## ISL77 - Parallel Programming Lab

**Part A: OpenMP Programs**

1. Write a program to sort an array on n elements using both sequential and parallel merge sort (using Section). Record the difference in execution time.
2. Estimate the value of pi using:

Parallelize the code by removing loop carried dependency and record both serial and parallel execution times.

1. Write an OpenMP program that divides the Iterations into chunks containing 2 iterations, respectively (OMP\_SCHEDULE=static,2). Its input should be the number of iterations, and its output should be which iterations of a parallelized for loop are executed by which thread. For example, if there are two threads and four iterations, the output might be the following:
   1. Thread 0: Iterations 0 −− 1
   2. Thread 1: Iterations 2 −− 3
2. Write a program to calculate n Fibonacci numbers using Parallel Directive. Demonstrate elimination of the race condition using Schedule directive.
3. Write a program to find the prime numbers from 1 to n employing parallel for directive. Record both serial and parallel execution times.
4. Parallel Vector Addition
5. Write a program calculate the sum of first 100 Numbers. (Using critical directive)

**Part B: MPI programs**

1. Write a MPI program to implement Blocking send and receive functions.
2. Write a MPI program to demonstrate deadlock and implement a solution to avoid deadlock
3. Write a MPI program to implement Vector Addition of two vectors to generate resultant vector using Scatter and gather operation.
4. Write a MPI program to find factorial of a number using Broadcast and Reduce operation.
5. Write a MPI program to generate all prime numbers till ‘n’ (n being user input).

**Marks splitup:**

One program from each part.

Students have to execute 2 programs.

Part A Part B

Write up 4 4

Execution 17 17

MCQ 8

Change of Program: -4 marks/part