```
%Gustavo Grinsteins
%ASEN 5050
%HW3

%House Keeping
clc;
clear;

%Given values
mu_mars = 4.305*10^4; %km^3/s^2
mu_saturn = 3.794*10^7; %km^3/s^2
EQR_mars = 3397.2; %Km
EQR_saturn = 60268; %Km

%% Problem 1 Part a
fprintf('Problem 1 Part a \n')
R_1 = [-720000;670000;310000];
V_1 = [2.160;-3.360;0.620];
%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);
h_1 = norm(H_1);
Sp_Mech_E_1 = (v_1^2/2)-(mu_saturn/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_saturn)/(2*Sp_Mech_E_1);
fprintf('semi-major axis a at t1 = %4.2f Km\n',a_1)
%eccentricity
Ecc_1 = (cross(V_1,H_1))/(mu_saturn)-(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\n',AOP_1*(180/pi))

fprintf('Given Ralative 2BP assumptions we can use t1 orbital elements to calculate↙
impact values \n')
r_impact = EQR_saturn;
p_1 = a_1*(1-ecc_1^2); %Km
r_p = (p_1)/(1+ecc_1*cosd(0));%Km since r_p < r_impact we know the orbit is in collision↙
course
```

```
r_a = (p_1)/(1+ecc_1*cosd(180));
theta_star_impact = abs(acos((p_1-r_impact)/(r_impact*ecc_1)));
if dot(V_1,R_1)<0
    theta_star_impact = (-1)*theta_star_impact;
end
fprintf('True Anomaly at impact = %4.2f deg \n',theta_star_impact*(180/pi))
v_r_impact = (mu_saturn/h_1)*ecc_1*sin(theta_star_impact);%km/s
v_theta_impact = (mu_saturn/h_1)*(1 + (ecc_1*cos(theta_star_impact))); %km/s

%Calculating rotation matrix
theta_impact = theta_star_impact+AOP_1;%Degrees

R1 = [1,0,0;0,cos(i_1),sin(i_1);0,-sin(i_1),cos(i_1)];
R3_RAAN = [cos(RAAN_1),sin(RAAN_1),0;-sin(RAAN_1),cos(RAAN_1),0;0,0,1];
R3_theta = [cos(theta_impact),sin(theta_impact),0;-sin(theta_impact),cos(theta_impact),0;↙
0,0,1];

C = R3_theta*R1*R3_RAAN;

PositionRot_impact = [r_impact;0;0];
VelocityRot_impact = [v_r_impact;v_theta_impact;0];

%Transforming position from r,theta,h to XYZ
PositionXYZ_impact = C.'*PositionRot_impact;
fprintf('R vector in XYZ frame at impact is <%4.4f,%4.4f,%4.4f> Km\n',PositionXYZ_impact)

%Transforming velocity from r,theta,h to XYZ
VelocityXYZ_impact = C.'*VelocityRot_impact;
fprintf('V vector in XYZ frame at impact is <%4.4f,%4.4f,%4.4f> Km/s\n\n',↙
VelocityXYZ_impact)


%% Problem 2 Part b
fprintf('Problem 2 Part b \n')
e = 0.45454;
a = 6463.8; %Km
ideg = 74.924; %Degrees
RAANdeg = 1.2410; %Degrees
AOPdeg = 353.31; %Degrees
theta_star_deg = 199.38; %Degrees

%Orbital Period
OrbitPeriod = (((2*pi)*sqrt(a^3/mu_mars))/60)/60;%Hours
fprintf('Orbital Period is %4.2f hours \n',OrbitPeriod)

%Periapsis Altitude
p = a*(1-e^2); %Km
r_periapsis = (p)/(1+e*cosd(0));%Km
PeriapsisAltitude = r_periapsis - EQR_mars;%Km
fprintf('Periapsis Altitude is %4.4f Km \n\n',PeriapsisAltitude)

%% Problem 2 Part c
fprintf('Problem 2 Part c \n')
h = sqrt(mu_mars*p);%Km^2/s
r = (p)/(1+e*cosd(theta_star_deg)); %Km
v_r = (mu_mars/h)*e*sind(theta_star_deg);%km/s
v_theta = (mu_mars/h)*(1 + (e*cosd(theta_star_deg))); %km/s
```

```
%Calculating rotation matrix
theta = theta_star_deg+AOPdeg;%Degrees

R1 = [1,0,0;0,cosd(ideg),sind(ideg);0,-sind(ideg),cosd(ideg)];
R3_RAAN = [cosd(RAANdeg),sind(RAANdeg),0;-sind(RAANdeg),cosd(RAANdeg),0;0,0,1];
R3_theta = [cosd(theta),sind(theta),0;-sind(theta),cosd(theta),0;0,0,1];

C = R3_theta*R1*R3_RAAN;

PositionRot = [r;0;0];
VelocityRot = [v_r;v_theta;0];

%Transforming position from r,theta,h to XYZ
PositionXYZ = C.'*PositionRot;
fprintf('R vector in XYZ frame is <%4.4f,%4.4f,%4.4f> Km\n',PositionXYZ)

%Transforming velocity from r,theta,h to XYZ
VelocityXYZ = C.'*VelocityRot;
fprintf('V vector in XYZ frame is <%4.4f,%4.4f,%4.4f> Km/s\n\n',VelocityXYZ)

%Recover Orbital Elements
r_2 = norm(PositionXYZ);
v = norm(VelocityXYZ);
H = cross(PositionXYZ,VelocityXYZ);
h = norm(H);
Sp_Mech_E = (v^2/2)-(mu_mars/r_2);
%inclination
z_hat = [0,0,1];
i = acos((dot(z_hat,H))/(norm(z_hat)*norm(H)));
fprintf('inclination angle i = %4.2f deg \n',i*(180/pi))
%major-axis
a_2 = (-mu_mars)/(2*Sp_Mech_E);
fprintf('semi-major axis a = %4.2f Km\n',a_2)
%eccentricity
Ecc = (cross(VelocityXYZ,H))/(mu_mars)-(PositionXYZ/r_2);
ecc = norm(Ecc);
fprintf('e is %4.5f\n',ecc)
%RAAN
x_hat = [1,0,0];
y_hat = [0,1,0];
N = cross(z_hat,H);
RAAN = abs(acos((dot(x_hat,N))/(norm(x_hat)*norm(N))));
if dot(N,y_hat)<0
    RAAN = (-1)*RAAN;
end
fprintf('RAAN is %4.2f deg \n',RAAN*(180/pi))
%AOP
AOP = abs(acos((dot(Ecc,N))/(norm(Ecc)*norm(N))));
if dot(Ecc,z_hat)<0
    AOP = (-1)*AOP;
end
fprintf('Argument of Periapsis w = %4.2f deg \n',AOP*(180/pi))
%True Anomaly
theta_star = abs(acos((dot(PositionXYZ,Ecc))/(norm(Ecc)*norm(PositionXYZ))));
if dot(PositionXYZ,VelocityXYZ)<0
    theta_star = (-1)*theta_star;
end
fprintf('True Anomaly ThetaStar = %4.2f deg \n\n',theta_star*(180/pi))
```