

## TP 5 Systèmes concurrents et distribués

### Exercice 1 :

Soit l'implémentation d'une file d'attente vue au cours.

- Implémentez un programme de test. Créez plusieurs threads qui se partagent une file d'attente et observez la synchronisation avec des `System.out.println(...)` judicieusement disposés.

```
public class queue {  
    private int head = 0, tail = 0;  
    final int QSIZE=1; Item[QSIZE] items;  
  
    public synchronized void enq(Item x) {  
        while (this.tail { this.head == QSIZE)  
            this.wait();  
        this.items[this.tail++%QSIZE] = x;  
        this.notifyAll();  
    }  
  
    public synchronized Item deq(Item x) {  
        while (this.tail { this.head == 0)  
            this.wait();  
        Item inter = this.items[this.head++%QSIZE];  
        this.notifyAll();  
        return inter;  
    }  
}
```

- Remplacez `notifyAll()` par `notify()`. Combien de threads au minimum il faut pour observer un interblocage ?
- Même question qu'au-dessus mais avec `QSIZE=2`

- Modifiez les implémentations des méthodes *enq()* et *deq()* de telle manière que les threads utilisent uniquement *notify()* mais qu'il n'y ait pas d'interblocage. Il faut utiliser des files d'attentes distinctes pour les threads qui attendent que la file ne soit plus vide et ceux qui attendent que la file ne soit plus pleine.