```matlab
% Sounak (Shaun) Ghosh
% 11/05/19
% ECE 202 - Fall 2019 - MATLAB Exercise M6
% Equation source: http://www.convertalot.com/elastic_collision_calculator.html
% MATLAB script to determine the number of collisions in a system of cart
% using User-Defined Functions

clear      % clears all variables in the workspace; avoids common errors
clc        % clears all previous outputs in the command window

M = [240 120 360];  % mass of the carts in g
Vi = [30 15 -45];  % initial velocity of carts in cm/s

Momentum_i = sum(M.*Vi); % Initial Momentum of the total system.
Energy_i = sum(0.5*M.*Vi.^2); % Initial Energy of the total system.

% From the diagram M2 & M3 are going to collide first.
% After Collision a

[Va(2), Va(3)] = final_velocity(M(2), M(3), Vi(2), Vi(3));

Va(1) = Vi(1)                              % velocity of carts after collision a

CheckMomentum_a = sum(M.*Va) - Momentum_i              % The change in total energy
                                                          % momentum of the system before
CheckEnergy_a = sum(0.5*M.*Va.^2) - Energy_i           % & after the collision should
                                                          % be zero.
% The velocities after collision A indicate carts #1 & #2 are going to collide.
% After Collision b.

[Vb(1), Vb(2)] = final_velocity(M(1), M(2), Va(1), Va(2));

Vb(3) = Va(3)                                    % velocity of carts after collision a

CheckMomentum_b = sum(M.*Vb) - Momentum_i                 % The change in total energy
                                                          % momentum of the system before
CheckEnergy_b = sum(0.5*M.*Vb.^2) - Energy_i    % & after the collision should
                                                          % be zero.

% The velocities after collision B indicate carts #2 & #3 are going to collide.
% After Collision c.

[Vc(2), Vc(3)] = final_velocity(M(2), M(3), Vb(2), Vb(3));

```

```
Vc(1) = Vb(1)                                              % velocity of carts after collision a

CheckMomentum_c = sum(M.*Vc) - Momentum_i                  % The change in total energy
                                                           % momentum of the system before
CheckEnergy_c = sum(0.5*M.*Vc.^2) - Energy_i   % & after the collision should
                                                           % be zero.
                                                           % After Collision c

% The velocities after collision C indicate carts #1 & #2 are going to collide.
% After Collision D.

[Vd(1), Vd(2)] = final_velocity(M(1), M(2), Vc(1), Vc(2));

Vd(3) = Vc(3)                             % velocity of carts after collision a

CheckMomentum_d = sum(M.*Vd) - Momentum_i                  % The change in total energy
                                                           % momentum of the system before
CheckEnergy_d = sum(0.5*M.*Vd.^2) - Energy_i   % & after the collision should
                                                           % be zero.



function [v1f, v2f] = final_velocity(m1, m2, v1i, v2i)
    M = m1 + m2;
    v1f = (m1*v1i - m2*(v1i - 2*v2i))/M;    % final velocity of left cart
    v2f = (m1*(2*v1i - v2i) + m2*v2i)/M;    % final velocity of right cart
end

% The final velocities of the carts 1, 2 and 3 after collision D are -50
% cm/s, -35 cm/s and 25 cm/s respectively, which indicate that cart#1 adn
% cart#2 are moving in the same direction with cart#1 moving faster than
% cart 2, hence carts #1 and #2 will never collide, and cart#3 is moving in
% the opposite direction of cart#1 and cart#2, therefore cart#3 won't have
% a collision with either of the other carts. In  conclusion,
% we can determine that the carts will not have any further collisions.
```

```

Va =

    30   -75   -15


CheckMomentum_a =

     0


CheckEnergy_a =

     0


Vb =

   -40    65   -15


CheckMomentum_b =

     0


CheckEnergy_b =

     0


Vc =

   -40   -55    25


CheckMomentum_c =

     0


CheckEnergy_c =

     0

```

```
Vd =

   -50   -35    25


CheckMomentum_d =

     0


CheckEnergy_d =

     0

>>
```