Semaphore Wait/Down:

```
down(semaphore s){
    s = s-1;
    if(s<0){
        put process(PCB) into
        suspend_list;
        sleep()
    }
    else{
        return;
    }
}
```

Semaphore Signal/Up :

```
up(semaphore s){
    s = s+1;
    if(s<=0){
        select a process
        from suspend_list;
        Wake_up();
    }
    else{
        return;
    }
}
```

Producer Code:

```
down(empty);
down(s);
buffer[in] = item;
in = (in+1)%n;
up(s);
up(full);
```

Consumer Code:

```
down(full);
down(s);
item = buffer[out];
out = (out+1)%n;
up(s);
up(empty);
```

Dining Philosopher:

```
void Philosopher(void){
    while(True){
       thinking();
       down(chopstick[i]);
       down(chopstick[(i+1)%n]);
       eat();
       up(chopstick[i]);
       up(chopstick[(i+1)%n]);
    }
}
```