Topic: MPI Programming

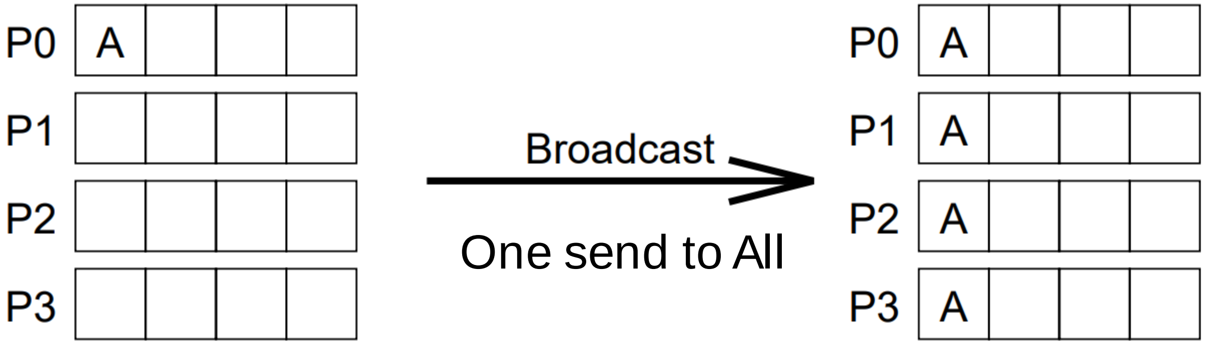
Objective

- MPI basic functions
- MPI Collective functions/communication

Template

```
program sample_mpi
 Implicit none
 Include ‘mpif.h’
[other includes]
 integer :: ierr, nproc, rank
 [other declarations]
 call mpi_init(ierr)
 call mpi_comm_size(MPI_COMM_WORLD, nproc, ierr)
 call mpi_comm_rank(MPI_COMM_WORLD, rank, ierr)
:
Main part of the code
:
 call mpi_finalize(ierr)
end program
```

Collective communication - Broadcast and reduction



P0 A

P1

P2

P3

Broadcast

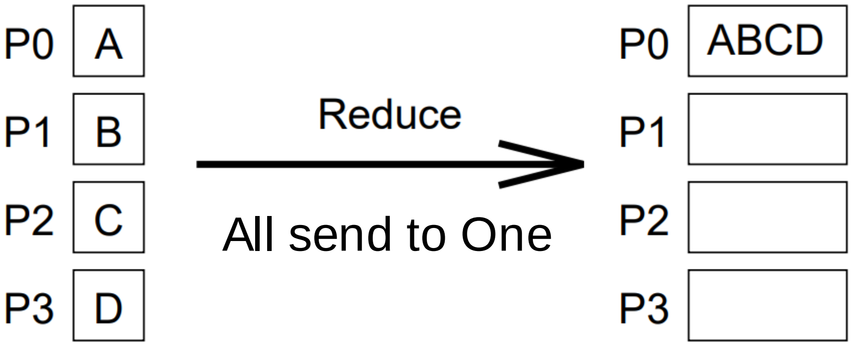
One send to All

P0 A

P1 A

P2 A

P3 A



P0 A

P1 B

P2 C

P3 D

Reduce

All send to One

P0 ABCD

P1

P2

P3

Topic: MPI Programming

Collective communication - Broadcast and reduction

```
MPI_Bcast(
   data,
   int count,
   MPI_Datatype datatype,
   int root,
   MPI_Comm communicator, ierr)
```

MPI_Datatype: MPI_LOGICAL, MPI_INTEGER, MPI_REAL, MPI_DOUBLE_PRECISION

```
MPI_Reduce(
   send_data,
   recv_data,
   int count,
   MPI_Datatype datatype,
   MPI_Op op,
   int root,
   MPI_Comm communicator, ierr)
```

MPI_Op: MPI_MAX, MPI_MIN, MPI_SUM, MPI_PROD , etc

Topic: MPI Programming

Examples – MPI Broadcast

MPI_Bcast(data, count, MPI_Datatype, root, MPI_Comm, ierr)

Examples:

- call MPI_Bcast(x, 1, MPI_Int, 0, MPI_Comm_World, ierr);
- call MPI_Bcast(numbers, 100, MPI_Double_Precision, 0, MPI_Comm_World, ierr);

