# Final Year B. Tech., Sem VII 2025-26

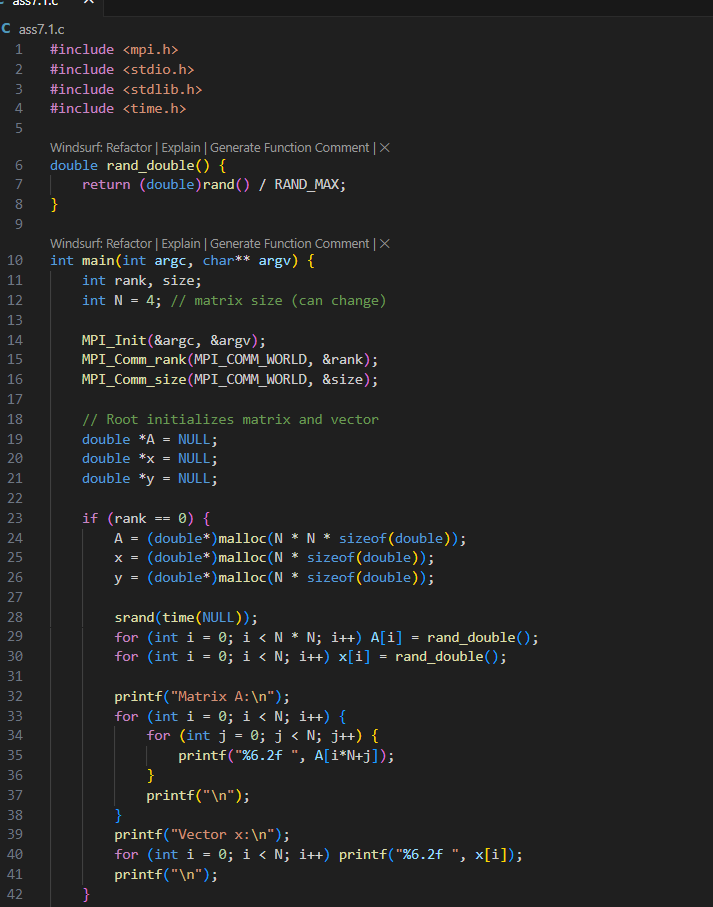
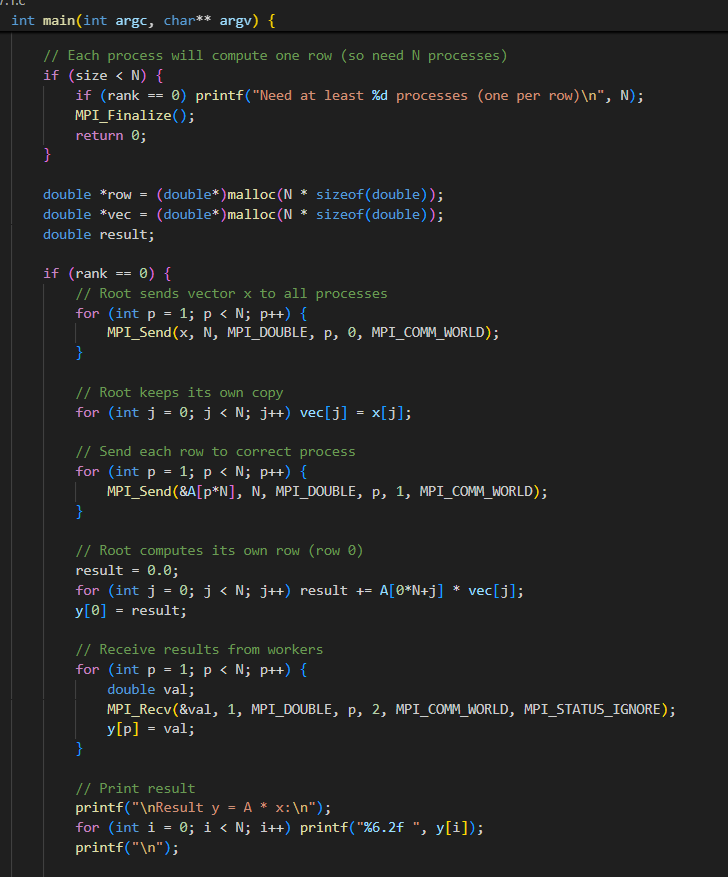
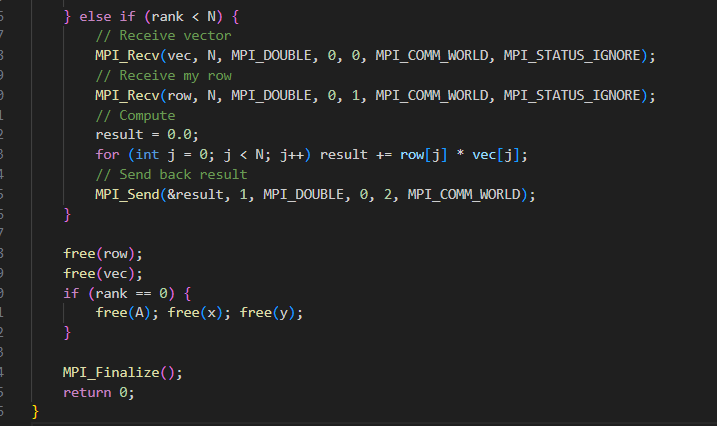
High Performance Computing Lab

