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

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

**Course:** High Performance Computing Lab

**Practical No. 4**

**Exam Seat No: 22510037**

**Name: Tejas Sanjay Chougule**

**Title of practical:**

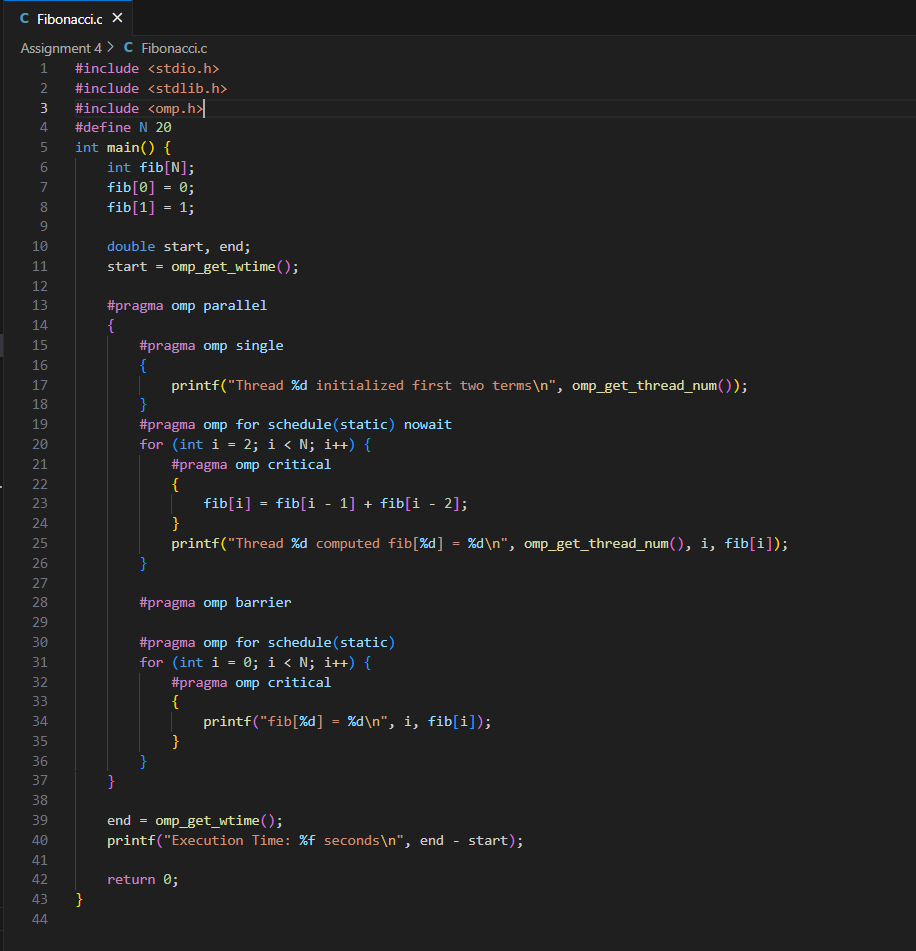
Study and Implementation of Synchronization

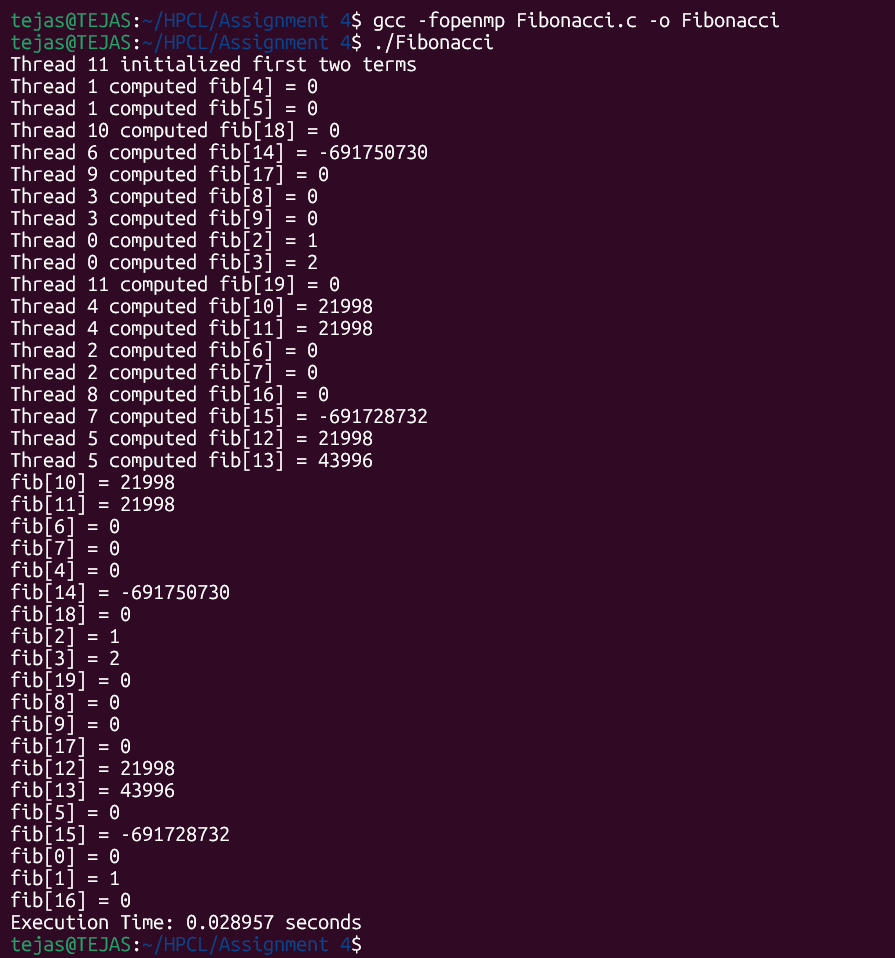
**Problem Statement 1:**

# Analyze and implement a Parallel code for below programs using OpenMP considering synchronization requirements. (Demonstrate the use of different clauses and constructs wherever applicable)

# Fibonacci Computation:

**Screenshots:**

****

****

**Information:**

Implemented parallel Fibonacci computation in C using OpenMP.

