- Everything works pretty well. The cave gets built correctly, solve() uses BFS like it's supposed to (no recursion), and the path to the mirror pool gets found if it exists. Both constructors (no-arg and file-based) are working. Also tested getPath() and it's giving the correct direction string.

- I used two test cases:
  One from the default constructor with a hardcoded layout
  One loaded from a file that followed the input format
  In both, there was a clear path to the mirror pool. I checked that the path was found, that the final layout printed correctly, and that the direction string was accurate.

- I learned how to implement a non-recursive BFS algorithm to explore 2D grids and track paths effectively using direction arrays and parent tracking. I also gained better understanding of when and why to use BFS over DFS.

- I learned how to implement a non-recursive BFS algorithm to explore 2D grids and track paths effectively using direction arrays and parent tracking. I also gained better understanding of when and why to use BFS over DFS.

- I liked that this project combined file I/O, 2D arrays, and algorithmic problem-solving in one. It felt very practical, like solving a real-world maze.

- Initially, the part about avoiding recursion was tricky, as DFS is often used for pathfinding. Also, managing both the layout and direction tracking without recursion took some trial and error.

- With extra time, I'd add support for multiple mirror pools or branching paths.