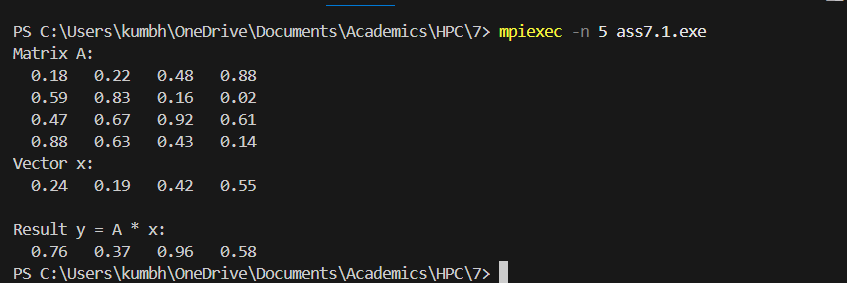
**Practical No. 7**

**Prn: 22510089**

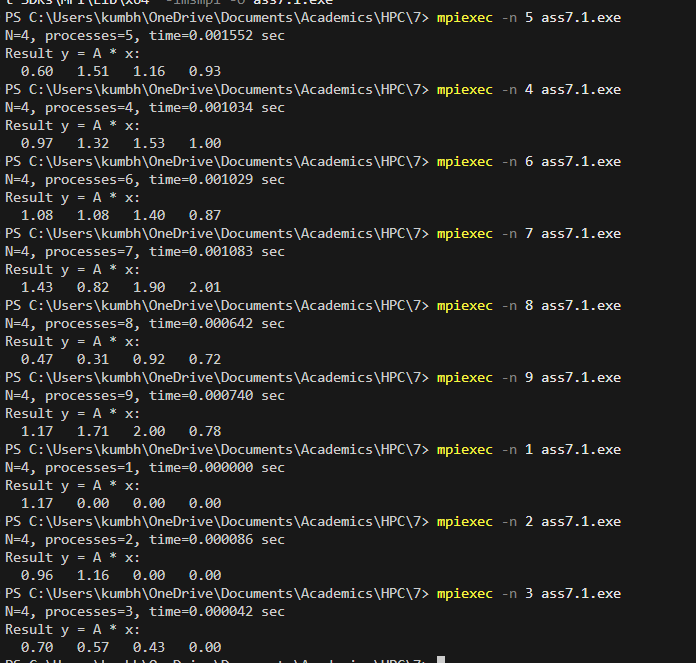
**Batch : B1**

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



