```
%Gustavo Grinsteins
%ASEN 5050
%HW4

%House Keeping
clc;
clear;

%Given values
mu_moon = 4902.799; %km^3/s^2
mu_saturn = 3.794*10^7; %km^3/s^2
EQR_moon = 1738; %Km
EQR_saturn = 60268; %Km

%% Problem 1
fprintf('Problem 1 \n')
%Given from HW3
R_1 = [-720000;670000;310000];%km
V_1 = [2.160;-3.360;0.620];%km/s
H_1 = cross(R_1,V_1);%km^2/s
r_1 = norm(R_1);
v_1 = norm(V_1);
Sp_Mech_E_1 = (v_1^2/2)-(mu_saturn/r_1);
a_1 = (-mu_saturn)/(2*Sp_Mech_E_1);
Ecc_1 = (cross(V_1,H_1))/(mu_saturn)-(R_1/r_1);
ecc_1 = norm(Ecc_1);
theta_star_1 = abs(acos((dot(R_1,Ecc_1))/(norm(Ecc_1)*norm(R_1))));
if dot(R_1,V_1)<0
    theta_star_1 = (-1)*theta_star_1;
end
fprintf('True Anomaly at t1 = %4.2f deg \n',theta_star_1*(180/pi))
theta_star_impact = (-13.17)*(pi/180);
fprintf('True Anomaly at impact = %4.2f deg \n',theta_star_impact*(180/pi))
%calculating eccentric anomalies
E_1 = 2*atan(sqrt((1-ecc_1)/(1+ecc_1))*tan(theta_star_1/2));
fprintf('Eccentric Anomaly at t1 = %4.2f deg \n',E_1*(180/pi))
E_impact = 2*atan(sqrt((1-ecc_1)/(1+ecc_1))*tan(theta_star_impact/2));
fprintf('Eccentric Anomaly at impact = %4.2f deg \n',E_impact*(180/pi))
n = sqrt(mu_saturn/(a_1^3));
fprintf('mean motion = %4.4f rad/s \n',n)
M_1 = E_1 - ecc_1*sin(E_1);
fprintf('Mean Anomaly at t1 = %4.4f radians \n',M_1)
M_impact = E_impact - ecc_1*sin(E_impact);
fprintf('Mean Anomaly at impact = %4.7f radians \n',M_impact)
T1_To_TImpact_Time = (M_impact - M_1)/n;
fprintf('Time from t1 to t_impact = %4.4f hours \n\n',T1_To_TImpact_Time/3600)

%% Problem 2 Part a b c
% Calculate the orbita elements at t1
fprintf('Problem 2 Part a b c \n')
R_1 = [-7.87701*10^2;-8.81425*10^2;1.43864*10^3];%km
V_1 = [0.98370;0.76950;1.01416];%km/s
%Calculating orbital elements at t1
%Recover Orbital Elements
r_1 = norm(R_1);
v_1 = norm(V_1);
H_1 = cross(R_1,V_1);
%fprintf('<%4.4f,%4.4f,%4.4f> \n',H_1)
```

```
h_1 = norm(H_1);
Sp_Mech_E_1 = (v_1^2/2)-(mu_moon/r_1);
%inclination
z_hat = [0,0,1];
i_1 = acos((dot(z_hat,H_1))/(norm(z_hat)*norm(H_1)));
fprintf('inclination angle i at t1 = %4.2f deg \n',i_1*(180/pi))
%major-axis
a_1 = (-mu_moon)/(2*Sp_Mech_E_1);
fprintf('semi-major axis a at t1 = %4.4f Km\n',a_1)
%eccentricity
Ecc_1 = (cross(V_1,H_1))/(mu_moon)-(R_1/r_1);
ecc_1 = norm(Ecc_1);
fprintf('e at t1 is %4.5f\n',ecc_1)
%RAAN
x_hat = [1,0,0];
y_hat = [0,1,0];
N_1 = cross(z_hat,H_1);
RAAN_1 = abs(acos((dot(x_hat,N_1))/(norm(x_hat)*norm(N_1))));
if dot(N_1,y_hat)<0
    RAAN_1 = (-1)*RAAN_1;
end
fprintf('RAAN at t1 is %4.2f deg \n',RAAN_1*(180/pi))
%AOP
AOP_1 = abs(acos((dot(Ecc_1,N_1))/(norm(Ecc_1)*norm(N_1))));
if dot(Ecc_1,z_hat)<0
    AOP_1 = (-1)*AOP_1;
end
fprintf('Argument of Periapsis w at t1 = %4.2f deg \n',AOP_1*(180/pi))
theta_star = abs(acos((dot(R_1,Ecc_1))/(norm(Ecc_1)*norm(R_1))));
if dot(R_1,V_1)<0
    theta_star = (-1)*theta_star;
end
p_1 = a_1*(1-ecc_1^2); %Km
r_p = (p_1)/(1+ecc_1*cosd(0));
r_a = (p_1)/(1+ecc_1*cosd(180));
fprintf('True Anomaly ThetaStar = %4.2f deg \n',theta_star*(180/pi))
fprintf('Moon Radius = %4.4f Km \n',EQR_moon)
fprintf('Moon orbit periapsis radius = %4.4f km \n',r_p)
fprintf('Moon orbit apoapsis radius = %4.4f km \n',r_a)
%Calculating Eccentric anomalies at ascending and descending nodes
theta_star_descending = AOP_1;
theta_star_ascending = pi-((-1)*AOP_1);
n = sqrt(mu_moon/(a_1^3));
period = 2*pi*sqrt((a_1^3)/mu_moon);
E_ascending = 2*atan(sqrt((1-ecc_1)/(1+ecc_1))*tan(theta_star_ascending/2));
tasc_minus_tp = (1/n)*(E_ascending - ecc_1*sin(E_ascending));%time from tp to asc
E_descending = 2*atan(sqrt((1-ecc_1)/(1+ecc_1))*tan(theta_star_descending/2));
tdesc_minus_tp = (1/n)*(E_descending - ecc_1*sin(E_descending));%time from desc to tp
tpos = period - abs(tdesc_minus_tp) - abs(tasc_minus_tp);
tneg = abs(tdesc_minus_tp) + abs(tasc_minus_tp);
fprintf('Tpos = %4.4f hours \n',tpos/3600)
fprintf('Tneg = %4.4f hours \n\n',tneg/3600)
%% Problem 2 Part d c
fprintf('Problem 2 Part d c \n')
E_1 = 2*atan(sqrt((1-ecc_1)/(1+ecc_1))*tan(theta_star/2));
fprintf('E at t1 = %4.2f degrees \n',E_1*(180/pi))
t1_minus_tp = (1/n)*(E_1 - ecc_1*sin(E_1));
fprintf('t1 - tp = %4.4f seconds \n',t1_minus_tp)
```

```
t2_minus_tp = t1_minus_tp + 30*(60);
fprintf('t2 - tp = %4.4f seconds \n',t2_minus_tp)
E_2 = NewtonRaphsonMethodForE(ecc_1,n,t2_minus_tp);
fprintf('E at t2 = %4.2f degrees \n',E_2*(180/pi))
theta_star_2 = 2*atan(sqrt((1+ecc_1)/(1-ecc_1))*tan((E_2)/2));
fprintf('True Anomaly at t2 = %4.2f deg \n',theta_star_2*(180/pi))
r_2 = (p_1)/(1+ecc_1*cos(theta_star_2));
fprintf('orbit radius at t2 = %4.4f km \n',r_2)
Altitude_t2 = r_2 - EQR_moon;
fprintf('Altitude at t2 = %4.4f km \n',Altitude_t2)
```