Examples – MPI Reduce

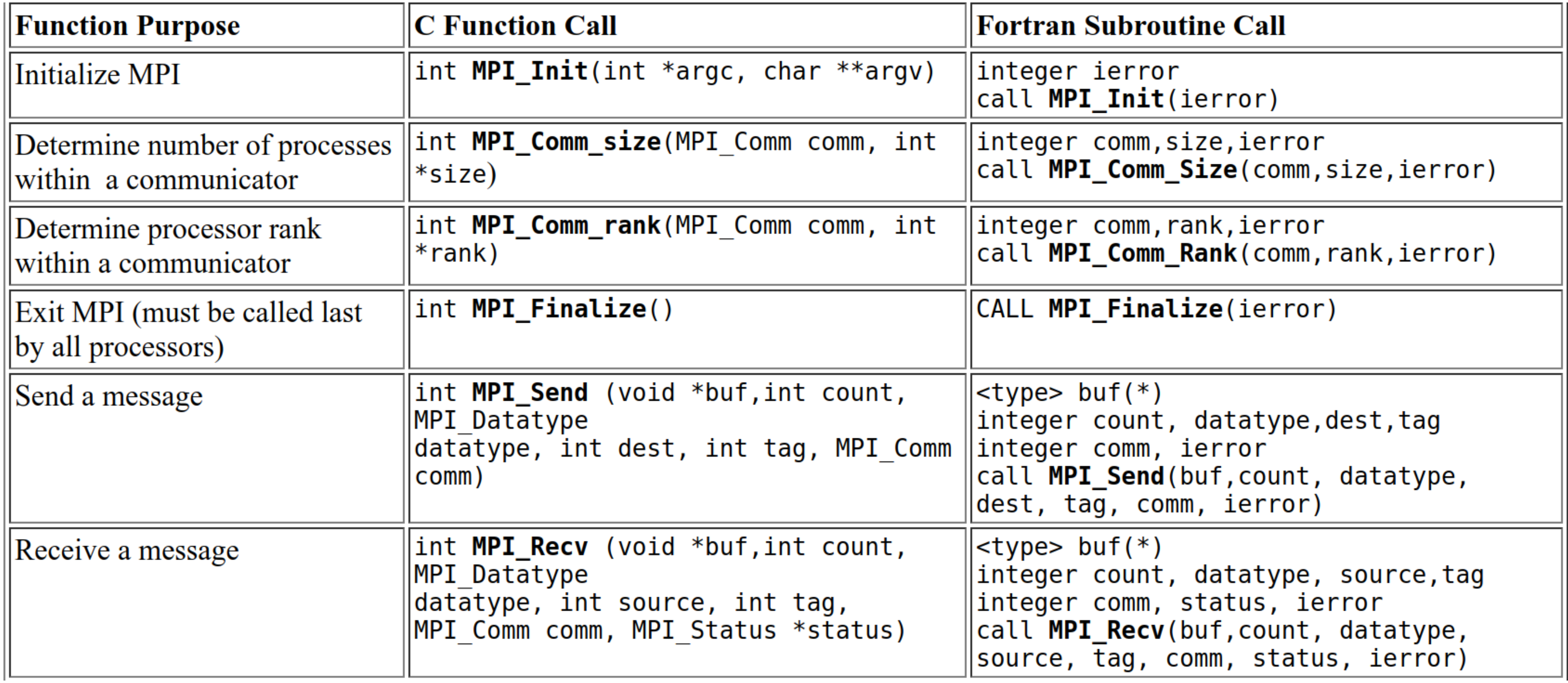
MPI_Reduce(send_data,recv_data, count, MPI_Datatype, MPI_Op, root, MPI_Comm, ierr)

Examples:

- call MPI_Reduce(avg_distance, new_avg_distance, 1, MPI_Real, MPI_Sum, 0, MPI_Comm_World, ierr)
- call MPI_Reduce(old, new, 100, MPI_Double_Precision, MPI_Sum, 0, MPI_Comm_World, ierr)

Topic: MPI Programming

Fortran and C



Function Purpose	C Function Call	Fortran Subroutine Call
Initialize MPI	int **MPI_Init**(int *argc, char **argv)	integer ierror call **MPI_Init**(ierror)
Determine number of processes within a communicator	int **MPI_Comm_size**(MPI_Comm comm, int *size)	integer comm,size,ierror call **MPI_Comm_Size**(comm,size,ierror)
Determine processor rank within a communicator	int **MPI_Comm_rank**(MPI_Comm comm, int *rank)	integer comm,rank,ierror call **MPI_Comm_Rank**(comm,rank,ierror)
Exit MPI (must be called last by all processors)	int **MPI_Finalize**()	CALL **MPI_Finalize**(ierror)
Send a message	int **MPI_Send** (void *buf,int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm)	<type> buf(*) integer count, datatype,dest,tag integer comm, ierror call **MPI_Send**(buf,count, datatype, dest, tag, comm, ierror)
Receive a message	int **MPI_Recv** (void *buf,int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm, MPI_Status *status)	<type> buf(*) integer count, datatype, source,tag integer comm, status, ierror call **MPI_Recv**(buf,count, datatype, source, tag, comm, status, ierror)

