

**Corso di Sistemi Distribuiti**  
**Prof. Rocco Aversa**  
**Prova scritta gennaio 2015**

1. Scrivere un programma MPI (con un numero di processi  $nproc=q*q$ ) che effettua l'elaborazione di una matrice di interi  $M(dim \times dim)$ , con  $dim$  multiplo intero di  $q$ , svolgendo le seguenti operazioni:

- Viene creata una topologia bidimensionale aperta ( $q \times q$ ).
- Il processo di rango 0 legge da file la matrice di interi  $M$  e la distribuisce a tutti i processi, compreso se stesso, inviando, in base alla topologia, ad ogni processo una matrice  $A(dim/q \times dim/q)$ .
- Il singolo processo localmente effettua il ribaltamento delle righe di  $A$ .
- Effettuare a questo punto uno shift verso il basso di lunghezza due dell'intera matrice  $M$  spostando tra i processi le intere matrici  $A$ .
- Calcolare, utilizzando le primitive di calcolo collettivo, il minimo degli elementi sulla diagonale secondaria di  $M$ .

P.S. Alcune primitive MPI utili allo svolgimento dell'esercizio:

```
int MPI_Cart_create(MPI_Comm comm_old, int ndims, const int dims[],  
                   const int periods[], int reorder, MPI_Comm *comm_cart)  
  
int MPI_Cart_rank(MPI_Comm comm, const int coords[], int *rank)
```