Below are several exercises that we will be covering in the upcoming tutorial (Friday, Jan. 21). We will release these exercises in advance of the tutorial so that you get a chance to attempt the exercises yourself before we discuss them in tutorial. The course staff running the tutorial will go through each problem (time permitting) and show their process for how they would complete these questions in an assignment/exam setting. Being familiar with the questions before attending tutorial will help to ensure that you get the most out of our explanations. Note that tutorials are not mandatory in CS135. We never cover any new material that you won’t have seen in lectures, and instead choose to highlight techniques and concepts from the most recent lectures to give you some extra practice. If you are very comfortable with the material that we covered and could easily complete the below questions, then you aren’t obligated to attend tutorials. Be very cautious with this though since sometimes questions can seem easy until you actually sit down and attempt them.

**Question 1: Rock, Paper, Scissors!**

You and your friend always settle disputes with a game of Rock Paper Scissors. Your friend always tries to cheat by changing the rules every time you play. To fix this problem, create a function \((\text{rps-result} \ \text{player1} \ \text{player2})\) that consumes two symbols (any of ‘rock, ‘paper, or ‘scissors), and produces a symbol denoting the outcome of the game (any of ‘p1-win,’ ‘p2-win,’ ‘draw’).

**Question 2: Rollercoaster Requirements**

As a roller coaster operator, you constantly decide if each person in line can be allowed on your ride. The amusement park you work at has some guidelines:

1. The height of a rider must be above 1.2 meters (120cm).
2. The rider must be at least 12 years old or be accompanied by an adult.
3. If the rider has a ride pass, they are to be allowed to ride even if they do not meet the previous requirements.

Write a function \((\text{able-to-ride?} \ \text{height} \ \text{age} \ \text{with-adult?} \ \text{pass})\) that consumes four parameters in this order: the rider’s height and age (in cm and years), whether they are accompanied by an adult (true or false), and whether they have a ride pass (‘pass or ‘no-pass). The function should produce true if the rider is allowed on the ride, and false otherwise.

*Example:*

\((\text{able-to-ride?} \ 180 \ 15 \ \text{false} \ \text{’no-pass}) \rightarrow \text{true}\)

*Hint: Try to reduce the number of cases in your cond expression as much as possible*

**Question 3: Squareception**

Write a function \((\text{inside/square?} \ \text{x1} \ \text{y1} \ \text{len1} \ \text{x2} \ \text{y2} \ \text{len2})\) which will consume 2 positions on a Cartesian plane, \((x1, y1)\) and \((x2, y2)\), as well as two lengths, \(\text{len1}\) and \(\text{len2}\). These parameters represent two squares, where each position represents the respective top-left point of each square, and each length represents the respective side length of each square. \(\text{inside/square?}\) should produce true if the second square is completely inside of the first. A square is considered to be completely inside of itself. Note that “square” in this context means a square with sides that are parallel to the x- and y-axes (i.e., not rotated in any way).

*Example:*

\((\text{inside/square?} \ 0 \ 3 \ 3 \ 1 \ 2 \ 1) \rightarrow \text{true}\)