CS4234 Homework #4

1. The purpose of this assignment is to implement and experiment with a parallel matrix-vector code. The main program should have the following features:
   - It assumes a square $n \times n$ matrix. You may assume that the number of processors $p$ evenly divides $n$.
   - The only thing it reads from stdin is the matrix dimension $n$ (process 0 does the reading).
   - It uses dynamic memory allocation for all data structures that depend on the size of the matrix.
   - Rather than reading matrix and vector elements from stdin, the data will be generated at run time, “in-place,” by functions Init_matrix() and Init_vector()

   http://courses.cs.vt.edu/~cs4234/fall.01/proj1/Init_matrix.c

   and Init_vector()

   http://courses.cs.vt.edu/~cs4234/fall.01/proj1/Init_vector.c

   that you create however you want.
   - The matrix and the vector are distributed by blocks of rows (of size $n/p$).
   - Each processor, working on a block $A[:,1:n]$ of rows of $A$, uses the algorithm:

     ```
     initialize $A[:,1:n]$, $x[:]$
     for $i := 1$ step 1 until $n/p$
       $y[i] = A[i,1:n] \times[1:n];$
     end
     ```

   The parallel_mat_vect.c

   http://people.cs.vt.edu/~ribens/t/4234/ppmpi_c/chap05/parallel_mat_vect.c

code that is described on page 83 of Pacheco can be used to implement the above mentioned algorithm.

Once your code is executing correctly you should gather data to evaluate the parallel performance of your code. Specifically:
   - Summarize the running time, parallel speedup, and parallel efficiency of your code for various problem sizes (e.g., $n = 10^3, 10^4, 10^5$) and numbers of processors (e.g., $p = 1, 2, 4, 8$). In computing speedup, be sure to compare against a sequential version of the code.
   - You can summarize the data in tables and plots.
   - Note: Please use comments to document your program and turn in a program listing, the output, and the requested tables and plots.