Task #1: Use Logical Clocks to Implement Mutual Exclusion

This task was not completed.

Task #2: Implementing the Bully Algorithm for Leader Election

a. The result of the sample run showed the results of the election process; more specifically, the election messages that were sent were shown after the status and process ID of each process was inputted into the command line.

b. There were no difficulties encountered during the implementation of this task. But I did learn the steps to take for implementing the algorithm.

c. The Ricart-Agrawala algorithm is simply an extension and optimization of Lamport's Mutual Exclusion Algorithm by removing the need for *"acknowledgement"* messages.

```
Enter the number of processes: 8
For process 1...
Status (1 for alive, 0 for dead): 1
Process id (1, 2, 3, .., n): 2
For process 2...
Status (1 for alive, 0 for dead): 0
Process id (1, 2, 3, .., n): 5
For process 3...
Status (1 for alive, 0 for dead): 1
Process id (1, 2, 3, .., n): 0
For process 4...
Status (1 for alive, 0 for dead): 1
Process id (1, 2, 3, .., n): 3
For process 5...
Status (1 for alive, 0 for dead): 1
Process id (1, 2, 3, .., n): 1
For process 6...
Status (1 for alive, 0 for dead): 1
Process id (1, 2, 3, .., n): 7
For process 7...
Status (1 for alive, 0 for dead): 0
Process id (1, 2, 3, .., n): 11
For process 8...
Status (1 for alive, 0 for dead): 1
Process id (1, 2, 3, .., n): 9
Which process will initiate election? 4
Election message is sent from 4 to 2
Election message is sent from 4 to 6
6 replies to 4
Election message is sent from 6 to 7
Election message is sent from 6 to 8
8 replies to 6
Election message is sent from 8 to 7
Election message is sent from 4 to 7
Election message is sent from 4 to 8
8 replies to 4
Election message is sent from 8 to 7
Final coordinator is 8
```