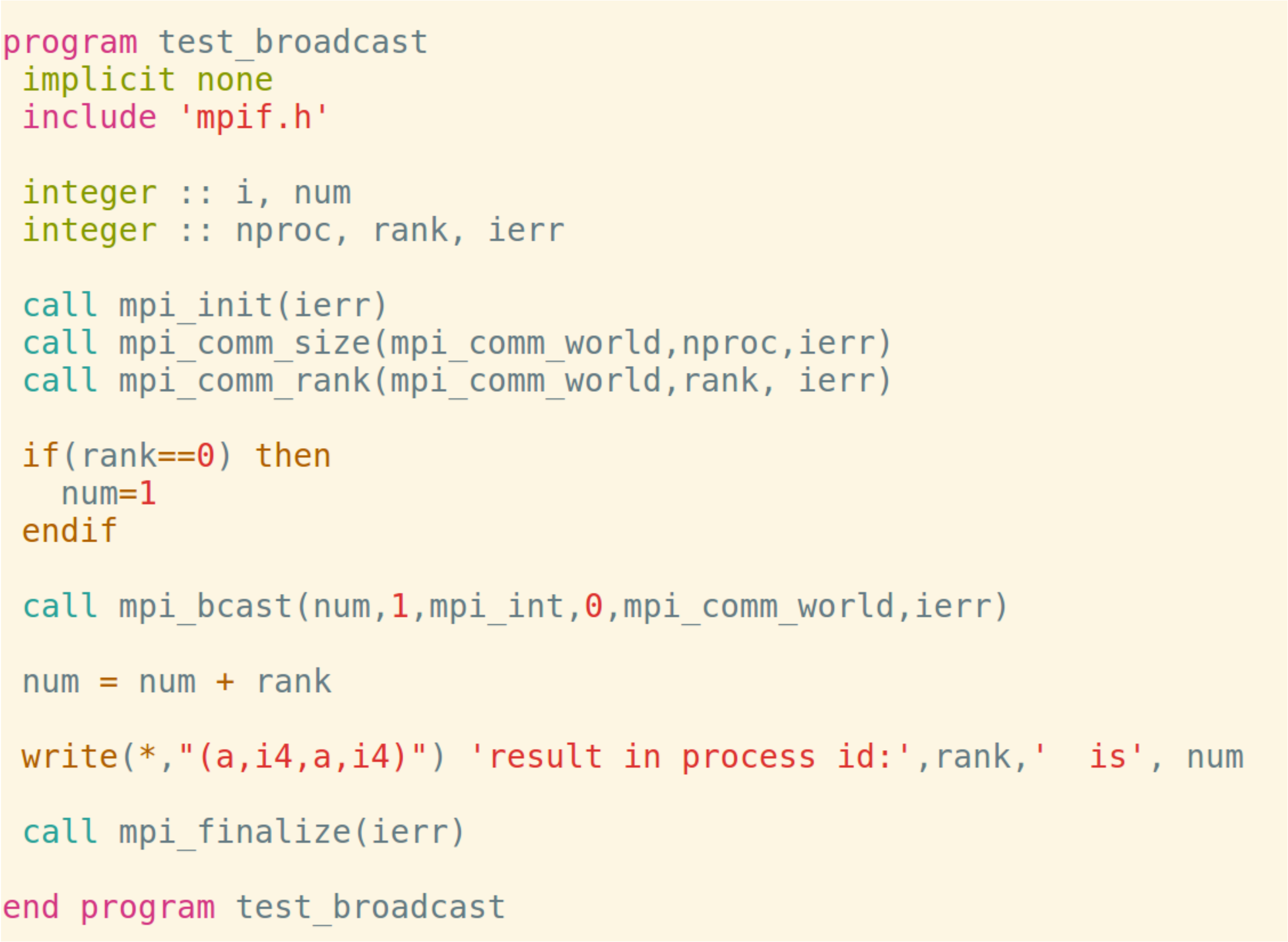
https://www.dartmouth.edu/~rc/classes/intro_mpi/basic_mpi_routines.htm

Standard input

Standard input is only accessible to the root process (rank=0)

Topic: MPI Programming

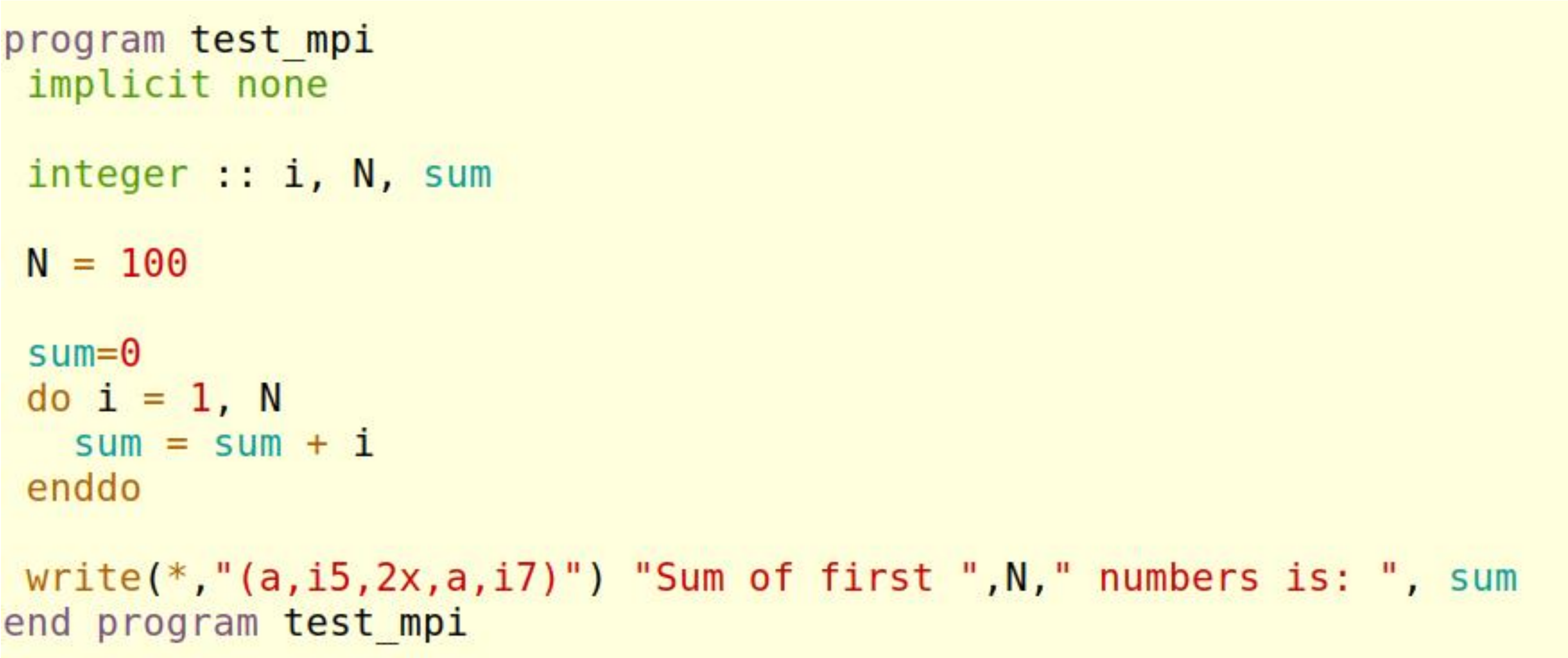
mpi_bcast



```
program test_broadcast
 implicit none
 include 'mpif.h'

 integer :: i, num
 integer :: nproc, rank, ierr

 call mpi_init(ierr)
 call mpi_comm_size(mpi_comm_world,nproc,ierr)
 call mpi_comm_rank(mpi_comm_world,rank, ierr)

 if(rank==0) then
   num=1
 endif

 call mpi_bcast(num,1,mpi_int,0,mpi_comm_world,ierr)

 num = num + rank

 write(*,"(a,i4,a,i4)") 'result in process id:',rank,'  is', num

 call mpi_finalize(ierr)

end program test_broadcast
```