**Observations:**

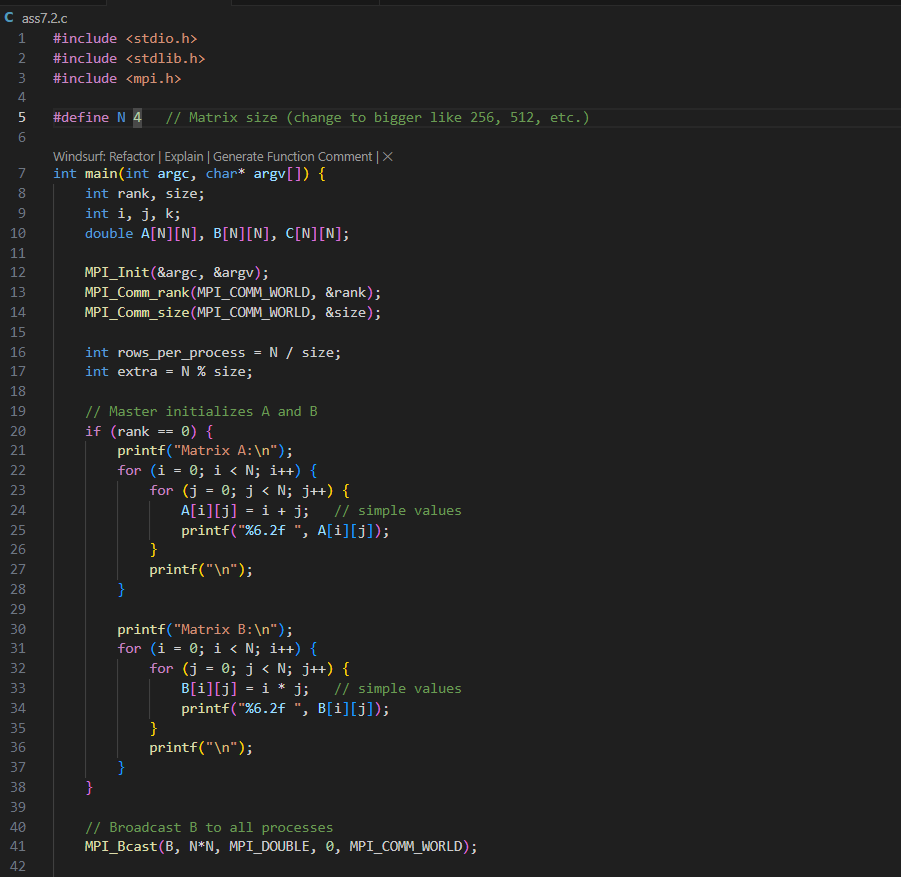
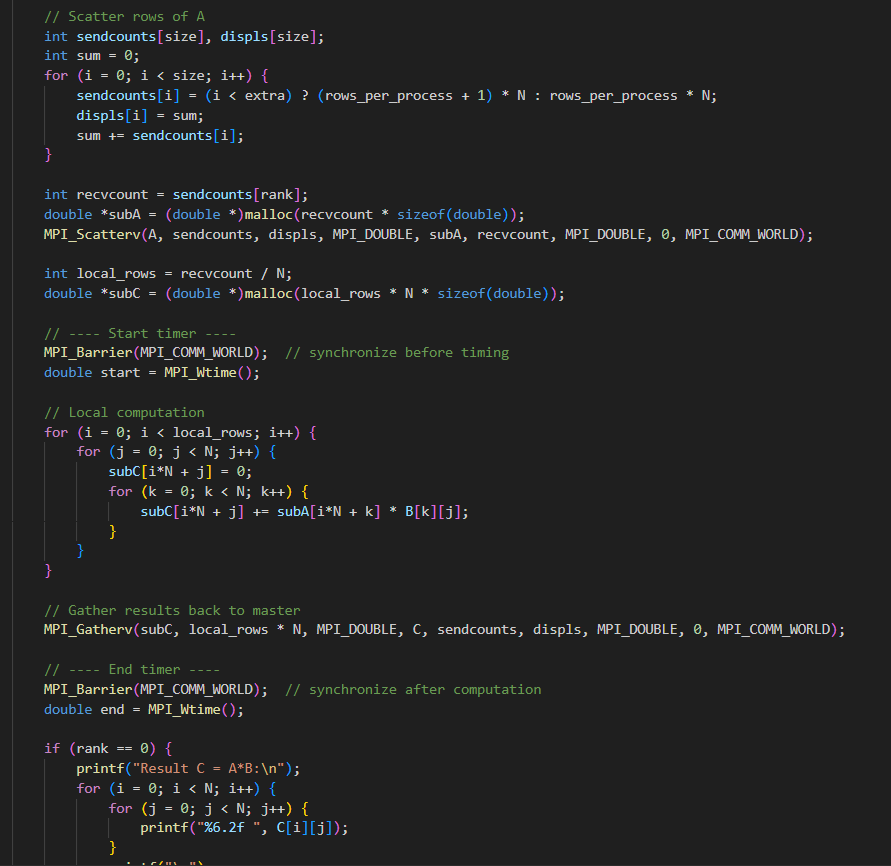
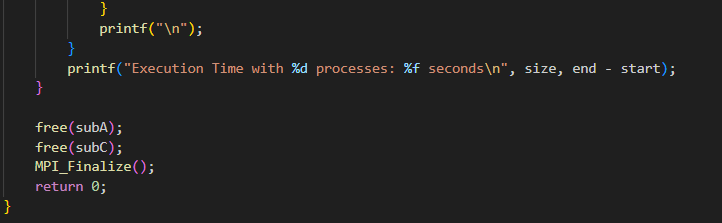


