Question 1:

Starting with the rate of change of true anomaly:

$$\dot{\nu} = \frac{h}{r^2}$$

The rate of change of true anomaly can be rewritten as:

$$\frac{d\nu}{dt} = \dot{\nu} = \frac{h}{r^2}$$

Solving for dt:

$$dt = \frac{r^2}{h} d\nu$$

We can use the substitutions to make the function depend only on ν, p, e, and μ:

$$r = \frac{p}{1+e\cos\nu} \quad \text{and} \quad h = \sqrt{\mu p}$$

$$dt = \frac{p^2}{\sqrt{\mu p}\,(1 + e\cos\nu)^2} d\nu$$

Integrating both sides gives the result:

$$\int_{t_1}^{t_2} dt = \int_{\nu_1}^{\nu_2} \frac{p^2}{\sqrt{\mu p}\,(1 + e\cos\nu)^2} d\nu$$

$$t_2 - t_1 = \int_{\nu_1}^{\nu_2} \frac{p^2}{\sqrt{\mu p}\,(1 + e\cos\nu)^2} d\nu$$

Question 2:

```
function f = timeChangeIntegrand(nu,p,e,mu)
% -------------------------------------------------------------%
% Integrand f(nu,p,e,mu) that is used to obtain the time change %
% deltat =t2 - t1 %
% Inputs: %
% nu: true anomaly (rad) %
% p: parameter (semi-latus rectum) %
% e: eccentricity %
% mu: gravitational paramter %
% Output: %
% f: value of function at (nu,p,e,mu) %
% -------------------------------------------------------------%

f = (p^2)./(sqrt(mu*p).*(1+e*cos(nu)).^2);

end


function deltat = timeChangeIntegral(f,nu1,nu2,p,e,mu,N)
% -------------------------------------------------------------%
% This function employs Legendre-Gauss quadrature to compute an %
% approximation to                                              %
%                             /nu2                              %
%                            /                                  %
%                           |                                   %
%        deltat = t2 - t1 = | f(nu,p,e,mu)dnu                   %
%                           |                                   %
%                          /                                    %
%                        /nu1                                   %
% where f(nu) is a function that used to define the change in   %
% The inputs and outputs of this function are as follows:       %
% Inputs:                                                       %
% f = a handle to the function to be integrated                 %
% nu1 = lower integration limit                                 %
% = initial true anomaly (rad)                                  %
% nu2 = upper integration limit                                 %
% = terminal true anomaly (rad)                                 %
% p = parameter (semi-latus rectum)                             %
% e = eccentricity                                              %
% mu = gravitational parameter                                  %
% N = number of Gauss points & weights                          %
% used to approximate the integral                              %
% Output:                                                       %
% deltat = Gauss quadrature approximation of                    %
% deltat, where deltat = t2 - t1 is the                         %
% time change from nu1 to nu2                                   %
% -------------------------------------------------------------%

[nus,w] = GaussPointsWeights(nu1,nu2,N);
nus=nus';
F = f(nus,p,e,mu);
F = F';
deltat = w.'*F;

end
```

