**Q1: What does process rank 5’s counter store at the end of the computation? (Pingpong activity 1)**

**Ans:** It stores 20.

**Q2: How many process ranks are used in the script above? (Pingpong activity 2)**

**Ans:** 4.

**Q3: What does process rank 5’s counter store at the end of the computation? (Ring activity 2)**

**Ans:** 40.

**Q4: Comparing Programming Activities #2 and #3, which was easier to implement? Explain. (Ring activity 3)**

**Ans:** Activity #3 was easier to implement because of using the non-blocking MPI\_Isend(...). With MPI\_Isend(...), we can use a single while(...) loop to implement the activity across all processes.

**Q5: Comparing Programming Activities #4 and #5, which was easier to implement? Explain. (Random activity 4 and 5)**

**Ans:** Activity #5 was easier than Activity #4 to implement. In Activity #5, as we use MPI\_ANY\_SOURCE, we can use simpler logic compared to Activity #4 for implementing the program. MPI\_ANY\_SOURCE allows a process to receive from a source without putting a restriction on the process rank which helps to implement Activity #5 with simpler program logic.