Question 1:

```
function oe = rv2oe_Hackbardt_Chris(rPCI,vPCI,mu)

% This function calculates the six orbital elements from the position vector
and velocity vector
% Function Call: oe = rv2oe_Hackbardt_Chris(rPCI,vPCI,mu)
%
% Inputs:
%    rPCI:  Cartesian planet-centered inertial (PCI) position (3 by 1)
%    vPCI:  Cartesian planet-centered inertial (PCI) velocity (3 by 1)
%    mu:    gravitational parameter of centrally attacting body
% Outputs:  orbital elements
%    oe(1): semi-major axis
%    oe(2): eccentricity
%    oe(3): longitude of the ascending node (rad)
%    oe(4): inclination (rad)
%    oe(5): argument of the periapsis (rad)
%    oe(6): true anomaly (rad)

rvec = rPCI;
vvec = vPCI;

Ix = [1;0;0];
Iy = [0;1;0];
Iz = [0;0;1];

hvec = cross(rvec,vvec);
h = norm(hvec);
p = h^2/mu;
r = norm(rvec);
evec = ((cross(vvec,hvec))/mu)-(rvec/r);
e = norm(evec);
a = p/(1-e^2);
nvec = cross(Iz,hvec);
n = norm(nvec);
Omega = atan2(dot(nvec,Iy),dot(nvec,Ix));
if Omega < 0
    Omega = Omega + (2*pi);
end
inc = atan2(dot(hvec,cross(nvec,Iz)),n*dot(hvec,Iz));
omega = atan2(dot(evec,cross(hvec,nvec)),h*dot(evec,nvec));
if omega < 0
    omega = omega + (2*pi);
end
nu = atan2(dot(rvec,cross(hvec,evec)),h*dot(rvec,evec));
if nu < 0
    nu = nu + (2*pi);
end

oe = [a; e; Omega; inc; omega; nu];
end
```

Question 2:

```matlab
function [rPCI,vPCI]  = oe2rv_Hackbardt_Chris(oe,mu)

% This function calculates the postion and velocity vector with respect to
the planet using orbital elements.
% Function Call: [rPCI,vPCI]  = oe2rv_Hackbardt_Chris(oe,mu)
% Input:  Orbital Elements: (6 by 1 column vector)
%   oe(1): Semi-major axis
%   oe(2): Eccentricity
%   oe(3): Longitude of the ascending node (rad)
%   oe(4): Inclination (rad)
%   oe(5): Argument of the periapsis (rad)
%   oe(6): True anomaly (rad)
%   mu:    Planet gravitational parameter     (scalar)
% Outputs:
%   rPCI:  Planet-Centered Inertial (PCI) Cartesian position
%          (3 by 1 column vector)
%   vPCI:  Planet-Centered Inertial (PCI) Cartesian inertial velocity
%          (3 by 1 column vector)

a = oe(1);
e = oe(2);
Omega = oe(3);
i = oe(4);
omega = oe(5);
nu = oe(6);

p = a*(1-e^2);
r = p/(1+(e*cos(nu)));

rvecP = [r*cos(nu);r*sin(nu);0];
vvecP = (sqrt(mu/p))*[-sin(nu);e+cos(nu);0];

T_NI = [cos(Omega),-sin(Omega),0;sin(Omega),cos(Omega),0;0,0,1];
T_QN = [1,0,0;0,cos(i),-sin(i);0,sin(i),cos(i)];
T_PQ = [cos(omega),-sin(omega),0;sin(omega),cos(omega),0;0,0,1];
T_PI = T_NI * T_QN * T_PQ;

rPCI = T_PI * rvecP;
vPCI = T_PI * vvecP;
end
```

Question 3:

Since angles between vectors are undefined when one or more of those vectors have a magnitude of zero.

1. Since the inclination of the orbit, i, is the angle from $\underline{I}_z$ to ${}^I\underline{h}$, i is undefined when ${}^I\underline{h}$ is zero, which happens when the orbit is rectilinear.

2. The longitude of the ascending node, Ω, is the angle between $\underline{I}_x$ and $\underline{n}$, Ω is undefined when $\underline{n}$ is zero. $\underline{n}$ is zero when inclination is zero, also known as an equatorial orbit.

3. The argument of the periapsis, ω, is the angle from $\underline{n}$ to $\underline{e}$. This means that ω is undefined when $\underline{n}$ or $\underline{e}$ is equal to zero. This happens when the orbit is equatorial or circular.

4. True anomaly, ν, is the angle between $\underline{e}$ and $\underline{r}$. ν is therefore undefined when $\underline{e}$ is equal to zero, which is a circular orbit.

