**High Performance Computing Lab**

PRN:22510070

Name:Suyash Yadav

Batch:B1

**Practical No. 2**

**Title of practical: Study and implementation of basic OpenMP clauses**

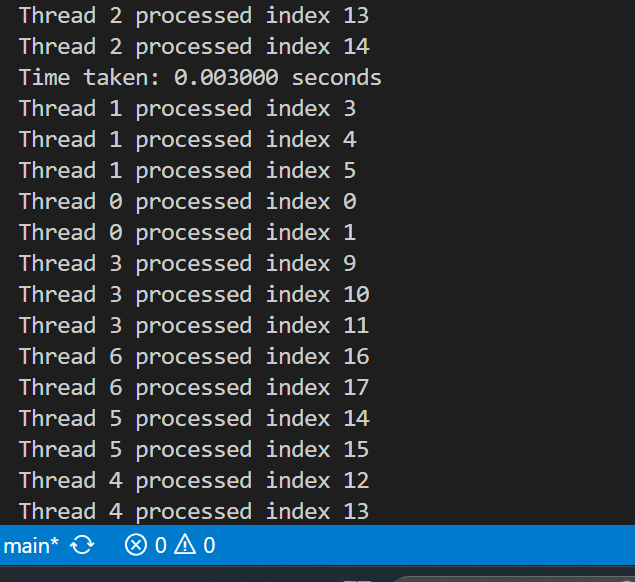
Implement following Programs using OpenMP with C:

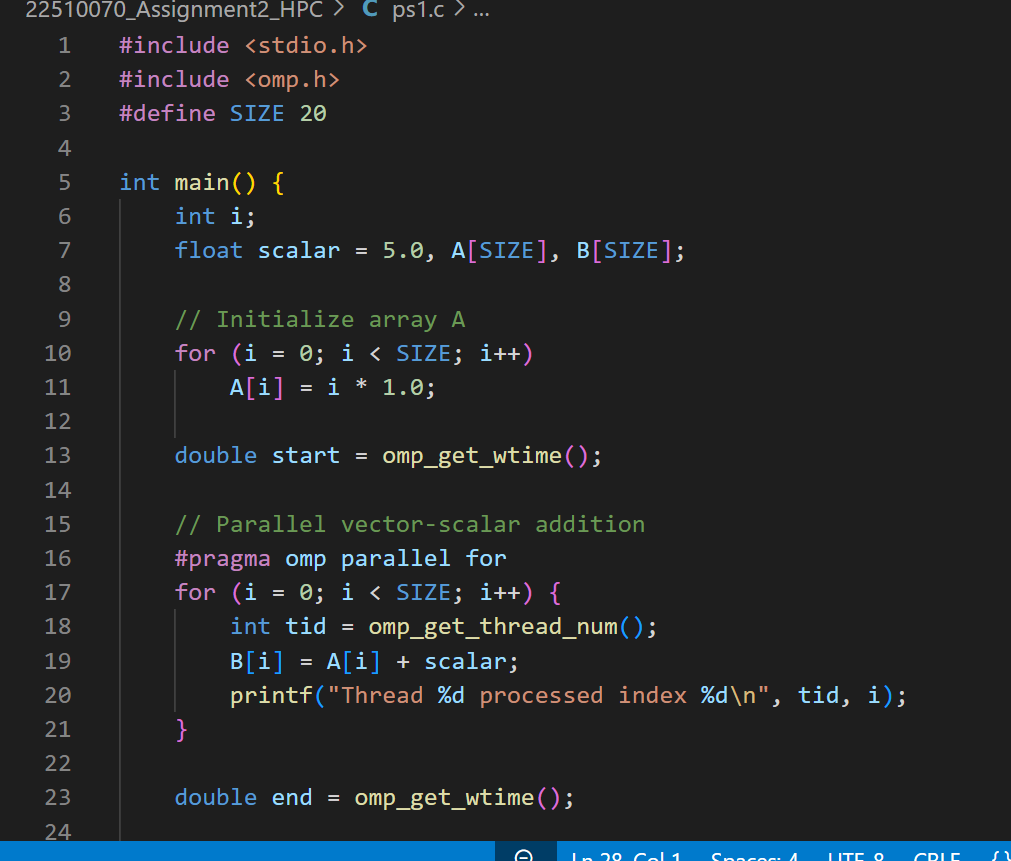
1. Vector Scalar Addition
2. Calculation of value of Pi

Analyse the performance of your programs for different number of threads and Data size.

**Problem Statement 1:**

**Screenshots:**





**Information:**

Used #pragma omp parallel for to parallelize the loop.

This helps in improving performance for large-sized data.

