**U19EC046 | OS | LAB 8**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**//semaphore.java**

**import java.time.LocalDateTime;**

**import java.time.format.DateTimeFormatter;**

**import java.util.concurrent.\*;**

**class MyThread extends Thread {**

**Semaphore s;**

**String threadName;**

**public static final String YELLOW = "\u001B[33m";**

**public static final String RESET = "\u001B[0m";**

**public MyThread(Semaphore s, String threadName) {**

**super(threadName);**

**this.s = s;**

**this.threadName = threadName;**

**}**

**static String timeStamp() {**

**LocalDateTime myDateObj = LocalDateTime.now();**

**DateTimeFormatter myFormatObj = DateTimeFormatter.ofPattern("HH:mm:ss");**

**return "[" + myDateObj.format(myFormatObj) + "] ";**

**}**

**@Override**

**public void run() {**

***// Process P***

**if (this.getName().equals("P")) {**

**try {**

**for (int i = 0; i < 50; i++) {**

***// acquiring the lock***

**System.out.println(timeStamp() + "P trying to aquire");**

**s.acquire(); *// P(S)***

**System.out.println(timeStamp() + "P aquired");**

**System.out.print(YELLOW + timeStamp() + "00\n" + RESET);**

**Thread.sleep(5000);**

**System.out.print(timeStamp() + "P realeased\n");**

**s.release(); *// V(S)***

**Thread.sleep(10);**

**}**

**} catch (InterruptedException exc) {}**

**}**

***// Process Q***

**else {**

**try {**

**Thread.sleep(1000);**

**for (int i = 0; i < 50; i++) {**

**System.out.println(timeStamp() + "Q trying to aquire");**

**s.acquire(); *// P(S)***

**System.out.print(timeStamp() + "Q Aquired\n");**

**System.out.print(YELLOW + timeStamp() + "11\n" + RESET);**

**Thread.sleep(5000);**

**System.out.print(timeStamp() + "Q realeased\n");**

**s.release(); *// V(S)***

**Thread.sleep(10);**

**}**

**} catch (InterruptedException exc) {}**

**}**

**}**

**}**

***// Driver class***

**public class semaphore {**

**public static void main(String args[]) throws InterruptedException {**

**Semaphore s = new Semaphore(1);**

**MyThread P = new MyThread(s, "P");**

**MyThread Q = new MyThread(s, "Q");**

**P.start();**

**Q.start();**

**}**

**}**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**OUTPUT**

