```
function [m_in1, m_in2, m_pr1, m_pr2, m0] = mass_function(Isp1, Isp2, X, delta1, delta2)
    % Returns the inert mass of stage 1, inert mass of stage 2, propellant
    % mass of stage 1, propellant mass of stage 2, and total mass given
    % specific impulse of stage 1, specific impulse of stage 2, and delV
    % fraction (delta1, delta2 are assumed to be 0.08)

    % Set constants based on mission requirements
    deltaV_tot = 12300; % m/s
    g0 = 9.81; % m/s^2
    m_pl = 26000; % kg

    % Solve for deltaV1 and deltaV2 from given delV fraction
    deltaV1 = X*deltaV_tot;
    deltaV2 = (1-X)*deltaV_tot;

    % Solve for mass ratio of stages 1 and 2 (m_i/m_f definition)
    R1 = exp(deltaV1/(g0*Isp1));
    R2 = exp(deltaV2/(g0*Isp2));

    % If (1 - R1*delta1) or (1 - R2*delta2) is less than 0, then the design
    % is structurally impossible based on the definitions
    % for m_2 and m_1 that follow, return NaN for all outputs
    if (1 - R1*delta1) <= 0 || (1 - R2*delta2) <= 0
        m_in1 = NaN; m_in2 = NaN; m_pr1 = NaN; m_pr2 = NaN; m0 = NaN;
        return
    end

    % Solve for total masses of stage 1 and stage 2
    m_2 = R2*m_pl/(1 - R2*delta2);
    m_1 = R1*m_2/(1 - R1*delta1);

    % Solve for inert masses of stage 1 and stage 2
    m_in1 = delta1*m_1;
    m_in2 = delta2*m_2;

    % Solve for propellant masses of stage 1 and stage 2
    m_pr1 = m_1 - m_in1 - m_2;
    m_pr2 = m_2 - m_pl - m_in2;

    % Assign total mass as the total mass of stage 1 (m_1 includes m_2)
    m0 = m_1;
end
```

```
Not enough input arguments.

Error in mass_function (line 13)
    deltaV1 = X*deltaV_tot;
              ^
```