HW6 solution

Q1:
Consider the following scenario for n=2.

Both p0 and p1 initiate entry at the same time, and they both read Number[0] = Number[1] = 0.

p1 completes the write Number[1] = 1 first, and then p1 enters the critical section before Number[0] is written as 1 by p0.

Then, p0 completes writes Number[0] = 1. Now, Number[1] = Number[0] and the process ids are used to break tie: (Number[1],1) > (Number[0],0) .

Then, p0 enters the critical section. The mutual exclusion property is violated, because p1 is also in the critical section.

Q2:
Following table shows an execution that results in a deadlock.

<table>
<thead>
<tr>
<th>P0 starts</th>
<th>P1 starts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line2</td>
<td>Line3</td>
</tr>
<tr>
<td>Line3</td>
<td>Line3</td>
</tr>
<tr>
<td>Stuck at line6</td>
<td>Line5, go back to line2</td>
</tr>
<tr>
<td></td>
<td>Stuck at line2</td>
</tr>
</tbody>
</table>

Q3:
(a) Eventual Consistency:
    7 possible values: [0, 1, 2, 3, 4, 5, 6]
(b) Read-my-write
    1 possible value: [6]

Q4:
(a) It is not possible. p will conclude q has crashed in bounded amount of time.

(b) It is possible. Say all messages have a delay that is 10 seconds. p would have concluded that q has crashed after t = 1 second