Quiz 1: what if?

Code for process pi:

```
initially:
        shared array R[0,..M-1] of 1WNR registers := [1,0,…,0]

upon read()
        for j = 0 to M-1 do
            if R[j].read() = 1 then return j

upon write(v)  // if i=1
        R[v].write(1)
        for j=0 to v-1 do R[j].write(0)
        return ok
```

Quiz 2: what if?

```
Code for process pi:
    initially:
            shared array R[0,..M-1] of 1WNR registers := [1,0,…,0]

    upon read()
            for j = 0 to M-1 do
                if R[j].read() = 1 then return j

   upon write(v)  // if i=1
            for j=v-1 down to 0 do R[j].write(0)
            R[v].write(1)
            return ok
```

Quiz 3: why not atomic?

- Can we find an execution that is not atomic?
  - ✓"new-old" inversion:
  - ✓R1 precedes R2
  - ✓R1 returns the new value, and R2 returns the old value

Quiz 4: atomic with safe?

- Does 2-process Peterson's lock work if we use regular registers instead of atomic?
- Does Lamport's Bakery algorithm work with safe registers?