Problem 1
Define a template for those gems include four attributes: name, hardness, density and colors. And then define the facts for each gem. Since we want to determine if the gem is Chyrsoberyl based on three facts, we use one rule to assert those three facts and one rule to decide if those three facts meet the characteristic of Chyrsoberyl.

Problem 2
For problem 2, we also need to define the template and facts like what we did in Problem 1. Then we use one rule to ask the user to enter the hardness, density and colors they want to use it to decide a gem name and one rule to compare the facts users enter and the facts we define before then get the result what the gem is.

Problem 3
In problem 3, we need to add the range of hardness and density. So I change the template of gems, I created two template one for gem's name and colors and one for gem's name, hardness's maximum, hardness's minimum, density's maximum and density's minimum. We need to have three rules to ask the user enter hardness, density and color separately because we need to create two rules to decide whether the number of hardness is between 1 to 10 and density is between 1 to 6. Then we create one rule to find gem's name depend the color user enter and use those gem's name to one rule decide what gems meet the number of hardness user enter and one rule decide what gems meet the number of density.

Problem 4
Based on the problem 3, I change the template. Just use one template includes gem's name, hardness's maximum, hardness's minimum, density's maximum, density's minimum and color. Then we also have rules to ask the user enter the facts they want to query and decide whether the input meet the requirement but we need to use retract when the fact user enter not meet the requirement. Then we find the gem depend on the user input if we find output the gem's name if not we need to create rules to decide which requirement meets depend on those input.