**Describe the state of your project, what works and what doesn’t.**

The project includes a CaveExplorer class that can create a 2D cave structure, display the initial and final layouts, and find a path from 'S' to 'M' while avoiding 'R' obstacles works correctly. The code includes constructors for creating a cave layout from a predefined string and reading it from a file. The solve method successfully finds a path, and the getPath method retrieves the path when one exists. My project is working without any error, and it met all the requirements.

**Describe how you tested your program, including tests that made you rethink your code. Include the layout you used.**

I tested the program by creating a CaveExplorer object with both predefined and file-based cave layouts. The testing involved printing the initial and final states of the cave, checking if there's a path, and if so, displaying the path. I try to check with various cave layouts, including simple and more complex ones, to verify my program. I used the following layout to test my code:

5 6

RRRRRR

R..SRR

R.RRRR

R..MRR

RRRRRR

**In a sentence or two, what did you learn?**

I learned how to implement a basic algorithm using breadth-first search. I also gained experience in reading data from files in Java.

**In a sentence or two, what did you like about this project?**

I liked that this project provided a practical application of data structures and algorithms. It was satisfying to see the code successfully navigate through the cave and find a path.

**In a sentence or two, what did you find confusing or would like to see done differently regarding this project?**

Initially I was confused about what the final layout would look like as there was not much description of the final layout. But eventually I figured out to mark the path travelled to the mirror as ‘V’ for visited.

**In a sentence or two, if you had another hour or two, what would you like to add to the project or how would you do things differently?**

If I had more time to work on this project, I would focus on the following:

* Optimizing the path retrieval to provide a more concise path representation, such as directions instead of individual moves.
* Expanding the test suite to include more diverse test cases, including edge cases and complex mazes, to thoroughly validate the program's functionality and correctness.