**Name:** Meet Vipul Gandhi

**PRN:** 2020BTECS00112

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

**Year:** 2023-24 **Semester:** 1

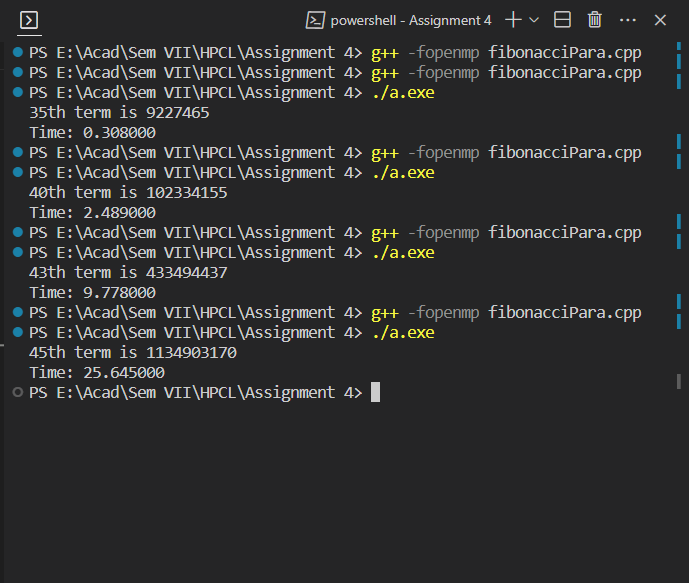
**Course:** High Performance Computing Lab

**Practical No. 4**

**Title of practical:** Study and Implementation of Synchronization. Analyze and implement a Parallel code for below programs using OpenMP considering synchronization requirements. (Demonstrate the use of different clauses and constructs wherever applicable)

**Problem Statement 1:** Fibonacci Computation

**Screenshots:**

****

**Information:**

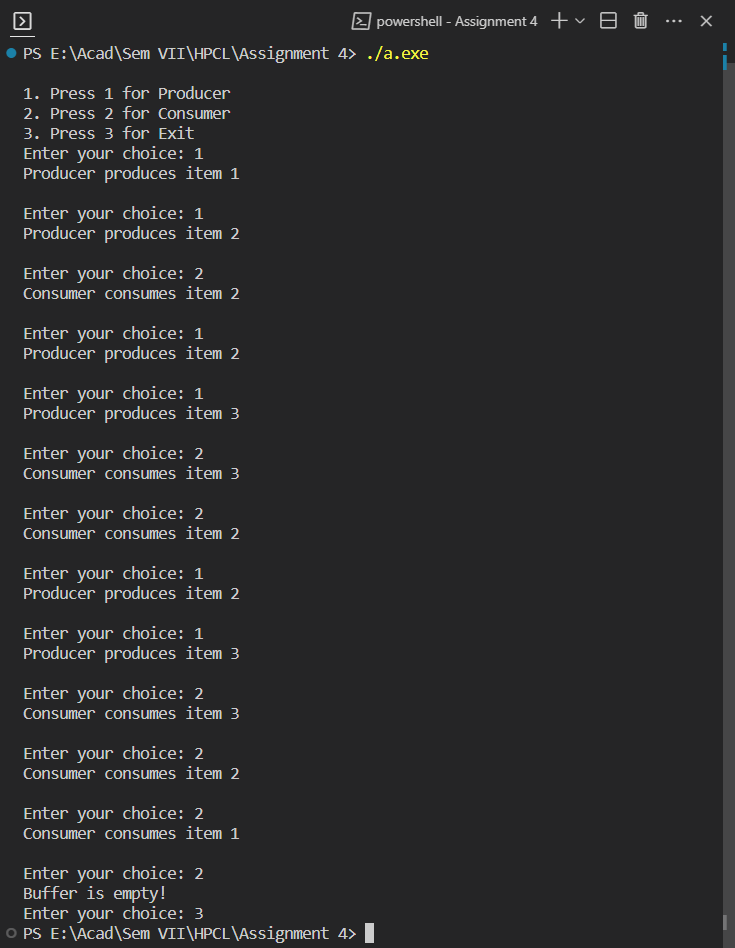
In this example, we're using tasks to parallelize the Fibonacci calculation. Here's how it works:

1. fibonacci(n) is a recursive function that calculates the nth Fibonacci term. If n is less than or equal to 1, it returns n.
2. In the recursive case, two tasks are created (x = fibonacci(n - 1) and y = fibonacci(n - 2)).
3. The taskwait directive ensures that the program waits until both tasks are completed before summing their results.
4. In the main function, a single parallel region is created to start the Fibonacci calculation.

Keep in mind that using tasks for Fibonacci might not always result in a performance improvement, as the overhead of task creation and management can sometimes outweigh the benefits of parallelization. For more complex problems, different algorithms (like memoization or iterative approaches) may be more suitable. Additionally, the cutoff condition for task creation (in this case if (n > 20)) might need to be adjusted based on your system's performance characteristics.

**Problem Statement 2:** Producer Consumer Problem

**Screenshots:**

****

**Information:**

1. We use ***#pragma omp critical*** sections to ensure that only one thread at a time can access the shared data structures (buffer in this case).
2. The producer produces 10 items, and the consumer consumes 10 items.
3. This is to properly synchronize the problem.

**GitHub Link:** <https://github.com/meetgandhi692/HPC-Lab/tree/79b911b51b14aed89f30d1a35fe33b2761b4459f/Assignment%204>