**Rubik’s Cube Response**

**To Represent a 3x3 Rubik's Cube in a program,**

We need to first consider the physical structure of the cube. The cube consists of six faces, each made up of nine smaller squares, for a total of 54 squares. Each square can be of one of six colors: red, blue, green, yellow, orange, and white.

**Data Structure(s) representing the cube and its parts**

To represent the cube in a program, we can use a 3-Dimensional (3D) Array. The first dimension would represent the face of the cube (top, bottom, left, right, front, and back), the second dimension would represent the row, and the third dimension would represent the column. Each element in the array would store the color of the square at that position.

**How to perform rotations on those data structures**

To perform rotations on the cube, we can define functions that update the colors in the array accordingly. For example, rotating the top face clockwise would involve swapping the colors of the top row of the front face with the top row of the right face, the top row of the right face with the top row of the back face, the top row of the back face with the top row of the left face, and the top row of the left face with the top row of the front face.

**How you could test your proposed solution**

To test our proposed solution, we can create a series of test cases that involve performing various rotations on the cube and checking that the resulting cube state matches our expectations. For example, we can test rotating the top face clockwise, counter-clockwise/anticlockwise, twice, etc. and verifying that the resulting cube state is correct.

Overall, this approach should work well for representing a Rubik's Cube in a program and performing rotations on it.