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

Practical No. 4

Exam Seat No:2018BTECS00100

1. Exam Seat Number – Prakash Singh

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)

Screenshot #:

```
prax@prakx-ideapad:~/Desktop/HPC/4$ gcc -fopenmp 1.c
prax@prakx-ideapad:~/Desktop/HPC/4$ ./a.out
34
```

```c
//Fibonacci Series using Dynamic Programming
#include<stdio.h>
int fib(int n)
{
    /* Declare an array to store Fibonacci numbers. */
    int f[n+2]; // 1 extra to handle case, n = 0
    int i;
    /* 0th and 1st number of the series are 0 and 1*/
    f[0] = 0;
    f[1] = 1;

    #pragma omp ordered

    for (i = 2; i <= n; i++)
    {
    /* Add the previous 2 numbers in the series
    and store it */

    f[i] = f[i-1] + f[i-2];
    }
    return f[n];
}
int main ()
{
    int n = 9;
    printf("%d", fib(n));
    getchar();
    return 0;
}

```

Information #:
Used `#pragma omp ordered` to run to compute the fibonacci sum sequentially.

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)

Screenshot #:

```
prax@prakx-ideapad:~/Desktop/HPC/4$ ./a.out

1. Press 1 for Producer
2. Press 2 for Consumer
3. Press 3 for Exit
Enter your choice:1

Producer producesitem 1
Enter your choice:1

Producer producesitem 2
Enter your choice:1

Producer producesitem 3
Enter your choice:2

Consumer consumes item 3
Enter your choice:2

Consumer consumes item 2
Enter your choice:2

Consumer consumes item 1
Enter your choice:2
Buffer is empty!
Enter your choice:
```

Information #:
Using `#pragma omp critical` , we use the concept of parallel programming and Critical Section to implement the Producer-Consumer problem in C language using OpenMP.

Github Link:https://github.com/prakx1/HPC-LAB/tree/master/4