Question 1. MPI Programming.

(a) Explain, and give the full prototype for the following MPI routines: MPI_Bcast(), MPI_Reduce(), MPI_Send(), MPI_Recv().

(10 marks)

(b) Develop an MPI function for the compare and exchange operation. The prototype of the method should be as follows

*int MPI_Exchange(int n, int *a, int rank1, int rank2, MPI_Comm comm)*

where the arguments are as follows:

n - the array size;

a – the array

rank1, rank2 – the processors to exchange

comm – the communicator

Note: You do not have to write a routine to merge two arrays.

(20 marks)

(c) Justify why MPI_Exchange is an efficient solution for the compare and exchange operation.

(10 marks)

Question 2. Parallel Algorithms

(a) Give an explanation of how the odd-even sort works and write a method for it. The prototype of this method should be as follows:

*int MPI_Sort(int n, int *a, int root, MPI_Comm comm)*

(20 marks)

(b) Evaluate the theoretical complexity of MPI_Sort considering both communication ($T_{startup}$, T_{comm}) and computation (T_{com}). Then explain the communication and computation overheads of the method.

(10 marks)

(c) Prove that MPI_Sort finishes the sorting in *size* repetitions, where *size* represents the number of processors.

(10 marks)