Walchand College of Engineering, Sangli
Department of Computer Science and Engineering

Class: Final Year (Computer Science and Engineering)
Year: 2021-22
Semester: 1
Course: High Performance Computing Lab
Batch: B4

Practical No. 04

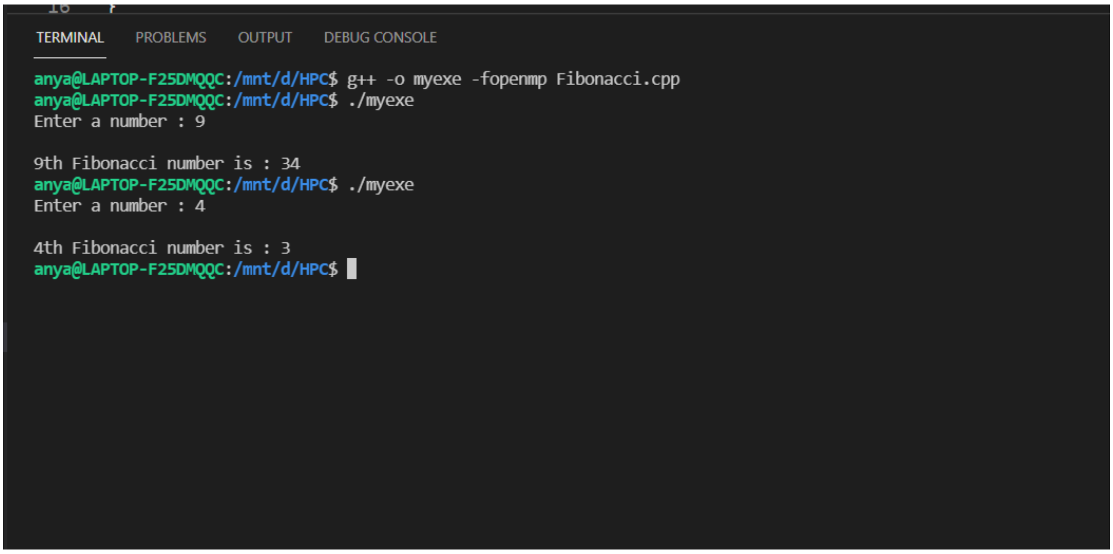
- ## Exam Seat No :

2018BTECS00044 - Aniruddha Sanjay Palekar

- ## Problem Statement 1 :

Fibonacci Series using Dynamic Programming :

The next Fibonacci number depends on the previous one. So it should be serial, Hence ordered construct can be used as shown in below



```
anya@LAPTOP-F25DMQQC:/mnt/d/HPC$ g++ -o myexe -fopenmp Fibonacci.cpp
anya@LAPTOP-F25DMQQC:/mnt/d/HPC$ ./myexe
Enter a number : 9

9th Fibonacci number is : 34
anya@LAPTOP-F25DMQQC:/mnt/d/HPC$ ./myexe
Enter a number : 4

4th Fibonacci number is : 3
anya@LAPTOP-F25DMQQC:/mnt/d/HPC$ 
```

- ## Problem Statement 2 :

