Homework Assignment #9
Due: April 4, 2025 at 5:00 p.m.

1. We want to design a concurrent implementation of a high-water register (HWR) for n processes. The HWR stores an integer value (initially 0) and supports the following two operations.

- WRITE(v) changes the value stored in the HWR to v if v is larger than its current value; otherwise it has no effect.
- READ returns the value stored in the HWR.

[3] **(a)** Consider the following implementation that uses a single shared-memory location X.

```
1: WRITE(v)
2: | if v > X then
3: | | X ← v
4: READ : int
5: | return X
```

Explain why this implementation is not linearizable.

[3] **(b)** Consider the following implementation that uses an array $A[1..n]$ of shared-memory locations. Assume the processes are numbered 1 to n and the following code is executed by process i.

```
1: WRITE(v)
2: | if v > A[i] then
3: | | A[i] ← v
4: READ : int
5: | result ← 0                          ▷ result is a local variable of the process
6: | for j ← 1..n do
7: | | result ← max(result, A[j])
8: | return result
```

Explain why this implementation is not linearizable.

[3] **(c)** Give a simple linearizable implementation of an HWR using a snapshot object. You should specify the linearization point of each operation, but you do not have to give a detailed proof of correctness.