```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function P1
%% In EMOS Units
        r1 = 1; % Mars Semi-Major Axis [AU]
        r2 = 1.5237; % Mars Semi-Major Axis [AU]
        mu = 1; % Standard Gravitation of Sun [AU^3/TU^2]

        disp('Normalized Units')
        [a_H,t_H,dv_1, dv_2, dv_tot] = HohmannTransfer(r1,r2,mu);


        mfmo = exp(-dv_tot/0.1);
        mpmo = 1-mfmo;

        eps = 1/7;
        mo=100;
        mp = 84.71;

        ms = (eps*mp)/(1-eps);
        ml = 100-(mp+ms);

%% In Standard Units
        r1 = 149600000; % Earth Semi-Major Axis [km]
        r2 = 227940000; % Mars Semi-Major Axis [km]
        mu = 1.327e11; % Standard Gravitation of Sun [km^3/s^2]

        disp('Standard Units')
        [a_H,t_H,dv_1, dv_2, dv_tot] = HohmannTransfer(r1,r2,mu);

end

%%%%%%%%%%%%%%%%%%%%%%%
% Hohmann Transfer
%%%%%%%%%%%%%%%%%%%%%%%

function [a_H,t_H,dv_1, dv_2, dv_tot] = HohmannTransfer(r1,r2,mu)
        a_H = (r1+r2)/2;
        t_H = pi*sqrt((a_H^3)/mu);

        dv_1 = sqrt(mu*((2/r1)-(1/a_H))) - sqrt(mu/r1);
        dv_2 = sqrt(mu/r2) - sqrt(mu*((2/r2)-(1/a_H)));
        dv_tot = abs(dv_1) + abs(dv_2);
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function P2
        r1 = 400+6378;
        r2 = 382900;
        mu = 3.985e5;

%% Quasi-Hohmann
        [a_H,t_H,dv_1, dv_2, dv_totH] = HohmannTransfer(r1,r2,mu);

%% Quasi-Bi-Elliptic
        ri = 3*0.3844e6;
        [a_1,a_2,dv_1,dv_i,dv_2,dv_totB,R,t1,t2,ttot] = BiEllipticTransfer(r1,r2,ri,mu);
        (1-(dv_totB/dv_totH))*100
end

%%%%%%%%%%%%%%%%%%%%%%%
% Bi-Elliptic Transfer
%%%%%%%%%%%%%%%%%%%%%%%

function [a_1,a_2,dv_1,dv_i,dv_2,dv_tot,R,t1,t2,ttot] = BiEllipticTransfer(r1,r2,ri,mu)
        a_1 = (r1+ri)/2;
        a_2 = (ri+r2)/2;

        dv_1 = sqrt(mu*((2/r1)-(1/a_1))) - sqrt(mu/r1);
        dv_i = sqrt(mu*((2/ri)-(1/a_2))) - sqrt(mu*((2/ri)-(1/a_1)));
        dv_2 = sqrt(mu/r2) - sqrt(mu*((2/r2)-(1/a_2)));
        dv_tot = abs(dv_1) + abs(dv_2) + abs(dv_i);

        R = r2/r1;

        t1 = pi*sqrt((a_1^3)/mu);
        t2 = pi*sqrt((a_2^3)/mu);
        ttot = t1+t2;
end
```

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 3A
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function P3A
        % Program to optimize two-stage rocket mass
        x0=[20000,5000,3000];
        %options=optimset('LargeScale','off','display','iter');
        options=optimset('LargeScale','off');
        x=fmincon(@objfun3,x0,[],[],[],[],[],[],@confuneq3,options);
        x(1)
        x(2)
        x(3)
        f=objfun3(x)
        [c,ceq]=confuneq3(x);
end

% objfun3.m
% Objective function (total mass) for optimal rocket problem
function f=objfun3(x)
        mp=1000;
        f=x(1)+x(2)+x(3)+mp;
end

% confuneq3.m
% constraint equation for optimal rocket problem
function [c,ceq]=confuneq3(x)
        c1 = 3500;
        c2 = 3800;
        c3 = 4100;
        e1 = 0.10;
        e2 = 0.12;
        e3 = 0.09;
        mp = 1000;
        vfinal = 8500;

        ceq =   c1*log((x(1)+x(2)+x(3)+mp)/(e1*x(1)+x(2)+x(3)+mp))+...
                c2*log((x(2)+x(3)+mp)/(e2*x(2)+x(3)+mp))+...
                c3*log((x(3)+mp)/(e3*x(3)+mp))-vfinal;
        c=[];
end
```

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 3B
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function P3B
% Program to optimize two-stage rocket mass
        x0=[20000,5000,3000];
        %options=optimset('LargeScale','off','display','iter');
        options=optimset('LargeScale','off');
        x=fmincon(@oobjfun3,x0,[],[],[],[],[],[],@cconfuneq3,options);
        x(1)
        x(2)
        x(3)
        f=oobjfun3(x)
        [c,ceq]=cconfuneq3(x);
end

% objfun3.m
% Objective function (total mass) for optimal rocket problem
function f=oobjfun3(x)
        mp=1000;
        f=x(1)+x(2)+x(3)+mp;
end

% confuneq3.m
% constraint equation for optimal rocket problem
function [c,ceq]=cconfuneq3(x)
        c1 = 3500;
        c2 = 3800;
        c3 = 4100;
        e1 = 0.10;
        e2 = 0.12;
        e3 = 0.09;
        mp = 1000;
        vfinal = 8500;
        g = 9.81;
        t = 90;

        ceq =   c1*log((x(1)+x(2)+x(3)+mp)/(e1*x(1)+x(2)+x(3)+mp))-g*t+...
                c2*log((x(2)+x(3)+mp)/(e2*x(2)+x(3)+mp))+...
                c3*log((x(3)+mp)/(e3*x(3)+mp))-vfinal;

        c=[];
end
```