JUNAID GIRKAR
60004190057
SE COMPS A-3

OPERATING SYSTEMS
EXPERIMENT - 6
Code & Output

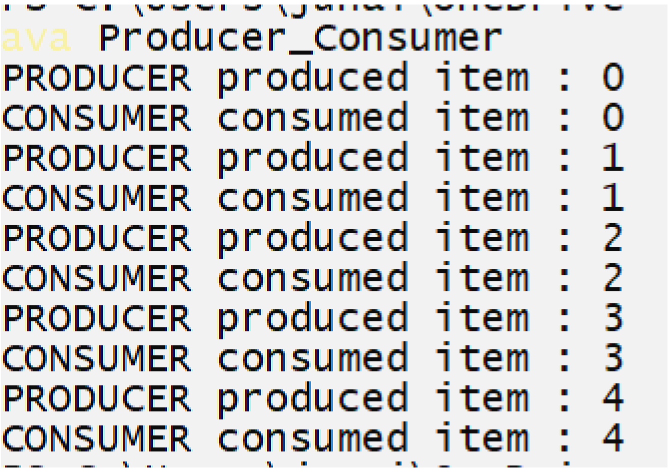
AIM: Solving the Producer Consumer problem using Semaphores

CODE:

```
import java.util.concurrent.Semaphore;

class Q {

    int item;

    // semCon initialized with 0 permits
    // to ensure put() executes first
    static Semaphore semCon = new Semaphore(0);

    static Semaphore semProd = new Semaphore(1);

    // to get an item from buffer
    void get()
    {
        try {
            // Before consumer can consume an item,
            // it must acquire a permit from semCon
            semCon.acquire();
        }
        catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }

        // consumer consuming an item
        System.out.println("CONSUMER consumed item : " + item);

        // After consumer consumes the item,
        // it releases semProd to notify producer
        semProd.release();
    }

    // to put an item in buffer
    void put(int item)
    {
        try {
            // Before producer can produce an item,
            // it must acquire a permit from semProd
            semProd.acquire();
        }
        catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }

        // producer producing an item
        this.item = item;

        System.out.println("PRODUCER produced item : " + item);

        // After producer produces the item,
        // it releases semCon to notify consumer
        semCon.release();
    }
}

// Producer class
class Producer implements Runnable {
    Q q;
    Producer(Q q)
    {
        this.q = q;
        new Thread(this, "PRODUCER").start();
    }

    public void run()
    {
        for (int i = 0; i < 5; i++)
            // producer put items
            q.put(i);
    }
}

// Consumer class
```

```java
class Consumer implements Runnable {
    Q q;
    Consumer(Q q)
    {
        this.q = q;
        new Thread(this, "CONSUMER").start();
    }

    public void run()
    {
        for (int i = 0; i < 5; i++)
            // consumer get items
            q.get();
    }
}

// Driver class
class Producer_Consumer {
    public static void main(String args[])
    {
        // creating buffer queue
        Q q = new Q();

        // starting consumer thread
        new Consumer(q);

        // starting producer thread
        new Producer(q);
    }
}
```

OUTPUT:



```
ava  Producer_Consumer
PRODUCER produced item : 0
CONSUMER consumed item : 0
PRODUCER produced item : 1
CONSUMER consumed item : 1
PRODUCER produced item : 2
CONSUMER consumed item : 2
PRODUCER produced item : 3
CONSUMER consumed item : 3
PRODUCER produced item : 4
CONSUMER consumed item : 4
```