

## Distributed Computing Lab Programs

1. Demonstration of MPI\_Send and MPI\_Recv
2. Demonstration of MPI\_Scatter and MPI\_Gather
3. Demonstration of MPI Broadcast operation
4. Demonstration of MPI\_Reduce and MPI\_Allreduce (MPI\_MAX, MPI\_MIN, MPI\_SUM, MPI\_PROD).
5. Write an OpenMP program that computes the sum of the first N integers using a parallel for loop. Use the `#pragma omp` directive along with the private and reduction clauses.
6. Write an OpenMP program to compute the Fibonacci sequence using task parallelism. The program should use a recursive function where each Fibonacci computation for  $\text{fib}(n-1)$  and  $\text{fib}(n-2)$  is created as an independent task.
7. Estimate the value of pi using: Parallelize the code by removing loop carried dependency.
8. Write a parallel C program using OpenMP in which each thread obtains its thread number and adds it to a global shared variable. Use the `#pragma omp critical` directive to avoid race conditions. Print the intermediate updates and the final sum.
9. Write a program to sort an array of n elements using both sequential and parallel merge sort (using Section). Record the difference in execution time.