

## Solution 2

### Solution 2

C

```
#include <stdlib.h>
#include <stdio.h>
#include <mpi.h>

#define NDATA 10000

int main(int argc, char* argv[]){

    int me, nprocs, i = 0;
    MPI_Status status;

    float a[NDATA];
    float b[NDATA];

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &me);

    /* Initialize data */
    for(i=0;i<NDATA;++i){
        a[i] = me;
    }
    /* Protect against the use with a number of processes != 2 */
    if(nprocs!=2 && me==0){
        printf("\n\tThis program must run on 2 processors");
        return 0;
    }
    /* Send and Receive data */
    if(me==0){
        MPI_Send(a, NDATA, MPI_REAL, 1, 0, MPI_COMM_WORLD);
        MPI_Recv(b, NDATA, MPI_REAL, 1, 0, MPI_COMM_WORLD, &status);
    }
    else{
```

```

    MPI_Recv(b, NDATA, MPI_REAL, 0, 0, MPI_COMM_WORLD, &status);
    MPI_Send(a, NDATA, MPI_REAL, 0, 0, MPI_COMM_WORLD);
}

printf("\tI am task %d and I have received b(0) = %1.2f \n", me, b[0]);

MPI_Finalize();
return 0;
}

```

## FORTRAN

```

program hello

use mpi

implicit none

integer ierr, me, nprocs, errcode
integer status(MPI_STATUS_SIZE)

integer,parameter :: ndata = 10000
real                :: a(ndata)
real                :: b(ndata)

call MPI_INIT(ierr)
call MPI_COMM_SIZE(MPI_COMM_WORLD, nprocs, ierr)
call MPI_COMM_RANK(MPI_COMM_WORLD, me, ierr)

!$ Initialize data
a = me
!$ Protect against the use with a number of processes != 2
if (nprocs .ne. 2) then
    print *, "This program must run on 2 processors"
    call MPI_ABORT(MPI_COMM_WORLD, errcode, ierr)
    stop
endif
!$ Send and Receive data
if (me==0) then
    call MPI_SEND(a,ndata,MPI_REAL,1,0,MPI_COMM_WORLD,ierr)
    call MPI_RECV(b,ndata,MPI_REAL,1,0,MPI_COMM_WORLD,status,ierr)
else
    call MPI_RECV(b,ndata,MPI_REAL,0,0,MPI_COMM_WORLD,status,ierr)
    call MPI_SEND(a,ndata,MPI_REAL,0,0,MPI_COMM_WORLD,ierr)
endif

print *, 'I am proc ',me,' and I have received b(1) = ',b(1)
call MPI_FINALIZE(ierr)

```

```
end program hello
```

- 
- [Q/A Exercise 2](#)

---

[< Exercise 2](#)[up](#)[Q/A Exercise 2 >](#)

---

© Copyright 2012 SCAI - SuperComputing Applications and Innovation - CINECA