Math 447/847 Spring 2011 **Homework #3** Due ______________

The following exercises are intended to make you comfortable with your computing environment. Any programming language or computing environment (such as Sage, Matlab, Maple, C, Python, etc) is fine, as long as you don't use built-in features that make the problem trivial (and aren't given expression to do so). When in doubt, ask.
When submitting an assigment that requires coding, submit a printout of the code, as well as examples of the code functioning correctly. Code should be commented; comments should add to the reader's understanding and not merely repeat what can be read from the code. Choose judiciously examples that show different paths through the code; more examples do not necessarily correlate with more correct.

1. Let

$$A = \begin{pmatrix} 0 & -1 & -17 & -2 & 1 & -2 & -2 \\ -1 & 2 & 0 & -3 & -2 & 10 & -2 \\ 0 & -1 & 0 & -7 & -1 & 0 & -1 \\ -3 & -75 & 0 & -12 & -2 & 0 & -1 \\ 1 & 0 & 0 & 1 & 2 & -4 & 1 \\ -1 & -20 & 1 & 0 & 7 & 1 & 0 \\ 0 & 0 & -1 & 5 & 0 & 1 & 0 \\ -1 & -2 & 1 & -1 & 1 & 0 & 0 \\ 1 & -1 & -14 & 0 & 1 & 0 & 0 \\ -1 & 0 & 3 & -1 & 0 & 6 & 1 \end{pmatrix}$$

(This 10×7 matrix is available on the course webpage.)

   (a) Determine the rank of A, and bases for the columnspace, rowspace, and nullspace.
   (b) Find the singular value decomposition of A, using the built-in eigenvalue and eigenvector commands. Compare with the built-in SVD command.

2. Write a function that takes an arbitrary $m \times n$ real matrix and determines if every entry is non-negative.
3. The Hilbert matrix is a square $m \times m$ matrix A where $a_{ij} = 1/(i + j - 1)$ (note that this notation is 1-indexed). Write a function that has m as a parameter, and returns the corresponding Hilbert function. For several values of m, compute the reduced row echelon form (RREF) using your computer environment's built-in capabilities

   (a) for several values of m (such as $4, 10, 25, 50$),
   (b) exactly (using `QQ` in Sage), using standard double floating point arithmetic (`RDF` in Sage), and using reduced precision floating point arithmetic (`RealField(`k`)` in Sage, where k is the number of bits of precision). Comment qualitatively on how accurate the methods using floating point arithmetic are.

4. An $m \times n$ real matrix A is said to be *totally unimodular* if every square submatrix has determinant 0, 1, or -1. (Note that submatrices do not have to have consecutive rows or columns.) Write a function that takes an arbitrary $m \times n$ matrix and determines if it is totally unimodular. (*Hint*: A recursive function makes it easy to check all square submatrices by removing just one row and/or column at a time.)