

```

{
// Cálculo de PI
int n,i,rank,size,bsize,remain,beg,end;
double x;

MPI_Init ( &argc, &argv );
MPI_Comm_rank ( MPI_COMM_WORLD, &rank );
MPI_Comm_size ( MPI_COMM_WORLD, &size );

if ( rank == 0 )
{
printf("Introduce la precisión del cálculo (n > 0): ");
fflush(stdout);
scanf("%d",&n);
}
MPI_Bcast ( &n, 1, MPI_INT, 0, MPI_COMM_WORLD );
bsize= n/size;
remain= n%size;
if ( rank == 0 )
{
beg= 0;
end= bsize + remain;
}
else
{
beg= rank*bsize + remain;
end= beg + bsize;
}

double PI25D = 3.141592653589793238462643;
double h = 1.0 / (double) n;

```

```

if ( rank == 0 )
{
    beg= 0;
    end= bsize + remain;
}
else
{
    beg= rank*bsize + remain;
    end= beg + bsize;
}

double PI25D = 3.141592653589793238462643;
double h = 1.0 / (double) n;
double sum = 0.0;
for (i = beg; i < end; i++) {
    x = h * ((double)i + 0.5);
    sum += (4.0 / (1.0 + x*x));
}
double res;
MPI_Reduce ( &sum, &res, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD );
if ( rank == 0 )
{
    double pi = res * h;
    printf ("El valor aproximado de PI es: %f, con un error de %f\n",pi,fabs(pi - PI25D));
}

MPI_Finalize ();

return 0;
}

```



**Problema:** Donat el següent programa paral·lel en MPI que calcula la integral d'una funció amb una precisió  $n$  i el resultat s'emmagatzema en tots els processos: (Dado el siguiente programa paralelo en MPI que calcula la integral de una función con una precisión  $n$  y el resultado se almacena en todos los procesos:)

```
int main(void) {
    int rank, sz, local_n, n;
    double a = 0.0, b = 5.0, h, local_a, local_b;
    double local, total;
    int source;
    MPI_Init(NULL, NULL);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &sz);
    if(rank==0){
        printf("Introduce la precisión del cálculo (n > 0): ");
        scanf("%d",&n);
    }
    MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    h = (b-a)/n; /* h es el mismo para todos los procesos */ 2 FLOP
    local_n = n/sz; /* es el número de trapezoides */
    local_a = a + rank*local_n*h; 3 FLOP
    local_b = local_a + local_n*h; 2 FLOP
    local = Trap(local_a,local_b,local_n, h); //Coste= 5*local_n+2 FLOP
    if (rank == 0) {
        total = local;
        for (source = 1; source < sz; source++) {
            MPI_Recv(&local, 1, MPI_DOUBLE, source, 0, MPI_COMM_WORLD,
                MPI_STATUS_IGNORE);
            total += local; 1 FLOP
        }
    } else {
        MPI_Send(&local, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD);
    }
    if (rank == 0) //para el coste no se tiene en cuenta la E/S
        printf("La integral de %f a %f = %.15e\n", a, b, total);
    MPI_Bcast(&total, 1, MPI_DOUBLE, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    MPI_Finalize();
    return 0;
} /* main */
```