Topic: MPI Programming

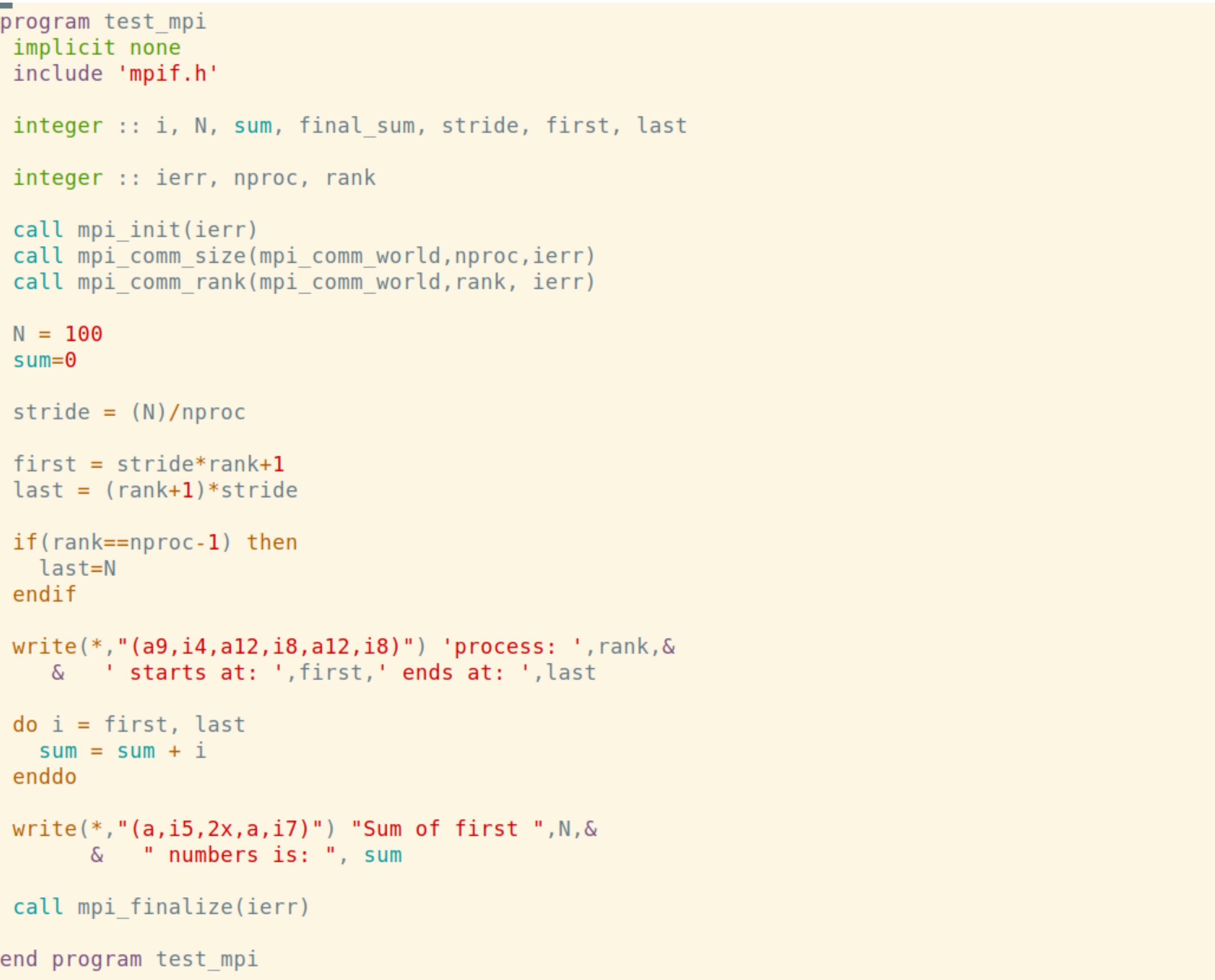
Problem - 1



```
program test_mpi
 implicit none

 integer :: i, N, sum

 N = 100

 sum=0
 do i = 1, N
   sum = sum + i
 enddo

 write(*,"(a,i5,2x,a,i7)") "Sum of first ",N," numbers is: ", sum
end program test_mpi
```

