HW 4

1: Double trouble Suppose $A \in \mathbb{R}^{n\times n}$ is a real matrix. Unless A has some special structure, there will usually either be either a unique real eigenvalue with largest modulus or a complex conjugate pair of eigenvalues with largest modulus. We will write a routine based on a two-vector orthogonal iteration that

1. Determines whether there is a dominant real eigenvalue or complex conjugate pair of A.
2. Returns the eigenvector in the real case *or* an orthonormal basis for a two-dimensional eigenspace corresponding to a complex conjugate pair of eigenvalues.
3. Returns the eigenvalue λ or a 2-by-2 matrix L whose eigenvalues are the dominant complex conjugate pair of eigenvalues for A.

In addition to the normal return values, you should think about ways in which the code can fail and describe them in your writeup. In the case of a detectable failure (e.g. a failure of the iteration to meet the desired error criterion within the maximum iteration count), you should provide an error message using the MATLAB `error` function. You may also want to use the `warning` function to report anything "suspicious." Please make your error messages informative. Include test cases that exercise all paths through your code, including any error or warning cases.

Your code should have the following interface:

```
function [V,L] = p1eig(A, tol, maxit)
%
% For a real matrix A, use two-vector orthogonal iteration
% to compute an orthonormal basis for either a dominant
% one-dimensional subspace (if the maximal modulus eigenvalue
% is unique) or a dominant two-dimensional subspace (if there
% is a complex conjugate pair).  Raises an error condition if
% the criterion norm(A*V-V*L, 'fro') < tol cannot be met
% within maxit iterations.
```

This exercise will be counted as two problems for grading.

2: Extracting eigenvectors Suppose $A = U^*TU$ is a given Schur decomposition and $\lambda_k = t_{kk}$ is a simple eigenvalue. Write an $O(n^2)$ routine to compute left and right eigenvectors associated with λ_i. Your routine should have the form

```
function [w, v] = p2vectors(T, U, k)
%
% Return w and v such that
%   w'*A = T(k,k)*w'
%   A*v = v*T(k,k)
```

3: Converging QR Plot the diagonal of the iterates in the unshifted QR iteration for the following matrices

1. A 6-by-6 matrix of standard normals (`A = randn(6);`)
2. `B = randn(6); A = B*diag(1:6)/B;`
3. `A = [1, 10; -1, 1];`

In each case, take enough iterations that the curves converge or cycle. What do you observe?