**Aim –** To implement Mutual Exclusion Algorithm (Lamport’s algorithm) in java.

**Theory:**

Lamport’s Distributed Mutual Exclusion Algorithm is a permission-based algorithm proposed by Lamport as an illustration of his synchronization scheme for distributed systems. In permission-based timestamp is used to order critical section requests and to resolve any conflict between requests. In Lamport’s Algorithm critical section requests are executed in the increasing order of timestamps i.e a request with smaller timestamp will be given permission to execute critical section first than a request with larger timestamp.

**Algorithm:**

**•To enter Critical section:**

•When a site Si wants to enter the critical section, it sends a request message Request(tsi, i) to all other sites and places the request on request\_queuei. Here, Tsi denotes the timestamp of Site Si

•When a site Sj receives the request message REQUEST(tsi, i) from site Si, it returns a timestamped REPLY message to site Si and places the request of site Si on request\_queuej

**•To execute the critical section:**

•A site Si can enter the critical section if it has received the message with timestamp larger than (tsi, i) from all other sites and its own request is at the top of request\_queuei

**•To release the critical section:**

•When a site Si exits the critical section, it removes its own request from the top of its request queue and sends a timestamped RELEASE message to all other sites

•When a site Sj receives the timestamped RELEASE message from site Si, it removes the request of Si from its request queue

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