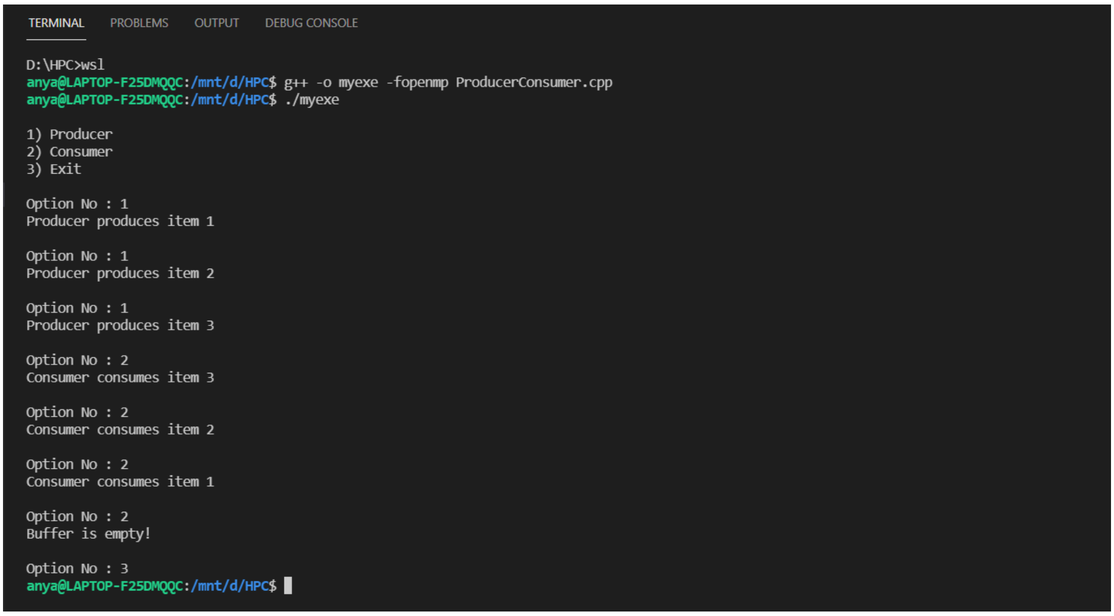
Analyse and implement a Parallel code for below programs using OpenMP
considering synchronization requirements. (Demonstrate the use of different
clauses and constructs wherever applicable)

Producer Consumer Problem –

Solution 1 :

Using variables as counting semaphores. Here the synchronization will be
achieved as the processes will be called through the critical section.

Output :

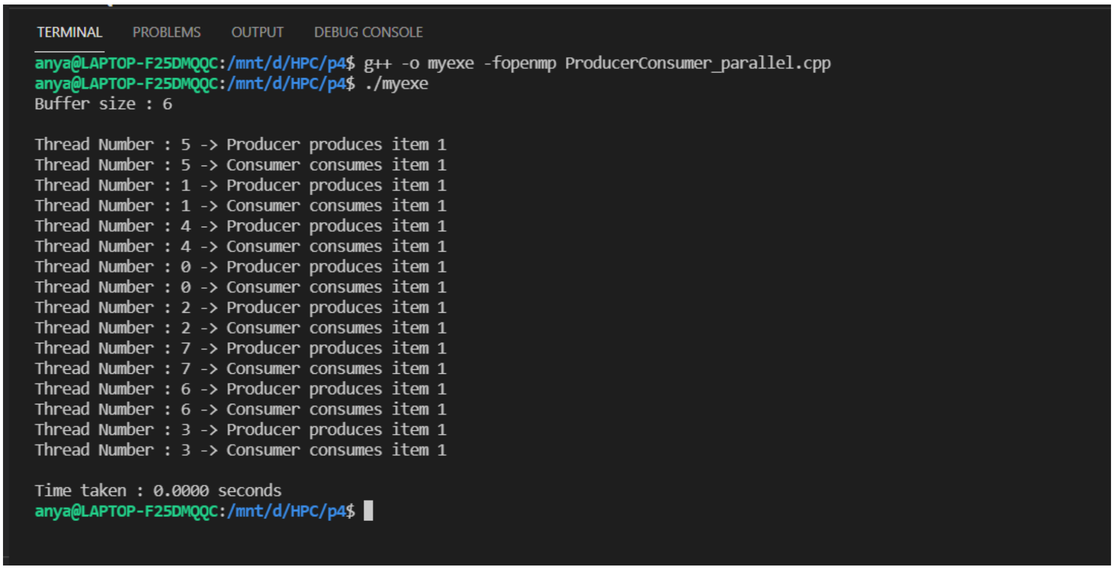


```
D:\HPC>wsl
anya@LAPTOP-F25DMQQC:/mnt/d/HPC$ g++ -o myexe -fopenmp ProducerConsumer.cpp
anya@LAPTOP-F25DMQQC:/mnt/d/HPC$ ./myexe

1) Producer
2) Consumer
3) Exit

Option No : 1
Producer produces item 1

Option No : 1
Producer produces item 2

Option No : 1
Producer produces item 3

Option No : 2
Consumer consumes item 3

Option No : 2
Consumer consumes item 2

Option No : 2
Consumer consumes item 1

Option No : 2
Buffer is empty!

Option No : 3
anya@LAPTOP-F25DMQQC:/mnt/d/HPC$
```

