**IT301 – Parallel Computing**

Assignment – 7

**Name:** Niraj Nandish

**Roll No:** 191IT234

1. Graphical user interface, text

   Description automatically generatedProgram 1 – Simple Hello World program
2. Program 2 – MPI\_Send() and MPI\_Recv()
   1. Source is process 0, destinations is process 1 and the tag is 55

A screenshot of a computer

Description automatically generated with medium confidence

* 1. Text

     Description automatically generatedSending the string "PCLAB"
  2. Sending array of elements

Graphical user interface, text

Description automatically generated

1. Program 3 – MPI\_ANY\_SOURCE and MPI\_ANY\_TAG
   1. Observation – In the MPI\_Recv() function, the source parameter is given the value of MPI\_ANY\_SOURCE which means accept the data from any source and the tag parameter is given the value of MPI\_ANY\_TAG which means accept the data with any tag value.

Text

Description automatically generated

1. Program 4 – MPI\_Send() and MPI\_Recv() with mismatched tag
   1. Graphical user interface, text

      Description automatically generatedObservation – Since the tag value was different in MPI\_Send() and MPI\_Recv() functions, hence the data wasn't parsed in the receiving end. But when the tag value is corrected, it gets parsed, and we see the output.
2. Program 5 - MPI\_Send() and MPI\_Recv() standard mode
   1. Observation – In process 0, we are first sending array "x" with the tag 1 and array "y" with the tag 2. In process 1, the first MPI\_Recv() function waits for data with tag 2(Array "y" in P0) and stores that in array "x" and the second MPI\_Recv() function waits for data with tag 1(Array "x" in P0) and stores that in array "y". So the first MPI\_Send() function was blocked as the first MPI\_Recv() function had a different tag value.

Graphical user interface, text

Description automatically generated