Project2:
Parallel Sorting Using QuickSort

For project #2 your assignment is to develop a parallel quick sort algorithm. In this project you will design an application to sort 10 million numbers (sort a smaller number for debugging, but the final project report should include results for sorting at least 10 million values). The values to sort will be obtained using the following code (which you will need to parallelize in the parallel version of this project).

```
const int N=10000000 ;

double V[N] ;
for(int i=0;i<N;++i) {
  double t = double(i)/double(N) ;
  V[N] = t*t*sin(2000.0*t*t) ;
}
```

You will need to develop the parallel quick sort algorithm based on the algorithm described in section 9.4.3 of the text. In particular, implement the distributed memory version in MPI. Please use the previous projects for guidance on generating makefiles and so on. In addition to implementing this algorithm, implement the bitonic sort and sample sort algorithms (you can use the sample sort algorithm as described in the MPI chapter.) I would recommend that you use the std C++ `std::sort()` algorithm when serial sorting is required.

You will be required to turn in the source code and a project report as described in the previous projects. In this report, please compare the parallel quick sort algorithm run times to the bitonic and sample sort algorithms. In your report be sure to include a description of the analysis of your algorithm, both in terms of timing and scalability. Try to confirm if your implementation performance behaves as predicted by your analysis.

Also, try different pivot selection methods and report on their effect on sorting performance. Based on measurements and analysis, make recommendations as to which pivot selection may work best.

I will be grading the undergraduate submissions more leniently than the graduate students, but you are required to implement the same algorithms.