Programming HW #3
Due: 23:59 June 29

Please use the provided `trigonometry.py` starter file for exercise 0.

The TAs will be in the labs starting at 20:30 June 29 to go over your code. Please print out and submit your report for exercise 1 to the homework submission folder.

0. Trigonometry

To practice a bit of the trigonometry you've been using (and to write a function that may prove useful for your OD code), do the following:

Define a function that takes the values (in decimal degrees) of two sides and the included angle of a spherical triangle as parameters and, using the law of cosines and law of sines (and the provided `findQuadrant` function), calculates and returns the values (also in decimal degrees) of the other side and angles.

The TAs will have you use the following values to test your solution:

- if a = 106°, B = 114°, and c = 42°, then b = 117.804°, A = 83.110°, and C = 43.715°

1. Code Archeology

This exercise will be a little different than usual. The TAs recently unearthed (metaphorically speaking) a Python program written by an SSP student in 2006. The code, under the file name `OrbitViz.py`, appears to have been written to use VPython to visualize the orbit of an asteroid around the Sun. It's difficult to be sure, as the code not only fails to run, but lacks comments altogether. Your task is both simple and open-ended: make it work. Your goal is to create a working visualization of an asteroid orbit. The details of the visualization (i.e. how many solar system bodies, scale, labeling, etc.) are up to you.

You have been provided with the source code and a document discovered alongside the code that appears to be the basis for its computations. In addition to your fixed (and possibly improved) visualization code, you'll turn in a typewritten report describing the following:

- Any errors you fixed
- Any changes you made to the structure of the code or the computations it performs, and why those were necessary or beneficial
- Any additional features you added to the visualization
- How you verified your visualization was correct (or close enough)

Consider including the original and final versions of the code in your report in order to illustrate what you changed.

Some things to be aware of:

- `OrbitViz.py` does not follow the ephemeris generation guide exactly
- The visualization depends of VPython. Find a tutorial here: http://www.glowscript.org/docs/VPythonDocs/VPython_Intro.pdf and full documentation here: http://www.glowscript.org/docs/VPythonDocs/index.html. Note that the tutorial and documentation are written for running VPython in a web browser-based environment called GlowScript rather than writing a standalone script as you will be doing. All the example code should still apply, you'll just need to ignore references to the GlowScript interface.
- The lack of comments in `OrbitViz.py` makes it very difficult to understand — don't make the same mistake in your own code!
- `OrbitViz.py` also puts a bunch of math operations all on the same line, making them nearly impossible to decipher — avoid this!
- `OrbitViz.py` does not use `numpy`, but it probably should.