION message to: ' + str((i + 1 + init)
                % noOfNodes)
        largest = biggerNodes[0]
    for i in biggerNodes:
        if largest < biggerNodes[i]:
            largest = biggerNodes[i]
```

```
    for i in range(0, noOfNodes):
        if i != initiatorNode:
            print str(initiatorNode) + ' sends COORDINATOR message to ' \
                + str(i) + ' that ' + str(largest) \
                + ' is the coordinator node'


if __name__ == '__main__':
    ring_algorithm()
```

```
student@dslab-12:~/Documents/190905156_DS/Lab7$ python3 ring.py 5 3
Node: 3 notices the coordinator: 5 has failed
[3]
3 sent ELECTION message to: 4
4 sent ELECTION message to: 0
0 sent ELECTION message to: 1
1 sent ELECTION message to: 2
2 sent ELECTION message to: 3
3 sends COORDINATOR message to 0 that 4 is the coordinator node
3 sends COORDINATOR message to 1 that 4 is the coordinator node
3 sends COORDINATOR message to 2 that 4 is the coordinator node
3 sends COORDINATOR message to 4 that 4 is the coordinator node
student@dslab-12:~/Documents/190905156_DS/Lab7$
0 ⚠ 0
```