Question 4:

Finding the rate of change of specific angular momentum:

$$^{\mathrm{I}}\boldsymbol{h} = \boldsymbol{r} \times {}^{I}\boldsymbol{v}$$

Taking the derivative and applying the chain rule:

$$\frac{^{\mathrm{I}}d}{dt}\left({}^{\mathrm{I}}\boldsymbol{h}\right) = \frac{^{\mathrm{I}}d}{dt}(\boldsymbol{r} \times {}^{I}\boldsymbol{v}) = \frac{^{\mathrm{I}}d\boldsymbol{r}}{dt} \times {}^{I}\boldsymbol{v} + \boldsymbol{r} \times \frac{^{\mathrm{I}}d}{dt}({}^{I}\boldsymbol{v})$$

$$^{I}\boldsymbol{v} = \frac{^{\mathrm{I}}d\boldsymbol{r}}{dt} \qquad \text{and} \qquad {}^{I}\boldsymbol{a} = \frac{^{\mathrm{I}}d}{dt}({}^{I}\boldsymbol{v}) = -\frac{\mu}{r^3}\boldsymbol{r}$$

Substituting these values in and evaluating:

$$\frac{^{\mathrm{I}}d}{dt}\left({}^{\mathrm{I}}\boldsymbol{h}\right) = \boldsymbol{r} \times \left(-\frac{\mu}{r^3}\boldsymbol{r}\right) = 0$$

Finding the rate of change of the eccentricity vector:

$$\boldsymbol{e} = \frac{^{I}\boldsymbol{v} \times {}^{\mathrm{I}}\boldsymbol{h}}{\mu} - \frac{\boldsymbol{r}}{r}$$

Rearranging constants:

$$\mu \boldsymbol{e} = {}^{I}\boldsymbol{v} \times {}^{\mathrm{I}}\boldsymbol{h} - \frac{\mu}{r}\boldsymbol{r}$$

Taking the derivative using the chain rule:

$$\frac{{}^{\mathrm{I}}d}{dt}(\mu \boldsymbol{e}) = \frac{{}^{\mathrm{I}}d}{dt}({}^{I}\boldsymbol{v}) \times {}^{\mathrm{I}}\boldsymbol{h} + {}^{I}\boldsymbol{v} \times \frac{{}^{\mathrm{I}}d}{dt}\left({}^{\mathrm{I}}\boldsymbol{h}\right) - \frac{\mu}{r}\frac{{}^{\mathrm{I}}d\boldsymbol{r}}{dt}$$

Using previous substitutions:

$$\frac{{}^{\mathrm{I}}d}{dt}(\mu \boldsymbol{e}) = {}^{I}\boldsymbol{a} \times {}^{\mathrm{I}}\boldsymbol{h} - \frac{\mu}{r}{}^{I}\boldsymbol{v}$$

$$\frac{{}^{\mathrm{I}}d}{dt}(\mu \boldsymbol{e}) = -\frac{\mu}{r^3}\boldsymbol{r} \times (\boldsymbol{r} \times {}^{I}\boldsymbol{v}) - \frac{\mu}{r}{}^{I}\boldsymbol{v}$$

Applying the anticommutative property of the cross product:

$$\frac{{}^{\mathrm{I}}d}{dt}(\mu \boldsymbol{e}) = -\left(\frac{\mu}{r^3}\boldsymbol{r} \cdot {}^{I}\boldsymbol{v}\right)\boldsymbol{r} + \left(\frac{\mu}{r^3}\boldsymbol{r} \cdot \boldsymbol{r}\right){}^{I}\boldsymbol{v} - \frac{\mu}{r}{}^{I}\boldsymbol{v}$$

Knowing that perpendicular vectors dotted is equal to zero and a vector dotted with itself is equal to the magnitude squared:

$$\frac{{}^{\mathrm{I}}d}{dt}(\boldsymbol{e}) = \frac{\mu}{r}{}^{I}\boldsymbol{v} - \frac{\mu}{r}{}^{I}\boldsymbol{v} = \frac{0}{\mu} = 0$$

Finding the rate of change of the line of nodes:

$$\boldsymbol{n} = \boldsymbol{I}_z \times {}^{I}\boldsymbol{h}$$

Taking the derivative and applying the chain rule:

$$\frac{{}^{\mathrm{I}}d}{dt}(\boldsymbol{n}) = \frac{{}^{\mathrm{I}}d}{dt}(\boldsymbol{I}_z \times {}^{I}\boldsymbol{h}) = \frac{{}^{\mathrm{I}}d}{dt}(\boldsymbol{I}_z) \times {}^{I}\boldsymbol{h} + \boldsymbol{I}_z \times \frac{{}^{\mathrm{I}}d}{dt}({}^{I}\boldsymbol{h})$$

