Problem 1

Part A

Listing 1: Function Definition for Banded

```
function B = Banded(A, Type)
%    Input: An nxn matrix A, and a string
%    corresponding to one of 6 banded/sparse
%    matrix types.
%    Output: A sparse matrix of type 'Type'.

switch Type
    case 'Diagonal'
        B = diag(diag(A));
    case 'Tridiagonal'
        B = diag(diag(A,-1))+diag(diag(A))+diag(diag(A,1));
    case 'Upper Triangular'
        B = triu(A);
    case 'Upper Hessenberg'
        B = triu(A, -1);
    case 'Lower Triangular'
        B = tril(A);
    case 'Lower Hessenberg'
        B = tril(A, 1);
    otherwise
        warning('Unknown type, no output created.');
end
end
```

Part B

Listing 2: TestBanded Defintion and Sample Output

```
%% TestBanded = @(A, Type) isequal(A, Banded(A, Type));


>> A = ones(9)
>> TestBanded = @(A, Type) isequal(A, Banded(A, Type));
>> TestBanded(A, 'Lower Triangular')
ans = 0
>> B = Banded(A, 'Lower Triangular')
>> C = Banded(A, 'Upper Hessenberg')
>> TestBanded(B, 'Lower Triangular')
ans = 1
>> TestBanded(C, 'Upper Hessenberg')
ans = 1
>> TestBanded(A, 'Diagonal')
ans = 0
```

Problem 2

Part A

Listing 3: Function Definition for YXZ Rotation

```
function R = RotationYXZ( phi, theta, psi )
% Input: three angles corresponding to rotation
% about Z (phi), X (theta), and Y (psi).
% The function returns a rotation matrix that
% will rotate any 3-vector in the order of YXZ.
P = [cos(psi) 0   -sin(psi)
     0      1       0
     sin(psi) 0    cos(psi)];

R = [1      0       0
     0    cos(theta) -sin(theta)
     0    sin(theta)  cos(theta)];

Y = [cos(phi) -sin(phi) 0
     sin(phi)  cos(phi) 0
     0      0       1  ];

R = P*R*Y;

end
```

Listing 4: Function Definition for ZXZ Rotation

```
function R = RotationZXZ( phi, theta, psi )
% Input: three angles corresponding to rotation
% about Z (phi), X (theta), and Z (psi).
% The function returns a rotation matrix that
% will rotate any 3-vector by the desired angles
% in order of ZXZ.
Y1= [cos(psi) -sin(psi) 0
     sin(psi)  cos(psi) 0
     0      0       1  ];

R = [1      0       0
     0    cos(theta) -sin(theta)
     0    sin(theta)  cos(theta)];

Y2= [cos(phi) -sin(phi) 0
     sin(phi)  cos(phi) 0
     0      0       1  ];

R = Y1*R*Y2;
end
```

Sample output has been omitted for the sake of structure and brevity. See appendix for sample output.

Part B

Prove that $R_{123}(\phi,\theta,\psi)^{-1} = R_{123}(-\psi,-\theta,-\phi)$:

Proof. Let $R_{123}(\phi,\theta,\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} cos(\phi) & -sin(\phi) & 0 \\ sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Then,

$$\begin{aligned}
R_{123}(\phi,\theta,\psi)^{-1} &= R_{123}(\phi,\theta,\psi)^T && \text{(by orthogonality)} \\
&= [R_{12}(\psi)R_{23}(\theta)R_{12}(\phi)]^T && \text{(by definition)} \\
&= [(R_{12}(\psi)R_{23}(\theta))R_{12}(\phi)]^T && \text{(by associativity of multiplication} \\
&= R_{12}(\phi)^T[R_{12}(\psi)R_{23}(\theta)]^T && \text{(by properties of transpose)} \\
&= R_{12}(\phi)^T R_{23}(\theta)^T R_{12}(\psi)]^T && \text{(by properties of transpose)} \\
&= \begin{bmatrix} cos(\phi) & sin(\phi) & 0 \\ -sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & sin(\theta) \\ 0 & -sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= [R_{12}(-\phi)R_{23}(-\theta)R_{12}(-\psi)] && \text{(since sin is odd, cos is even)} \\
&= R_{123}(-\psi,-\theta,-\phi) && \square
\end{aligned}$$

Problem 3

Prove that $det(A'B') = \overline{det(A)det(B)}$.

Proof. Let A, B be real symmetric matrices of equal size. Then,

$$\begin{aligned}
det(A'B') &= det(A')det(B') && \text{(by properties of determinants)} \\
&= det(\overline{A^T})det(\overline{B^T}) && \text{(by definition of Hermitian transpose)} \\
&= det(\overline{A})det(\overline{B}) && \text{(since A, B are symmetric)} \\
&= det(A)det(B) && \text{(since A, B are real) (note: we've just proved they're also Hermitian)} \\
&= \overline{det(A)det(B)} && \text{(determinants of Hermitian matrices are real)} \quad \square
\end{aligned}$$

Problem 4

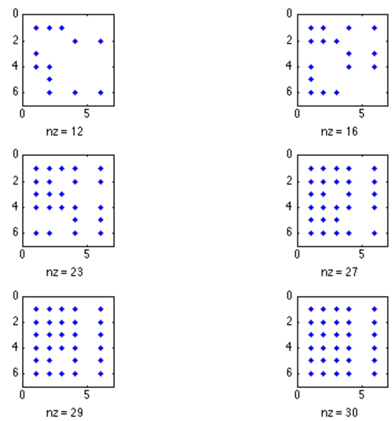
Part A: Moler (2004), Exercise 10.1

magic(4): Singular.
hess(magic(4)): in Hessenberg form.
schur(magic(5)): in Schur form.
pascal(6): Symmetric.
hess(pascal(6)): Tridiagonal.
schur(pascal(6)): Diagonal.
orth(gallery(3)): Orthogonal.
gallery(5): Singular.
gallery('frank',12): in Hessenberg form.
$[110; 021; 003]$: in Schur form.
$[210; 021; 002]$: Defective.

Part B: Moler (2004), Exercise 2.31

a. The number of nonzeros in G^p stops increasing at p = 7.
b. The proportion of nonzeros in G^p is $\frac{30}{36}$.

Figure 1: Sub-plots of G^2 to G^7



c. See figure 1.
d. Yes.

Problem 5

a. The function call MatMatOuter(ones(3,2), ones(2,4)) results in a 3x4 matrix in which each entry is the number 2.

b. I am confused by this question. I thought that the tensor product was a generalization of matrix multiplication? The operations are identical to matrix multiplication when performed on matrices.

Appendix

Listing 5: Sample Output for Exercise 2a

```
>> R = RotationYXZ(pi/2, pi/4, pi/6)

R =

   -0.3536   -0.8660   -0.3536
    0.7071    0.0000   -0.7071
    0.6124   -0.5000    0.6124

>> R = RotationZXZ(pi/2, pi/4, pi/6)

R =

   -0.3536   -0.8660    0.3536
    0.6124   -0.5000   -0.6124
    0.7071    0.0000    0.7071

>> Q = RotationZXZ(-pi/6, -pi/4, -pi/2)

Q =

   -0.3536    0.6124    0.7071
   -0.8660   -0.5000    0.0000
    0.3536   -0.6124    0.7071

>> P = inv(R)

P =

   -0.3536    0.6124    0.7071
   -0.8660   -0.5000    0.0000
    0.3536   -0.6124    0.7071
```