Program BB models a “bounded-buffer” of size $N$.
Awaits are weak (i.e., a thread passes \texttt{await (B) S} if B holds continuously).
Parameter $j$ is an integer in $1..N$.

program BB():
  \texttt{N: positive integer}
  num ← 0
  function cAdd(j):
    await (num ≤ N - j)
    num ← num + j
  function cRmv(j):
    await (num ≥ j)
    num ← num - j

1. **[25 points]** Implement program BB (including its progress) using locks and condition variables as the only synchronization constructs. Your answer will consist of
   - Definitions of additional variables (e.g., locks, condition variables).
   - Pseudocode bodies of functions cAdd$(j)$ and cRmv$(j)$. Each function must be less than 12 lines.
2. [15 points] Implement program BB using semaphores as the only synchronization constructs. Your solution must ensure priority for awakened threads, i.e., if a thread is awakened at a gate, it must not get blocked again.

Your answer will consist of

- Definitions of additional variables (e.g., semaphores).
- Brief description of function bodies. No need for pseudocode.