Topic: MPI Programming

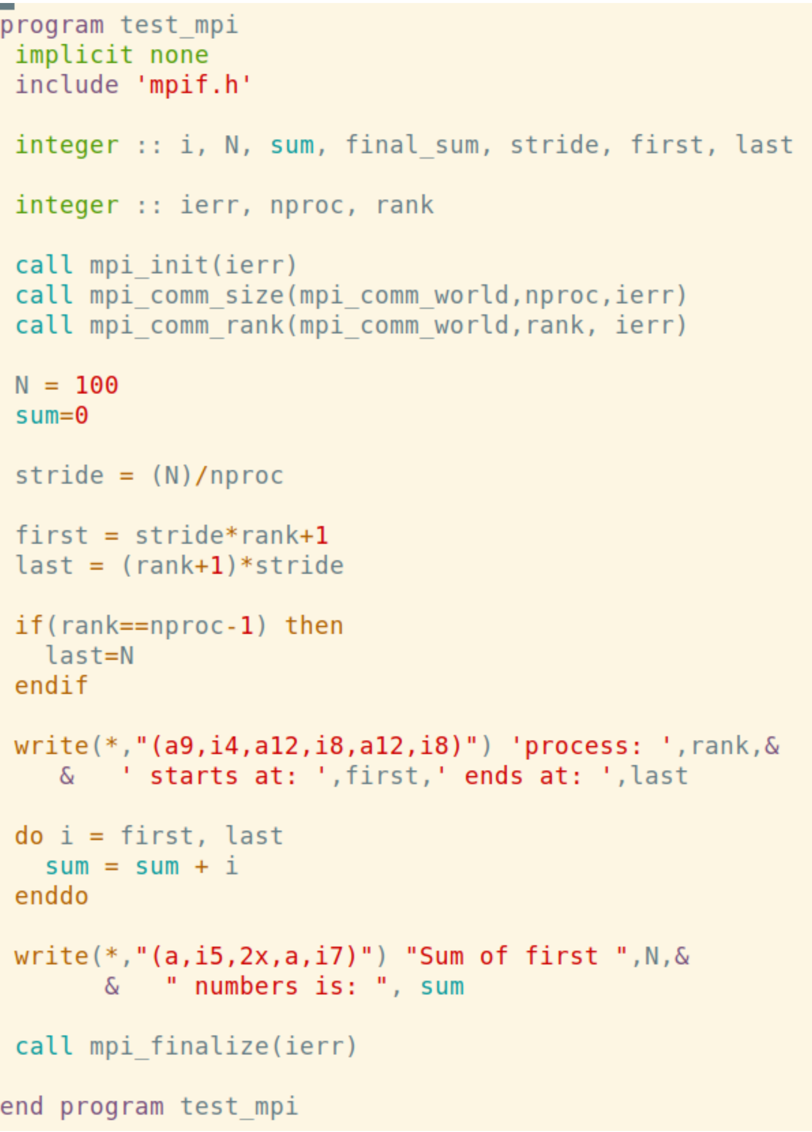
Problem - 1



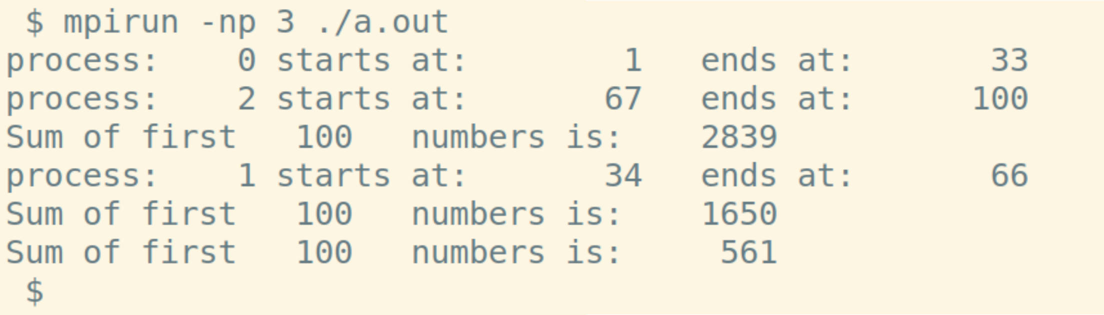
```
program test_mpi
 implicit none
 include 'mpif.h'

 integer :: i, N, sum, final_sum, stride, first, last

 integer :: ierr, nproc, rank

 call mpi_init(ierr)
 call mpi_comm_size(mpi_comm_world,nproc,ierr)
 call mpi_comm_rank(mpi_comm_world,rank, ierr)

 N = 100
 sum=0

 stride = (N)/nproc

 first = stride*rank+1
 last = (rank+1)*stride

 if(rank==nproc-1) then
   last=N
 endif

 write(*,"(a9,i4,a12,i8,a12,i8)") 'process: ',rank,&
    &   ' starts at: ',first,' ends at: ',last

 do i = first, last
   sum = sum + i
 enddo

 write(*,"(a,i5,2x,a,i7)") "Sum of first ",N,&
       &   " numbers is: ", sum

 call mpi_finalize(ierr)

end program test_mpi
```

