```
clear all
clc

% Parameters
syms m M g L F real
% Variables
syms x  theta1  theta2 real
syms dx dtheta1 dtheta2 real

% Define symbolic variable q for the generalized coordinates
% x, theta1 and theta2
q = [x; theta1; theta2];
% Define symbolic variable dq for the derivatives
% of the generalized coordinates
dq = [dx; dtheta1; dtheta2];
% Write the expressions for the positions of the masses
p{1} = [x - L*sin(theta1); -L*cos(theta1)];
p{2} = p{1} + [-L*sin(theta2); -L*cos(theta2)];

% Kinetic energy of the cart
T = 0.5 * m * dx^2;
% For loop that adds the kinetic energies of the masses
dp = cell(length(p), 1);

for k = 1:length(p)
    dp{k} = jacobian(p{k},q)*dq; % velocity of mass k
    T = T + 0.5 * dp{k}' * M * dp{k}; % add kinetic energy of mass k
end
T = simplify(T);

% Potential energy of the cart
V = 0;
% For loop that adds the potential energies of the masses
for k = 1:length(p)
    V = V + M*g*p{k}(2); % add potential energy of mass k
end
V = simplify(V);

% Generalized forces
Q = [F; 0; 0];

% Lagrangian
Lag = T - V;

Lag_q = simplify(jacobian(Lag,q)).';
Lag_qdq = simplify(jacobian(Lag_q.',dq));
Lag_dq = simplify(jacobian(Lag,dq)).';
Lag_dqdq = simplify(jacobian(Lag_dq.',dq));

% The equations have the form W*q_dotdot = RHS, with
W = Lag_dqdq;
RHS = Q + simplify(Lag_q - Lag_qdq*dq);
```

```
state = [q;dq];
param = [m;M;L;g];

matlabFunction(p{1},p{2}, 'file','PendulumPosition','vars',{state,
 param});
matlabFunction(W,RHS, 'file','PendulumODEMatrices','vars',
{state,F,param});
```

Published with MATLAB® R2019a