```
function TA_end = orbit_prop(e,TA,a,t,mu)
%ORBIT_PROP Propogates orbit forward t seconds
%   Inputs are:
%   h       :a scalar specific angular momentum in m^2/s
%   e       :a scalar orbital eccentricy
%   RA      :a scalar right ascension of the ascending node in rad
%   incl    :a scalar orbital inclination
%   W       :a scalar argument of perigee in rad
%   TA      :a scalar true anomaly in rad
%   a       :a scalar semi major axis in m
%   t       :a scalar propogation time in seconds
%   mu      :an optional scalar gravitational parameter in m^3/s^2 (default
%            earth)
%
%   Outputs are:
%   TA      :a scalar propogated true anomaly in rad

    arguments
        e {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        TA {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        a {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        t {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal}
        mu {mustBeScalarOrEmpty, mustBeNumeric, mustBeReal} = 3.98600442e14
    end

    E = 2*atan(sqrt((1-e)/(1+e))*tan(TA/2));
    M = E-e*sin(E);

    M_end = sqrt((mu)/a^3)*t + M;

    E_end = kepler_M2E(e,M_end);
    TA_end = 2*atan(sqrt((1+e)/(1-e))*tan(E_end/2));
end
```