

Exercise 14

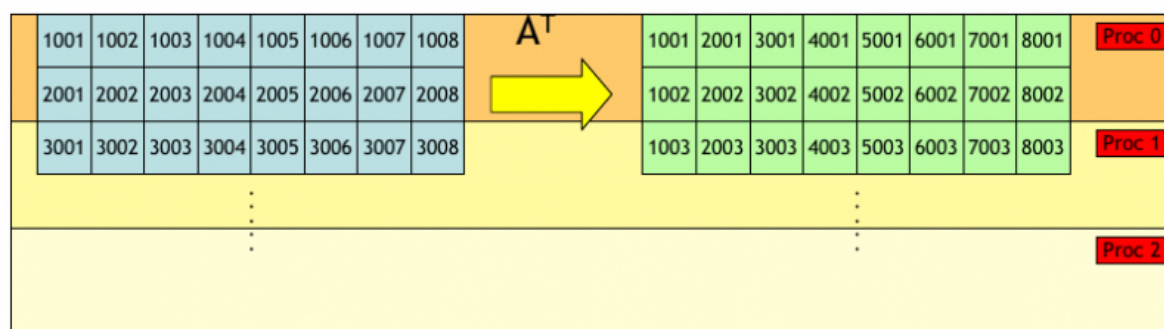
Write a program that performs the transpose of a square matrix A of arbitrary dimension (multiple of the number of tasks).

The matrix is split among the tasks and initialized so that each element is unique (use row-column number and task rank).

Thus you need to evaluate B of the form:

$$B = A^T$$

Follows a visual representation of the matrices A and B (8x8 in the example):



Solve the problem distributing A and B by lines (or columns) over the tasks and using the collective communication `MPI_ALLTOALL`. Please note that this collective function works with data stored contiguously in memory, therefore choose carefully whether to store the matrix by lines or columns. Can you tell why? ([answer](#))

We ask you to:

- initialize A
- evaluate $B = A^T$

- check if the operation was successful

HINTS:

C

MPI_ALLTOALL	<code>int MPI_Alltoall(void* sendbuf, int sendcount, MPI_Datatype sendtype, void* recvbuf, int recvcount, MPI_Datatype recvtype, MPI_Comm comm)</code>
MPI_INIT	<code>int MPI_Init(int *argc, char ***argv)</code>
MPI_COMM_SIZE	<code>int MPI_Comm_size(MPI_Comm comm, int *size)</code>
MPI_COMM_RANK	<code>int MPI_Comm_rank(MPI_Comm comm, int *rank)</code>
MPI_FINALIZE	<code>int MPI_Finalize(void)</code>
MPI_WTIME	<code>double MPI_Wtime(void)</code>
MPI_ABORT	<code>int MPI_Abort (MPI_Comm comm)</code>

FORTRAN

MPI_ALLTOALL	<code>MPI_ALLTOALL(SENDBUF, SENDCOUNT, SENDTYPE, RECVBUF, REVCOUNT, RECVTYPE, COMM, IERROR)</code> <code><type> SENDBUF(*), RECVBUF(*)</code> <code>INTEGER SENDCOUNT, SENDTYPE, REVCOUNT, RECVTYPE, COMM, IERROR</code>
MPI_INIT	<code>MPI_INIT(IERROR)</code> <code>INTEGER IERROR</code>
MPI_COMM_SIZE	<code>MPI_COMM_SIZE(COMM, SIZE, IERROR)</code> <code>INTEGER COMM, SIZE, IERROR</code>
MPI_COMM_RANK	<code>MPI_COMM_SIZE(COMM, SIZE, IERROR)</code> <code>INTEGER COMM, SIZE, IERROR</code>
MPI_FINALIZE	<code>MPI_FINALIZE(IERROR)</code> <code>INTEGER IERROR</code>
MPI_WTIME	<code>MPI_WTIME()</code>

MPI_ABORT`MPI_ABORT(COMM, ERRCODE)``INTEGER COMM, ERRCODE`

[< Solution 13](#)[up](#)[Solution 14 >](#)

© Copyright 2012 SCAI - SuperComputing Applications and Innovation - CINECA