Since $\boldsymbol{I}_z$ is fixed in I, the rate of change is zero, and the rate of change of ${}^{I}\boldsymbol{h}$ has already been proven to be zero:

$$\frac{{}^{\mathrm{I}}d}{dt}(\boldsymbol{n}) = \boldsymbol{0}$$

Question 5-6:

```
clc;clear;

rvec = [0.7;0.6;0.3];
vvec = [-0.8;0.8;0];
mu = 1;

oe = rv2oe_Hackbardt_Chris(rvec,vvec,mu);
a = oe(1);
e = oe(2);
Omega = oe(3);
i = oe(4);
omega = oe(5);
nu = oe(6);

tau = (2*pi)*sqrt(a^3/mu);
hvec = cross(rvec,vvec);
h = norm(hvec);
p = h^2/mu;
energy = -mu/(2*a);

calculatedOE = [a; e; Omega; i; omega; nu];
[calculatedRvec,calculatedVvec]  = oe2rv_Hackbardt_Chris(calculatedOE,mu);

fprintf('Semi-Major Axis (a) [AU]: \t\t\t\t\t\t %16.8f\n',a);
fprintf('Eccentricity (e) [AU]: \t\t\t\t\t\t\t %16.8f\n',e);
fprintf('Longitude of the Ascending Node (Omega) [rad]:   %16.8f\n',Omega);
fprintf('Orbital Inclination (i) [rad]: \t\t\t\t\t %16.8f\n',i);
fprintf('Argument of the Periapsis (omega) [rad]:  \t\t %16.8f\n',omega);
fprintf('True Anomaly (nu) [rad]: \t\t\t\t\t\t %16.8f\n\n',nu);

fprintf(2,'Orbital Period []: \t\t\t\t\t\t\t\t %16.8f\n\n',tau);

fprintf('Semi-Latus Rectum (p) [AU]: \t\t\t\t\t %16.8f\n\n',p);

fprintf('X Component Angular Momentum Vector [kg m^2/s]:  %16.8f\n',hvec(1));
fprintf('Y Component Angular Momentum Vector [kg m^2/s]:  %16.8f\n',hvec(2));
fprintf('Z Component Angular Momentum Vector [kg m^2/s]:
%16.8f\n\n',hvec(3));
fprintf('Magnitude of Angular Momentum [kg m^2/s]: \t\t %16.8f\n\n',h);

fprintf('Orbital Energy  [kg km^2 s^-2]: \t\t\t\t %16.8f\n\n',energy);

fprintf('Using the calculated orbital elements to recalculate the position
and velocity vectors:\n');
fprintf('X Component Position [AU]:  %16.8f\n',calculatedRvec(1));
fprintf('Y Component Position [AU]:  %16.8f\n',calculatedRvec(2));
fprintf('Z Component Position [AU]:  %16.8f\n\n',calculatedRvec(3));

fprintf('X Component Velocity [AU/TU]:  %16.8f\n',calculatedVvec(1));
fprintf('Y Component Velocity [AU/TU]:  %16.8f\n',calculatedVvec(2));
fprintf('Z Component Velocity [AU/TU]:  %16.8f\n\n',calculatedVvec(3));
```

Semi-Major Axis (a) [AU]:	1.27739617
Eccentricity (e) :	0.25118540
Longitude of the Ascending Node (Omega) [rad]:	5.49778714
Orbital Inclination (i) [rad]:	0.31545875
Argument of the Periapsis (omega) [rad]:	1.86539229
True Anomaly (nu) [rad]:	5.91559204
Orbital Period [hours]:	9.07127393
Semi-Latus Rectum (p) [AU]:	1.19680000

X Component Angular Momentum Vector [kg m^2/s]:	-0.24000000
Y Component Angular Momentum Vector [kg m^2/s]:	-0.24000000
Z Component Angular Momentum Vector [kg m^2/s]:	1.04000000
Magnitude of Angular Momentum [kg m^2/s]:	1.09398355

Orbital Energy [kg km^2 s^-2]: -0.39142125

Using the calculated orbital elements to recalculate the position and velocity vectors:

X Component Position [AU]:	0.70000000
Y Component Position [AU]:	0.60000000
Z Component Position [AU]:	0.30000000

X Component Velocity [AU/TU]:	-0.80000000
Y Component Velocity [AU/TU]:	0.80000000
Z Component Velocity [AU/TU]:	0.00000000

