

Q/A Exercise 3

ANSWERS:

Q- Take a careful look at the output; is the task's order guaranteed? Try to run the code 3 times.

A- The processes are independent and their termination order is basically not guaranteed.

Q- What function call can you use instead of the pair ISEND/Irecv?

A- A send-receive operation is very useful for executing a shift operation across a chain of processes:

MPI_SENDRECV

```
int MPI_Sendrecv(void *sendbuf, int sendcount, MPI_Datatype sendtype, int dest,
int sendtag, void *recvbuf, int recvcount, MPI_Datatype recvtype, int source, int
recvtag, MPI_Comm comm, MPI_Status *status)
```

```
MPI_SENDRECV(SENDBUF, SENDCOUNT, SENDTYPE, DEST, SENDTAG, RECVBUF, REVCOUNT,
RECVTYPE, SOURCE, RECVTAG, COMM, STATUS, IERROR)
<type> SENDBUF(*), RECVBUF(*)
INTEGER SENDCOUNT, SENDTYPE, DEST, SENDTAG, REVCOUNT, RECVTYPE, SOURCE, RECV
TAG, COMM, STATUS(MPI_STATUS_SIZE), IERROR
```

Try to modify the code in order to call the SENDRECV function instead of the SEND and RECV calls.

