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

**Year:** 2022-23 **Semester:** 1

**Course:** High Performance Computing Lab

**Practical No. 4**

**Exam Seat No:**

1. 2019BTECS00038 – Sadaf Najeem Mulla

**Title of practical:**

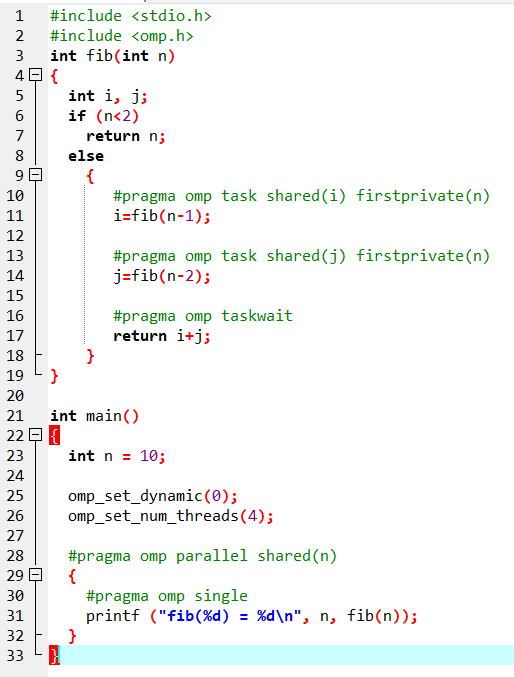
Study and Implementation of Synchronization

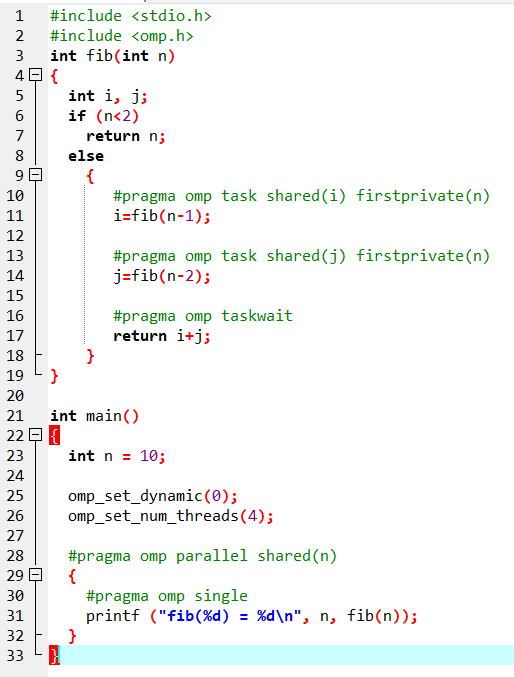
**Problem Statement 1:**

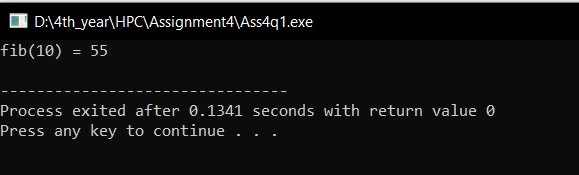
# Analyse and implement a Parallel code for below programs using OpenMP considering synchronization requirements. (Demonstrate the use of different clauses and constructs wherever applicable)

# Fibonacci Computation:

**Screenshots:**







**Information :**

The call to fib(n) generates two tasks, indicated by the task directive. One of the tasks computes fib(n-1) and the other computes fib(n-2), and the return values are added together to produce the value returned by fib(n). Each of the calls to fib(n-1) and fib(n-2) will in turn generate two tasks. Tasks will be recursively generated until the argument passed to fib() is less than 2.

The taskwait directive ensures that the two tasks generated in an invocation of fib() are completed (that is. the tasks compute i and j) before that invocation of fib() returns.

