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 1

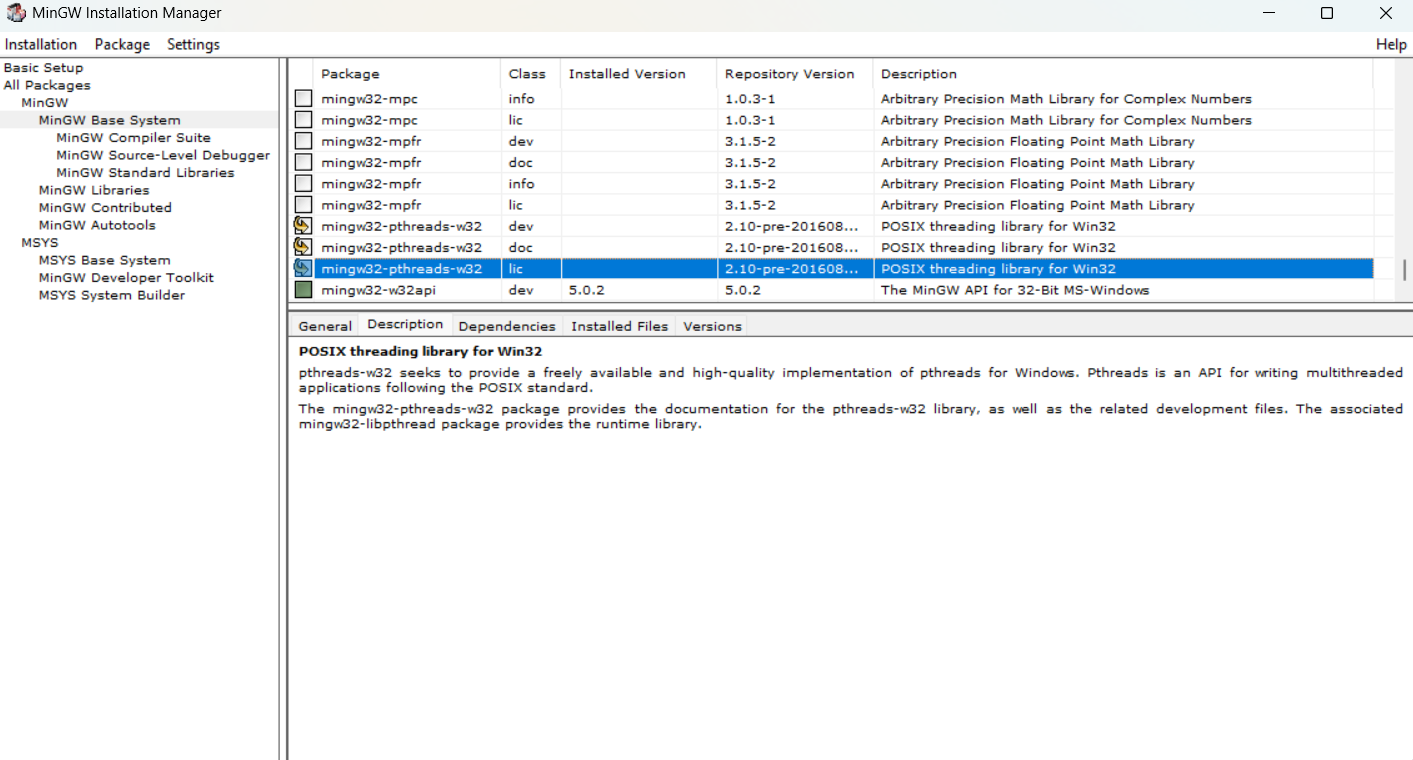
PRN: 22510058

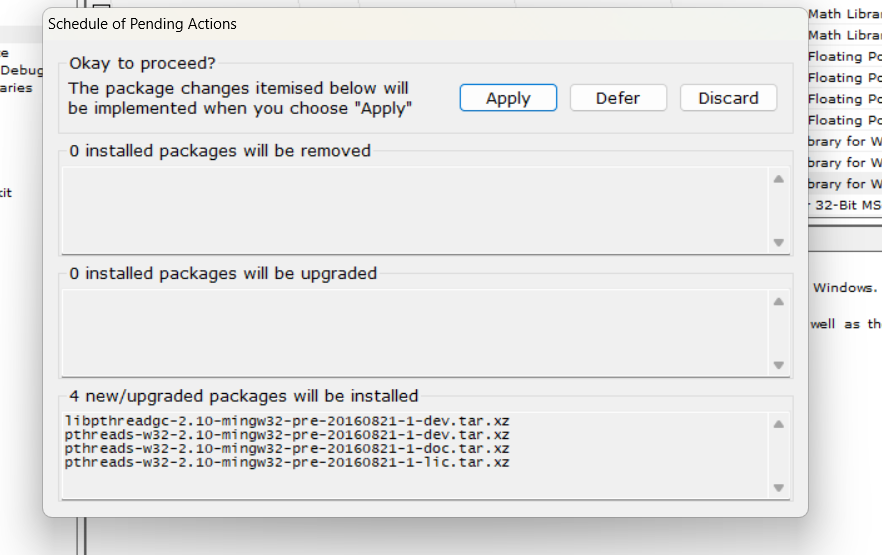
Name: Vaishnavi Hanmant Katkar.

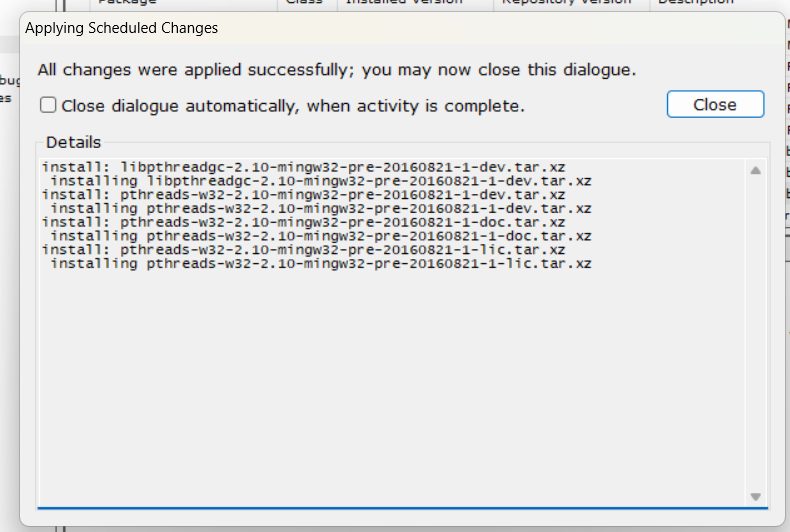
Batch: B3

**Title: Introduction to OpenMP**

Problem