

```
r""" This script is the inner workings of the RV2COE 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 constants  
import pdb
```

```
filename = input("Name File to Interpret: ")  
with open(filename) as file, open("RV2COE_Results.txt", "w") as  
finalfile:
```

```
    print ("For Inputting Lines: Line n = n-1")  
    start = input("First Line to Evaluate: "),  
    end = input("Last Line to Evaluate: "),  
    mu = float(input("Input mu Value: "))  
    for count in range(int(start[0]), int(end[0])):  
  
        line = open (filename, "r").readlines()[count].split()  
        Line_0 = ['Line {} \n\t{} \n'.format(count+1, line)]  
  
        r_i = float(line[0])  
        r_j = float(line[1])  
        r_k = float(line[2])  
        v_i = float(line[3])  
        v_j = float(line[4])  
        v_k = float(line[5])  
  
        r_input = [r_i, r_j, r_k]  
        v_input = [v_i, v_j, v_k]  
  
        Line_1 = ("The mu Constant: \n\t", str(mu), "\n", 'Radius {}  
(km) \n\t{} \n'.format(count+1, r_input))  
        Line_2 = ('Velocity {} (km/s) \n\t{}  
\n'.format(count+1, v_input))  
        r"""Inner Workings, Pulling outside functions"""  
        h_input = RV2COE.ang_momentum(r_input, v_input)  
  
        e_vector = RV2COE.eccentricity(mu, r_input, v_input, h_input)  
  
        n_vector = RV2COE.line_of_nodes(RV2COE.unit_vector(h_input))
```

```

p = RV2C0E.semi_latus_rectum(mu, h_input)

theta = RV2C0E.true_anom(r_input, h_input, e_vector)

i = RV2C0E.inclination(RV2C0E.unit_vector(h_input))

a = RV2C0E.semi_major_axis(p, np.linalg.norm(e_vector))

raan = RV2C0E.R_A_A_N(n_vector)

w = RV2C0E.arg_of_periapsis(n_vector, e_vector, h_input)

r_p = RV2C0E.rad_peri(p, np.linalg.norm(e_vector))

r_a = RV2C0E.rad_apo(p, np.linalg.norm(e_vector))

gamma = RV2C0E.flight_ang(np.linalg.norm(e_vector), theta)

period = RV2C0E.Orbit_Period(mu, a)

if (theta < 0):
    theta = 2 * np.pi + theta
else:
    theta = theta
if (raan < 0):
    raan = 2 * np.pi + raan
else:
    raan = raan
if (w < 0):
    w = 2 * np.pi + w
else:
    w = w
if (i < 0):
    i = 2 * np.pi + i
else:
    i = i

r""""Print The Results in Full""""
Line_3 = ["Semi Major Axis in Kilometers:\n\t", str(a), "\n"]
Line_4 = ["Eccentricity:\n\t", str(np.linalg.norm(e_vector)),
"\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 = ["\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.close()

print ("Complete")

```