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

int main (int argc, char *argv[])
{
int position, i, j, a[2], myrank, num_proc;
char buff[1000];
MPI_Status stat;
MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &num_proc );
MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
if (myrank == 0)
{
/* SENDER CODE */
       i = 1; j = 2; a[0] = 3;
       printf( "Proc %d: sending %u %u %u.th portion to proc 0.\n", myrank, i, j, a );
position = 0;
MPI_Pack(&i, 1, MPI_INT, buff, 1000, &position, MPI_COMM_WORLD);
MPI_Pack(&j, 1, MPI_INT, buff, 1000, &position, MPI_COMM_WORLD);
MPI_Send( buff, position, MPI_PACKED, 1, 0, MPI_COMM_WORLD);
}
else /* RECEIVER CODE */
{
       MPI_Recv( a, 2, MPI_INT, 0, 0, MPI_COMM_WORLD,&stat);
       printf( "Proc %d: reciving %u %u %u.th portion to proc 0.\n", myrank, i, j,
a[0] );
}
MPI_Finalize();
return 0;
}
```

```fortran
program main
implicit none
include 'mpif.h'
integer a(0:1)
integer position, i, j, a(0:1), rank, numtasks;
character buff(0:1000);
integer stat(MPI_STATUS_SIZE)
call MPI_INIT(ierr)
call MPI_COMM_RANK(MPI_COMM_WORLD, rank, ierr)
call MPI_COMM_SIZE(MPI_COMM_WORLD, numtasks, ierr)
if (rank == 0) then
!/* SENDER CODE */
position = 0;
i=1;j=2;
CALL MPI_Pack(i, 1, MPI_INTEGER, buff, 1000, position, MPI_COMM_WORLD, ierr);
CALL MPI_Pack(j, 1, MPI_INTEGER, buff, 1000, position, MPI_COMM_WORLD, ierr);
CALL MPI_Send( buff, position, MPI_PACKED, 1, 0, MPI_COMM_WORLD, ierr);
else !/* RECEIVER CODE */
CALL MPI_Recv( a, 2, MPI_INTEGER, 0, 0, MPI_COMM_WORLD, ierr);
position = 0;
CALL MPI_Unpack(a, 2, position, i, 1, MPI_INTEGER, MPI_COMM_WORLD, ierr);
CALL MPI_Unpack(a, 2, position, j, 1, MPI_INTEGER, MPI_COMM_WORLD, ierr);
Write(*,*) 'i , j = ' , i,j
END IF
call MPI_FINALIZE(ierr)
end program main
```