Used single for initialization, for with schedule(static) nowait for parallel work, critical for safe updates/printing, and barrier for synchronization.

Observed that synchronization ensures correctness but adds overhead, reducing speedup for small tasks.

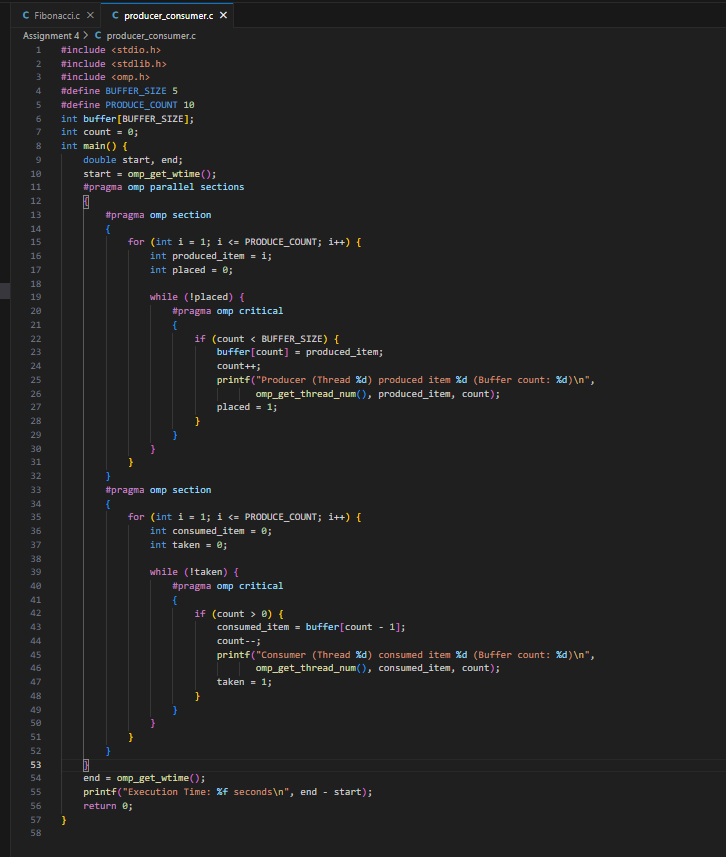
Concluded that minimal synchronization gives better performance while maintaining correctness.

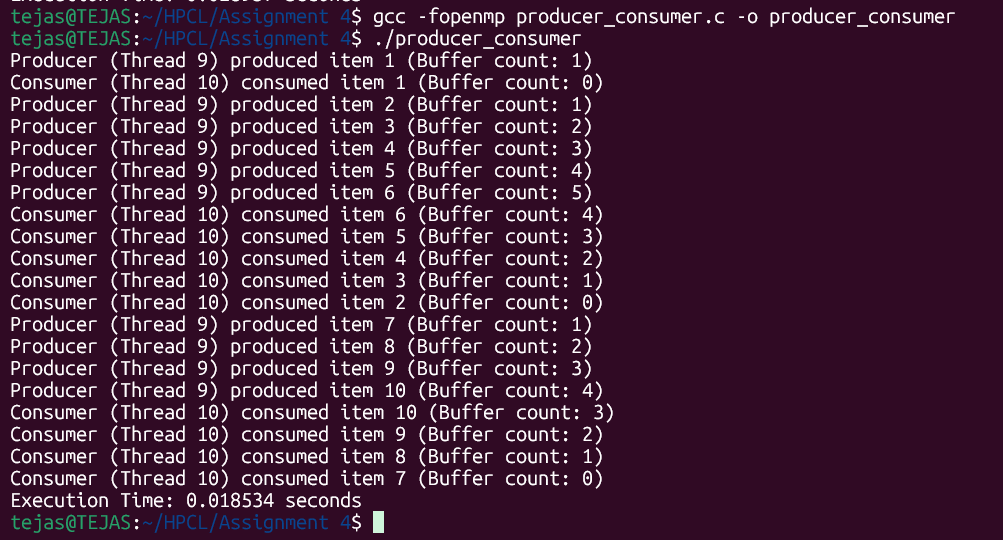
**Problem Statement 2:**

# Analyze and implement a Parallel code for below programs using OpenMP considering synchronization requirements. (Demonstrate the use of different clauses and constructs wherever applicable)

## Producer Consumer Problem

**Screenshots:**

****

****

**Information:**

Without critical, multiple threads could modify count or buffer at the same time, causing data corruption.

For a small buffer, synchronization overhead dominates.

The while loops simulate blocking behavior until a slot/item is available.

OpenMP is not ideal for producer–consumer compared to message queues, but works for demonstration.

**Github Link: https://github.com/TSC2004/Hpcl**