%%%%%README%%%%%
%%%%%%%%%%%%%%%

This homework covers number 40-47

num42
Many files here:

- vecrot- part a. Function
- rotmat- part b. Function
- axisAngle- part d. Fuction
- num42_tests- this tests whether the above files are working for some cases.

- vecrot takes the vector cross product given an axis n, theta in degrees, and vector r.
- rotmat finds the rotation matrix about an axis n and for rotation of theta.
- axisAngle takes a rotation matrix and finds the unit axis of rotation and the theta of rotation in degrees.

So the tests for each case is given by testVecrot, testRotmat, and testAxisAngle. testVecrot and testRotmat is combined into testbc, in which the first column is testVecrot and the second column is testRotmat. The row number of testbc corresponds to the test number. Basically, checked that all the unit vectors subjected to the rotation about x,y,z, and [1,1,1]' mapped correctly.

For part d, similar test structure, but checking that the axis and angle were properly mapped.

One thing I noticed was that the output of the functions were sometimes doubles/ not ints. To deal with this, round was used within the num42_tests.
Stupid mistake that took me way too long to catch: used n instead of nhat.

num43
Using Plot3 to draw a box with sides b,w, and h and a line through the box in direction [1 1 1]' and rotating it by 120 degrees. Fun times animation.

Hard part of this is drawing the box and figuring out the edges to draw. Looking back, I should've made this a function because I needed to make this in num45 and num47.

num44_eulerAngles
Finds the Euler angles using 2 ways:
1. Rotation about the fixed x axis, then fixed y axis, then fixed z axis.
2. ROtation about the fixed z axis, then new y axis, then new x axis.

Rnet_fixed is the combined rotation matrix required to do 1.
Rx_new is the combined rotation matrix required to do 2.

I compared the two by subtracting the two from each other, do simplifying that resulting matrix. I'm not sure why they are equivalent by this metric, but looking at the component on the 3rd column and 1st row they seem different, even though I used simplify for both Rnet_fixed and Rx_new.

This outputs to a matlab function because you need it for num45_animation. I throw a thing in here to convert the angle input from rads to degrees to match the fact that I do all my functions with degrees.

num45_animation
Most of this code is the same as num43, except the rotation matrix comes from num44_eulerAngle and three time varying rotation. So basically combines 43 and 44.

The figure it outputs is the animation.

num47_eulerEquations
This draws a long box that has an I = [1 0 0; 0 2 0; 0 0 3] and initial conditions angular velocity omegaVector_0 = [0.01, 1, 0]' and initial rotation matrix of eye(3). This required using ode45 to iterate to find the rotation matrix at each timestep. Hardest part was trying to get the dimensions of all the vectors and tensors right for putting into ode45 and within ode45.

The figure it outputs is the animation.