Question 7:

```
clc;clear;

a = 15307.548;
e = 0.7;
Omega = 194;
i = 39;
omega = 85;
nu = 48;
mu = 398600;

Omega = deg2rad(Omega);
i = deg2rad(i);
omega = deg2rad(omega);
nu = deg2rad(nu);

oe = [a; e; Omega; i; omega; nu];

[rvec,vvec]  = oe2rv_Hackbardt_Chris(oe,mu);

fprintf('X Component Position [km]:  %16.8f\n',rvec(1));
fprintf('Y Component Position [km]:  %16.8f\n',rvec(2));
fprintf('Z Component Position [km]:  %16.8f\n\n',rvec(3));

fprintf('X Component Velocity [km/s]:  %16.8f\n',vvec(1));
fprintf('Y Component Velocity [km/s]:  %16.8f\n',vvec(2));
fprintf('Z Component Velocity [km/s]:  %16.8f\n\n',vvec(3));
```

X Component Position [km]: 4249.24395473

Y Component Position [km]: -2054.84062287

Z Component Position [km]: 2446.99585787

X Component Velocity [km/s]: 9.07117614

Y Component Velocity [km/s]: 5.81566502

Z Component Velocity [km/s]: -2.79245828

Question 8:

```
clc;clear;

a = 19133.333;
e = 0.5;
Omega = 30;
i = 45;
omega = 45;
nu = 0;
mu = 398600;

Omega = deg2rad(Omega);
i = deg2rad(i);
omega = deg2rad(omega);
nu = deg2rad(nu);

oe = [a; e; Omega; i; omega; nu];

[rvec,vvec]  = oe2rv_Hackbardt_Chris(oe,mu);

fprintf('X Component Position [km]:  %16.8f\n',rvec(1));
fprintf('Y Component Position [km]:  %16.8f\n',rvec(2));
fprintf('Z Component Position [km]:  %16.8f\n\n',rvec(3));

fprintf('X Component Velocity [km/s]:  %16.8f\n',vvec(1));
fprintf('Y Component Velocity [km/s]:  %16.8f\n',vvec(2));
fprintf('Z Component Velocity [km/s]:  %16.8f\n\n',vvec(3));
```

X Component Position [km]: 3466.69624109

Y Component Position [km]: 7524.81548702

Z Component Position [km]: 4783.33325000

X Component Velocity [km/s]: -6.81755776

Y Component Velocity [km/s]: 0.62817226

Z Component Velocity [km/s]: 3.95279202

Question 9:

```
clc;clear;

a = 20000;
e = 0.45;
Omega = 59;
i = 27;
omega = 94;
nu = 58;
mu = 398600;

Omega = deg2rad(Omega);
i = deg2rad(i);
omega = deg2rad(omega);
nu = deg2rad(nu);

oe = [a; e; Omega; i; omega; nu];

[rvec,vvec]  = oe2rv_Hackbardt_Chris(oe,mu);

fprintf('X Component Position [km]:  %16.8f\n',rvec(1));
fprintf('Y Component Position [km]:  %16.8f\n',rvec(2));
fprintf('Z Component Position [km]:  %16.8f\n\n',rvec(3));

fprintf('X Component Velocity [km/s]:  %16.8f\n',vvec(1));
fprintf('Y Component Velocity [km/s]:  %16.8f\n',vvec(2));
fprintf('Z Component Velocity [km/s]:  %16.8f\n\n',vvec(3));
```

X Component Position [km]: -10474.46193267

Y Component Position [km]: -6972.51466993

Z Component Position [km]: 2744.94389648

X Component Velocity [km/s]: 1.12638675

Y Component Velocity [km/s]: -6.03283073

Z Component Velocity [km/s]: -2.07511343

Question 10:

```
clc;clear;

a = 1.6;
e = 0.4;
Omega = 287;
i = 46;
omega = 28;
nu = 139;
mu = 1;

Omega = deg2rad(Omega);
i = deg2rad(i);
omega = deg2rad(omega);
nu = deg2rad(nu);

oe = [a; e; Omega; i; omega; nu];

[rvec,vvec]  = oe2rv_Hackbardt_Chris(oe,mu);

fprintf('X Component Position:  %16.8f\n',rvec(1));
fprintf('Y Component Position:  %16.8f\n',rvec(2));
fprintf('Z Component Position:  %16.8f\n\n',rvec(3));

fprintf('X Component Velocity:  %16.8f\n',vvec(1));
fprintf('Y Component Velocity:  %16.8f\n',vvec(2));
fprintf('Z Component Velocity:  %16.8f\n\n',vvec(3));
```

X Component Position: -0.26075057

Y Component Position: 1.88182968

Z Component Position: 0.31152557

X Component Velocity: -0.46004409

Y Component Velocity: 0.23163947

Z Component Velocity: -0.38544253