

```

#include "mpi.h"
#include <stdio.h>

int main(int argc, char **argv)
{
    int rank, size;
    int root ;
    int left, right, interval ;
    int number, start, end, sum, GrandTotal;
    int mystart, myend;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    root = 0;
    left =1;
    right=1000;

    if (rank == root)
    {
        printf("Proceso 0: valor inicial %d, final : %d\n", left, right);
    }

    MPI_Bcast(&left, 1, MPI_INT, root, MPI_COMM_WORLD); //Envia los limites a
    todos los procesos
    MPI_Bcast(&right, 1, MPI_INT, root, MPI_COMM_WORLD);

    if (((right - left + 1) % size) != 0)
        interval = (right - left + 1) / size + 1 ; /*Fija límites locales de
suma*/
    else
        interval = (right - left + 1) / size;

    mystart = left + rank*interval ;
    myend = mystart + interval ;
    /* establece los límites de los intervalos correctamente */

    if (myend > right)
        myend = right + 1 ;
    sum = root; /* Suma localmente en cada proceso MPI */

    if (mystart <= right)
        for (number = mystart; number < myend; number++)

            printf("Proceso: %d Suma acumulada: %d Suma del proceso:
%d\n",rank,sum,number);

    /* Hace la reducción en el proceso raíz */
    MPI_Reduce(&sum, &GrandTotal, 1, MPI_INT, MPI_SUM, root, MPI_COMM_WORLD) ;
    MPI_Barrier(MPI_COMM_WORLD);
    /* El proceso raíz retorna los resultados */
    if(rank == root)
        printf("Suma total: Grand total = %d \n", GrandTotal);

    MPI_Finalize();
}

```