Notes on homework 0

- Use exact notation from homework (dp, rank, comm, etc)
- Must run correctly for any n and $p \geq 2$

1 MPI Overview

1. Point-to-point routines
   - blocking: `mpi_send`, `mpi_ssend`, `mpi_recv`
   - non-blocking: `mpi_isend`, `mpi_issend`, `mpi_irecv`
2. Collective routines: involves all processors in the `comm` (i.e. `mpi_barrier`, `mpi_bcast`)
3. MPI 1-sided routines (vs Cray's shmem (shared memory) routines; complicated but very fast)
4. MPI I/O; quite useful, but complicated ($\approx$ 100 pages); Parallel I/O
5. MPI process creation (not really useful to most people)

2 Point-to-Point Communication

1. blocking
   - `mpi_ssend`
   - `mpi_rsend`
   - `mpi_bsend`
   - `mpi_send`
   - `mpi_recv`
   - `mpi_sendrecv`
   - `mpi_sendrecv_replace`
2. non-blocking
   - `mpi_issend`
   - `mpi_irsend`
   - `mpi_ibsend`
   - `mpi_isend`
   - `mpi_irecv`

2.1 Difference between blocking and non-blocking routines

When is the statement following the `send` call executed? For blocking routines, it waits until it is safe to modify the variable containing the data to be passed. Non-blocking routines do not offer this safety?

Why have non-blocking routines? It can hep avoid deadlocks and allow communication and execution to occur at the same time.

2.2 Blocking sends

- `mpi_ssend`: synchronize send; slow for small messages, but performs fine with large messages; great for debugging
- `mpi_rsend`: ready send; supposed to provide a very fast send; sends message immediately and the corresponding `recv` must be "posted" (waiting for the message); if the `recv` is not posted the results are undefined