ASTR222
Project 1.5, Due Wednesday, November 7, 2018

1. Write a program in Python that will accept the following inputs and return the following outputs:

   Inputs:
   i. Geographic location on Earth, provided as latitude and longitude in decimal degrees. (Remember that east is positive and west is negative.)
   ii. Time, provided as clock time.
   iii. Altitude and Azimuth, provided as decimal degrees.

   Outputs:
   i. The astronomical coordinates at which you are pointed, as RA and Dec.
   ii. The most likely astronomical object for the given input. Assume that pointing error in the original input is hand-held accuracy, 10 arcminutes.

2. Now it's time to solve the problem in reverse. Write a program in Python that accepts the following inputs and return the following outputs:

   Inputs:
   i. The name of an astronomical object.
   ii. The Altitude and Azimuth of that object, assuming the same pointing error as in part 1.
   iii. Time, provided as GMT.

   Output:
   Your geographic location on the Earth as latitude and longitude.

3. For part 1, describe the methodology you used to determine most probable astronomical source.

4. For part 2, how does the uncertainty in your geographic position compare with the uncertainty in astronomical coordinates from the previous problem?

   Note on Python packages: you may use astroquery for things involving the names of astronomical objects. You may not use astropy.coordinates or anything similar to do the coordinate transformation steps.

   Note on corrections: Atmospheric refraction ranges from zero at the zenith to ~5 arcminutes at 10º altitude. You may assume that all observations will be conducted at substantially greater than 10º altitude, so the error introduced by atmospheric refraction

is small compared to the pointing error and may thus be ignored. Similarly, the ellipsoidal shape of the Earth can introduce a difference between Geodetic latitude and geocentric latitude as large as 10 arcminutes. For our purposes, we will assume observations are taken at high enough latitudes to bring this error within the range of pointing error, and will thus choose to ignore it also.