**Name:** Meet Vipul Gandhi

**PRN:** 2020BTECS00112

**Class:** Final Year (Computer Science and Engineering)

**Year:** 2023-24 **Semester:** 1

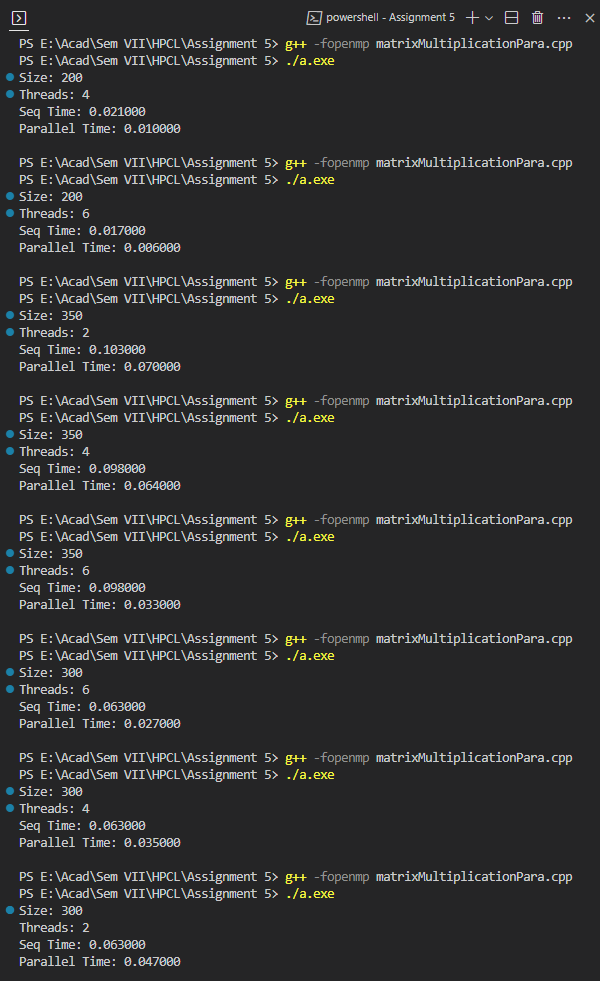
**Course:** High Performance Computing Lab

**Practical No. 5**

**Title of practical:** Implementation of OpenMP programs.

**Problem Statement 1:** Implementation of Matrix-Matrix Multiplication.

**Screenshots:**

****

**Information:**

1. Matrix Multiplication has very high complexity of O(N3) so on parallelizing we can reduce the time by a lot.
2. Few variables (I, j, k) are shared privately to each thread whereas the matrices A, B and C result matrix are shared.
3. Sizes are changed along with the number of threads.

**Analysis:**

****

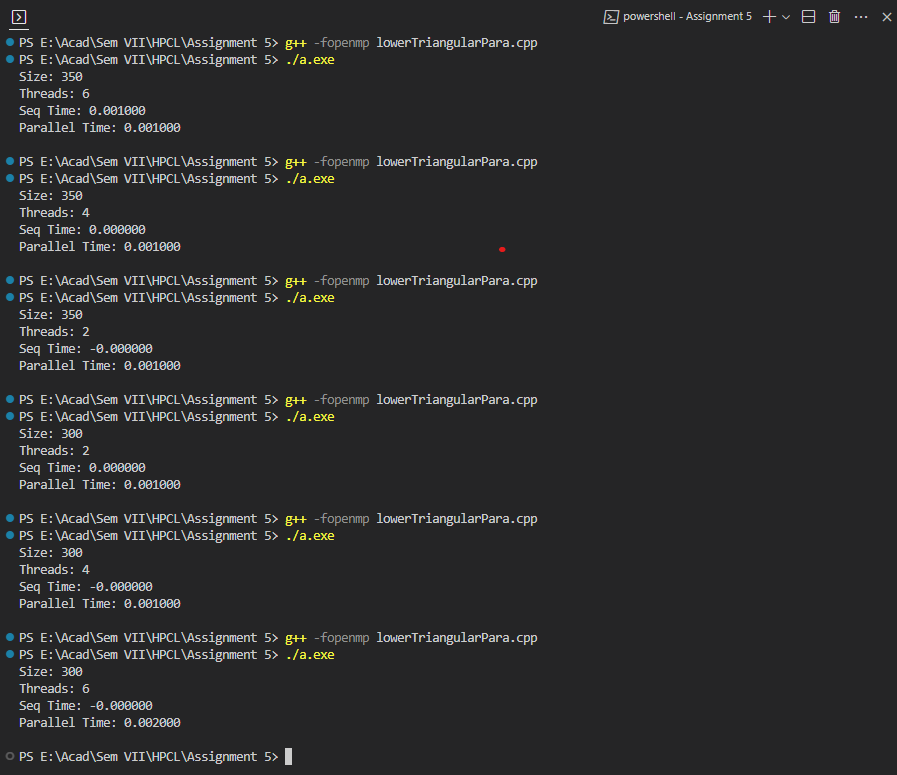
**Speedup:**

1. For matrix size of 200x200 minimum time is 0.006 secs for 10 threads.
2. For matrix size of 300x300 minimum time is 0.002 secs for 10 threads.
3. For matrix size of 350x350 minimum time is 0.003 secs for 10 threads.

****

**Problem Statement 2:** Implementation of sum of two lower triangular matrices.

**Screenshots:**

****

**Information:**

1. Lower triangular matrix is a matrix where all the cells above the diagonals are zero.
2. Cells below the diagonal of matrix can be zero as well as non-zero.
3. Just like above problem we used private sharing for (i, j, k) and shared for (A, B, C) for all the threads.
4. Paralleling the above problem doesn’t affect much to the execution time.

**Analysis:** Here we can see that on increasing the matix size and number of threads doesn’t result in decrease in execution time. So, for small matrix size parallelizing is not a good option.

**GitHub Link:** <https://github.com/meetgandhi692/HPC-Lab/tree/52fd41746f81d008d559dd997530e81d3f6707d3/Assignment%205>