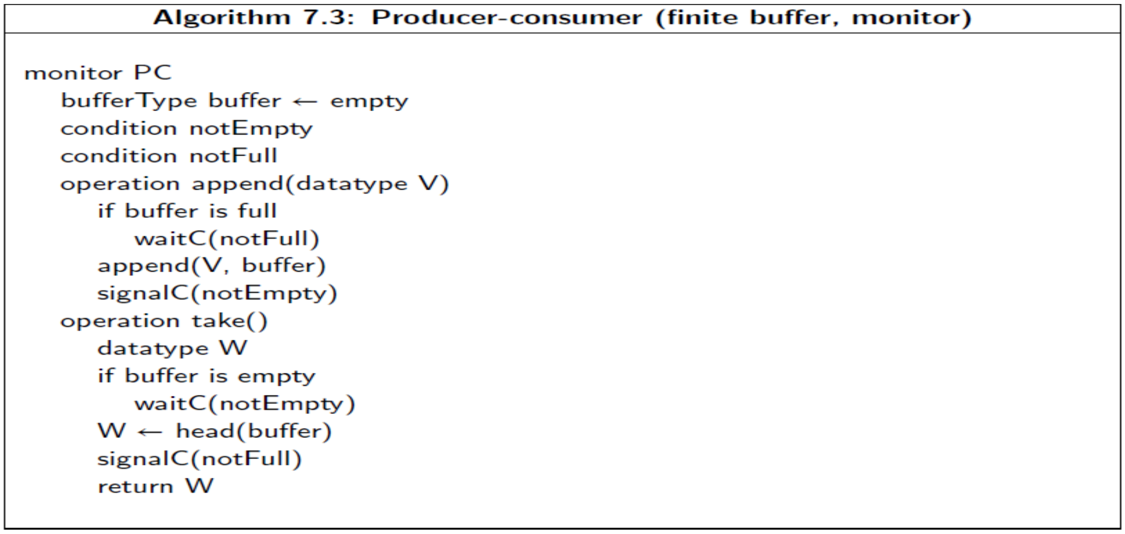
B. TECH - COMPUTER SCIENCE AND ENGINEERING (V SEMESTER)