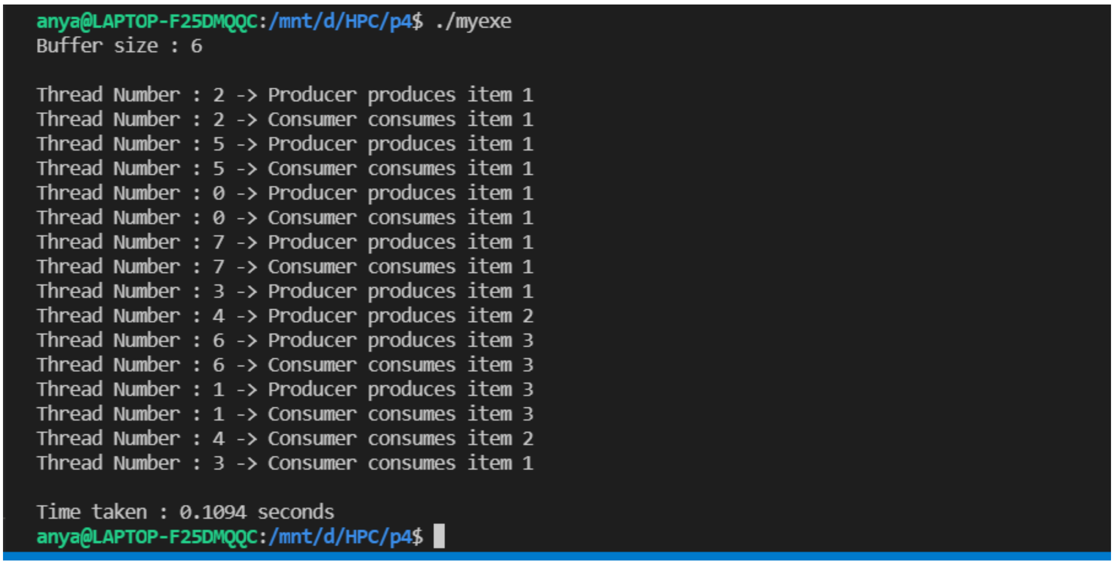
Solution 2 :

Using lock construct as mutex (Lock Variable) and calling producer and consumer processes in parallel using simply parallel construct. Here the synchronization will be achieved using the lock variable.

