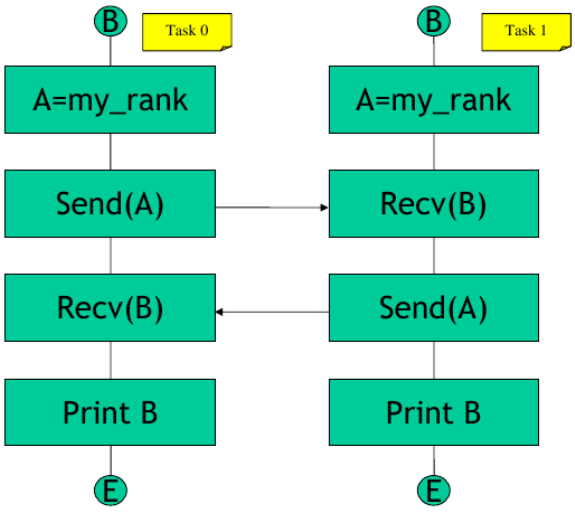
Exercise 2

Exercise 2

Write a code using point to point communication that makes two processes send each other an array of floats containing their rank. Each of the processes will declare two float arrays, A and B, of a fixed dimension (10000). All of the elements of the array A will be initialized with the rank of the process. Then, A and B will be used as the buffers for SEND and RECEIVE, respectively. The program terminates with each process printing out one element of the array B.

The program should follow this scheme:



The **output** should look like

```
I am task 0 and I have received b(0) = 1.00

I am task 1 and I have received b(0) = 0.00
```

HINTS:

C

MPI_SEND

```
int MPI_Send(void* buf, int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm)
```

MPI_RECV

```
int MPI_Recv(void* buf, int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm, MPI_Status *status)
```

MPI_INIT

```
int MPI_Init(int *argc, char ***argv)
```

MPI_COMM_SIZE

```
int MPI_Comm_size(MPI_Comm comm, int *size)
```

MPI_COMM_RANK

```
int MPI_Comm_rank(MPI_Comm comm, int *rank)
```

MPI_FINALIZE

```
int MPI_Finalize(void)
```

FORTRAN

MPI_SEND

```
MPI_SEND(BUF, COUNT, DATATYPE, DEST, TAG, COMM, IERROR)
<type> BUF(*)
INTEGER COUNT, DATATYPE, DEST, TAG, COMM, IERROR
```

MPI_RECV

```
MPI_RECV(BUF, COUNT, DATATYPE, SOURCE, TAG, COMM, STATUS, IERROR)
<type> BUF(*)
INTEGER COUNT, DATATYPE, SOURCE, TAG, COMM, STATUS(MPI_STATUS_SIZE), IERROR
```

MPI_INIT

```
MPI_INIT(IERROR)
INTEGER IERROR
```

MPI_COMM_SIZE

```
MPI_COMM_SIZE(COMM, SIZE, IERROR)
INTEGER COMM, SIZE, IERRO
```

MPI_COMM_RANK

```
MPI_COMM_SIZE(COMM, SIZE, IERROR)
INTEGER COMM, SIZE, IERROR
```

MPI_FINALIZE

```
MPI_FINALIZE(IERROR)
INTEGER IERROR
```

QUESTIONS:

Q- What happens if the order of the send/receive in task 1 is inverted? (answer)

Q- Try to reduce the SEND buffer A to just one element and invert the order of send/receive in task 1. What happens when you run the code? (answer)

Q- Rewrite the code without any IF statement and any deadlock. What do you need to use instead of the blocking SEND? (answer)

‹ Solution 1 up Solution 2 ›