CST 303Concurrent and Parallel Programming Lab

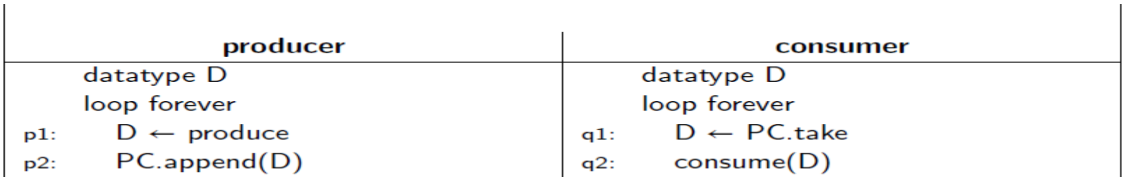
Week: 7

1. Implement the solution of producer-consumer bounded buffer problem with a monitor.



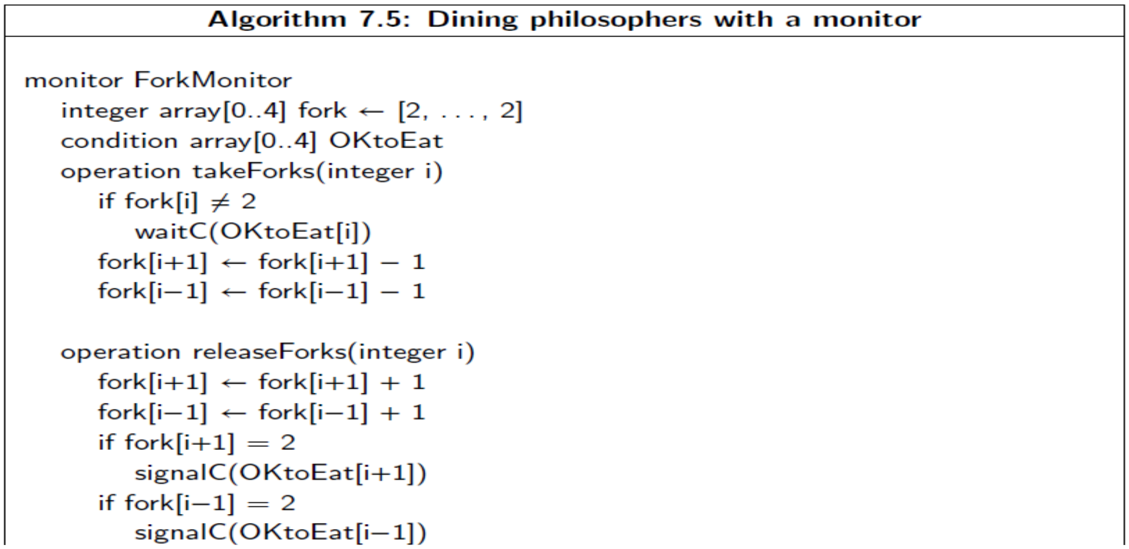
Algorithm 7.3: Producer-consumer (finite buffer, monitor)

```
monitor PC
    bufferType buffer ← empty
    condition notEmpty
    condition notFull
    operation append(datatype V)
        if buffer is full
            waitC(notFull)
        append(V, buffer)
        signalC(notEmpty)
    operation take()
        datatype W
        if buffer is empty
            waitC(notEmpty)
        W ← head(buffer)
        signalC(notFull)
        return W
```



```
              producer                              consumer
        datatype D                            datatype D
        loop forever                          loop forever
p1:         D ← produce               q1:         D ← PC.take
p2:         PC.append(D)              q2:         consume(D)
```

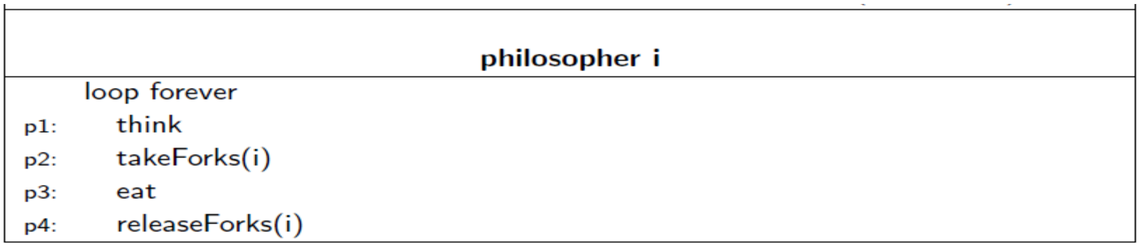
2. Implement the solution of Dining philosophers with a monitor.



Algorithm 7.5: Dining philosophers with a monitor

```
monitor ForkMonitor
    integer array[0..4] fork ← [2, . . . , 2]
    condition array[0..4] OKtoEat
    operation takeForks(integer i)
        if fork[i] ≠ 2
            waitC(OKtoEat[i])
        fork[i+1] ← fork[i+1] − 1
        fork[i−1] ← fork[i−1] − 1

    operation releaseForks(integer i)
        fork[i+1] ← fork[i+1] + 1
        fork[i−1] ← fork[i−1] + 1
        if fork[i+1] = 2
            signalC(OKtoEat[i+1])
        if fork[i−1] = 2
            signalC(OKtoEat[i−1])
```



```
                    philosopher i
    loop forever
p1:     think
p2:     takeForks(i)
p3:     eat
p4:     releaseForks(i)
```

3. Consider a system consisting of processes P1, P2, ..., Pn, each of which has a unique priority number. Write a monitor that allocates three identical line printers to these processes, using the priority numbers for deciding the order of allocation.

4. Develop a simulation of monitors by semaphores.