**Aim –** To implement Mutual Exclusion Algorithm (Raymond tree algorithm) in java.

**Theory –**

Mutual exclusion is a concurrency control property which is introduced to prevent race conditions. It is the requirement that a process can not enter its critical section while another concurrent process is currently present or executing in its critical section i.e only one process is allowed to execute the critical section at any given instance of time.

In Distributed systems, we neither have shared memory nor a common physical clock and there for we can not solve mutual exclusion problem using shared variables. To eliminate the mutual exclusion problem in distributed system approach based on message passing is used. A site in distributed system do not have complete information of state of the system due to lack of shared memory and a common physical clock.

Requirements of Mutual exclusion Algorithm:

No Deadlock:  
Two or more site should not endlessly wait for any message that will never arrive.

No Starvation:  
Every site who wants to execute critical section should get an opportunity to execute it in finite time.

Fairness:  
Each site should get a fair chance to execute critical section.

Fault Tolerance:  
In case of failure, it should be able to recognize it by itself in order to continue functioning without any disruption.

Solution to distributed mutual exclusion:.

Message passing is a way to implement mutual exclusion. Below are the three approaches based on message passing to implement mutual exclusion in distributed systems:

Token Based- A unique token is shared among all the sites. If a site possesses the unique token, it is allowed to enter its critical section. This approach uses sequence number to order requests for the critical section. Eg- Raymond tree Algorithm

Non Token Based -These types of algorithm use timestamps instead of sequence numbers to determine which sites should execute a critical section next. Sites communicate with each other to exchange timestamps, and multiple rounds of messages are required. When a site requests access to the critical section, it is assigned a timestamp, which is also used to resolve conflicts between multiple requests.

**Conclusion-** Successfully Implemented mutual exclusion algorithm.