**Course:** High Performance Computing Lab

**Practical No. 4**

PRN: 22510057

Name: Ashutosh Gundu Birje

Batch: B8

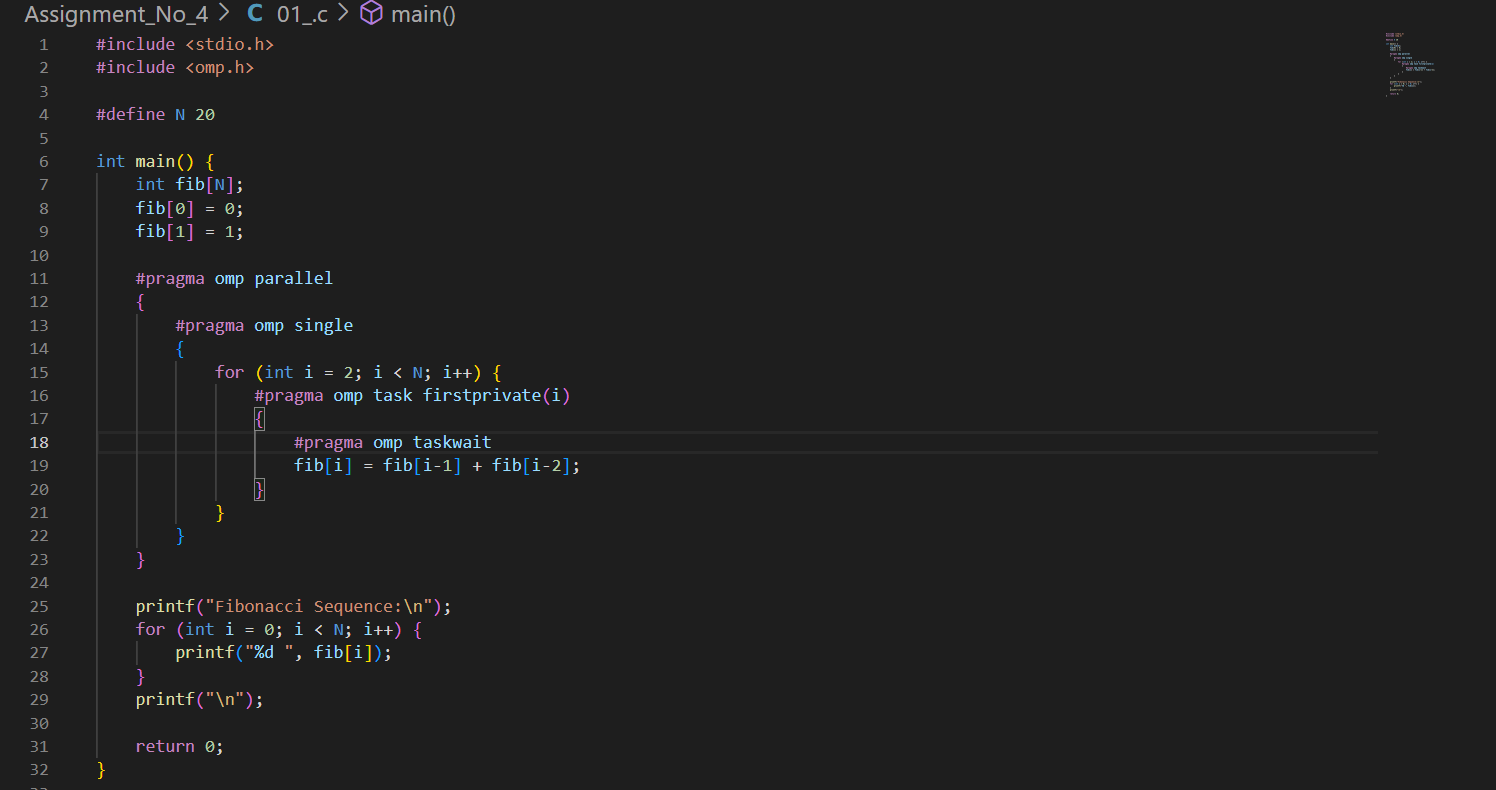
**Title of practical:**

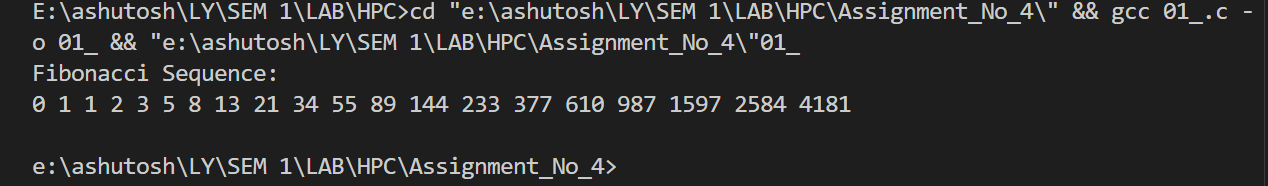
Study and Implementation of Synchronization

**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:**

**Synchronization:**In Fibonacci computation, each element fib[i] depends on the previous two (fib[i-1] and fib[i-2]). Without proper synchronization, threads might read uninitialized values.

