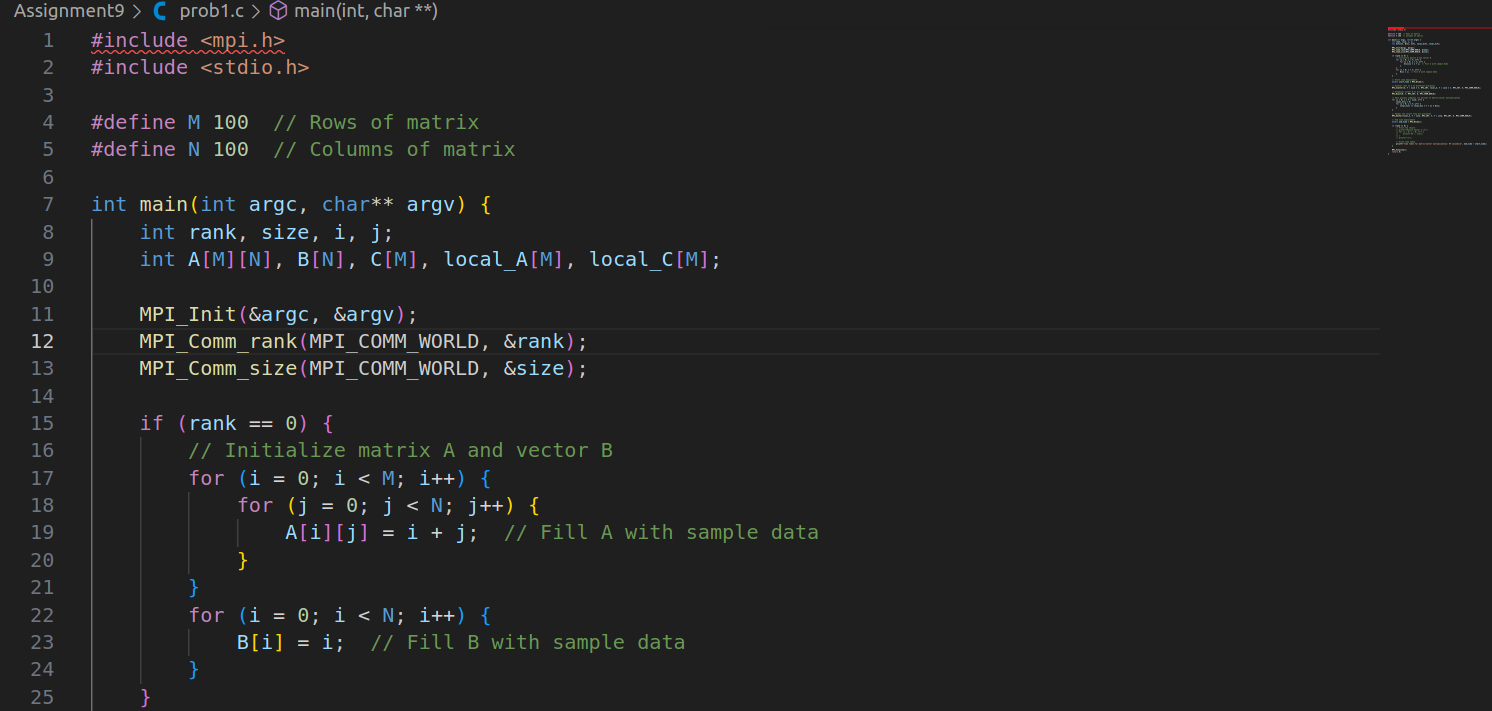
# Final Year B. Tech., Sem VII 2024-

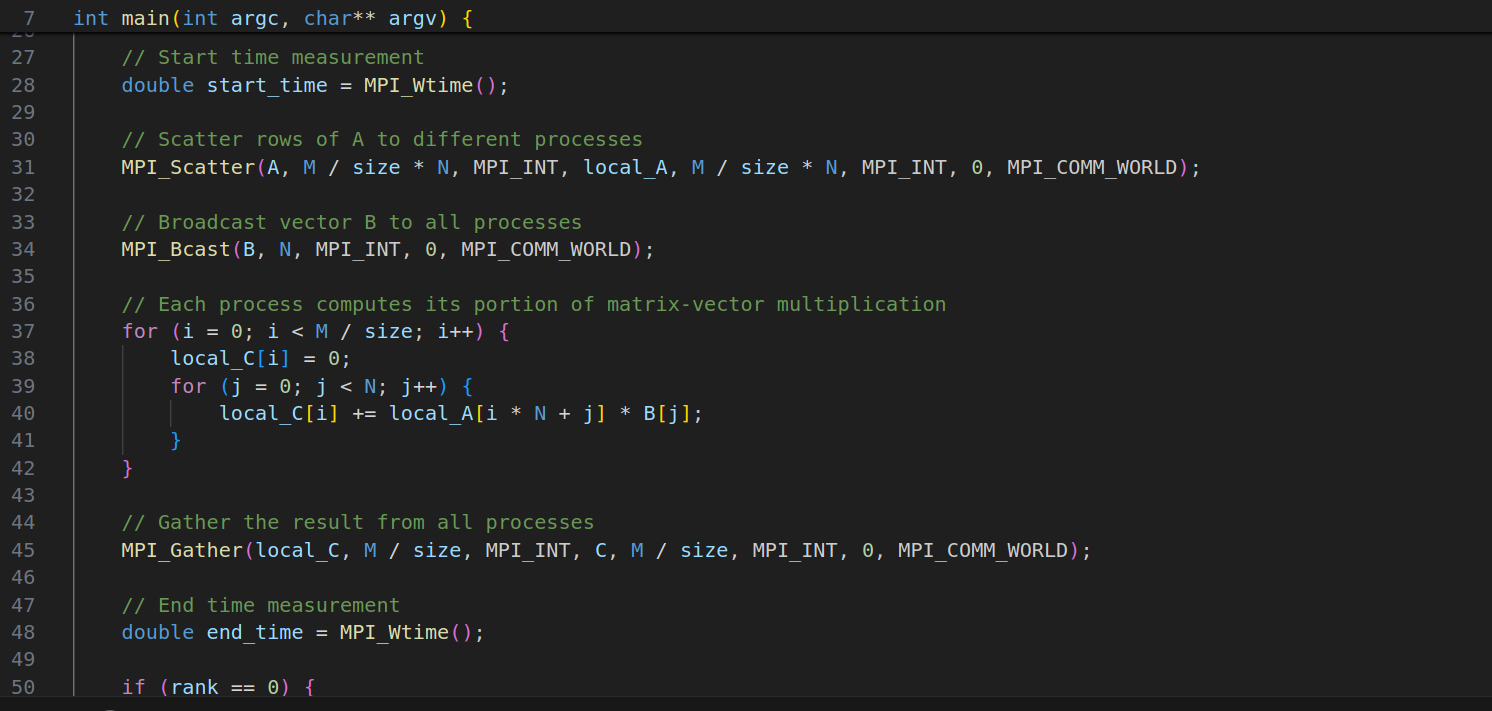
High Performance Computing Lab

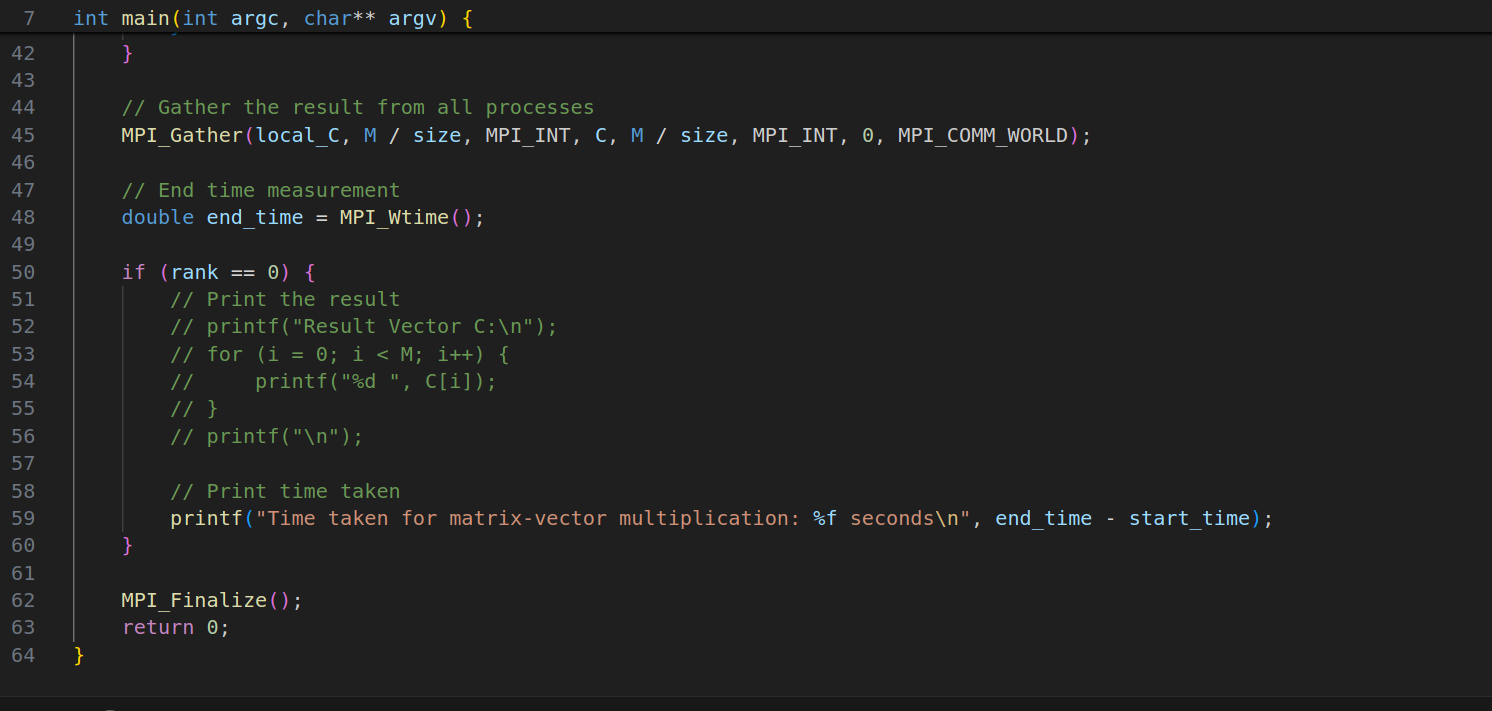
**Practical No. 9**

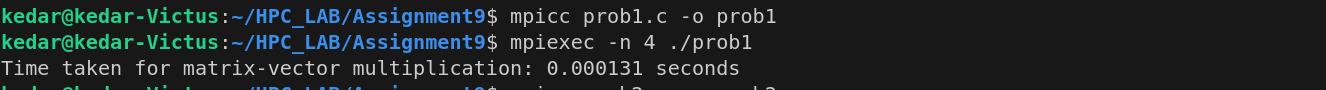
**PRN : 21510011**

## Implement Matrix-Vector Multiplication using MPI. Use different number of processes and analyze the performance.



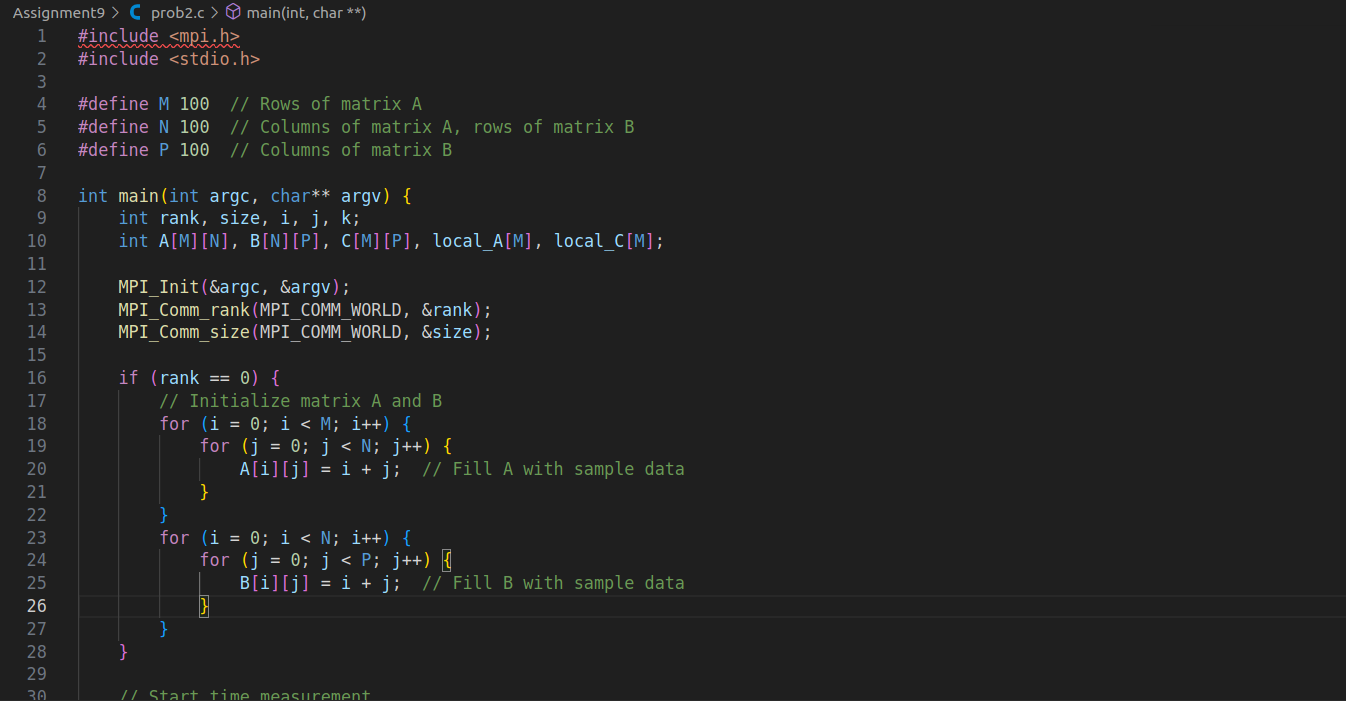


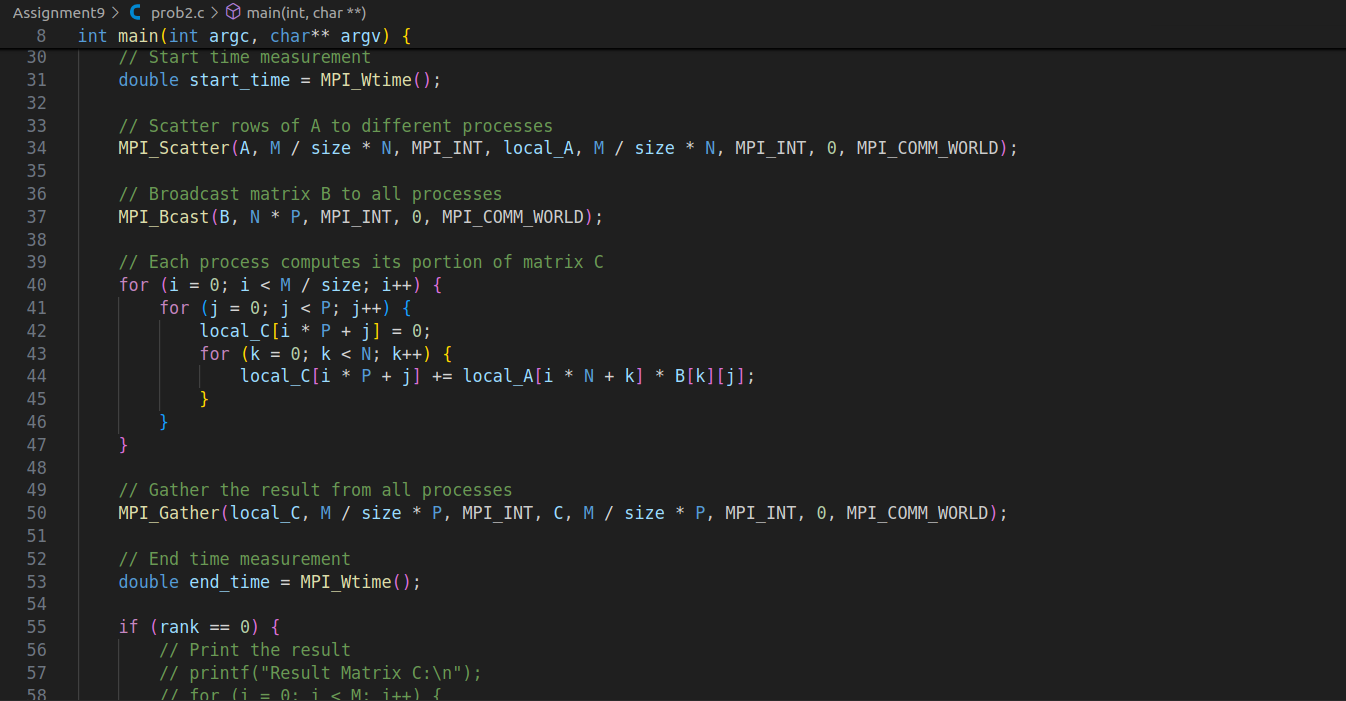


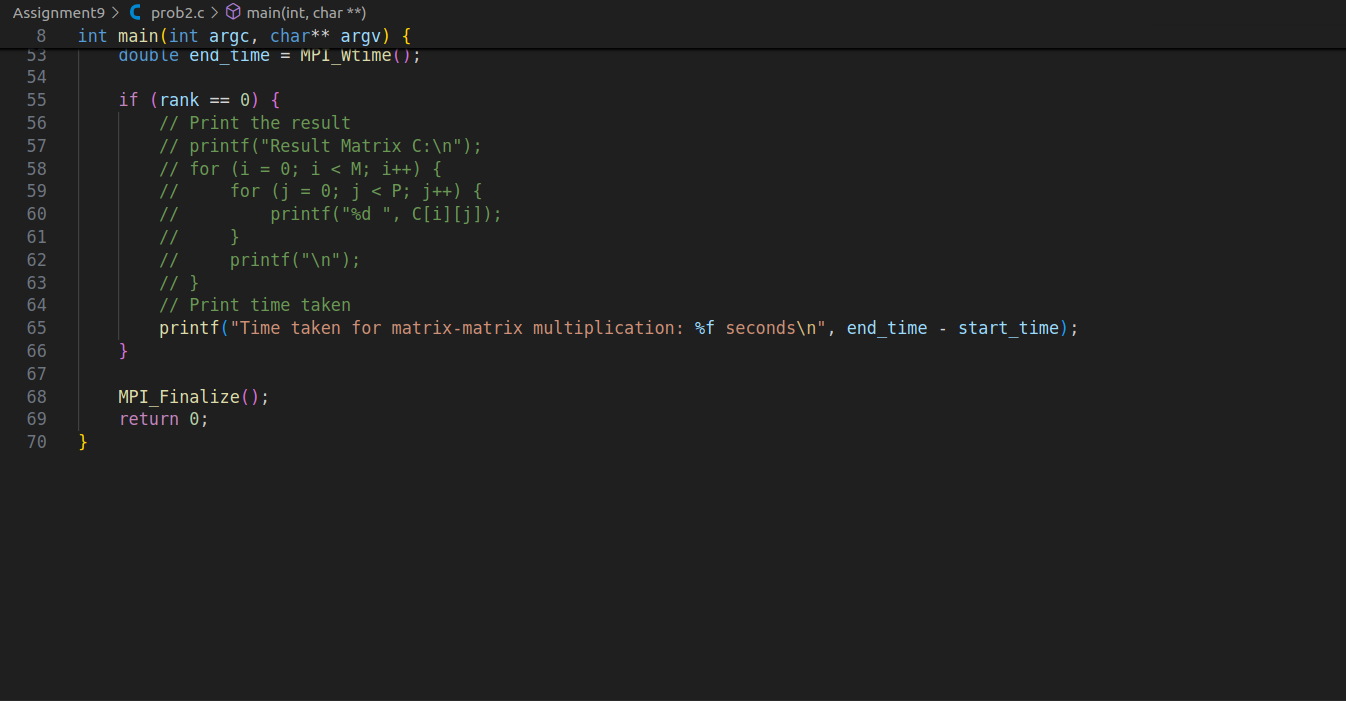


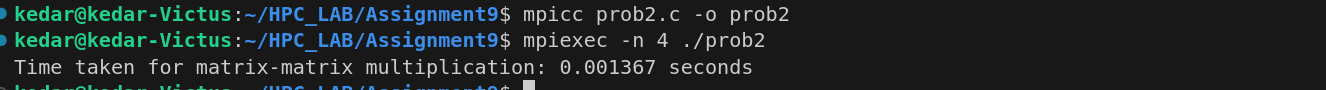
It can be observed from the program that as the processes are increased from 1 to 4, the speedup consistently increases. The execution time was around 1.245 seconds for a single process. it was reduced to 0.678 seconds, For 2 processes (speed up ~1.83x). In the end, with 4 processes, it was even reduced to 0.355 seconds resulting in speed up of approximately 3.5. As impressive as the reductions are for the computation time, more processes also result to more communication overhead which brings the speed up away from perfect linear rise. This discussion elaborates on the benefits of executing such large scale operations as matrix computations in parallel, yet it also brings forth the issue of communication cost.

## Implement Matrix-Matrix Multiplication using MPI. Use different number of processes and analyze the performance.







Analysis :

The MPI Matrix-Matrix Multiplication application is a good example of how useful extensions of matrix operations can be for parallel processing, particularly when not more than 1 or 2 processes. The considerable speedup achieved suggests that workload has been succès in its distribution. But with the increase in the number of processes assigned, performance improvements appear to come under the ceiling because of the communication overhead, load imbalance and memory bandwidth constraints. This thus calls for better design of parallel algorithms to avoid waste of available resources as well as the complaints of process communication and management.

Github: [Assignment9\_21510011](https://github.com/kedar2953/Assignment9_HPC.git)