Statement 1 – Demonstrate Installation and Running of OpenMP code in C  
Installed MinGW and installed the necesarry packages.



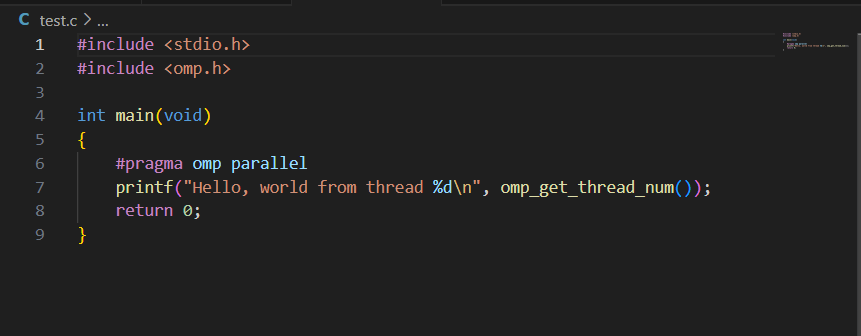


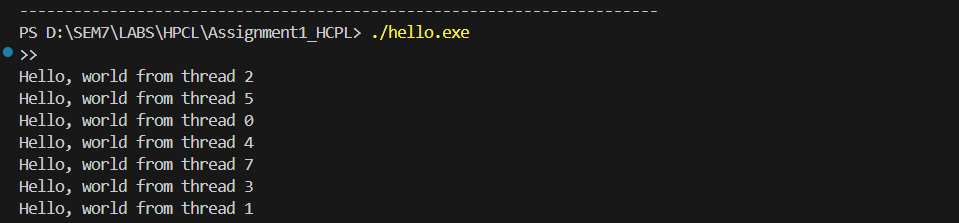


Then, to run a program in OpenMP, we have to pass a flag `-fopenmp`.