**OpenMP Constructs Used:**

|  |  |
| --- | --- |
| **#pragma omp parallel** | **Creates multiple threads for parallel execution.** |
| **#pragma omp single** | **Ensures only one thread initiates the tasks.** |
| **#pragma omp task** | **Defines independent units of work to be executed in parallel.** |
| **#pragma omp taskwait** | **Waits for all created tasks to complete before proceeding.** |

**Clauses Demonstrated:**

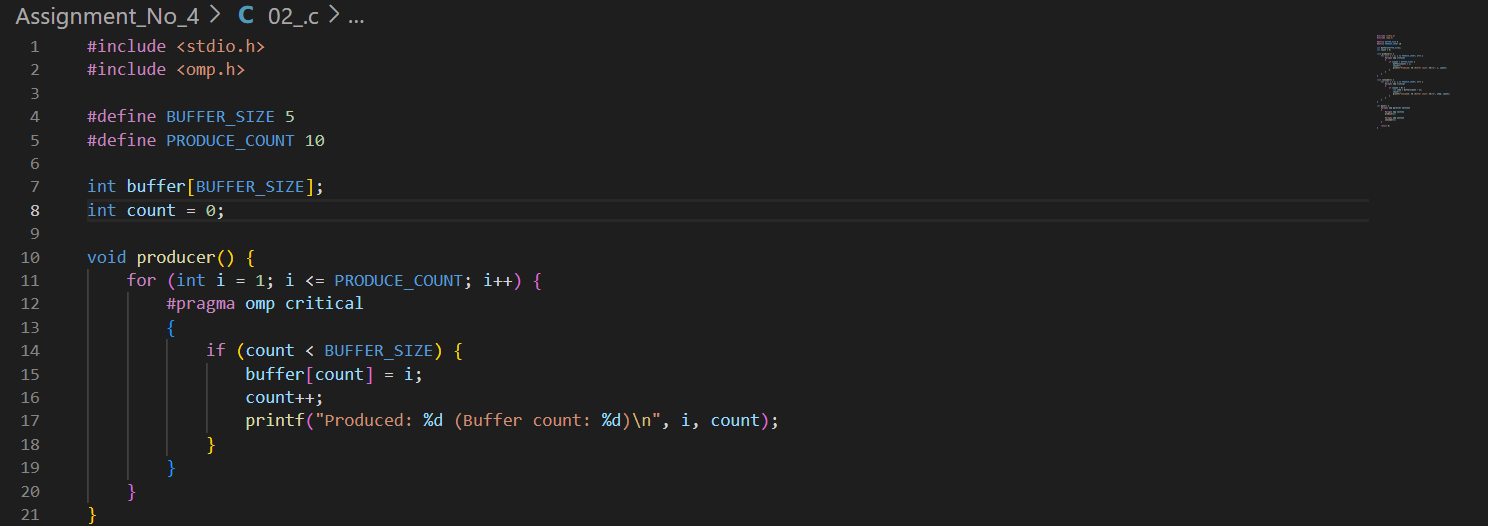
firstprivate(i): Ensures each task gets its own copy of i.

