1. Free/minimum time trajectory optimization problems are usually nonconvex, except for when the dynamics are linear.

   (a) true

   (b) false

   Solution: False, even with linear dynamics, having Δt as a decision variable makes the equality constraints nonlinear, making the problem nonconvex.

2. Direct trajectory optimization with a NonLinear Programming (NLP) solver like SNOPT/IPOPT can just as easily use explicit or implicit integration schemes.

   (a) true

   (b) false

   Solution: True. In direct trajectory optimization we are optimizing over both x's and u's and we use equality constraints to enforce the dynamics. This means that we can just as easily write down these constraints for an explicit integrator $x_{k+1} = f_e(x_k, u_k)$:

$$\begin{bmatrix} x_2 - f_e(x_1, u_1) \\ x_3 - f_e(x_2, u_2) \\ \vdots \\ x_N - f_e(x_{N-1}, u_{N-1}) \end{bmatrix} = 0 \tag{1}$$

   or an implicit integrator $f_i(x_k, u_k, x_{k+1}) = 0$:

$$\begin{bmatrix} f_i(x_1, u_1, x_2) \\ f_i(x_2, u_2, x_3) \\ \vdots \\ f_i(x_{N-1}, u_{N-1}, x_N) \end{bmatrix} = 0 \tag{2}$$

   Remember that an implicit integrator is defined by some residual that must be equal to 0, look at HW1 Q1 if you need to remember how you did this.

3. Trajectory optimization with DDP/iLQR can just as easily use explicit or implicit integration schemes.

   (a) true

   (b) false

   Solution: False, when we use DDP/iLQR we are doing rollouts (forward simulations in time), and taking derivatives of the discrete dynamics (A's and B's) for the backwards pass. Both of these things are easier with explicit integrators. When we use implicit integrators, our forward rollout involves using Newton's method to solve for each new state (which is more expensive than explicit integration), and we have to use the implicit function theorem to compute our dynamics derivatives (A's and B's). Both of these things are more difficult/expensive than with explicit integrators.

4. Both DDP/iLQR methods and direct trajectory optimization allow for warm starting of the controls (initializing the solver with a "guess" control trajectory).

   (a) true

   (b) false

   Solution: True, we can warmstart the controls with both methods.

5. Both DDP/iLQR methods and direct trajectory optimization allow for warm starting of the states (initializing the solver with a "guess" state trajectory).

(a) true
(b) false

Solution: False, we can only warmstart the controls with DDP/iLQR, but with direct trajectory optimization we can warmstart either controls or states or both.

6. NonLinear Program (NLP) solvers like SNOPT and IPOPT are guaranteed to find a **feasible** solution to a nonconvex problem if it exists.

   (a) true
   (b) false

   Solution: False, we are not guaranteed to find anything with a nonconvex optimization problem. Imagine if you gave an NLP solver a trajectory optimization problem through a maze, it is extremely likely the solver will be unable to find a feasible solution without an extremely good initialization.

7. NonLinear Program (NLP) solvers like SNOPT and IPOPT are guaranteed to find a **locally optimal** solution to a nonconvex problem if it exists.

   (a) true
   (b) false

   Solution: False. A locally optimal solution implies constraint satisfaction, so all of our concerns with the feasibility guarantee transfer to this as well.

8. NonLinear Program (NLP) solvers like SNOPT and IPOPT are guaranteed to find a **globally optimal** solution to a nonconvex problem if it exists.

   (a) true
   (b) false

   Solution: False. We are not guaranteed to find a feasible or locally optimal solution, so we definitely can't expect a guarantee to find a globally optimal one.

9. How many degrees of freedom exist in the group of 3D rotations?

   (a) 3
   (b) 4
   (c) 9

   Solution: 3. Any rotation can be described with thee-parameters (think roll pitch yaw in euler angles). I prefer thinking about this with the axis-angle parameterization, where any rotation can be described with a simple rotation about one axis.

10. There exists a singularity-free three-parameter attitude representation.

    (a) true
    (b) false

    Solution: False. Euler angles can be singular as soon as $90°$, axis-angle and Rodrigues Parameters are singular at $180°$, Modified Rodrigues Parameters are singular at $360°$. These are just the most common ones but they all have singularities.

11. Simulating the kinematics of a quaternion does not require any trigonometric functions.

    (a) true
    (b) false

    Solution: True, no trigonometry just quaternion multiplication (so a cross product and dot product but no trig). $\dot{q} = \frac{1}{2} q \odot \hat{\bar{\omega}}$.

12. Every unique attitude can be described by how many unique quaternions?

   (a) 1
   (b) 2
   (c) 4

   Solution: 2. Each $+q$ and $-q$ describe the same attitude. It is important to note that they describe different rotations but the same attitude. This is because one of the quaternions rotations $< 180°$, and the other quaternion rotates the opposite direction $>= 180°$. This is the 3D version of how $+90°$ and $-270°$ describe the same angle.

13. These two quaternions describe the same attitude but different rotations. Which rotation is "shorter"?

$$q_1 = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ 0 \\ 0 \\ \frac{\sqrt{2}}{2} \end{bmatrix} \qquad q_2 = \begin{bmatrix} -\frac{\sqrt{2}}{2} \\ 0 \\ 0 \\ -\frac{\sqrt{2}}{2} \end{bmatrix} \tag{3}$$

   (a) 1
   (b) 2

   Solution: A. The quaternion with the positive scalar component is the shorter of the 2.