Topic: MPI Programming

Problem – 1



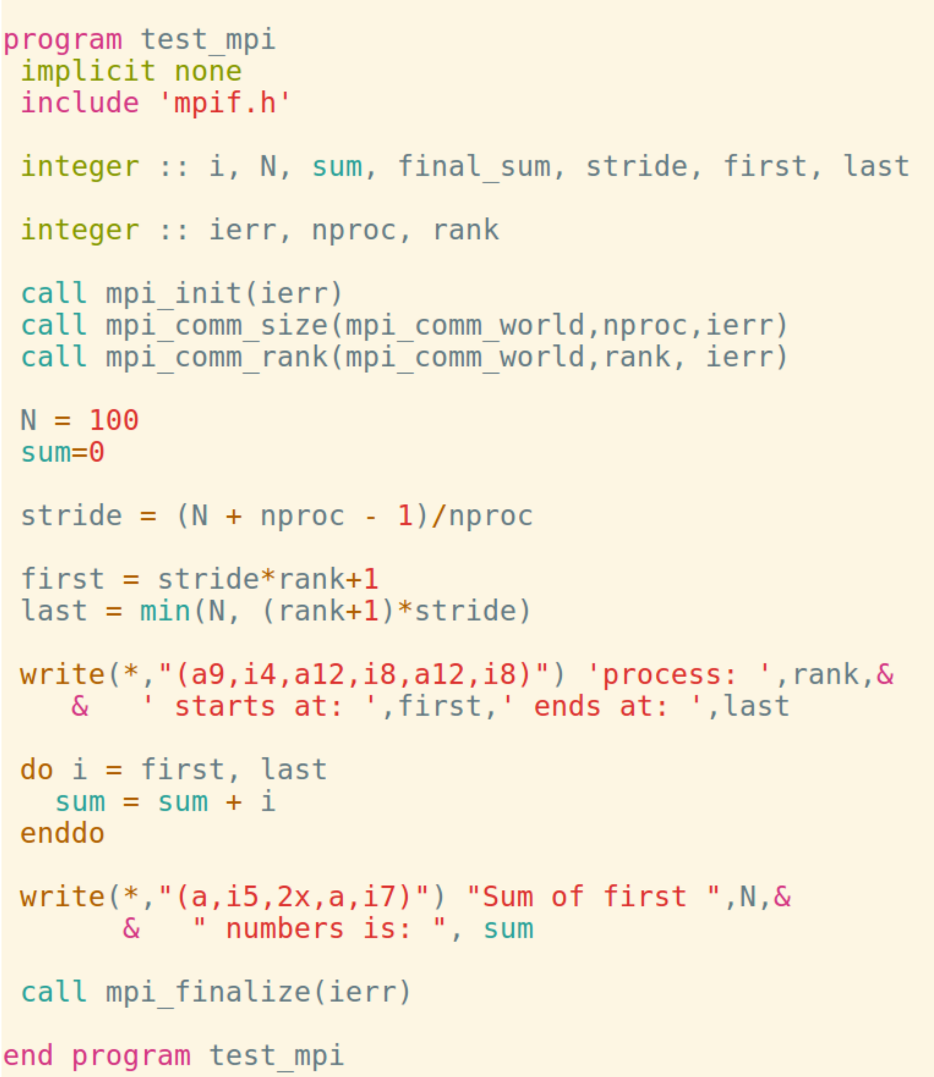
```fortran
program test_mpi
 implicit none
 include 'mpif.h'

 integer :: i, N, sum, final_sum, stride, first, last

 integer :: ierr, nproc, rank

 call mpi_init(ierr)
 call mpi_comm_size(mpi_comm_world,nproc,ierr)
 call mpi_comm_rank(mpi_comm_world,rank, ierr)

 N = 100
 sum=0

 stride = (N)/nproc

 first = stride*rank+1
 last = (rank+1)*stride

 if(rank==nproc-1) then
   last=N
 endif

 write(*,"(a9,i4,a12,i8,a12,i8)") 'process: ',rank,&
    &   ' starts at: ',first,' ends at: ',last

 do i = first, last
   sum = sum + i
 enddo

 write(*,"(a,i5,2x,a,i7)") "Sum of first ",N,&
       &   " numbers is: ", sum

 call mpi_finalize(ierr)

end program test_mpi
```



```
 $ mpirun -np 3 ./a.out
process:    0 starts at:        1   ends at:       33
process:    2 starts at:       67   ends at:      100
Sum of first   100   numbers is:   2839
process:    1 starts at:       34   ends at:       66
Sum of first   100   numbers is:   1650
Sum of first   100   numbers is:    561
 $
```