* Times are **not monotonic** (don’t decrease smoothly with more processes).
* Sometimes 2 or 3 processes look “faster” than 4–7.
* That’s because when problem size is small, startup/communication overhead + scheduling noise is larger than the computation itself.
* **For small matrices** (N=4, N=10, etc.):
  + Parallelization **does not help**.
  + Communication overhead dominates.
  + Speedup < 1 (worse than serial).
* **For larger matrices** (N=256, 512, 1024+):
  + Each process does much more computation.
  + Overhead becomes negligible compared to work.
  + You should see actual speedup as number of processes increases.
* **Scalability behavior**:
  + With very small workloads → no gain.
  + With medium workloads → some improvement up to certain process count.
  + With very large workloads → close to linear speedup until communication overhead catches up again.

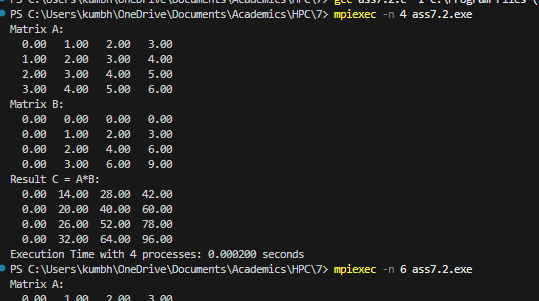
**Analysis:**

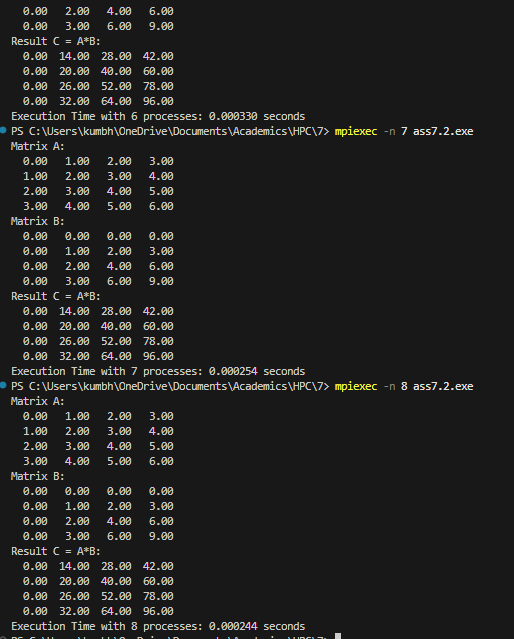
* Your experiment shows that for **tiny matrices (N=4)**, parallelism with MPI is **not efficient**.
* To properly analyze **performance scaling**, you need to run with **much larger matrices** (e.g. 512×512, 1024×1024).

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

Observation:





Github Link :

https://github.com/Vru01/HPC\_22510092