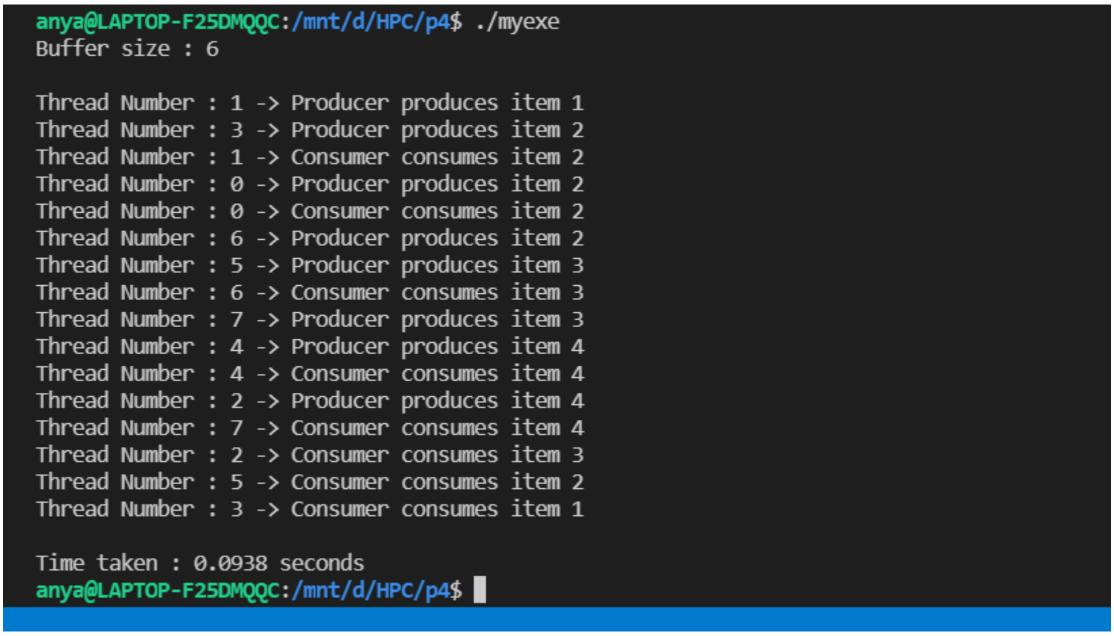
Output :



