Final Project Options

1. **N-body integrator:** Create an integrator that solves the gravitational N-body problem forward in time for up to 520 time steps, in 2 dimensions.
    a. Part I: 2-body problem, showcase the simplest system using the parameters for Earth and the Sun
        i. You'll need: masses and positions of particles, initial velocity of particles, acceleration due to gravity between particles
        ii. What you need to do: print out relevant data in an understandable way
    b. Part II: Integrate up to 100 particles in a system
    c. BONUS: Add a large mass in the center of your 100 particle system (like a black hole in the centre of a globular cluster) – what happens?

2. **Automated Database:** Create a program that will initialize, read in, and manipulate a database of cars and car accessories.
    a. Part I: Your inventory is only of cars.
        i. You'll need: the model number, year, colour, make, model, body type, and quantity of each car.
        ii. What you need to do: Your inventory needs to take in data about the cars and perform the following actions as per the user's request: load data, find index (given the model number of the car), add car to inventory, remove car from inventory, show inventory, output inventory, and quit (ie. stop program).
    b. Part II: Include a separate inventory for car accessories. Each accessory will need the following information: accessory name, model number (for car it matches), year, colour, make, model, body type, and quantity. You'll now need to include operations for finding cars that match accessories.
    c. BONUS: Add suggestions for buying car accessories and cars to the user when certain thresholds are reached or particular "necessary" accessories and models are not in the initial inventory.

3. **Cancer cell growth:** Create an algorithm that will estimate if patients will develop a cancerous growth within 10 years and if so, how that growth progresses in time and if it spreads.
    a. Part I: only one patient, with a given probability of developing cancer (between 0-100%).
        i. You'll need: weighting on IF the patient develops cancer after each time step (1 week), how fast the cancer grows once developed, weighting on IF the cells become dislodged, and the weighting on IF dislodged cells create a tumour elsewhere.
        ii. What you need to do: print out relevant data in an understandable way

b. Part II: Input probabilities from up to 100 patients at a time.
c. BONUS: Include probabilities for a patient receiving treatment and how well that treatment works – does it stop the cancer by the end of the 10 year period?

All projects must be accompanied with supporting documentation on use (user manual, training documentation, help files), a method for generating input files, and a report that discusses:

- Pros and cons of using python for this problem and why
- Difficulties (computationally, mathematically, conceptually, etc.) with the problem
- Brief history of problem being solved on computers
- Best current version of publically available code for this problem
- Your logic for solving the problem and implementing the code
- Struggles or issues you noticed while writing your code or issues that are still present and why

Important Dates:

- April 3rd 2018: Choose a project (notify in-person or by email)
- June 4th 2018: Code, supporting documentation, and reports due by midnight (email submission)
    - Grading:
        - Knowledge: Using skills from course in code – 30%
        - Inquiry: Background knowledge, mechanics of code – 15%
        - Communication: Comments, Report, supporting documentation – 25%
        - Application: Implementing the above 3 into the code – 30%
- June 5th 2018: 10 minute presentation on results and background information (in-class)
    - Grading:
        - Knowledge: Highlighting methods used – 10%
        - Inquiry: Presenting logic and reasoning for input choices – 20%
        - Communication: presentation skills – 50%
        - Application: showcasing visuals/results – 20%