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

Class: Final Year (Computer Science and Engineering)

Year: 2025-26 **Semester:** 1

Course: High Performance Computing Lab

Practical No. 4

Exam Seat No: 22510021

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:

Code:

```
#include <stdio.h>
#include <omp.h>
int fib(int n)
{
  int i, j;
  if (n<2)
    return n;
  else
    {
       #pragma omp task shared(i) firstprivate(n)
       i=fib(n-1);

       #pragma omp task shared(j) firstprivate(n)
       j=fib(n-2);

       #pragma omp taskwait
       return i+j;
    }
}
```

```
int main()
{
  int n = 10;

  omp_set_dynamic(0);
  omp_set_num_threads(5);

  #pragma omp parallel shared(n)
  {
    #pragma omp single
    printf ("fib(%d) = %d\n", n, fib(n));
  }
}
```

Screenshots:

Information:

- The program computes the nth Fibonacci number (fib(10)) using recursion and parallelism with OpenMP tasks.
- OpenMP tasks are used to run the two recursive calls fib(n-1) and fib(n-2) at the same time.
- Taskwait is used to wait until both tasks finish before adding their results.
- The program sets 5 threads using omp_set_num_threads(5).
- Dynamic adjustment is turned off using omp_set_dynamic(0) so that OpenMP always uses exactly 5 threads.

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:

Github Link: