Lab #3 MPI Programming II

Purpose: to learn how to write more complex MPI programs.

1. (40 points) Write a MPI program to perform matrix-matrix multiplication (A x B = C). A, B and C are *nxn* matrices. You program should

a. Take a master-slave model. All three matrices exist only on the master process at the start and finish of the computation. The master does no calculation other than distributing blocks of the matrices and collecting computation results.

b. Take a row-wise splitting strategy to partition matrix A into *n/(npes-1)* blocks, where *npes* is the number of processes. Use MPI functions to distribute each block of A and the whole matrix B to the slave processes. The slave processes do the multiplication, then use MPI functions to return their computation results to the master.

c. If *n* cannot be perfectly divided by *npes-1*, one of the slave processes should be able to take care of the extra rows (i.e. *n%(npes-1)* number of rows).

d. If n <= 16, your program should print out the matrix A, B, and C.

e. Use the master process to record the total computation time. Set appropriate *n* and *npes* shown in the following table. Run your program using four cluster nodes and fill out the form with relevant computation time.

	4 processes	8 processes	16 processes
n=256			
n=512			

Note: defining your matrix A, B, and C as ***one-dimensional*** arrays would make the programming much easier and lead to much better performance.