```
% *** Aidan Chin, 9/22/23, etc.

    %   *** Label carts from left to right as 1, 2, 3
    %   *** Label "collisions" A, B, C, etc. until everything is final
    %   *** do not use loops or functions
    %   *** use arrays to keep track of everything
    %   *** Formulas from M1, http://hyperphysics.phy-astr.gsu.edu/hbase/elacol2.html

clear

% ----- Getting Started -----

% givens

m = [ 300 60 240 ] % mass of cars in g from left to right
v0 = [ 36 9 -45 ] % velocity of cars in cm/s from left to right

% set up total masses for two types of collisions

m23 = m(2)+m(3); % total mass of carts 2 and 3 in g
m12 = m(1)+m(2); % total mass of carts 1 and 2 in g

% set up the checks by computing total energy and momentum

KE0 = sum(.5.*m.*v0.^2); %calculates the total kenetic energy initially
P0 = sum(m.*v0);         %calculates the total potential energy initially


% ----- Collision #1 -----

% *** carts 2 and 3 will collide

vA = [ 0 0 0 ];    % initialize vA

vA(1) = v0(1);  %no interaction - velocity of cart 1 remains constant in cm/s
vA(2) = (m(2)-m(3))/m23*v0(2) + (2*m(3)/m23)*v0(3); %resultant velocity of
                                                     %cart 2 in cm/s
vA(3) = 2*m(2)/m23*v0(2) - (m(2)-m(3))/m23*v0(3)  %resultant velocity of
                                                     %cart 3 in cm/s

% check energy and momentum (expectations)

KEA = sum(.5.*m.*vA.^2); %calculates the total kinetic energy of situation
checkKE_A = KEA-KE0 % Should be zero because no kenetic energy
                                    % is added or removed
PA = sum(m.*vA); %calculates the total potential energy of situation
checkP_A = PA-P0          % should be zero because no potential energy
                                    % added or removed

% check to see if there is another collision and output result

```

```
    %   *** check BOTH pairs of adjacent carts, even if you know which is next
    %   *** use IF/ELSE to output something simple yet meaningful.
    %   *** end the IF/ELSE statement before starting the next collision

if vA(1) <= vA(2) && vA(3) >= vA(2) %checks if carts will never collide again
    fprintf("There are no more collisions\n")
elseif vA(1) > vA(2) || vA(3) < vA(2) % checks if carts will collide
    fprintf("There is another collision\n")
end


% ----- Collision #2 -----

% *** cart 1 and 2 will collide

vB = [ 0 0 0 ]; % initialize vB

vB(1) = (m(1)-m(2))/m12*vA(1) + (2*m(2))/m12*vA(2); %resultant velocity of
                                                     %cart 1 in cm/s
vB(2) = (2*m(1)/m12)*vA(1) - (m(1)-m(2))/m12*vA(2); %resultant velocity of
                                                     %cart 2 in cm/s
vB(3) = vA(3) %no interaction - velocity of cart 3 remains constant in cm/s

% check energy and momentum (expectations)

KEB = sum(.5.*m.*vB.^2);%calculates the total kinetic energy of situation
checkKE_B = KEB-KE0 % Should be zero because no kenetic energy
                                  % is added or removed
PB = sum(m.*vB); %calculates the total potential energy of situation
checkP_B = PB-P0          % should be zero because no potential energy
                                  % added or removed

% check to see if there is another collision and output result

if vB(1) <= vB(2) && vB(3) >= vB(2) %checks if carts will never collide again
    fprintf("There are no more collisions\n")
elseif vB(1) > vB(2) || vB(3) < vB(2) % checks if carts will collide
    fprintf("There is another collision\n")
end


% ----- Collision #3 -----

% *** carts 2 and 3 will collide

vC = [ 0 0 0 ];    % initialize vC

vC(1) = vB(1); %no interaction - velocity of cart 1 remains constant in cm/s
vC(2) = (m(2)-m(3))/m23*vB(2) + (2*m(3)/m23)*vB(3); %resultant velocity of
                                                     %cart 2 in cm/s
vC(3) = (2*m(2)/m23)*vB(2) - (m(2)-m(3))/m23*vB(3)  %resultant velocity of
```

```
                                                    %cart 3 in cm/s

% check energy and momentum (expectations)

KEC = sum(.5.*m.*vC.^2);%calculates the total kinetic energy of situation
checkKE_C = KEC-KE0       % Should be zero because no kenetic energy
                                    % is added or removed
PC = sum(m.*vC); %calculates the total potential energy of situation
checkP_C = PC-P0          % should be zero because no potential energy
                                    % added or removed

% check to see if there is another collision and output result

if vC(1) <= vC(2) && vC(3) >= vC(2) %checks if carts will never collide again
    fprintf("There are no more collisions\n")
elseif vC(1) > vC(2) || vC(3) < vC(2) % checks if carts will collide
    fprintf("There is another collision\n")
end

% ----- Collision #4 -----

% *** cart 1 and 2 will collide

vD = [ 0 0 0 ]; % initialize vD

vD(1) = (m(1)-m(2))/m12*vC(1) + (2*m(2))/m12*vC(2); %resultant velocity of
                                                    %cart 1 in cm/s
vD(2) = (2*m(1)/m12)*vC(1) - (m(1)-m(2))/m12*vC(2); %resultant velocity of
                                                    %cart 2 in cm/s
vD(3) = vC(3) %no interaction - velocity of cart 3 remains constant in cm/s

% check energy and momentum (expectations)

KED = sum(.5.*m.*vD.^2);%calculates the total kinetic energy of situation
checkKE_D = KED-KE0 % Should be zero because no kenetic energy
                                    % is added or removed
PD = sum(m.*vD); %calculates the total potential energy of situation
checkP_D = PD-P0          % should be zero because no potential energy
                                    % added or removed

% check to see if there is another collision and output result

if vD(1) <= vD(2) && vD(3) >= vD(2) %checks if carts will never collide again
    fprintf("There are no more collisions\n")
elseif vD(1) > vD(2) || vD(3) < vD(2) % checks if carts will collide
    fprintf("There is another collision\n")
end

% ----- Collision #5 -----

% *** carts 2 and 3 will collide
```

```

vE = [ 0 0 0 ];    % initialize vE

vE(1) = vD(1); %no interaction - velocity of cart 1 remains constant in cm/s
vE(2) = (m(2)-m(3))/m23*vD(2) + (2*m(3)/m23)*vD(3); %resultant velocity of
                                                    %cart 2 in cm/s
vE(3) = (2*m(2)/m23)*vD(2) - (m(2)-m(3))/m23*vD(3)  %resultant velocity of
                                                    %cart 3 in cm/s

% check energy and momentum (expectations)

KEE = sum(.5.*m.*vE.^2);%calculates the total kinetic energy of situation
checkKE_E = KEE-KE0       % Should be zero because no kenetic energy
                                    % is added or removed
PE = sum(m.*vE); %calculates the total potential energy of situation
checkP_E = PE-P0          % should be zero because no potential energy
                                    % added or removed

% check to see if there is another collision and output result

if vE(1) <= vE(2) && vE(3) >= vE(2) %checks if carts will never collide again
    fprintf("There are no more collisions\n")
elseif vE(1) > vE(2) || vE(3) < vE(2) % checks if carts will collide
    fprintf("There is another collision\n")
end

% *** Keep adding similar code until there are no more collisions

% *** There are 5 collisions before no more are possible
```