Name: Brijesh Mavani
CWID: A20406960
University: Illinois Institute of Technology
Course: Parallel and Distributed Processing
Assignment: 3 - README

The code (MPICollective.c) is already compiled on a comet system with executable file MPICollective for Collective communication.
Following steps can be executed for code execution:

1) C codes are already compiled as mentioned above. In case re-compilation is required you can execute the following command:
mpicc -o <executable file name> <C code file name>
eg: mpicc -o MPICollective MPICollective.c -- for Collective communication

2) For code execution there are two ways. Both ways are provided below:

1) Creating a job and submitting.
   a. Create the job file for execution. There are jobs created for each processor size (1,2,4,8,12,16).
   File name format : <MPIP2P/MPICollective><no of processors>_job.sh
   eg:
   MPICollective1_job.sh -- For MPI collective communication for 1 processor
   MPICollective2_job.sh -- For MPI collective communication for 2 processors

   Sample .sh file content

```
#!/bin/bash
#SBATCH --job-name="<executable file>"
#SBATCH --output="MPICollective_1Proc.%j.%N.out"
#SBATCH --partition=compute
#SBATCH --nodes=X
#SBATCH --ntasks-per-node=1
#SBATCH --export=ALL
#SBATCH -t 00:10:00

ibrun -np X ./<executable file> Y
```

   ----

   eg:ibrun -np 1 ./MPICollective 100000000
   where,
   - X is number of processors i.e. 1,2,4,8,12,16.
   - executable file -- name of the executable file name given during compilation. eg. MPICollective

- **Y -- Array Size. It should be multiple of the number of processors. And the max size should be 100000000**

b. Execute the job with sbatch <job name>.
eg: $ sbatch MPICollective1_job.sh

c. The output file will be created with a unique name for each job execution. For simplicity, run the command: ls –ltr to find out which file created at last.

d. cat <output file name> -- To view the contents of the job output file. As output file can be large, you can use the following command to view certain lines of the output file.

tail -n 10 <output file name> -- for viewing only last 10 lines of the output file.

2) Alternatively, below command can be executed to execute the code without submitting the job:
mpirun –np X ./<executable file> Y
where,

- X is number of processors i.e. 1,2,4,8,12,16.
- executable file -- name of the executable file name given during compilation. eg. MPICollective
- **Y -- Array Size. It should be multiple of the number of processors. And the max size should be 100000000**