Game of Life

For this problem we need to update the board simultaneously based on Conway's rule, preferably in place.

Key Challenge:

- You must update all cells simultaneously, meaning you cannot directly overwrite cells, since future cells depend on original state.
- In-place trick: Encode both the old and new states into one integer:
  - 0 → dead → dead (0)
  - 1 → live → live (1)
  - 1 → live → dead (2) (old live, new dead)
  - 0 → dead → live (3) (old dead, new live)

Why?

- state % 2 gives original state (0 or 1).
- state // 2 or additional logic can decode the new state later.

Rules Recap:

For each cell (i, j):

1. Count live neighbors (original states).
2. Apply rules:
   - Live cell (1):
     - Fewer than 2 or more than 3 neighbors → dies (mark as 2).
     - Else → stays alive
   - Dead cell (0):
     - Exactly 3 neighbors → becomes alive (mark as 3).

In-Place Algorithm:

1. Traverse matrix and count live neighbors (check board[x][y] % 2 to get original state).
2. Apply rules using encoded values (2 and 3).
3. After traversal, update all cells: · If board[i][j] == 2 → 0 (dead)
   · If board[i][j] == 3 → 1 (alive)

Complexity:

- Time: $O(m \times n)$ (each cell visited once, neighbor check constant 8).
- Space: $O(1)$ (in place, no extra storage)

Follow-up (Infinite Board):

- For an infinite board, track only live cells (sparse representation):
  - Use unordered_set<pair<int, int>> or set.
  - Update only cells that are live or neighbors of live cells.