Topic: MPI Programming

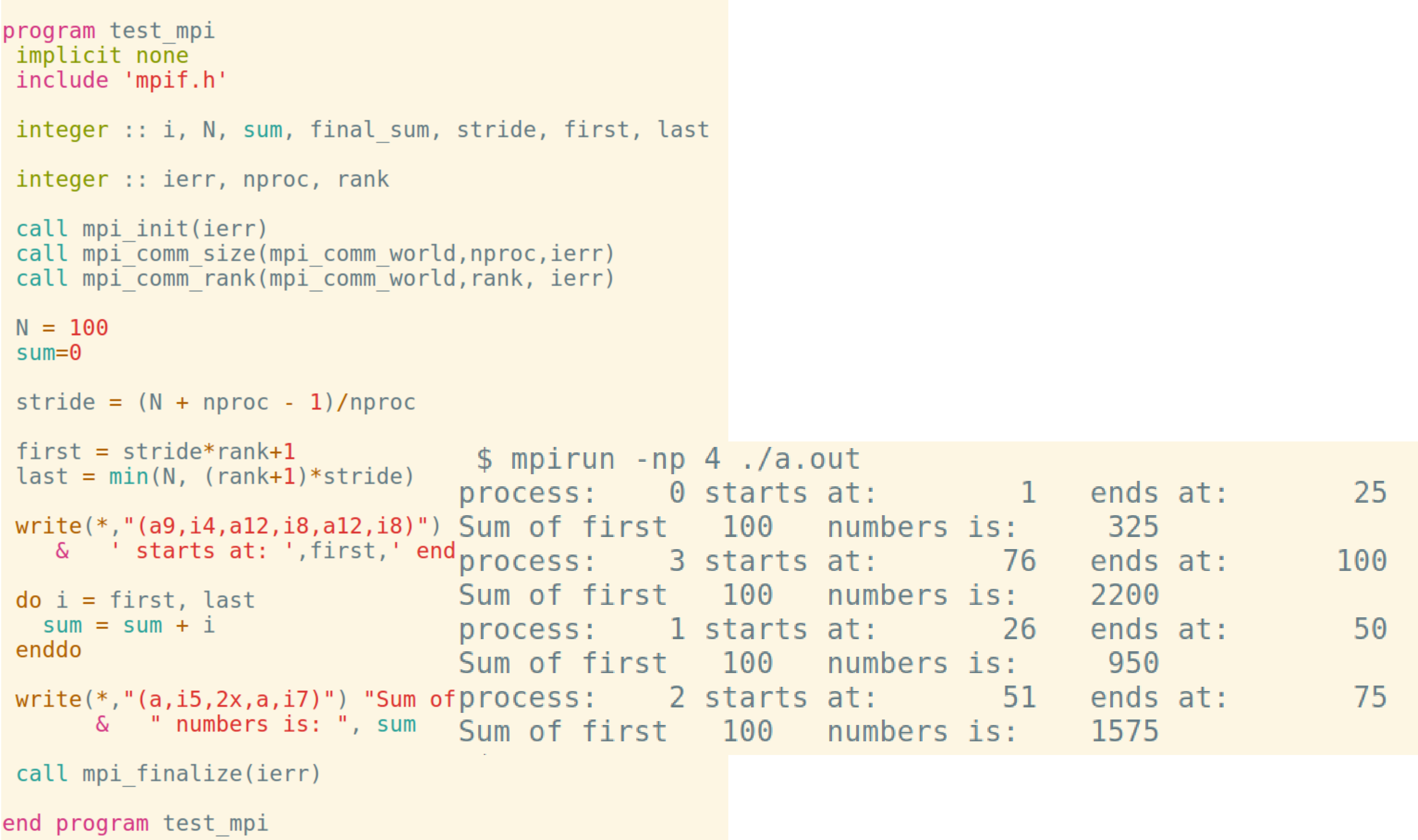
Problem - 1



```
program test_mpi
 implicit none
 include 'mpif.h'

 integer :: i, N, sum, final_sum, stride, first, last

 integer :: ierr, nproc, rank

 call mpi_init(ierr)
 call mpi_comm_size(mpi_comm_world,nproc,ierr)
 call mpi_comm_rank(mpi_comm_world,rank, ierr)

 N = 100
 sum=0

 stride = (N + nproc - 1)/nproc

 first = stride*rank+1
 last = min(N, (rank+1)*stride)

 write(*,"(a9,i4,a12,i8,a12,i8)") 'process: ',rank,&
    &   ' starts at: ',first,' ends at: ',last

 do i = first, last
   sum = sum + i
 enddo

 write(*,"(a,i5,2x,a,i7)") "Sum of first ",N,&
       &   " numbers is: ", sum

 call mpi_finalize(ierr)

end program test_mpi
```

