```matlab
clear all
close all
clc

%%% FILL IN ALL PLACES LABELLED "complete"

syms rho theta psi real
syms drho dtheta dpsi real

A      = [rho;theta;psi];
dA     = [drho;dtheta;dpsi];

% rotation about x
R{1} = [1    0          0;
        0    cos(rho)   -sin(rho);
        0    sin(rho)   cos(rho)];

% rotation about y
R{2} = [cos(theta)  0    sin(theta);
        0           1    0;
        -sin(theta) 0    cos(theta)];

% rotation about z
R{3} = [cos(psi)    -sin(psi)   0;
        sin(psi)    cos(psi)    0;
        0           0           1];

%Rotation matrix
Rba = simplify(R{1} * R{2} * R{3});

%Time deriviatve of the rotation matrix (Hint: use
%the function "diff" to differentiate the matrix w.r.t. the angles
%rho, theta, psi one by one, and form the whole time derivative using the chain rule and summing the deriviatives)
dRba = diff(Rba, rho) * drho + diff(Rba, theta) * dtheta + diff(Rba, psi) * dpsi;

% Use the formulat relating Rba, dRba and OmegaX (skew-symmetric matrix underlying the angular velocity omega)
OmegaX_b = Rba.' * dRba;

% Extract the angular velocticy vector omega (3x1) from the matrix OmegaX (3x3)
omega = [OmegaX_b(3,2); OmegaX_b(1,3); OmegaX_b(2,1)];

% This line generates matrix M in the relationship omega = M*dA
M = jacobian(omega,dA)

% This line creates a Matlab function returing Rba and M for a given A  = [rho;theta;psi], can be called using [Rba,M] = Rotations(state);
matlabFunction(Rba,M,'file','Rotations','vars',{A})
```