

Solution of Serie 12

Exercise 1 *Since the messages are moving in only one direction, there is for sure no deadlock. Notice that the three initialization lines can be found in almost all MPI codes. The first line is absolutely required; the two next ones define the number of processors and the rank of the processor which is running the code, which are essential informations. All the declared variables have a copy on each processor; hence, the content of the variable `num` must be passed by MPI from one processor to the other one, contrary to what happens on a shared memory machine. Notice that the printing order is not chronological, because there is a buffer which empties only when it is full or at the end of the job .*

Exercise 2 *One can notice that the array `A` is declared on each processor. In a true application code with large arrays, one should allocate memory space for `A` dynamically (`malloc`) only on the processor 0 and one should proceed in the same way for the array `AB`, but on the other processors. In the applications requiring a lot of memory, one tries if possible that each processor generates its own data. One could proceed like this for the original matrix, by sending then one block at once for printing by the processor 0, and next transposing the blocks and again send one block at once for printing the transposed matrix.*