For the first challenge my idea was simple calculate a ray’s outcome by moving it without representing a board so I checked in every position what will happen by calculating the positions that an atom will affect whether it is a hit or a reflection. I tested it by calculating multiple interactions and checking if the outcome is the expected one. I calculated some of them by hand to make sure they are good. I tested it with different board sizes and different atom positions.

For the second challenge I first tried the brute forcing method, but it took way too much time. Then I read about threshold approach, and I tried that which became my final solution. I assigned different values to tiles and the bigger the tile the higher chance that it is an atom. For testing I choose different board sizes and different places for an atom and then using challenge 1 to calculate the interactions I checked if it was good or not and if it is within the time limit.

For the third challenge my idea was to first convert the expression to lambda form with the correct placement of parentheses. After converting it I did the Scott encoding, I found it easier this way than by doing directly before converting it. I tested the program by manually calculating different lambda expressions and seeing if my prettyPrint would give the expected outcome and if it would transform any suitable lambda expressions to natural numbers.

In a nutshell, I tried testing every function with different values that might show me unexpected errors.