

Programming Assignment 1

Here is a dry run of the programming assignment question 1. Suppose the final state that is demanded in subpart 4 [This is not a real subpart] is the following diagram:

```
XX00XX
XX00XX
000000
XX00XX
XX00XX
```

Where x is the black cell and o is white. Suppose you are to get the above state in 1 iteration. As shown in the tutorial, the rule can be the one where the cell goes black if there is at least 1 black cell in its neighborhood.

Initial State

The initial state that you must store in the file `config.txt` will be the representation of the state with only the corner cells as black. The initial state will be:

```
X0000X
000000
000000
000000
X0000X
```

The configuration for the above state will be:

```
6 6 4
1 1
1 6
6 1
6 6
```

Rule

The rule is to set a cell to black if there is at least one cell in the neighborhood that is black:

```
def rule(cell_value: int, neighbors: list[int]) -> int:
    """
    Representing the cell values as an integer
    - 0 is white
    - 1 is black
    The return value of the function is the new value the cell must take.
    """
    for n in neighbors:
        if n == 1:
            return 1
    return 0

# Other functionality to run the simulation and render the state
```

Execution

When the command `python q1/rule.py` is executed, the program must render the initial state based on `q1/config.txt`. It must prompt the user to input the number of iterations and must take the input via `stdin`. Entering the iteration number must render the diagram on the screen and must also save the newly generated state in a file in the [X, O] format and save it in `q1/`.

The folder structure has been mentioned in the assignment document.