

[ABOUT US](#)[RESOURCES](#)[SERVICES](#)[FOR USERS](#)[TRAINING](#)[PROJECTS](#)[Home](#) > [Training MPI](#)

## Solution 4

### Solution 4

**C**

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

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

    int me, nprocs, left, right, count;
    MPI_Status status;

    float a;
    float b;
    float sum;

    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &me);
    /* Initialize workspace */
    a  = me;
    b  = -1;
    sum = a;
    /* Compute neighbour ranks */
    right = (me+1)%nprocs;
    left  = (me-1+nprocs)%nprocs;

    /* Circular sum*/
    for(count = 1; count < nprocs; count++)
    {
        MPI_Sendrecv(&a, 1, MPI_REAL, left, 0, &b, 1, MPI_REAL, right, 0, MPI_COMM_WORLD, &status);
        /* Set "a" value to the newly received rank */
        a  = b;
        /* Update the partial sum */
        sum += a;
    }
```

```
printf("\tI am proc %d and sum(0) = %1.2f \n", me, sum);

MPI_Finalize();

}
```

**FORTRAN**

```
program hello

use mpi

implicit none

integer ierr,me,nprocs,left,right
integer status(MPI_STATUS_SIZE)

integer,parameter :: ndata = 1000

real :: a
real :: b
real :: somma

integer i

call MPI_INIT(ierr)
call MPI_COMM_SIZE(MPI_COMM_WORLD, nprocs, ierr)
call MPI_COMM_RANK(MPI_COMM_WORLD, me, ierr)
!$ Initialize workspace
a = me
b = -1
!$ Compute neighbour ranks
right = mod(me + 1 , nprocs)
left = mod(me - 1 + nprocs , nprocs)
!$ Circular sum
somma = a

do i = 1,nprocs-1
    call MPI_SENDRECV(a,1,MPI_REAL,left,0, &
        b,1,MPI_REAL,right,0,MPI_COMM_WORLD,status,ierr)
    !$ Set "a" value to the newly received rank
    a = b
    !$ Update the partial sum
    somma = somma + a
enddo

print *, 'I am proc ',me,' and somma = ',somma
call MPI_FINALIZE(ierr)
```

```
end program hello
```

---

[< Exercise 4](#)[up](#)[Exercise 5 >](#)

---

© Copyright 2012 SCAI - SuperComputing Applications and Innovation - CINECA