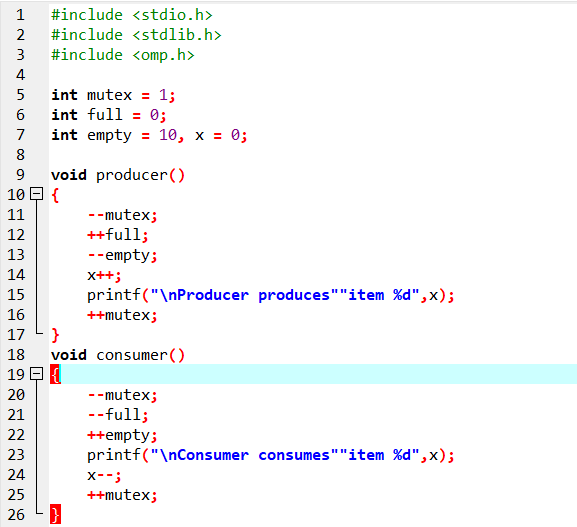
Note that although only one thread executes the single directive and hence the call to fib(n), all four threads will participate in executing the tasks generated.

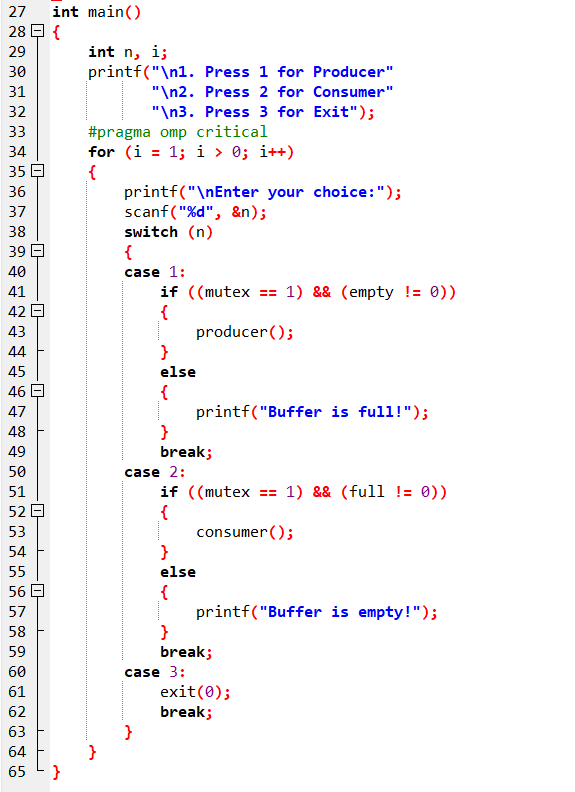
**Problem Statement 2:**

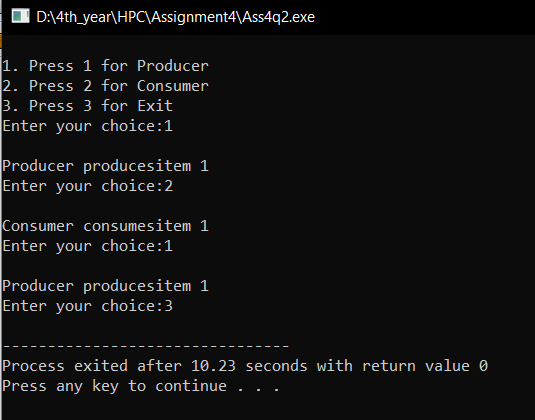
# Analyse and implement a Parallel code for below programs using OpenMP considering synchronization requirements. (Demonstrate the use of different clauses and constructs wherever applicable)

## Producer Consumer Problem:

**Screenshots:**







**Information :**

The producer is to either go to sleep or discard data if the buffer is full. The next time the consumer removes an item from the buffer, it notifies the producer, who starts to fill the buffer again. In the same manner, the consumer can go to sleep if it finds the buffer to be empty. The next time the producer puts data into the buffer, it wakes up the sleeping consumer.

**Github Link:** [**https://github.com/sadafmulla/HPC\_LAB/tree/main/Assignment4**](https://github.com/sadafmulla/HPC_LAB/tree/main/Assignment4)