# Quiz 1.1

- What if we reverse the order of the first two lines the 2-process Peterson's algorithm

```
P0:                          P1:
turn = 1;                    turn = 0;
flag[0] = true;              flag[1] = true;
…                            …
```

Would it work?

- Prove that Peterson's N-process algorithm ensures:
  - ✓mutual exclusion: no two processes are in the critical section at a time
  - ✓starvation freedom: every process in the trying section eventually reaches the critical section (assuming no process fails in the trying, critical, or exit sections)

# Quiz 1.2: safety

1. Let S be a safety property. Show that if all finite runs of an implementation I are safe (belong to S) then all runs of I are safe

2. Show that every unsafe run σ has an unsafe finite prefix σ’: every extension of σ’ is unsafe

3. Show that every property is an intersection of a safety property and a liveness property

# Quiz 1.3: liveness

- Show how the elements of the “periodic table of progress” are related to each other

  - ✓Hint: for each pair of properties, A and B, check if any run of A is a run of B (A is stronger than B), or if there exists a run of A that is not in B (A is not stronger than B)
  - ✓Can be shown by transitivity: if A is stronger than B and B is stronger than C, then A is stronger than C

# Quiz 1.4: linearizability

- Show that the sequential queue implementation considered before is linearizable and wait-free *as is* if used by two processes: one performing only enqueue operations and one performing only dequeue operations

- Devise a simple queue implementation shared by any number of processes in which enqueue and dequeue operations can run concurrently (data races between these operations are allowed)