Problem Set Extra: Due by November 14, 2006

Problem P1. (500 points)

Implement the algorithm for inversion on pages 19-20 of Subject 6 (Matrix Operations). Your algorithm should work with any dimension n input matrices A. Adjustments to the dimension should be internal; if the input is of dimension n so should the output even if internally you are using a dimension higher than n. The three function of Homework 3 (P1) `ReadMatrix, SetMatrix, PrintMatrix` can/must be reused for I/O. In order to avoid (and be able to deal with) problems with singularities you may wish to read the last page of the Subject 6 notes.

```
// n can be any integer dimension ; i.e. you have to take care and make it
//   a power of two if necessary
// *A, *B, *C are one dimensional arrays of n*n elements (floats)
// A[j*n+i] is the i-th row and j-th column element of a two dimensional array
// For Java use one dimensional arrays
RecursiveInverse(float *A, float *B, int n); //Find B=A**-1 Inverse per page 19-20 of Subject 6
MatrixMultiply(float *A, float *B, float *C, int n);  // C= A*B
ReadMatrix(float **A,int n, file input-file); //Allocates space for A and reads A
SetMatrix(float *A,float *B,int n); //Allocates space for A and reads A
PrintMatrix(float *A,int n,  file output-file,)//Prints A into file output-file

A matrix of the following form might be used as input for testing purposes.
Test Input Matrix ( float *mat )
  for(j=0;j<n;j++)
    for(i=0;i<n;i++) {
       if (i>j) mat[j*n+i] = (float) 0.5*i+1.0;
           else mat[j*n+i] = (float) 0.5*j+0.5;
  }
```

You need to implement the following interface

```
% ./reinverse input-A output-B
or
% java strassen input-A output-B
```

where `input-A, output-B` are files containing input/output matrices A, B. All have the same format (go to Problem Set 3 for details). For other assumptions, deviations or instructions, provide a `readme.txt` file with your code; none of the assumptions/deviations however should restrict the generality of the problem.