```
──────────────────── MODULE Blockchain ────────────────────
This module defines network initial conditions to be used by the p2p algorithm.

EXTENDS Integers, Sequences, TLC, Utils

Create a network with given number of peers, the blocks and connections to be established.
CreateNetwork(numPeers, blockCounts, connections) ≜
    [peer ∈ 1 . . numPeers ↦
        LET numBlocks ≜ blockCounts[peer]
            lastBlockHash ≜ IF numBlocks > 0
                THEN "blockhash" ∘ ToString(numBlocks)
             ELSE  "blockhash0"
              Construct peer_set as a sequence of other peers, seeder nodes have no connections.
            peerSet ≜ IF connections[peer] = TRUE THEN
                Remove(
                    Add all peers to the list.
                   [i ∈ 1 . . numPeers ↦ [
                     address ↦ "peer" ∘ ToString(i),
                     tip ↦ blockCounts[i],
                     established ↦ FALSE
                   ]],
                    Remove the current peer from the list.
                   [
                       address ↦ "peer" ∘ ToString(peer),
                       tip ↦ blockCounts[peer],
                       established ↦ FALSE
                   ])
             ELSE ⟨⟩
        IN  [
             peer ↦ "peer" ∘ ToString(peer),
             blocks ↦ ToSet([height ∈ 1 . . numBlocks ↦ [
                 height ↦ height,
                 hash ↦ "blockhash" ∘ ToString(height),
                 block ↦ "serialized block data " ∘ ToString(height)
             ]]),
             peer_set ↦ peerSet,
             chain_tip ↦ [height ↦ numBlocks, hash ↦ lastBlockHash]
        ]
    ]
───────────────────────────────────────────────────────────

2 peers network. 1 seeder with 1 block and no outbound connections and 1 peer with no blocks
and an outbound connection to the seeder.
Blockchain1 ≜ CreateNetwork(2, ⟨1, 0⟩, ⟨FALSE, TRUE⟩)
```

2 peers network. 1 seeder with 10 blocks and no outbound connections and 1 peer with no blocks and an outbound connection to the seeder.

$Blockchain2 \triangleq CreateNetwork(2, \langle 10, 0\rangle, \langle \text{FALSE}, \text{TRUE}\rangle)$

3 peers network. 1 seeder with 1 block and no outbound connections and 2 peers with no blocks and an outbound connection to the seeder.

$Blockchain3 \triangleq CreateNetwork(3, \langle 1, 0, 0\rangle, \langle \text{FALSE}, \text{TRUE}, \text{TRUE}\rangle)$

1 peer with or without connections or blocks is an assert

$Blockchain4 \triangleq CreateNetwork(1, \langle 0\rangle, \langle \text{TRUE}\rangle)$

2 or more peers without connections is a deadlock

$Blockchain5 \triangleq CreateNetwork(2, \langle 0, 0\rangle, \langle \text{FALSE}, \text{FALSE}\rangle)$

2 peers network. 2 connected to each other with the same amount of blocks.

$Blockchain6 \triangleq CreateNetwork(2, \langle 100, 100\rangle, \langle \text{TRUE}, \text{TRUE}\rangle)$

2 peers network. 2 connected to each other with different amount of blocks.

$Blockchain7 \triangleq CreateNetwork(2, \langle 2, 1\rangle, \langle \text{TRUE}, \text{TRUE}\rangle)$