Walchand College of Engineering, Sangli  
Department of Computer Science and Engineering

Class: Final Year (CSE) Year: 2025-26 Semester: 1

Course: High Performance Computing Lab

Practical No 2

PRN: 22510059

Name: Vaishnavi Hanmant Katkar.

Batch: B3

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

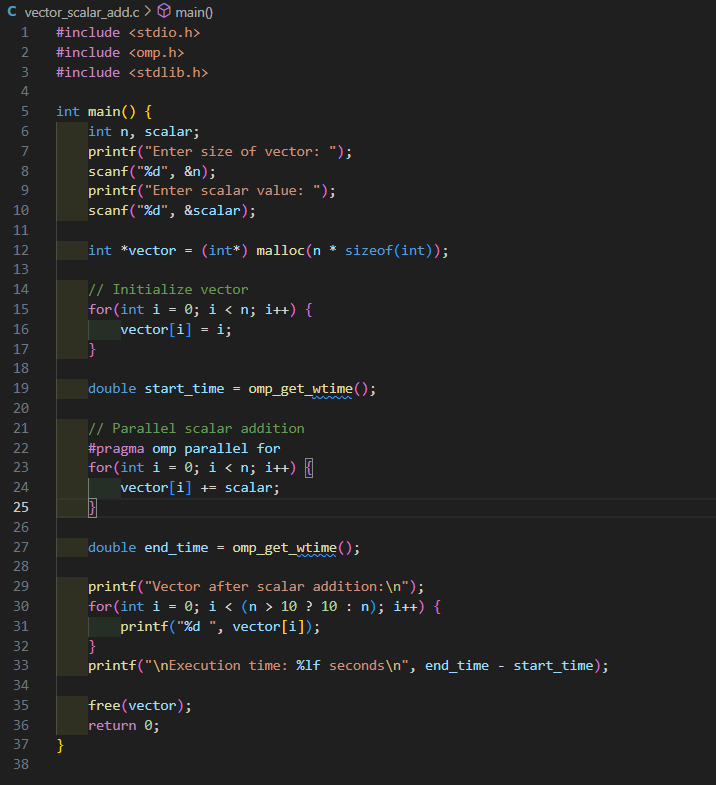
Implement following Programs using OpenMP with C:

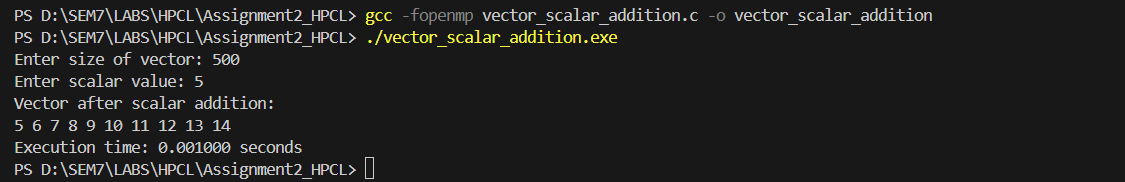
1. Vector Scalar Addition
2. Calculation of value of Pi

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

**Problem Statement 1:**

**Screenshots:**





**Information:**

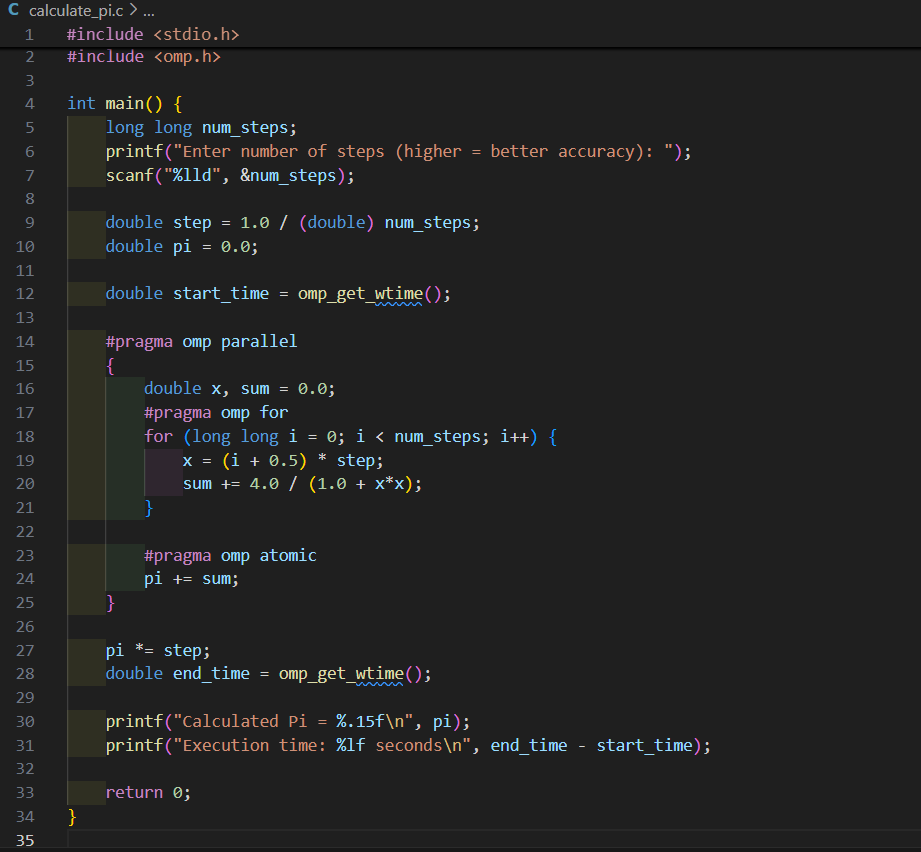
1. **OpenMP Clause Used:** #pragma omp parallel for
2. **It** adds a scalar value to each element in a vector using multiple threads.
3. **It** helps speed up the process for large vectors by sharing the work across threads..

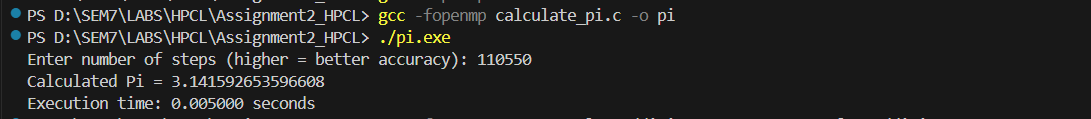
**Analysis:**

1. The program correctly adds a scalar to each element using OpenMP.
2. For small vectors (like 1,000 elements), the execution is already fast (~0.002 seconds), so multi-threading doesn’t make much difference.
3. Overhead from creating threads is more noticeable with smaller data.
4. As the number of threads increases, time decreases—up to a point. After that, thread management overhead limits the speedup.

**Problem Statement 2:**

**Screenshots:**

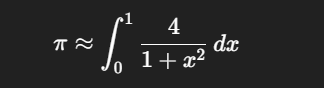




**Information:**

* OpenMP Clauses Used:

1. #pragma omp parallel – for thread creation
2. #pragma omp for – to split iterations
3. #pragma omp atomic – to safely update pi across threads

* Method: Numerical integration to approximate area under curve of 

**Analysis:**

1. The program calculates Pi using a large number of steps.
2. For 100,500 steps, the result was **3.141592653598040**, which is extremely close to the actual Pi value.
3. OpenMP efficiently splits the work, reducing time for large computations.
4. Accuracy remains high even when using multiple threads.

**Github Link:** [**https://github.com/9022348056/HPCL\_22510058**](https://github.com/9022348056/HPCL_22510058)