

## Solution 11

**C**

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

    int world_rank, row_rank, col_rank, cart_rank;
    int nprocs, row_size, col_size;
    int coords[2], sub_coords[2];
    int dims[2] = { 0, 0}, period[2] = { 1, 1};
    int src_rank, dst_rank;
    int sum, temp;
    float avg;

    MPI_Comm cart_grid, cart_row, cart_col;
    MPI_Status status;

    /* MPI Initialization */
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    /* Cartesian grid creation */
    MPI_Dims_create(nprocs, 2, dims);
    MPI_Cart_create(MPI_COMM_WORLD, 2, dims, period, 1, &cart_grid);

    /*Local world_rank initialization and comparison to global world_rank */
    MPI_Comm_rank(cart_grid, &cart_rank);

    printf ("I am world_rank %d in MPI_COMM_WORLD and world_rank %d in the cartesian communicator\n",
            world_rank, cart_rank);

    /* Coordinates creation and neighbour communication */
    MPI_Cart_coords(cart_grid, cart_rank, 2, coords);

    /* Communication south */
    sum = world_rank;
    MPI_Cart_shift(cart_grid, 1, 1, &src_rank, &dst_rank);
    MPI_Sendrecv(&world_rank, 1, MPI_INT, dst_rank, 0, &temp, 1, MPI_INT,
```

```
    src_rank, 0, cart_grid, &status);
sum += temp;

/* Communication north */
MPI_Cart_shift(cart_grid, 1, -1, &src_rank, &dst_rank);
MPI_Sendrecv(&world_rank, 1, MPI_INT, dst_rank, 0, &temp, 1, MPI_INT,
    src_rank, 0, cart_grid, &status);
sum += temp;

/*Communication east */
MPI_Cart_shift(cart_grid, 0, 1, &src_rank, &dst_rank);
MPI_Sendrecv(&world_rank, 1, MPI_INT, dst_rank, 0, &temp, 1, MPI_INT,
    src_rank, 0, cart_grid, &status);
sum += temp;

/*Communication west */
MPI_Cart_shift(cart_grid, 0, -1, &src_rank, &dst_rank);
MPI_Sendrecv(&world_rank, 1, MPI_INT, dst_rank, 0, &temp, 1, MPI_INT,
    src_rank, 0, cart_grid, &status);
sum += temp;

/*Neighbour's average */
avg = (float) sum / 5;
printf("Cart rank %d (%d, %d), neighbours average: %.2f\n", cart_rank, coords[0], coords[1], avg);

/*Row sub-communicator creation */
sum = 0;
sub_coords[0] = 1;
sub_coords[1] = 0;
MPI_Cart_sub(cart_grid, sub_coords, &cart_row);
MPI_Comm_size(cart_row, &row_size);
MPI_Comm_rank(cart_row, &row_rank);

/*Row sub-communicator's average calculation */
MPI_Reduce(&world_rank, &sum, 1, MPI_INT, MPI_SUM, 0, cart_row);

if (row_rank == 0) {
    avg = (float) sum / row_size;
    printf("Row %d, row average: %.2f\n", coords[1], avg);
}

/*Column sub-communicator creation */
sum = 0;
sub_coords[0] = 0;
sub_coords[1] = 1;
MPI_Cart_sub(cart_grid, sub_coords, &cart_col);
MPI_Comm_size(cart_col, &col_size);
MPI_Comm_rank(cart_col, &col_rank);

/*Column sub-communicator's average calculation */
MPI_Reduce(&world_rank, &sum, 1, MPI_INT, MPI_SUM, 0, cart_col);
```

```

if (col_rank == 0) {
    avg = (float) sum / col_size;
    printf("Column %d, column average: %.2f\n", coords[0], avg);
}

/* Finalization operations */
MPI_Comm_free(&cart_grid);
MPI_Comm_free(&cart_col);
MPI_Comm_free(&cart_row);
MPI_Finalize();

return 0;
}

```

## FORTRAN

```

program cartesian

use mpi

implicit none

INTEGER :: world_rank, row_rank, col_rank, cart_rank
INTEGER :: nprocs, row_size, col_size
INTEGER :: dims(2), coords(2)
LOGICAL :: period(2), sub_coords(2)
INTEGER :: src_rank, dst_rank
INTEGER :: sum, temp
REAL :: avg

INTEGER :: cart_grid, cart_row, cart_col, status, ierr

!$ MPI Initialization
CALL MPI_Init(ierr)
CALL MPI_Comm_size(MPI_COMM_WORLD, nprocs, ierr)
CALL MPI_Comm_rank(MPI_COMM_WORLD, world_rank, ierr)

!$ Cartesian grid creation
dims(1) = 0
dims(2) = 0
period(1) = .true.
period(2) = .true.