Topic: MPI Programming

Problem - 1

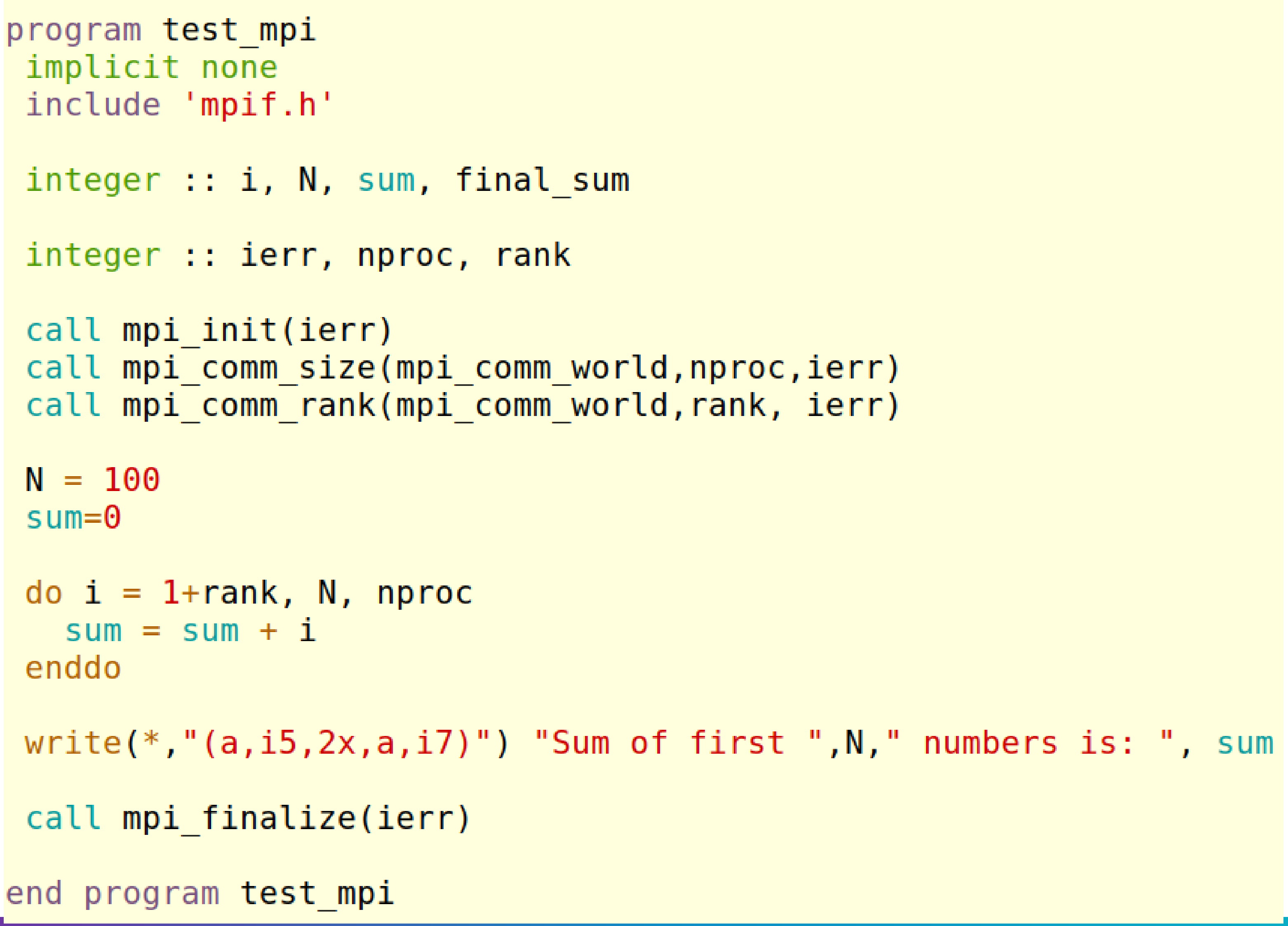


```
program test_mpi
 implicit none
 include 'mpif.h'

 integer :: i, N, sum, final_sum, stride, first, last

 integer :: ierr, nproc, rank

 call mpi_init(ierr)
 call mpi_comm_size(mpi_comm_world,nproc,ierr)
 call mpi_comm_rank(mpi_comm_world,rank, ierr)

 N = 100
 sum=0

 stride = (N + nproc - 1)/nproc

 first = stride*rank+1
 last = min(N, (rank+1)*stride)

 write(*,"(a9,i4,a12,i8,a12,i8)")
    &   ' starts at: ',first,' end

 do i = first, last
   sum = sum + i
 enddo

 write(*,"(a,i5,2x,a,i7)") "Sum of
       &   " numbers is: ", sum

 call mpi_finalize(ierr)

end program test_mpi
```

```
 $ mpirun -np 4 ./a.out
process:    0 starts at:        1   ends at:       25
Sum of first   100   numbers is:    325
process:    3 starts at:       76   ends at:      100
Sum of first   100   numbers is:   2200
process:    1 starts at:       26   ends at:       50
Sum of first   100   numbers is:    950
process:    2 starts at:       51   ends at:       75
Sum of first   100   numbers is:   1575
```

Topic: MPI Programming

Problem - 1



```
program test_mpi
 implicit none
 include 'mpif.h'

 integer :: i, N, sum, final_sum

 integer :: ierr, nproc, rank

 call mpi_init(ierr)
 call mpi_comm_size(mpi_comm_world,nproc,ierr)
 call mpi_comm_rank(mpi_comm_world,rank, ierr)

 N = 100
 sum=0

 do i = 1+rank, N, nproc
   sum = sum + i
 enddo

 write(*,"(a,i5,2x,a,i7)") "Sum of first ",N," numbers is: ", sum

 call mpi_finalize(ierr)

end program test_mpi
```

Topic: MPI Programming

Hands-on

1. Use MPI_Reduce function in the program given in the previous slide and print the result in the root process
2. Write a MPI Fortran program to read a five numbers from standard input and write the output to terminal.
3. Write a Fortran program to calculate the number of prime numbers in the range of 2 and N, and, implement MPI programming model.

Topic: MPI Programming

Assignment

1. Explain the code **pi_numerical_comments.f90**

2. Explain the code **pi_dartboard_comments.f90**