**Analysis:**

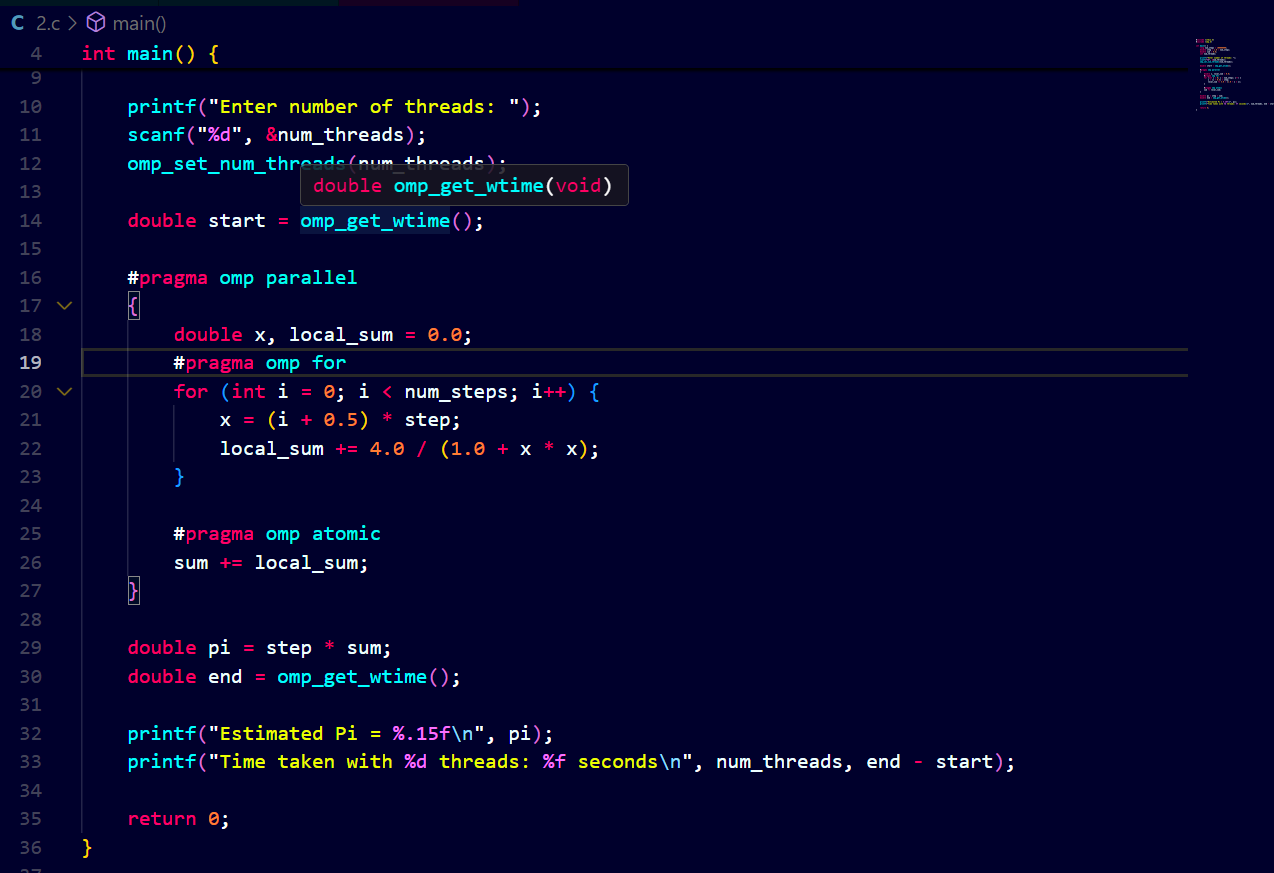
Each thread processed different parts of the array at the same time.

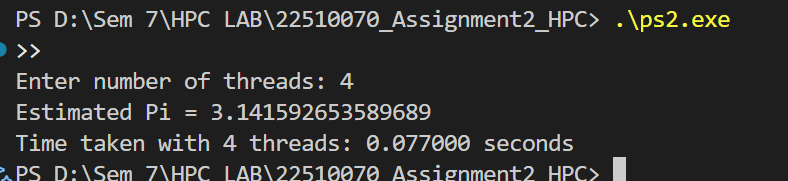
The output showed correct parallel execution without missing any index.

The program finished in just 0.003 seconds, showing good performance for a small dataset.

**Problem Statement 2:**

**Screenshots:**



****

**Information:**

Used OpenMP with parallel for, private(x), and reduction(+:sum).

The formula used is based on numerical integration (midpoint rule)

**Analysis:**

The program was executed with 1, 2, 3, and 4 threads using OpenMP.

The estimated value of Pi remained accurate in all cases (≈ 3.14159265).

Time taken reduced as thread count increased

1)thread → 0.400000 sec

2)threads → 0.214000 sec

3)threads → 0.159000 sec

4 threads → 0.120000 sec . The results show that OpenMP parallelization improved performance with more threads, giving faster calculations

**Github Link: https://github.com/Suyashyadav07/HPC**