

Exercise 8

Exercise 8

Task 0 initializes a one-dimensional array assigning to each cell the value of its index. This array is then divided into chunks and sent to other processes. After having received the proper chunk, each process updates it by adding its rank and then sends it back to root process. (Analyze the cases for equal and not equal chunks separately).

HINTS:

C

MPI_SCATTER

```
int MPI_Scatter(void* sendbuf, int sendcount, MPI_Datatype sendtype, void*
recvbuf, int recvcount, MPI_Datatype recvtype, int root, MPI_Comm comm)
```

MPI_GATHER

```
int MPI_Gather(void* sendbuf, int sendcount, MPI_Datatype sendtype, void* recvbuf,
int recvcount, MPI_Datatype recvtype, int root, MPI_Comm comm)
```

MPI_SCATTERV

```
int MPI_Scatterv(void* sendbuf, int *sendcounts, int *displs, MPI_Datatype
sendtype, void* recvbuf, int recvcount, MPI_Datatype recvtype, int root, MPI_Comm
comm)
```

MPI_GATHERV

```
int MPI_Gatherv(void* sendbuf, int sendcount, MPI_Datatype sendtype, void*
recvbuf, int *recvcounts, int *displs, MPI_Datatype recvtype, int root, MPI_Comm
comm)
```

MPI_INIT

```
int MPI_Init(int *argc, char ***argv)
```

MPI_COMM_SIZE

```
int MPI_Comm_size(MPI_Comm comm, int *size)
```

MPI_COMM_RANK

```
int MPI_Comm_rank(MPI_Comm comm, int *rank)
```

MPI_FINALIZE

```
int MPI_Finalize(void)
```

FORTTRAN

MPI_SCATTER	<pre> MPI_SCATTER(SENDBUF, SENDCOUNT, SENDTYPE, RECVBUF, REVCOUNT, RECVTYPE, ROOT, COMM, IERROR) <type> SENDBUF(*), RECVBUF(*) INTEGER SENDCOUNT, SENDTYPE, REVCOUNT, RECVTYPE, ROOT, COMM, IERROR </pre>
MPI_GATHER	<pre> MPI_GATHER(SENDBUF, SENDCOUNT, SENDTYPE, RECVBUF, REVCOUNT, RECVTYPE, ROOT, COMM, IERROR) <type> SENDBUF(*), RECVBUF(*) INTEGER SENDCOUNT, SENDTYPE, REVCOUNT, RECVTYPE, ROOT, COMM, IERROR </pre>
MPI_SCATTERV	<pre> MPI_SCATTERV(SENDBUF, SENDCOUNTS, DISPLS, SENDTYPE, RECVBUF, REVCOUNT, RECVTYPE, ROOT, COMM, IERROR) <type> SENDBUF(*), RECVBUF(*) INTEGER SENDCOUNTS(*), DISPLS(*), SENDTYPE, REVCOUNT, RECVTYPE, ROOT, COMM, IERROR </pre>
MPI_GATHERV	<pre> MPI_GATHERV(SENDBUF, SENDCOUNT, SENDTYPE, RECVBUF, REVCOUNTS, DISPLS, RECVTYPE, ROOT, COMM, IERROR) <type> SENDBUF(*), RECVBUF(*) INTEGER SENDCOUNT, SENDTYPE, REVCOUNTS(*), DISPLS(*), RECVTYPE, ROOT, COMM, IERROR </pre>
MPI_INIT	<pre> MPI_INIT(IERROR) INTEGER IERROR </pre>
MPI_COMM_SIZE	<pre> MPI_COMM_SIZE(COMM, SIZE, IERROR) INTEGER COMM, SIZE, IERR0 </pre>
MPI_COMM_RANK	<pre> MPI_COMM_SIZE(COMM, SIZE, IERROR) INTEGER COMM, SIZE, IERROR </pre>
MPI_FINALIZE	<pre> MPI_FINALIZE(IERROR) INTEGER IERROR </pre>

[< Solution 7](#)[up](#)[Solution 8 >](#)