```
r""" This script is the inner workings of the COMFIX Project

This code will simplify the functions for the user
It will connect the functions into one simple code
Allows for Text File reading
Includes final conversions to degrees
Prints Results to Text File

Author: Thomas J Susi
"""

import numpy as np
from astro import RV2COE
from astro import COMFIX
from astro import constants
import pdb

filename = input("Name File to Interpret: ")
with open(filename) as file, open("COMFIX_Results.txt", "w") as
finalfile:

    print ("For Inputting Lines: Line n = n-1")
    start = int(input("First Line to Evaluate: ")),
    num = int(input("Number of Data Sets: ")),
    end = start[0] + 2*num[0]
    mu = float(input("Input mu Value: "))

    import pdb; pdb.set_trace()

    for count in range(start[0], end, 2):

        if count % 2 == 0:

            line = open (filename, "r").readlines()[count].split()
            Line_0 = ['Data Set {}\n\t{}\n'.format(count,line)]

            """Site Info"""
            lat = np.deg2rad(float(line[0]))
            lon = np.deg2rad(float(line[1]))
            alt = float(line[2])/1000
            jd = float(line[3])

            """Satellite Info"""
            line = open (filename, "r").readlines()[count+1].split()
            ID = float(line[0])
            rang = float(line[1])
            azm = np.deg2rad(float(line[2]))
            elev = np.deg2rad(float(line[3]))
            rang_r = float(line[4])
            azm_r = np.deg2rad(float(line[5]))
```

```
            elev_r = np.deg2rad(float(line[6]))

            Line_1 = ('Site Info:\n\tLatitude(deg): {}
\tLongitude(deg): {}\tAltitude(m): {}\n\tJD: {}
\n'.format(lat,lon,alt,jd))
            Line_2 = ('Sat {} Info:\n\tRange(km): {}\tAzm(deg): {}
\tElv(deg): {}\n\tRange Rate(km/s): {}\tAzm Rate(deg/s): {}\tElv
Rate(deg/s): {}\n'.format(ID,rang,azm,elev,rang_r,azm_r,elev_r))

            """Define Vectors"""
            pr_sez, pv_sez = COMFIX.topo2rv(rang, azm, elev, rang_r,
azm_r, elev_r)

            rad_r_ecef = COMFIX.lla2ecef(lat, lon, alt)

            """Define Rotations"""
            sez2b, b2ecef = COMFIX.sez2ecef(lat, lon, alt)

            ecef2eci = COMFIX.ecef2eci(jd, lon)

            """Rotate pr and pv"""
            pr_b = np.dot(sez2b.transpose(),pr_sez)

            pv_b = np.dot(sez2b.transpose(),pv_sez)

            pr_ecef = np.dot(b2ecef.transpose(),pr_b)

            pv_ecef = np.dot(b2ecef.transpose(),pv_b)

            pr_eci = np.dot(ecef2eci,pr_ecef)

            pv_eci = np.dot(ecef2eci,pv_ecef)

            """Rotate rad_r"""
            rad_r_eci = np.dot(ecef2eci,rad_r_ecef)

            """Find R and V"""
            w_E = [0,0,0.000072921151467]

            r_eci_array = pr_eci + rad_r_eci

            r_eci = [r_eci_array[0][0],r_eci_array[1]
[0],r_eci_array[2][0]]

            v_eci_array = pv_eci.transpose() + np.cross(w_E, r_eci)

            v_eci = np.array([v_eci_array[0][0],v_eci_array[0]
[1],v_eci_array[0][2]])

            a, e, e_vector, i, raan, w, theta, p, r_p, r_a, n_vector,
```

```
period = RV2COE.RV2COE(r_eci, v_eci, mu)

            r"""Print The Results in Full"""
            Line_3 = ["Semi Major Axis in Kilometers:\n\t", str(a),
"\n"]
            Line_4 = ["Eccentricity:\n\t", str(e), "\n"]
            Line_5 = ["Eccentricity Vector:\n\t", str(e_vector), "\n"]
            Line_6 = ["Inclination in Degrees:\n\t",
str(np.rad2deg(i)), "\n"]
            Line_7 = ["RAAN in Degrees:\n\t", str(np.rad2deg(raan)),
"\n"]
            Line_8 = ["Argument of Periapsis in Degrees:\n\t",
str(np.rad2deg(w)), "\n"]
            Line_9 = ["True Anomaly in Degrees:\n\t",
str(np.rad2deg(theta)), "\n"]
            Line_10 = ["Semi Latus Rectum in Kilometers:\n\t", str(p),
"\n"]
            Line_11 = ["Radius of Periapsis in Kilometers:\n\t",
str(r_p), "\n"]
            Line_12 = ["Radius of Apoapsis in Kilometers:\n\t",
str(r_a), "\n"]
            Line_14 = ["Line of Nodes Vector in Kilometers:\n\t",
str(n_vector), "\n"]
            Line_15 = ["Orbit Period in Seconds:\n\t", str(period),
"\n"]
            Line_16 = ["Sat Radius Vector (ECI):\n", str(r_eci), "\n"]
            Line_17 = ["Sat Velocity Vector (ECI):\n", str(v_eci),
"\n"]
            Line_18 = ["\n\n\n"]

            finalfile.writelines(Line_0)
            finalfile.writelines(Line_1)
            finalfile.writelines(Line_2)
            finalfile.writelines(Line_3)
            finalfile.writelines(Line_4)
            finalfile.writelines(Line_5)
            finalfile.writelines(Line_6)
            finalfile.writelines(Line_7)
            finalfile.writelines(Line_8)
            finalfile.writelines(Line_9)
            finalfile.writelines(Line_10)
            finalfile.writelines(Line_11)
            finalfile.writelines(Line_12)
            finalfile.writelines(Line_14)
            finalfile.writelines(Line_15)
            finalfile.writelines(Line_16)
            finalfile.writelines(Line_17)
            finalfile.writelines(Line_18)

            continue
```

```
finalfile.close()

print ("Complete")
```