To run a basic Hello World :

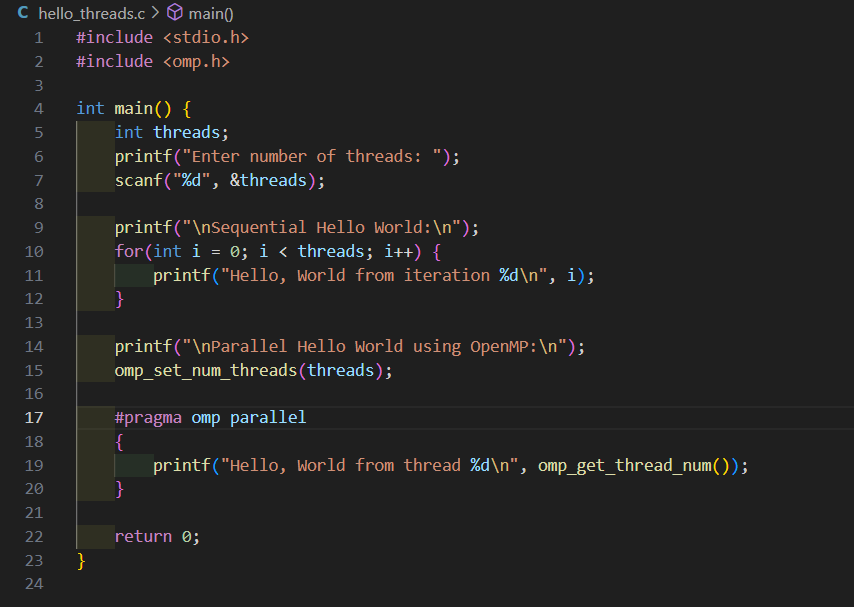
Code and output snapshot :



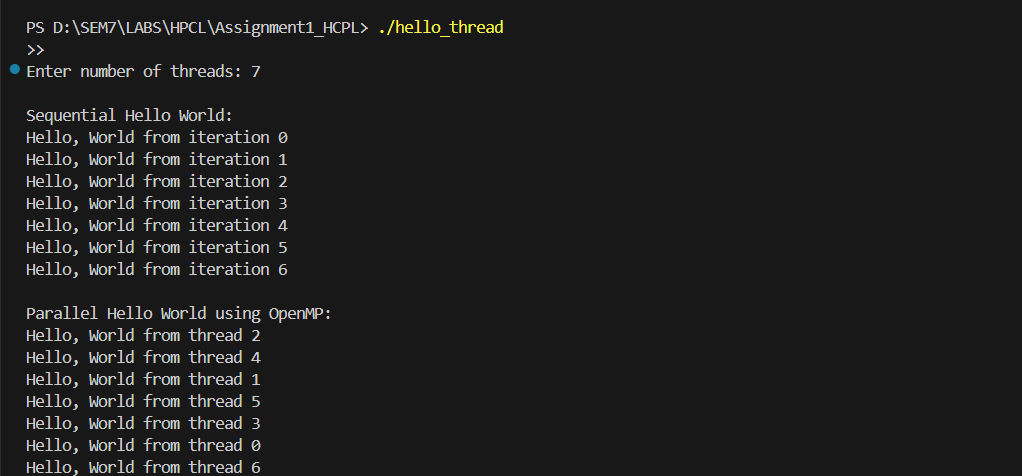


Problem Statement 2 – Print ‘Hello, World’ in Sequential and Parallel in OpenMP

Code snapshot:



Output snapshot:



Analysis:

1)In the sequential section, the program prints “Hello, World” one by one in the correct order because only the main thread is running.

2)In the parallel section, multiple threads run at the same time, so each thread prints its message. The order might change since all threads are working together.

3)This shows how OpenMP helps the program run faster by using several threads at once.

GitHub Link: <https://github.com/9022348056/HPCL_22510058>

Problem statement 3: Calculate theoretical FLOPS of your system on which you are running the above codes.

Cores: 4 cores per processor

Processors (sockets): 8

Total Cores = 4 × 8 = 32 cores

Clock Speed: 1.80 GHz = 1.80 × 10⁹ Hz (cycles/second)

FLOPs per cycle per core: 16

FLOPS = Total Cores × Clock Speed × FLOPs per Cycle per Core

= 32 × 1.80 × 10⁹ × 16

= 921.6 × 10⁹ FLOPS

= 921.6 GFLOPS