Assignment 1 (100 marks)

0	1	2	3	4	…	…	…	P-1

Write a program to perform data (M doubles) exchange with two ranks that are D1 and D2 distances away on the right (D1 < D2). Assume that the total number of processes = P. If rank + D2 > P-1, then the rank sends only to D1. If rank + D1 > P-1, then the rank (i.e. invalid sender) does not send data to any rank. The receivers of a sender rank (D1 and D2 distances away) perform computation as below and send back the data to the sender.

A receiver rank which is D1 distance away: receives **data_received** (M doubles), computes **data_at_D1** as shown below and sends **data_at_D1** to the sender.

```
for (ij = 0; ij<M; ij++)
  data_at_D1[ij] = data_received[ij] * data_received[ij];
```

A receiver rank which is D2 distance away: receives **data_received** (M doubles), computes **data_at_D2** as shown below and sends **data_at_D2** to the sender.

```
for (ij = 0; ij<M; ij++)
  data_at_D2[ij] = log (data_received[ij]);
```

The sender rank receives data_at_D1 (received D1 distance away) and data_at_D2 (received from receiver D2 distance away) and updates data_received before the next iteration as follows:

data to be sent to D1 in the next iteration is updated as:

```
for (ij = 0; ij<M; ij++)
 buffer_updated_for_D1[i] = (unsigned long long)data_received[i] % 100000;
```

data to be sent to D2 in the next iteration is updated as:

```
for (ij = 0; ij<M; ij++)
 buffer_updated_for_D2[i] = data_received[i]*100000;
```

Thus, the sequence for the sender ranks are communication(s) followed by computations at D1 and D2 followed by communication(s) followed by computations at the sender rank. This sequence of operations (you may select any order) occurs for T iterations. At the end of T iterations, all valid senders send the maximum of final **data_received_D1** (among M elements) and maximum of final **data_received_D2** (among M elements) to rank 0. Rank 0 prints the maximum of **data_received_D1** and maximum of **data_received_D2** (only from valid senders) and the total time (T iterations plus final communication/computation).

The initial data must be randomly generated using the value “seed” as shown below.

```
srand(seed); //include <stdlib.h>
for (int i=0; i<M; i++)
   data_received[i] = (double)rand()*(myrank+1)/10000.0;
```

The input to the program is M, D1, D2, T, random seed. (D1>0, D2>D1)
The output should be a **single line** containing three numbers: <maximumD1> <maximumD2> <time>

Final configurations for the report:

```
for execution in 1 to 5 // repeat each configuration 5 times
 for P (number of processes) in 8, 16, 32 // use only one process count in one script
  for M (doubles) in 262144, 1048576 // you may include both in one script
   for D1 in 2
    for D2 in 4
     for T in 10
      for seed in 1000
       mpirun –np P –f hostfile ./src M D1 D2 T seed
```

Use 16 processes per node.

You may only use MPI_Send and MPI_Recv for this assignment.

Plot the time (in seconds) for each data size per process count. Use boxplots (from the 5 executions) for every data point in a plot. Submit one plot file only. Time in seconds on the y-axis and processes (P) on the x-axis.

What to include in the report: Your report must include code description, code compilation and execution instructions and results and the observations from the results. Peruse this LaTeX file.

What to include in the submission: src.c, job scripts, plot script and the input (timing data file), job output files (SLURM), GroupXY.pdf (report), Makefile (if any). Include all your files in a folder named GroupXY and then compress it, please name it GroupXY.zip.

Submission instructions (*only leaders*): Upload GroupXY.zip to hello.iitk.ac.in and submit your code to P2rutor Assignment 1 event (details coming soon). Submissions via email will not be accepted. Multiple submission on MooKIT will incur -5 penalty, so ensure that you have included your final correct code and report in your GroupXY.zip file before uploading.

Approximate split of marks:

Coding (efficient, neat, compact) + execution (including hidden test cases) = 60%
Code documentation: 20%
Report: 20%

Penalty of -10 will be given for violation in specified instructions/naming conventions or omission of any of the mandatory files.

You may mention in the report (conclusions) regarding the percentage of work done by every member (especially in case of non-uniform workload sharing).

Plagiarism will NOT be tolerated, and CSE/IITK policy will be strictly followed. Inter-group discussions regarding anything even remotely related to the Assignment is not allowed. You must only discuss within your own group and ensure that none of your group members are discussing (not even any simple doubt) with group members of any other group.

Due date: 12-02-2026 (Thursday)

You may post any doubts regarding the assignment here:
https://forms.gle/J7GzKEmyKHN6jTH6A