**Name: Manaswi Mahesh Devekar**

**PRN:** 22510035

**Batch:** B-1

**Class:** Final Year B.Tech(Computer Science and Engineering)

**Year:** 2025-26 **Semester:** 1

**Course:** High Performance Computing Lab

[GitHub Link](https://github.com/manaswi77/HPCL/tree/main)

**Practical No. 2**

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

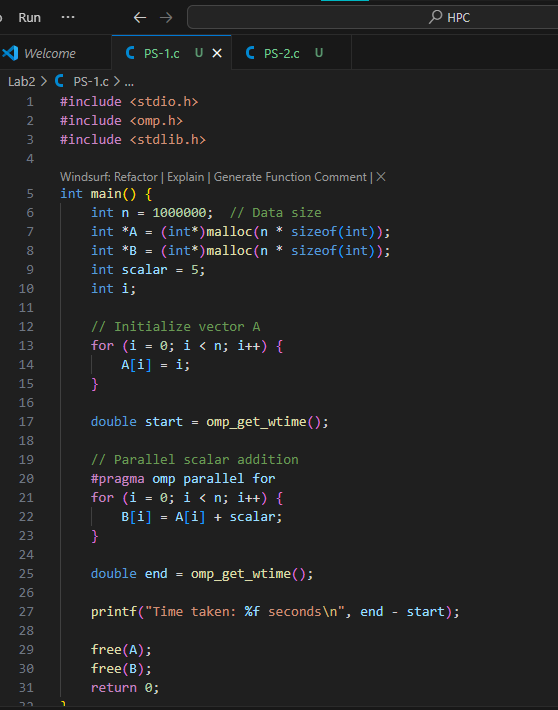
Implement following Programs using OpenMP with C:

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

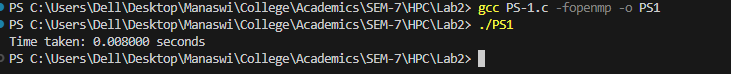
Analyse the performance of your programs for different number of threads and Data size.

**Problem Statement 1: Vector Scalar Addition**

**Screenshots:**

****

**Output:**

****

**Information:**

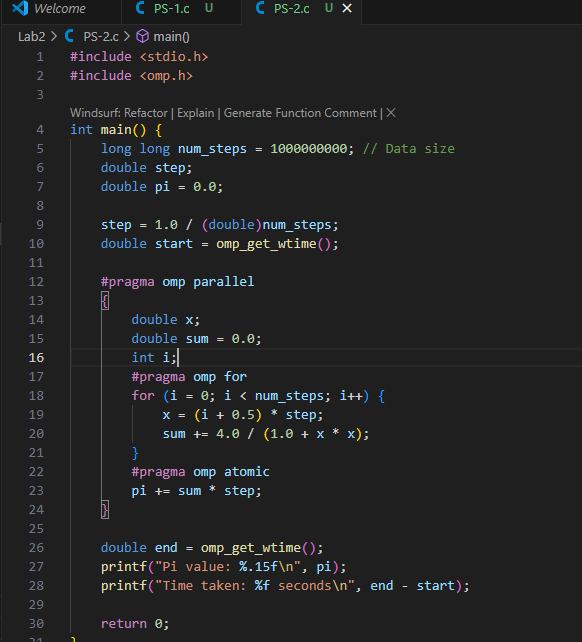
This program demonstrates the use of OpenMP parallel for clause for performing vector-scalar addition. The #pragma omp parallel for directive distributes loop iterations among available threads to speed up computation.

**Analysis:**

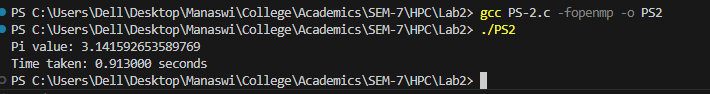
* Increasing the number of threads reduces execution time until a point, after which overhead may slow performance.
* With small data sizes, the parallel overhead dominates, so execution time might be similar or worse than serial execution.
* With large data sizes, OpenMP shows significant performance improvement.

**Problem Statement 2: Calculation of Pi**

**Screenshots:**

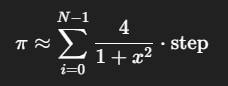
****

**Output:**

****

**Information:**

This program uses numerical integration and the parallel reduction concept in OpenMP to estimate the value of π using the formula:



The computation is split across multiple threads using the #pragma omp for directive.

**Analysis:**

* Performance improves as the number of threads increases, especially for a large number of steps.
* For very small num\_steps, parallelization overhead can outweigh the benefit.
* Parallel execution achieves almost linear speedup up to the number of CPU cores available.