

4. The `LightBoard` class models a two-dimensional display of lights, where each light is either on or off, as represented by a Boolean value. You will implement a constructor to initialize the display and a method to evaluate a light.

```
public class LightBoard
{
    /** The lights on the board, where true represents on and false represents off.
     */
    private boolean[][] lights;

    /** Constructs a LightBoard object having numRows rows and numCols columns.
     *   Precondition: numRows > 0, numCols > 0
     *   Postcondition: each light has a 40% probability of being set to on.
     */
    public LightBoard(int numRows, int numCols)
    { /* to be implemented in part (a) */ }

    /** Evaluates a light in row index row and column index col and returns a status
     *   as described in part (b).
     *   Precondition: row and col are valid indexes in lights.
     */
    public boolean evaluateLight(int row, int col)
    { /* to be implemented in part (b) */ }

    // There may be additional instance variables, constructors, and methods not shown.
}
```

(a) Write the constructor for the `LightBoard` class, which initializes `lights` so that each light is set to on with a 40% probability. The notation `lights[r][c]` represents the array element at row `r` and column `c`.

Complete the `LightBoard` constructor below.

```
/** Constructs a LightBoard object having numRows rows and numCols columns.
 *   Precondition: numRows > 0, numCols > 0
 *   Postcondition: each light has a 40% probability of being set to on.
 */
public LightBoard(int numRows, int numCols)
```

(b) Write the method `evaluateLight`, which computes and returns the status of a light at a given row and column based on the following rules.

- 1. If the light is on, return `false` if the number of lights in its column that are on is even, including the current light.
- 2. If the light is off, return `true` if the number of lights in its column that are on is divisible by three.
- 3. Otherwise, return the light's current status.

For example, suppose that `LightBoard sim = new LightBoard(7, 5)` creates a light board with the initial state shown below, where `true` represents a light that is on and `false` represents a light that is off. Lights that are off are shaded.

|   | 0     | 1     | 2     | 3     | 4     |
|---|-------|-------|-------|-------|-------|
| 0 | true  | true  | false | true  | true  |
| 1 | true  | false | false | true  | false |
| 2 | true  | false | false | true  | true  |
| 3 | true  | false | false | false | true  |
| 4 | true  | false | false | false | true  |
| 5 | true  | true  | false | true  | true  |
| 6 | false | false | false | false | false |

Sample calls to `evaluateLight` are shown below.

| Call to <code>evaluateLight</code>    | Value Returned     | Explanation                                                                             |
|---------------------------------------|--------------------|-----------------------------------------------------------------------------------------|
| <code>sim.evaluateLight(0, 3);</code> | <code>false</code> | The light is on, and the number of lights that are on in its column is even.            |
| <code>sim.evaluateLight(6, 0);</code> | <code>true</code>  | The light is off, and the number of lights that are on in its column is divisible by 3. |
| <code>sim.evaluateLight(4, 1);</code> | <code>false</code> | Returns the light's current status.                                                     |
| <code>sim.evaluateLight(5, 4);</code> | <code>true</code>  | Returns the light's current status.                                                     |

Class information for this question

```
public class LightBoard
private boolean[][] lights
public LightBoard(int numRows, int numCols)
public boolean evaluateLight(int row, int col)
```

Complete the `evaluateLight` method below.

```
/** Evaluates a light in row index row and column index col and returns a status
 *   as described in part (b).
 *   Precondition: row and col are valid indexes in lights.
 */
public boolean evaluateLight(int row, int col)
```

STOP

END OF EXAM