Question 3:

```
clc;clear;close all;

%Constants that are given
rvec = [5634.297397, -2522.807863, -5037.930889];
vvec = [8.286176, 1.815144, 3.624759];
mu = 398600;
Re = 6378.145;

%Calculates p, e, and orbital period
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);
tau = 2*pi*sqrt(a^3/mu);
period = tau/3600;

%Calculates the angles for ascending and descending nodes using argument of
%periapsis
nvec = cross([0,0,1],hvec);
argPeriapsis = atan2(dot(evec,cross(hvec,nvec)),h*dot(evec,nvec));
if argPeriapsis<0
    argPeriapsis = argPeriapsis + (2*pi);
end
nuAscendingNode = (2*pi)-argPeriapsis;
nuDescendingNode = nuAscendingNode + pi;

%Names the angles as nu1 and nu2 in radians and degrees
nu1rad = nuAscendingNode;
nu2rad = nuDescendingNode;
nu1deg = rad2deg(nuAscendingNode);
nu2deg = rad2deg(nuDescendingNode);

%Calculates deltat from ascending to descending node using functions from
%problem 2
N = [10,15,20,25];
deltat=zeros(length(N),1);
for i=1:length(N)
    deltat(i) =
timeChangeIntegral(@timeChangeIntegrand,nu1rad,nu2rad,p,e,mu,N(i));
end
deltat = deltat/3600;

%Calculates deltat from descending to ascending node only using
%GaussPointsWeights()
nu1rad=nu1rad+2*pi;
deltat2 = zeros(length(N),1);
for i=1:length(N)
    [nus,w] = GaussPointsWeights(nu2rad,nu1rad,N(i));
    for j=1:N(i)
        f = (p^2)/(sqrt(mu*p)*(1+e*cos(nus(j)))^2);
        deltat2(i) = deltat2(i) + (f*w(j));
```

```
    end
end
deltat2 = deltat2/3600;

%Plots the orbit using the orbit equation and the earth using the radius.
%Also marks the ascending and descending nodes
nu = 0:0.1:2*pi;
earth(length(nu))=Re;
earth(:)=Re;
orbitEquation = p./(1+e*cos(nu));
polarplot(nu,orbitEquation,'r')
hold on
polarplot(nu1rad,p/(1+e*cos(nu1rad)),'ro',nu2rad,p/(1+e*cos(nu2rad)),'rs')
hold on
polarplot(nu,earth,'b')

%Prints the results from the calculations
fprintf('----------------------------------------------------------\n');
fprintf('----------------------------------------------------------\n');
fprintf(' Part (a): \n');
fprintf('----------------------------------------------------------\n');
fprintf('----------------------------------------------------------\n');
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',hvec(3));
fprintf('----------------------------------------------------------\n');
fprintf('Magnitude of Angular Momentum [kg m^2/s]: \t %16.8f\n',h);
fprintf('----------------------------------------------------------\n');
fprintf(2,'Semi-Latus Rectum (p) [km]: \t\t\t %16.8f\n',p);
fprintf('----------------------------------------------------------\n');
fprintf('Magnitude of Radius at Given Point [km]:%16.8f\n',r);
fprintf('----------------------------------------------------------\n');
fprintf('X Component Eccentricity Vector:\t\t%16.8f\n',evec(1));
fprintf('Y Component Eccentricity Vector:\t\t%16.8f\n',evec(2));
fprintf('Z Component Eccentricity Vector:\t\t%16.8f\n',evec(3));
fprintf('----------------------------------------------------------\n');
fprintf(2,'Eccentricity (e): \t\t\t\t\t %16.8f\n',e);
fprintf('----------------------------------------------------------\n');
fprintf('Semi-Major Axis (a) [km]: \t\t\t %16.8f\n',a);
fprintf('----------------------------------------------------------\n');
fprintf('Orbital Period [seconds]: \t\t\t %16.8f\n',tau);
fprintf('----------------------------------------------------------\n');
fprintf(2,'Orbital Period [hours]: \t\t\t %16.8f\n', period);
fprintf('----------------------------------------------------------\n');
fprintf('----------------------------------------------------------\n');
fprintf(' Part (b): \n');
fprintf('----------------------------------------------------------\n');
fprintf('----------------------------------------------------------\n');
fprintf('True Anomaly of Ascending Node: %g deg\n',nu1deg);
fprintf('----------------------------------------------------------\n');
fprintf('True Anomaly of Descending Node: %g deg\n',nu2deg');
fprintf('----------------------------------------------------------\n');
fprintf('----------------------------------------------------------\n');
fprintf(' Part (c): Time change from nu1=%g deg to nu2=%g deg
\n',nu1deg,nu2deg);
fprintf('----------------------------------------------------------\n');
```

```
fprintf('-----------------------------------------------------------\n');
for i=1:length(N)
    fprintf('Time elapsed (%g,%g) deg [hours]
(N=%i):%16.8f\n',nu1deg,nu2deg,N(i),deltat(i));
end
fprintf('-----------------------------------------------------------\n');
fprintf('-----------------------------------------------------------\n');
fprintf(' Part (d): Time change from nu2=%g deg to nu1=%g deg
\n',nu2deg,nu2deg+180);
fprintf('-----------------------------------------------------------\n');
fprintf('-----------------------------------------------------------\n');
for i=1:length(N)
  fprintf('Time elapsed (%g,%g) deg [hours]
(N=%i):%15.8f\n',nu2deg,nu2deg+180,N(i),deltat2(i));
end
fprintf('-----------------------------------------------------------\n');
fprintf('-----------------------------------------------------------\n');
fprintf(' Part (f): \n');
fprintf(' The orbital period is half of the rotational period of   \n');
fprintf(' earth. This means the spacecraft will orbit the earth    \n');
fprintf(' twice everyday. It also means that the spacecraft will   \n');
fprintf(' cross periapsis over the same two points everyday,       \n');
fprintf(' switching back and forth between the two every crossing. \n');
fprintf('-----------------------------------------------------------\n');
fprintf('-----------------------------------------------------------\n');
```

Output:

Part (a):

X Component Angular Momentum Vector [kg m^2/s]:   -0.00048110

Y Component Angular Momentum Vector [kg m^2/s]: -62168.15222054

Z Component Angular Momentum Vector [kg m^2/s]: 31131.49108138

Magnitude of Angular Momentum [kg m^2/s]:   69527.32475413

Semi-Latus Rectum (p) [km]:   12127.56870915

Magnitude of Radius at Given Point [km]:   7968.09979316

X Component Eccentricity Vector: -0.00000014

Y Component Eccentricity Vector: -0.33055414

Z Component Eccentricity Vector: -0.66010137

Eccentricity (e): 0.73824106

Semi-Major Axis (a) [km]: 26653.98916786

Orbital Period [seconds]: 43306.64565920

Orbital Period [hours]: 12.02962379

Part (b):

True Anomaly of Ascending Node: 90 deg

True Anomaly of Descending Node: 270 deg

Part (c): Time change from nu1=90 deg to nu2=270 deg

Time elapsed (90,270) deg [hours] (N=10): 11.09570134

Time elapsed (90,270) deg [hours] (N=15): 11.10162308

Time elapsed (90,270) deg [hours] (N=20): 11.10156632

Time elapsed (90,270) deg [hours] (N=25): 11.10156681

Part (d): Time change from nu2=270 deg to nu1=450 deg

Time elapsed (270,450) deg [hours] (N=10): 0.92805699

Time elapsed (270,450) deg [hours] (N=15): 0.92805699

Time elapsed (270,450) deg [hours] (N=20): 0.92805699

Time elapsed (270,450) deg [hours] (N=25): 0.92805699

Part (f):

The orbital period is half of the rotational period of earth. This means the spacecraft will orbit the earth twice every day. It also means that the spacecraft will cross periapsis over the same two points every day, switching back and forth between the two every crossing.