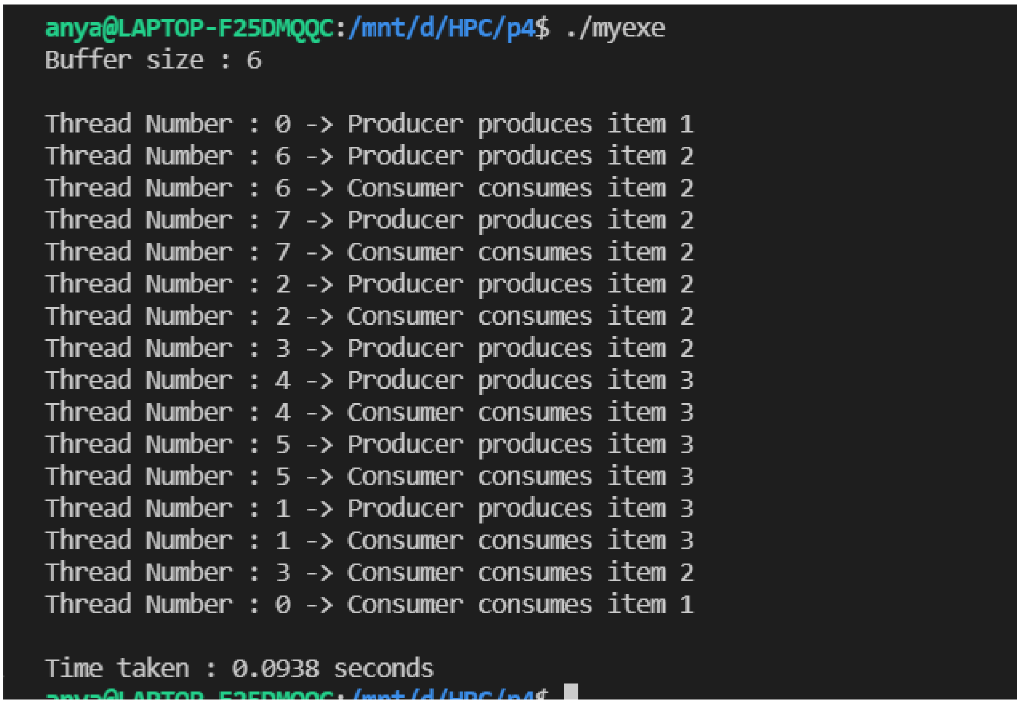
```
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/p4$ g++ -o myexe -fopenmp ProducerConsumer_parallel.cpp
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/p4$ ./myexe
Buffer size : 6

Thread Number : 5 -> Producer produces item 1
Thread Number : 5 -> Consumer consumes item 1
Thread Number : 1 -> Producer produces item 1
Thread Number : 1 -> Consumer consumes item 1
Thread Number : 4 -> Producer produces item 1
Thread Number : 4 -> Consumer consumes item 1
Thread Number : 0 -> Producer produces item 1
Thread Number : 0 -> Consumer consumes item 1
Thread Number : 2 -> Producer produces item 1
Thread Number : 2 -> Consumer consumes item 1
Thread Number : 7 -> Producer produces item 1
Thread Number : 7 -> Consumer consumes item 1
Thread Number : 6 -> Producer produces item 1
Thread Number : 6 -> Consumer consumes item 1
Thread Number : 3 -> Producer produces item 1
Thread Number : 3 -> Consumer consumes item 1

Time taken : 0.0000 seconds
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/p4$
```



```
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/p4$ ./myexe
Buffer size : 6

Thread Number : 2 -> Producer produces item 1
Thread Number : 2 -> Consumer consumes item 1
Thread Number : 5 -> Producer produces item 1
Thread Number : 5 -> Consumer consumes item 1
Thread Number : 0 -> Producer produces item 1
Thread Number : 0 -> Consumer consumes item 1
Thread Number : 7 -> Producer produces item 1
Thread Number : 7 -> Consumer consumes item 1
Thread Number : 3 -> Producer produces item 1
Thread Number : 4 -> Producer produces item 2
Thread Number : 6 -> Producer produces item 3
Thread Number : 6 -> Consumer consumes item 3
Thread Number : 1 -> Producer produces item 3
Thread Number : 1 -> Consumer consumes item 3
Thread Number : 4 -> Consumer consumes item 2
Thread Number : 3 -> Consumer consumes item 1

Time taken : 0.1094 seconds
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/p4$
```



```
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/p4$ ./myexe
Buffer size : 6

Thread Number : 1 -> Producer produces item 1
Thread Number : 3 -> Producer produces item 2
Thread Number : 1 -> Consumer consumes item 2
Thread Number : 0 -> Producer produces item 2
Thread Number : 0 -> Consumer consumes item 2
Thread Number : 6 -> Producer produces item 2
Thread Number : 5 -> Producer produces item 3
Thread Number : 6 -> Consumer consumes item 3
Thread Number : 7 -> Producer produces item 3
Thread Number : 4 -> Producer produces item 4
Thread Number : 4 -> Consumer consumes item 4
Thread Number : 2 -> Producer produces item 4
Thread Number : 7 -> Consumer consumes item 4
Thread Number : 2 -> Consumer consumes item 3
Thread Number : 5 -> Consumer consumes item 2
Thread Number : 3 -> Consumer consumes item 1

Time taken : 0.0938 seconds
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/p4$
```



```
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/p4$ ./myexe
Buffer size : 6

Thread Number : 0 -> Producer produces item 1
Thread Number : 6 -> Producer produces item 2
Thread Number : 6 -> Consumer consumes item 2
Thread Number : 7 -> Producer produces item 2
Thread Number : 7 -> Consumer consumes item 2
Thread Number : 2 -> Producer produces item 2
Thread Number : 2 -> Consumer consumes item 2
Thread Number : 3 -> Producer produces item 2
Thread Number : 4 -> Producer produces item 3
Thread Number : 4 -> Consumer consumes item 3
Thread Number : 5 -> Producer produces item 3
Thread Number : 5 -> Consumer consumes item 3
Thread Number : 1 -> Producer produces item 3
Thread Number : 1 -> Consumer consumes item 3
Thread Number : 3 -> Consumer consumes item 2
Thread Number : 0 -> Consumer consumes item 1

Time taken : 0.0938 seconds
anya@LAPTOP-F25DMQQC:/mnt/d/HPC/p4$
```

- **Github Link : https://github.com/aniruddhapalekar/HPC**