Grok Academy - take home exam

Background

Imagine a row of boxes which can be either empty or filled, like so:

```
[ ][ ][ ][X][ ][ ][ ]
```

This row of boxes represents the world, with all boxes being empty except the middle one, which is filled. In principle the world can be any size, we've just used seven boxes here for simplicity.

This world evolves over time, in steps, with boxes becoming empty or filled based on the state of their neighbours. At each step, we apply some rules to each box, calculating the next generation of the world. We then apply the rules again to the new generation, creating another new generation, and so on.

The rules for updating each box are:

- If a box is filled, and either its left neighbour or both neighbours are filled, it becomes empty.
- If a box is empty, and exactly one neighbour is filled, it becomes filled.
- Otherwise, its state remains the same.

The task

We would like you to create a program which calculates the first ten generations of the world above and prints them to the console, with the following specifications:

- The world should be 21 boxes wide.
- Initially all boxes should be empty except for the middle one.
- Each generation should be printed to the console on a new line. If a box is empty, print a space. If a box is filled, print an uppercase X.

For example, printing the first two generations would give the following result:

```
  X
 XXX
```

Submission instructions

Please share a link to a GitHub repo with your submission. Be sure to include instructions for how to run your code.
Feel free to use any programming language you wish.