CALL MPI_Dims_create(nprocs, 2, dims, ierr)
CALL MPI_Cart_create(MPI_COMM_WORLD, 2, dims, period, .true., cart_grid, ierr)

!$ Local world_rank initialization and comparison to global world_rank
CALL MPI_Comm_rank(cart_grid, cart_rank, ierr)

```

```

WRITE (*,'(a,i1,a,i1,a)') 'I am world_rank ', world_rank, ' in MPI_COMM_WORLD and world_rank ', cart_rank, ' in the &
cartesian communicator'

!$ Coordinates creation and neighbour communication
CALL MPI_Cart_coords(cart_grid, cart_rank, 2, coords, ierr)

!$ Communication south
sum = world_rank
CALL MPI_Cart_shift(cart_grid, 1, 1, src_rank, dst_rank, ierr)
CALL MPI_Sendrecv(world_rank, 1, MPI_INTEGER, dst_rank, 0, temp, 1, MPI_INTEGER, src_rank, 0, cart_grid, status, ierr)
sum = sum + temp

!$ Communication north
CALL MPI_Cart_shift(cart_grid, 1, -1, src_rank, dst_rank, ierr)
CALL MPI_Sendrecv(world_rank, 1, MPI_INTEGER, dst_rank, 0, temp, 1, MPI_INTEGER, src_rank, 0, cart_grid, status, ierr)
sum = sum + temp

!$ Communication east
CALL MPI_Cart_shift(cart_grid, 0, 1, src_rank, dst_rank, ierr)
CALL MPI_Sendrecv(world_rank, 1, MPI_INTEGER, dst_rank, 0, temp, 1, MPI_INTEGER, src_rank, 0, cart_grid, status, ierr)
sum = sum + temp

!$ Communication west
CALL MPI_Cart_shift(cart_grid, 0, -1, src_rank, dst_rank, ierr)
CALL MPI_Sendrecv(world_rank, 1, MPI_INTEGER, dst_rank, 0, temp, 1, MPI_INTEGER, src_rank, 0, cart_grid, status, ierr)
sum = sum + temp

!$ Neighbour's average
avg = REAL(sum)/5
WRITE (*,'(a,i2,a,i1,a,i1,a,f6.2)') 'Cart rank ', cart_rank, ' (', coords(1), ', ', coords(2), &
    '), neighbours average: ', avg

!$ Row sub-communicator creation
sum = 0
sub_coords(1) = .true.
sub_coords(2) = .false.
CALL MPI_Cart_sub(cart_grid, sub_coords, cart_row, ierr)
CALL MPI_Comm_size(cart_row, row_size, ierr)
CALL MPI_Comm_rank(cart_row, row_rank, ierr)

!$ Row sub-communicator's average calculation
CALL MPI_Reduce(world_rank, sum, 1, MPI_INTEGER, MPI_SUM, 0, cart_row, ierr)

if (row_rank.eq.0) then
    avg = REAL(sum) / row_size
    WRITE (*,'(a,i1,a,f6.2)') 'Row ', coords(2), ' row average: ', avg
endif

!$ Column sub-communicator creation
sum = 0

```

```
sub_coords(1) = .false.
sub_coords(2) = .true.
CALL MPI_Cart_sub(cart_grid, sub_coords, cart_col, ierr)
CALL MPI_Comm_size(cart_col, col_size, ierr)
CALL MPI_Comm_rank(cart_col, col_rank, ierr)

!$ Column sub-communicator's average calculation
CALL MPI_Reduce(world_rank, sum, 1, MPI_INTEGER, MPI_SUM, 0, cart_col, ierr)

if (col_rank.eq.0) then
  avg = REAL(sum) / col_size
  WRITE (*,'(a,i1,a,f6.2)') 'Column ',coords(1),' column average: ',avg
endif

!$ Finalization operations
CALL MPI_Comm_free(cart_grid, ierr)
CALL MPI_Comm_free(cart_col, ierr)
CALL MPI_Comm_free(cart_row, ierr)
CALL MPI_Finalize(ierr)

end program cartesian
```

[< Exercise 11](#)[up](#)[Exercise 12 >](#)