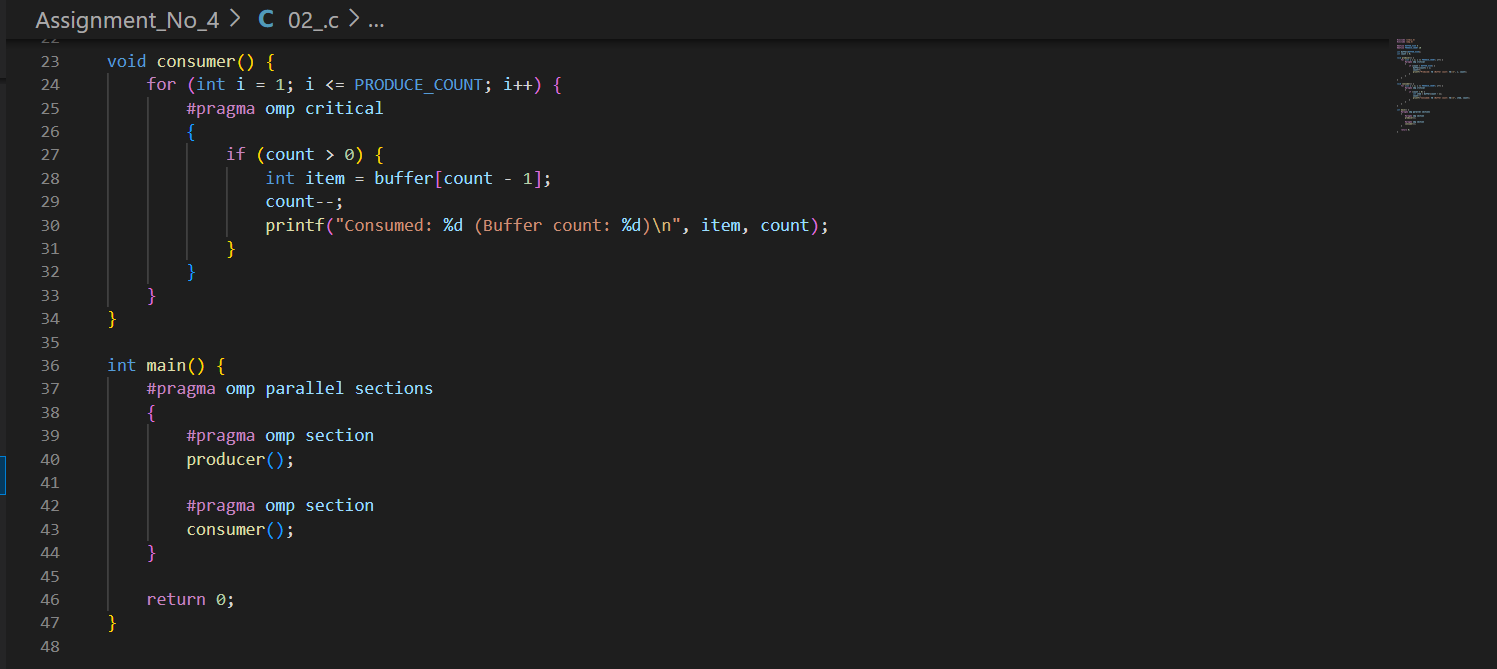
taskwait: Synchronizes dependent calculations.

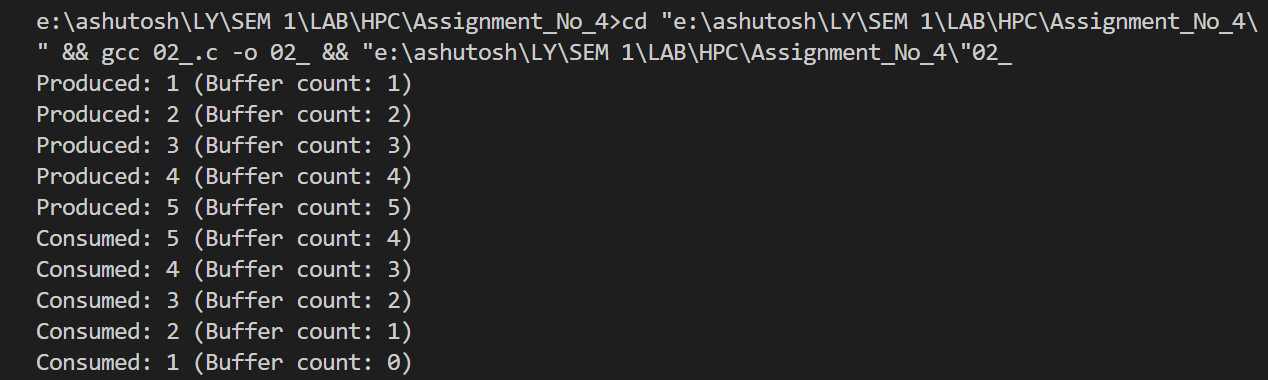
**Problem Statement 2:**

# 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:**

**Synchronization:**

Multiple threads try to modify shared variables (buffer[] and count). Without synchronization, race conditions occur.

The critical section ensures only one thread modifies buffer and count at a time.

**OpenMP Constructs Used:**

|  |  |
| --- | --- |
| **#pragma omp parallel sections** | **Divides work between producer and consumer.** |
| **#pragma omp section** | **Assigns specific blocks to different threads.** |
| **#pragma omp critical** | **Ensures mutual exclusion when accessing shared data.** |

**Clauses Demonstrated:**

critical: Prevents simultaneous access to the shared buffer.

sections/section: Used to split producer and consumer logic between threads.

**Github Link:**[**https://github.com/Ashutoshbirje/HPC-LAB/tree/master/Assignment\_No\_4**](https://github.com/Ashutoshbirje/HPC-LAB/tree/master/Assignment_No_4)